PERSONAEDIT: Representative Sample Selection for Personalized Model Editing

You-Mei Huang,[1] Chung-Chi Chen,[2] An-Zi Yen[1]
[1] Department of Computer Science, National Yang Ming Chiao Tung University, Taiwan
[2] Human-Agent Ally Lab (HAA Lab), National Institute of Informatics (NII), Japan
ymh636.cs13@nycu.edu.tw, chen@nii.ac.jp, azyen@nycu.edu.tw

Abstract

Personalization has attracted growing interest in LLM applications, yet existing retrieval-based approaches depend heavily on retrieval quality and degrade in long-term interactions. Model editing, which directly modifies internal model parameters to incorporate new knowledge, has demonstrated effective knowledge modification capabilities in factual knowledge editing tasks and may provide a potential solution for personalization. However, scaling model editing to personalization is non-trivial. Editing large amounts of user data increases computational cost and causes interference among edits, motivating the need for effective sample selection. To address this issue, we propose, PERSONAEDIT, a hidden representation clustering strategy that selects representative editing samples through proportional stratified sampling. Experiments show that model editing is effective for personalization, and that our selection strategy preserves most of the performance while substantially reducing the number of required editing samples. Beyond standalone editing, we find that combining model editing with retrieval-based prompt augmentation further improves personalization, as edited knowledge and retrieved context provide complementary information. These results demonstrate the potential of model editing as an efficient and scalable approach for LLM personalization.

1 Introduction

With the rapid advancement of large language models (LLMs) in dialogue systems and user-interactive applications, there has been growing interest in personalized LLM systems that adapt responses according to individual user preferences and behaviors. Personalization has shown potential for improving user satisfaction and response quality in personalized response generation tasks (Zhang et al., 2026; Qiu et al., 2025b). Existing approaches commonly rely on context-based methods, including In-context Learning (ICL) and Retrieval-Augmented Generation (RAG) (Zhang et al., 2026; Qiu et al., 2025b; Kim and Yang, 2025). However, these methods suffer from three fundamental limitations: (1) strong dependence on retrieval quality and prompt engineering (Lewis et al., 2020); (2) inherent constraints imposed by the limited context window (Ratner et al., 2023); (3) performance degradation in long-term interaction settings (Tan et al., 2025), as the model lacks the ability to truly internalize user preferences.

Recent studies have begun to explore more structured personalization approaches, such as user embeddings, soft prompt tuning and representation editing (Doddapaneni et al., 2024; Liu et al., 2025; Zhang et al., 2025). Among these approaches, model editing provides a promising direction by directly modifying model parameters to incorporate user-specific knowledge and preferences into the model, potentially enabling more persistent personalized behavior without relying entirely on external retrieval during inference (Meng et al., 2022a). However, compared with factual knowledge editing, the application of model editing to user-level personalization remains relatively less explored. Moreover, incorporating all user interaction data into the editing process introduces significant computational overhead and scalability challenges (Gupta et al., 2024). Therefore, this study investigates the following question: *Can effective personalization be achieved by editing the model using only a small set of representative samples?*

To address this issue, we investigate sample selection strategies for personalization through model editing, including topic-based and hidden representation approaches. Using OpinionQA (Santurkar et al., 2023) as a testbed, we analyze how different selection strategies influence the generalization behavior of model editing. Our analysis shows that human-annotated topics are insufficient to effectively characterize editing generalization, while hid-

Findings of the Association for Computational Linguistics: EMNLP 2026, pages 18945–18960
October 24-29, 2026 ©2026 Association for Computational Linguistics

den representation similarity better captures transferable editing behavior for sample selection. Further experiments demonstrate that the proposed approach improves sample efficiency while preserving personalization capability.

In addition, we observe that personalization performance varies across users and is influenced by both user stance consistency and the discrepancy between model preferences and user preferences.

Our contributions are summarized as follows:

- **Applying model editing to personalization:** We demonstrate that model editing can effectively improve personalization performance, and that incorporating additional user information, such as persona descriptions and few-shot examples, further enhances performance.[1]
- **Hidden representation-based sample selection for model editing:** We propose a hidden representation clustering strategy for selecting representative editing samples. Experimental results show that the approach reduces required editing samples while preserving personalization performance.
- **Analysis of user factors affecting model editing effectiveness:** We analyze how user-specific characteristics, including preference consistency and preference gaps between users and models, affect the effectiveness of model editing-based personalization.

2 Related Work

2.1 LLM Personalization

LLM personalization aims to generate responses that better align with individual users' preferences, viewpoints, or behavioral patterns. Early approaches primarily rely on prompt engineering or profile conditioning, incorporating user profiles and historical interaction records into prompts, or leveraging LLMs to iteratively construct personalized prompts to guide response generation (Kim and Yang, 2025). These methods are training-free and thus do not require additional model fine-tuning; however, their effectiveness is highly dependent on prompt design quality (Sclar et al., 2024) and the model's in-context reasoning capabilities.

More recent approaches increasingly adopt Retrieval-Augmented Generation (RAG) (Qiu et al., 2025b), which retrieves user-relevant information during inference and incorporates it into the context window to enhance personalization performance. However, retrieval-based approaches still suffer from several limitations, including strong dependence on retrieval quality, constraints imposed by the limited context window (Ratner et al., 2023), and performance degradation in long-term interaction settings (Tan et al., 2025).

Beyond retrieval-based methods, recent studies have also explored latent personalization approaches, which represent user preferences in latent space through mechanisms such as user embeddings (Liu et al., 2025), soft prompts (Qiu et al., 2025a), and representation editing (Zhang et al., 2025).

However, these methods still rely on latent conditioning or representation-level intervention during inference, requiring personalized information to be provided or maintained externally rather than incorporated into the model itself. This motivates the exploration of approaches that directly encode personalized knowledge into model parameters, such as model editing.

2.2 Model Editing

Model editing aims to inject new knowledge into the parameters of pre-trained language models through direct parameter modifications. Existing approaches can generally be categorized into two paradigms: locate-then-edit methods, represented by ROME (Meng et al., 2022a), and hypernetwork-based methods, represented by MEND (Mitchell et al., 2021), which learn parameter update directions through additional training. In this work, we primarily focus on the former approach.

Locate-then-edit methods first localize the transformer layers and hidden representations associated with target knowledge, typically through causal tracing (Meng et al., 2022a,b; Fang et al., 2025), and then modify the corresponding parameters to steer the model toward the desired outputs. Among these approaches, Meng et al. (2022a) introduce ROME, a single-layer editing method designed for individual factual associations. Subsequently, Meng et al. (2022b) present MEMIT, which extends this framework to batch editing of multiple facts. Building upon MEMIT, Fang et al. (2025) propose AlphaEdit, which incorporates a null-space constraint to reduce interference from parameter updates and mitigate undesired impacts on the model's existing knowledge and capabilities.

Recent studies have further extended model edit-

[1]The prompts and code are available at at https://github.com/NYCU-NLP-Lab/PersonaEdit

ing to LLM personalization. Huang et al. (2025) use clustering to augment each target preference with multiple semantically similar samples, in order to improve the editing accuracy of individual preferences.

However, scaling model editing to a large number of personalized preferences remains a challenge. Prior studies have shown that applying model editing to a large number of samples may lead to forgetting and downstream performance degradation (Gupta et al., 2024). This issue is particularly relevant in LLM personalization, where a user may have a large number of personalized preferences. Directly incorporating all user interaction data into model editing not only incurs substantial computational overhead, but may also worsen catastrophic forgetting and reduce the scalability of model editing.

Therefore, our work considers a user with multiple personalized preferences and uses hidden-representation clustering to select a compact set of representative preferences before editing. This strategy aims to reduce editing cost while maintaining personalization effectiveness and overall model performance. Although both approaches employ clustering, their purposes are different: Huang et al. (2025) use clustering for preference augmentation, whereas we use it to select representative preferences under a limited editing budget.

3 PERSONAEDIT

3.1 Editing Sample Selection Strategy

Prior work has shown that the effects of model editing can transfer to related inputs, suggesting that editing effects are not limited to the edited instance (Meng et al., 2022a,b; Fang et al., 2025). In addition, existing studies have observed that editing facts with similar subjects may lead to interference effects, suggesting potential dependencies among editing outcomes across different inputs (Dong et al., 2025). Motivated by these findings, we hypothesize that similarity in hidden representations may be associated with partially shared internal computational pathways, which in turn may affect the locality of editing effects.

The overall procedure can be formulated as the following three stages.

3.1.1 Extract Hidden Representations

For each training sample, we feed the input prompt containing the subject s_i into a baseline language model and extract the hidden state of the last token corresponding to the subject from an intermediate layer used in model editing methods, resulting in a hidden representation vector $\mathbf{h}_i \in \mathbb{R}^d$. The choice of this layer follows the design settings in prior model editing methods (e.g., AlphaEdit (Fang et al., 2025)), ensuring that the extracted hidden representations are aligned with the model layer on which subsequent editing operations are performed.

3.1.2 Clustering via K-Means

To aggregate information and reduce potential overlap in computational pathways, we apply K-means clustering to the feature vectors $\{\mathbf{h}_i\}_{i=1}^{N}$, partitioning them into K disjoint clusters by minimizing the intra-cluster variance:

$$\arg\min_{\mathcal{C}} \sum_{k=1}^{K} \sum_{\mathbf{h}_i \in C_k} \|\mathbf{h}_i - \boldsymbol{\mu}_k\|^2 \tag{1}$$

where $\boldsymbol{\mu}_k$ denotes the centroid of cluster C_k. Through this clustering process, input prompts with similar hidden representations are grouped into the same cluster, reflecting potentially shared local computational structures within the model.

3.1.3 Proportional Stratified Sampling

To maintain overall coverage of the representation space under a limited number of selected samples while reducing sampling bias, we perform proportional stratified sampling within each cluster. Given the full training pool $\mathcal{D}$ containing $N = 300$ samples, we consider different settings for the number of selected samples M. The number of samples assigned to each cluster C_k is defined as:

$$n_k = \frac{|C_k|}{N} \cdot M \tag{2}$$

In practice, we round n_k to integers and further adjust the allocation to ensure that $\sum_k n_k = M$. We then rank samples within each cluster based on their Euclidean distance to the centroid $\boldsymbol{\mu}_k$ and select the top-n_k nearest samples, forming the final editing set $\mathcal{D}_{\text{edit}}$.

4 Experiments

In this section, we compare model editing with retrieval-based and prompt-based personalization methods to evaluate their relative effectiveness and potential complementarity in personalization tasks.

4.1 Dataset Preprocessing and Subsampling

We conduct experiments using OpinionQA (Santurkar et al., 2023), which contains longitudinal user-level question-answering data collected across 15 interaction waves, capturing individuals' opinions on various social and political topics. Due to its large scale, we select users who participate in all 15 interaction waves and further perform stratified sampling based on the performance of the original model. Specifically, we sample 10 users from each of the top-performing group (top 10), middle group (middle 10), and lowest-performing group (bottom 10), resulting in a total of 30 users and 31,017 interaction instances, to ensure diversity in user characteristics.

In the dataset construction, we consider 30 respondents whose question sets partially overlap. We therefore sample 300 questions from those commonly answered by all users to form the training set for subsequent clustering, while the remaining questions are used as the test set for personalized evaluation. In the test set, most users have over 700 test samples. Overall, the final dataset consists of 9,000 training samples and 22,017 test samples.

After selecting the experimental subset, we use the Gemini 3 Flash model to identify the subject entity from each original OpinionQA question. We then construct structured subject–relation–object (s, r, o) triplets by treating the remaining question text as the relation and the respondent's original answer as the object, in order to support locate-then-edit based model editing methods (Meng et al., 2022a). For example, the question "How much, if at all, do you worry about the following happening to you? Being the victim of a terrorist attack" with the response "Worry a little" is converted into (victim of a terrorist attack, How much, if at all, do you worry about the following happening to you? Being the, Worry a little). The subject is identified from the original question, while the remaining question text and the original response form the relation and object, respectively.

Additionally, to evaluate the generalization ability of model editing under semantic rephrasing, we also generate corresponding paraphrased queries for each question. The prompts used for this transformation are provided in Appendix B.

4.2 Experimental Setup

We compare the proposed method against several baselines, including Vanilla, Few-shot, Persona, All-info (Persona + Few-shot) and FERMI (Kim and Yang, 2025). The experiments are conducted using LLaMA 3.1 8B (Meta, 2024) under the k=18, training set size=200 setting.

In the few-shot setting, we retrieve the top-3 relevant samples for each task using BM25 and Contriever, and incorporate them into the prompt. In the persona setting, we use basic demographic information provided by OpinionQA (Santurkar et al., 2023) (e.g., gender, age, and income) as personalization conditions. The all-info setting combines both the top-3 retrieved samples and persona information in the prompt. FERMI adopts an iterative prompt optimization approach for inference (Kim and Yang, 2025).

Method	Accuracy	Gain	# Gain
Vanilla	33.38%	–	–
Few-shot$_{BM25}$	43.00%	9.67%	30
Few-shot$_{contriever}$	42.84%	9.49%	30
Persona	41.44%	8.05%	30
All info$_{BM25}$	44.60%	11.28%	30
All info$_{contriever}$	44.51%	11.15%	30
FERMI	34.52%	1.16%	16
PersonaEdit	44.67%	11.31%	30
+ persona	45.35%	12.01%	30
+ Few-shot$_{BM25}$	<u>45.73%</u>	<u>12.39%</u>	30
+ All info$_{BM25}$	**47.00%**	**13.65%**	30

Table 1: Evaluation results on the OpinionQA dataset. Bold and underlined values indicate the best and second-best performance, respectively. "Gain" denotes the average accuracy improvement per respondent compared to the Vanilla baseline, and "# Gain" indicates the number of respondents with performance gains.

4.3 Main Personalization Results

We first evaluate personalization performance using accuracy, where the final prediction is determined by matching the option in the model output. The experimental results are presented in Table 1. The results show that model editing achieves comparable or better performance than retrieval-based and prompt-based personalization methods. In particular, the combination of model editing and All-info achieves the best accuracy of 47.00%, outperforming all baseline approaches. Respondent-level analysis further demonstrates consistent improvements across users, with gains observed for all 30 respondents under model editing.

Discussion on FERMI Reproduction. As shown in Table 1, FERMI achieves lower performance than the other personalization baselines. We believe this discrepancy may stem from several dif-

Prompt	Vanilla	PersonaEdit	Gain	# Gain
w/o prompt	33.38%	44.67%	11.31%	30
$FS_{(BM25)}$	43.00%	45.35%	2.72%	25
Persona	41.44%	45.73%	3.96%	25
All $info_{(BM25)}$	44.60%	47.00%	2.51%	25

Table 2: Comparative analysis of Vanilla and PersonaEdit across different prompt augmentation strategies on the OpinionQA dataset. Bold and underlined values indicate the best and second-best performance within columns. FS denotes few-shot prompting. "Gain" denotes the average accuracy improvement per respondent brought by model editing, and "# Gain" indicates the number of respondents who achieved gains.

Prompt	PersonaEdit	GPT-5.4 mini	GPT-5.4
w/o prompt	44.67%	43.89%	37.63%
Few-$shot_{(BM25)}$	45.35%	47.64%	48.39%
Persona	45.73%	47.01%	49.67%
All $info_{(BM25)}$	47.00%	49.72%	51.51%
Total Cost	–	$9.63	$57.58

Table 3: Comparison of personalization performance between PersonaEdit and closed-source GPT-5.4 mini and GPT-5.4 models.

ferences between our setup and the original FERMI framework.

First, the original FERMI study primarily evaluates its method using larger-scale models, such as ChatGPT and Llama 2 70B, whereas our experiments are conducted mainly on the smaller Llama 3.1 8B Instruct model. During implementation, we observe that smaller models are more likely to exhibit unstable output formatting behavior during iterative prompt optimization. In particular, the optimized prompts occasionally fail to effectively constrain the model to select from predefined options, causing the model to generate verbose free-form responses rather than directly outputting the corresponding option, which negatively affects the final answer matching accuracy.

In addition, to ensure a consistent evaluation protocol across all personalization methods, our experimental setting requires every method to generate full answer content instead of producing a single option token (e.g., A/B/C/D). However, the original FERMI framework is primarily evaluated under a constrained option prediction setting. Under full-answer generation settings, FERMI's prompt optimization process appears to have greater difficulty maintaining output formatting, particularly when applied to smaller models.

A second possible reason stems from differences in the experimental data setting. The original FERMI study mainly focuses on small-scale personalization scenarios, where each user typically contains only a small number of training and testing samples (e.g., 20 training samples and 30 testing samples per user). In contrast, our experimental setting contains approximately 200 training samples and approximately 700 testing samples for each user. Since the data scale and personalization setting differ substantially from those used in the original FERMI study, the method design of FERMI may not directly generalize to larger-scale personalization scenarios.

Finally, because the official implementation of FERMI has not been released, our experiments are based on a reimplementation derived from the methodological descriptions provided in the original paper. As a result, implementation details may differ from those of the original system.

4.4 Effect of Prompt Augmentation

Beyond the comparison with existing personalization baselines, we further investigate whether model editing can be effectively combined with retrieval-based and prompt-based augmentation strategies. As shown in Table 2, the edited model consistently outperforms the non-edited model under all prompt settings, with statistically significant improvements. Respondent-level analysis reveals that 25 out of 30 respondents exhibit performance gains when using the edited model.

Comparison with Frontier Closed-Source Models. To further contextualize prompting-based personalization, we evaluate GPT-5.4 and GPT-5.4 mini under the same prompting settings. As shown in Table 3, under the strongest prompting setting (All-$info_{(BM25)}$), GPT-5.4 and GPT-5.4 mini achieve 51.51% and 49.72% accuracy, respectively, compared with 47.00% for PersonaEdit. While stronger closed-source models achieve higher accuracy, prompting-based personalization requires user information to be repeatedly provided during inference, or risks persona forgetting in multi-term interactions (Huang et al., 2025). In contrast, model editing directly incorporates user-specific preferences into model parameters for open-weight models.

4.5 Ordinal Evaluation

Additionally, since some questions in OpinionQA exhibit ordinal relationships among options, ac-

Method	Accuracy	MMAE (↓)	CEM (↑)
Vanilla	35.48%	1.0728	0.5356
Few-shot$_{(BM25)}$	41.47%	1.0842	0.5370
Few-shot$_{(Contriever)}$	40.33%	1.0802	0.5381
Persona	38.52%	1.0952	0.5377
All info$_{(BM25)}$	42.50%	1.1067	0.5350
All info$_{(Contriever)}$	40.97%	1.0986	0.5366
FERMI	34.56%	1.1237	0.5232
PersonaEdit	43.00%	**1.0669**	0.5382
+ Persona	42.84%	1.0744	0.5379
+ Few-shot$_{(BM25)}$	43.76%	1.0825	**0.5385**
+ All info$_{(BM25)}$	**44.95%**	1.0733	0.5371

Table 4: Evaluation results on the OpinionQA dataset focusing on questions with ordinal options. Bold and underlined values indicate the best and second-best performance within each column, respectively. ↑ indicates higher values are better, while ↓ indicates lower values are better.

curacy alone cannot capture the distance between predictions and ground-truth labels. Therefore, for these cases, we additionally adopt macro-averaged MAE (MMAE) (Baccianella et al., 2009) and the Closeness Evaluation Measure (CEM) (Amigó et al., 2020) as evaluation metrics.

Macro-averaged MAE measures the average distance between predicted and ground-truth labels. CEM is an ordinal classification evaluation metric proposed by Amigó et al. (2020), designed to account for the informational closeness between ordinal classes.

The results are shown in Table 4. In terms of accuracy, the model editing approach significantly outperforms the other methods. For the MMAE metric, editing-based approaches also achieve significantly better performance than the baselines. In terms of CEM, Model Editing with Few-shot$_{(BM25)}$ significantly outperforms the other approaches as well. Detailed statistical significance test results for this section are provided in Appendix E.

5 Discussion

We conduct a series of experiments and analyses to evaluate the effectiveness of the proposed hidden representation clustering-based sample selection method for personalized model editing tasks.

- **RQ1:** How effective is model editing for personalization across different model architectures and editing methods?
- **RQ2:** How do different sample selection strategies affect personalization performance in model editing?
- **RQ3:** How do the number of clusters and the sample selection size affect personalization performance?
- **RQ4:** What factors determine the effectiveness of model editing for personalization?

5.1 Model Editing for Personalization (RQ1)

This section evaluates the performance of LLaMA 3.1 8B Instruct and GPT-J-6B (Wang and Komatsuzaki, 2021) under three model editing methods (ROME, MEMIT, and AlphaEdit) on the personalized dataset constructed in the previous sections.

Following prior model editing studies (Fang et al., 2025), we adopt four evaluation criteria: efficacy, paraphrase, fluency, and locality. In the locality evaluation, we further use tasks from the General Language Understanding Evaluation (GLUE) benchmark (Wang et al., 2018), including SST-2 (Socher et al., 2013), MRPC (Dolan and Brockett, 2005), RTE (Bentivogli et al., 2009), and CoLA (Warstadt et al., 2019), as well as natural language inference (NLI) tasks (Williams et al., 2018). In addition, we include MMLU (Hendrycks et al., 2020) as a supplementary benchmark to assess the model's reasoning ability.

As shown in Table 5, across both LLaMA 3.1 8B Instruct and GPT-J-6B, MEMIT and AlphaEdit consistently outperform ROME on most metrics, including efficacy, paraphrase, and fluency. A further comparison between MEMIT and AlphaEdit reveals that the two methods achieve comparable performance, while exhibiting different advantages across model architectures. On LLaMA 3.1 8B Instruct, AlphaEdit significantly outperforms MEMIT in paraphrase robustness (paired Wilcoxon test, $p < 0.001$). On GPT-J-6B, AlphaEdit shows significant improvements over MEMIT in multiple metrics, including efficacy, fluency, and locality (paired Wilcoxon test, $p < 0.005$).

Furthermore, in the locality evaluation, the differences among methods are relatively small, indicating that model editing approaches can improve the accuracy of target knowledge while having limited impact on the model's original language and reasoning capabilities, without degrading the model's inherent performance. Detailed locality evaluation results are provided in Appendix C.

5.2 Effect of Sample Selection Strategies (RQ2)

In this section, we investigate how different sample selection strategies affect personalization performance in model editing, with a particular focus on

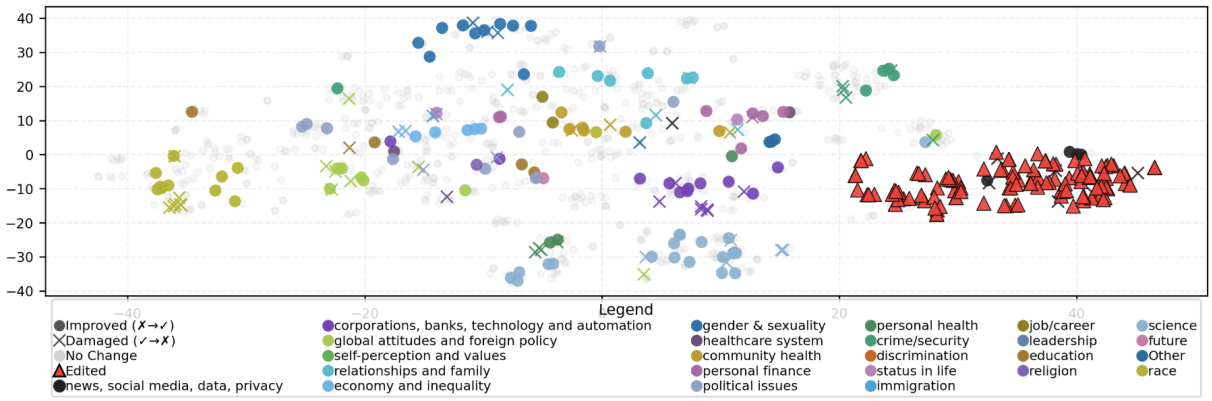

Figure 1: Semantic similarity visualization illustrating the transfer effects of model editing. Red triangles denote the edited samples used for model editing, while black markers represent other samples from the same semantic topic category. Circles indicate samples whose predictions improved after editing, whereas crosses denote samples whose predictions became incorrect after editing.

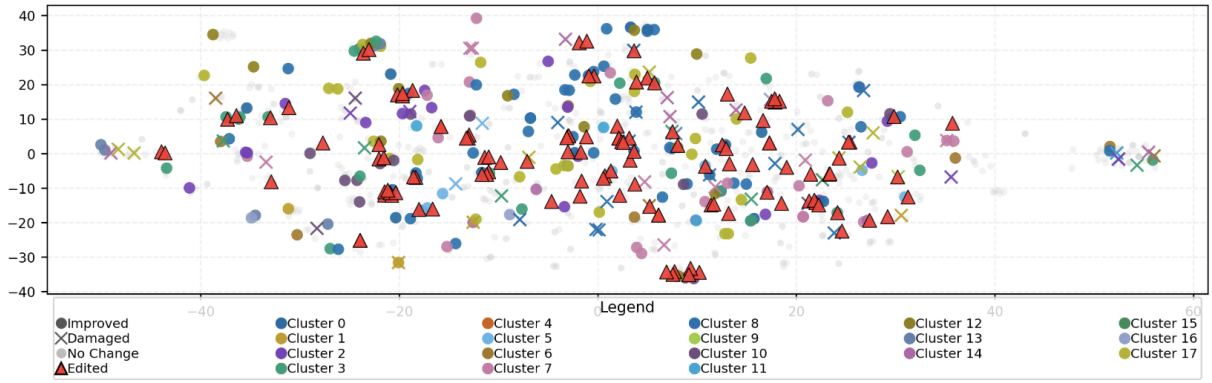

Figure 2: Hidden representation visualization illustrating the transfer effects of model editing. Red triangles denote the edited samples used for model editing. Circles indicate samples whose predictions improved after editing, whereas crosses denote samples whose predictions became incorrect after editing. Different colors represent clusters constructed from hidden representation similarity.

human-annotated topic-based selection.

We use the topic annotations provided in the original OpinionQA dataset, which categorizes questions into 23 social and political topics with varying numbers of samples per topic, and randomly select 10 respondents who participated in all 15 survey waves. For each setting, we perform model editing using 120 samples from the topic containing the largest number of questions and evaluate the edited models on both same-topic and cross-topic hold-out samples.

As shown in Table 6, editing effects vary substantially across respondents even when both editing and evaluation samples are drawn from the same topic, with both performance improvements and degradations observed. In contrast, the cross-topic setting exhibits relatively more consistent behavior. These findings suggest that human-annotated topics may not reliably capture the generalization behavior of model editing. More detailed per-respondent results are provided in Appendix D.

To further analyze this phenomenon, we examine the relationship between editing influence and the cosine similarity between edited and hold-out samples using embeddings generated by all-MiniLM-L6-v2. We further visualize the embedding distribution using t-SNE. As shown in Figure 1, there is no consistent monotonic relationship between semantic similarity and editing influence. Although some affected samples exhibit higher semantic similarity to the edited samples, samples with low similarity can also undergo substantial changes. For example, several samples from semantically distant categories, such as *race*, are also affected after editing, suggesting that editing transferability may not align well with human-annotated semantic topics. Compared with the hidden representation clustering visualization in Figure 2, these observations

Model	Method	Eff.	Para.	Flu.	Loc.
LLaMA	Vanilla	39.65	41.02	641.15	62.85
	ROME	39.28	40.83	**634.62**	62.51
	MEMIT	**89.67**	64.68	631.45	**62.95**
	AlphaEdit	89.47	**66.33**	630.93	62.88
GPT-J	Vanilla	27.18	23.92	621.41	45.45
	ROME	28.32	23.68	**618.69**	**45.53**
	MEMIT	95.75	47.30	609.33	43.57
	AlphaEdit	**97.35**	**47.83**	610.48	44.03

Table 5: Comparison of model editing methods across different base models. Locality is reported as the average performance across GLUE and MMLU benchmarks. Eff., Para., Flu., and Loc. denote efficacy, paraphrase, fluency, and locality, respectively. Bold and underlined values indicate the best and second-best performance within each model group, respectively (excluding the Vanilla baseline for Fluency ranking).

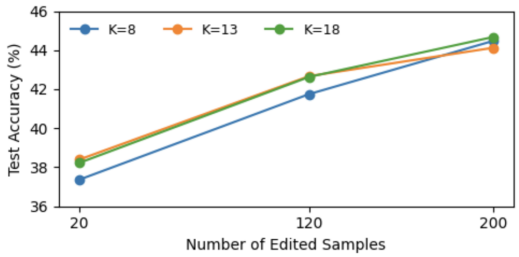

Figure 3: Test accuracy improvement under different clustering hyperparameters K and editing data sizes.

suggest that editing influence may be better characterized by hidden representation similarity than by human-annotated semantic similarity.

Motivated by this observation, we adopt hidden representation clustering as a sample selection strategy, grouping samples according to similarity in the model's internal representations to better capture editing transfer behavior.

To further assess the effectiveness of hidden representation clustering, we compare it with random sample selection using the same preference pool as in our main experiments (N=300) and an editing size of M=20. For random selection, we use two different random seeds and report the average performance. Our clustering-based selection achieves higher test accuracy than random sampling (0.3823 vs. 0.3756), and a paired significance test across respondents indicates that the improvement is statistically significant (p=0.0451).

5.3 Clustering and Sample Size Effects (RQ3)

In this section, we analyze the effects of different K values (8, 13, and 18) and editing data sizes (20, 120, and 200 samples). Figure 3 illustrates the test accuracy under different settings.

According to the results, the most influential factor affecting model editing performance is the amount of editing data. Regardless of the choice of K, performance on the test set consistently improves as the number of editing samples increases, yielding approximately 5%–7% gains overall.

However, the performance improvement begins to plateau as the editing data size increases. In particular, increasing the number of editing samples from 120 to 200 yields only marginal gains (approximately 1%–2%), suggesting that a moderate number of representative samples is sufficient to preserve most of the personalization performance.

In contrast, the choice of K does not substantially affect editing performance. Under the same editing data size, changing K from 8 to 18 results in only minor fluctuations in accuracy (mostly within 1%–2%), without showing a consistent positive or negative trend. Detailed editing-related evaluation metrics are provided in Appendix F.

Finally, we conduct an additional analysis to examine whether the effectiveness of sample selection depends on the size of the preference pool. We evaluate our method with a smaller pool of N=100, while keeping the editing size fixed at M=20. Our method achieves comparable test accuracy under the smaller pool setting (0.3770 for N=100 vs. 0.3823 for N=300), suggesting that the selection strategy remains effective when fewer user preferences are available.

5.4 Factors of Editing Effectiveness (RQ4)

In this section, we analyze the effects of model editing from two perspectives: (1) how the alignment between users and the original model influences editing performance, and (2) how the consistency of user responses within the same cluster affects model accuracy.

5.4.1 Effect of User-Model Alignment

First, we analyze how user-model alignment influences the effectiveness of model editing. As described in Section 4.1, we conduct editing experiments on the top 10, middle 10, and bottom 10 users ranked by original model accuracy. Table 7 presents the average accuracy before and after editing for each group.

The results show that all three groups achieve noticeable improvements after model editing. Among them, the Low Accuracy group exhibits the largest improvement, while the High Accuracy group

Evaluation Set	Mean Δ Acc.	Std	Min Δ	Max Δ	# Improved	# Degraded	# Unchanged
Edit set	+49.58	10.09	+30.83	+61.67	10	0	0
Same-topic hold-out	+1.60	11.07	-11.76	+22.58	3	4	3
Other-topic hold-out	+3.54	2.85	-1.32	+7.79	9	1	0

Table 6: Editing influence across different hold-out settings under topic-based sample selection. Mean, standard deviation, minimum, and maximum Δ Acc. are computed across 10 respondents. # Improved, # Degraded, and # Unchanged indicate the number of respondents (N=10).

Group	Before	After	Improvement
Low Accuracy	0.280	0.401	+0.121
Mid Accuracy	0.335	0.447	+0.113
High Accuracy	0.387	0.492	+0.105

Table 7: Average editing performance before and after editing across different accuracy groups.

shows relatively smaller gains. This suggests that model editing tends to provide greater benefits when the alignment between user preferences and the original model is relatively low.

5.4.2 Effect of Intra-Cluster Consistency

In this section, we investigate whether user response consistency within hidden representation clusters influences personalization performance after model editing. For each user response, we extract token embeddings from the language model output embedding space (i.e., `lm_head`), average them to obtain the answer representation, and apply L_2 normalization. We then compute the average pairwise cosine similarity among all answer representations within each cluster as the intra-cluster answer consistency score, and compute a weighted average based on the number of questions in each cluster to obtain a respondent-level consistency score.

We further examine the relationship between this consistency score and the accuracy improvement achieved through model editing using Spearman rank correlation analysis. The results reveal a positive correlation between response consistency and personalization improvement, with statistically significant correlation on the training set ($\rho = 0.4218, p = 0.0202$) and a moderate positive trend on the test set ($\rho = 0.2984, p = 0.1090$). These results suggest that users with more consistent response patterns may benefit more from model editing personalization. Conversely, users with more heterogeneous or conflicting preferences may be more challenging to personalize, as the selected representative samples may not adequately capture the diversity of their preferences.

6 Conclusion

Recent studies have begun to explore model editing for personalization. In this paper, we investigate how sample selection strategies influence editing generalization and personalization performance. Experiments on OpinionQA show that hidden representation similarity provides a more effective basis for editing sample selection than human-annotated semantic topics, while the proposed clustering strategy reduces required editing samples while preserving most personalization performance. We further observe that retrieval-based prompt augmentation and model editing complement each other in personalization tasks, and that user-level characteristics influence editing effectiveness.

Despite these promising results, the current study is limited to a single benchmark and restricted editing settings. Future work may further investigate editing transferability across broader personalization scenarios.

Limitations

Although the experimental results demonstrate the effectiveness of the proposed sample selection strategy for personalization-oriented model editing, several limitations remain. First, our evaluation is primarily based on OpinionQA and is conducted on a subset of 30 users, with a focus on social and political topics. Moreover, personalization is formulated as a discrete option prediction task. As a result, the current evaluation setting does not fully capture more open-ended personalization scenarios, such as long-form dialogue generation, continual interaction, or dynamic shifts in user preferences over time.

Second, although hidden representation-based clustering empirically improves editing sample selection, we do not extensively compare it with other sample selection strategies. In particular, methods that cluster samples using external text embeddings could provide alternative selection signals. Comparing against a broader range of selection strate-

gies would help better assess the effectiveness and advantages of model-internal representations for representative sample selection.

Finally, our experiments are conducted on a limited set of open-weight language models. Since editing behavior may vary across model architectures and training settings, the generalizability of the current findings remains limited.

In addition, our study evaluates personalized model editing independently for each user under a controlled experimental setting and does not specify a concrete deployment architecture. In practical deployment, maintaining a separately edited model or model delta for each user may introduce additional challenges, including update frequency, edit rollback, privacy protection, and cumulative interference from repeated edits. Personalization through model editing may also introduce potential risks related to privacy leakage, behavioral manipulation, or unintended amplification of user biases if deployed without appropriate safeguards. Addressing these issues is beyond the scope of the current study and requires further investigation. Future work should further investigate editing generalization in broader personalization settings, practical deployment strategies, and safer and more controllable personalization mechanisms for real-world use.

Acknowledgments

This research was partially supported by National Science and Technology Council, Taiwan, under grants NSTC 114-2221-E-A49-057-MY3, NSTC 114-2221-E-002-070-MY3 and NSTC 115-2634-F-002-012-, and Ministry of Education (MOE) in Taiwan, under grants 115L900901.

References

Enrique Amigó, Julio Gonzalo, Stefano Mizzaro, and Jorge Carrillo-de Albornoz. 2020. An effectiveness metric for ordinal classification: Formal properties and experimental results. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 3938–3949.

Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2009. Evaluation measures for ordinal regression. In *2009 Ninth international conference on intelligent systems design and applications*, pages 283–287. IEEE.

Luisa Bentivogli, Peter Clark, Ido Dagan, and Danilo Giampiccolo. 2009. The fifth pascal recognizing textual entailment challenge. *TAC*, 7(8):1.

Sumanth Doddapaneni, Krishna Sayana, Ambarish Jash, Sukhdeep Sodhi, and Dima Kuzmin. 2024. User embedding model for personalized language prompting. In *Proceedings of the 1st Workshop on Personalization of Generative AI Systems (PERSONALIZE 2024)*, pages 124–131.

William B Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the third international workshop on paraphrasing (IWP2005)*.

Zilu Dong, Xiangqing Shen, and Rui Xia. 2025. Memitmerge: Addressing memit's key-value conflicts in same-subject batch editing for llms. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7952–7960.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. Alphaedit: Null-space constrained knowledge editing for language models. In *International Conference on Learning Representations*, volume 2025, pages 16366–16396.

Akshat Gupta, Anurag Rao, and Gopala Anumanchipalli. 2024. Model editing at scale leads to gradual and catastrophic forgetting. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15202–15232.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Baixiang Huang, Limeng Cui, Jiapeng Liu, Haoran Wang, Jiawei Xu, Zhuiyue Tan, Yutong Chen, Chen Luo, Yi Liu, and Kai Shu. 2025. Towards effective model editing for llm personalization. *arXiv preprint arXiv:2512.13676*.

Jaehyung Kim and Yiming Yang. 2025. Few-shot personalization of llms with mis-aligned responses. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 11943–11974.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Jiongnan Liu, Yutao Zhu, Shuting Wang, Xiaochi Wei, Erxue Min, Yu Lu, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. 2025. Llms+ persona-plug= personalized llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9373–9385.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *arXiv preprint arXiv:2210.07229*.

Meta. 2024. Llama 3. https://llama.meta.com/llama3/. Large language model release.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. 2021. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*.

Yilun Qiu, Tianhao Shi, Xiaoyan Zhao, Fengbin Zhu, Yang Zhang, and Fuli Feng. 2025a. Latent inter-user difference modeling for llm personalization. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 10610–10628.

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

Nir Ratner, Yoav Levine, Yonatan Belinkov, Ori Ram, Inbal Magar, Omri Abend, Ehud Karpas, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. Parallel context windows for large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6383–6402.

Shibani Santurkar, Esin Durmus, Faisal Ladhak, Cinoo Lee, Percy Liang, and Tatsunori Hashimoto. 2023. Whose opinions do language models reflect? In *International conference on machine learning*, pages 29971–30004. PMLR.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. Quantifying language models' sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting. In *International Conference on Learning Representations*, volume 2024, pages 25055–25083.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, and 1 others. 2025. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8416–8439.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP*, pages 353–355.

Ben Wang and Aran Komatsuzaki. 2021. GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model. https://github.com/kingoflolz/mesh-transformer-jax.

Alex Warstadt, Amanpreet Singh, and Samuel R Bowman. 2019. Neural network acceptability judgments. *Transactions of the Association for Computational Linguistics*, 7:625–641.

Adina Williams, Nikita Nangia, and Samuel Bowman. 2018. A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122.

Robert F Woolson. 2007. Wilcoxon signed-rank test. *Wiley encyclopedia of clinical trials*, pages 1–3.

Yijing Zhang, Dyah Adila, Changho Shin, and Frederic Sala. 2025. Personalize your llm: Fake it then align it. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7302–7316.

Yingyi Zhang, Pengyue Jia, Derong Xu, Yi Wen, Xianneng Li, Yichao Wang, Wenlin Zhang, Xiaopeng Li, Weinan Gan, Huifeng Guo, and 1 others. 2026. Personalize before retrieve: Llm-based personalized query expansion for user-centric retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 16406–16414.

A Implementation Details and Dataset Information

Our implementation is primarily based on the publicly available AlphaEdit framework. Experiments are conducted on NVIDIA A6000 and H100 GPUs using LLaMA 3.1 8B Instruct and GPT-J-6B under multiple model editing settings. The runtime of the editing pipeline on an H100 GPU is summarized in Table 8.

Unless otherwise specified, the editing hyperparameter settings follow those used in prior model editing studies.

We use the publicly released OpinionQA dataset, which is constructed from the American Trends

Table 8: Runtime of the model editing pipeline on an NVIDIA H100 GPU. The AlphaEdit projection computation is performed once and reused across editing experiments.

Component	Runtime
AlphaEdit projection (one-time)	∼25 hours
Editing 200 samples	∼2 hours

Panel (ATP) surveys conducted by Pew Research. The dataset consists of 15 survey waves covering a wide range of social and political topics, including guns, race, and other public opinion topics. Each respondent is associated with persona-related attributes such as sex, age, and education, together with coarse-grained and fine-grained topic annotations for each question.

An example question from OpinionQA is: *"How safe, if at all, would you say your local community is from crime? Would you say it is (Options: Very safe, Somewhat safe, Not too safe, Not at all safe, Refused)"*

All datasets, models, and model editing frameworks used in this work are publicly available and used under their respective licenses and intended research usage conditions. Details regarding data collection procedures and participant consent are described in the original OpinionQA dataset paper. The dataset does not include personally identifying information, and we do not collect or process any additional personal information.

B Prompt Details

Previous studies on model editing commonly represent knowledge in the form of (s, r, o) triplets, where s, r, and o denote the subject, relation, and object, respectively. This representation is adopted because prior locate-then-edit studies have shown that editing based on the hidden representation of the last token of the subject generally achieves superior performance (Meng et al., 2022a).

Therefore, in this study, we first transform each question into an (s,r,o)-style representation and identify the corresponding subject entity in each question. Specifically, we use Gemini 3 flash to automatically extract the subject for each question.

The prompt used for subject extraction is provided below.

Task: Extract a "Continuous Discriminative Subject" for Model Editing.

Question: "{question}"

Rules:

- No pronouns such as "I", "you", or "my".
- Include semantic context (e.g., gender or race) when relevant.
- The subject must be a continuous span appearing verbatim in the question.
- Avoid leading or trailing filler phrases.
- Ensure the subject is uniquely identifiable.

Examples:

- "How safe ... is your local community from crime?" → local community
- "... treated unfairly because of your gender ..." → treated unfairly because of your gender
- "Do you now smoke cigarettes?" → now smoke cigarettes

Output: `{"subject": "..."}`

Furthermore, paraphrase questions are commonly used in model editing evaluations to verify whether the model has genuinely retained the edited knowledge rather than merely memorizing a specific input formulation (Meng et al., 2022a). Therefore, we utilize Gemini 3 flash to rewrite the original questions and generate paraphrase questions, which are subsequently used as evaluation data for model editing.

The prompt used for paraphrase generation is provided below.

Task: For the provided survey question (Q) and its options (Options), generate a structured JSON object for each option.

Core Rules:

- **Variable Fields:** Only `target` and `input_attribute` should change according to the option content.
  - `target`: Must exactly match the text in the Options list.
  - `input_attribute`: A first-person self-description simulating the user. This must be written based on the target option.
- **Fixed Fields:** For the same question, `question`, `question_paraphrased`, and `implicit_question` must remain identical across all generated objects.
- **Output Format:** Output only a pure JSON list `[...]`.

Model	Method	SST	MRPC	CoLA	RTE	NLI	MMLU
LLaMA 3.1 8B	Vanilla	95.99$_{\pm 0.00}$	**69.81**$_{\pm 0.00}$	69.38$_{\pm 0.00}$	22.77$_{\pm 0.00}$	61.62$_{\pm 0.00}$	**57.52**$_{\pm 0.00}$
	ROME	**96.13**$_{\pm 1.41}$	69.52$_{\pm 1.52}$	68.75$_{\pm 0.62}$	**22.82**$_{\pm 1.61}$	**62.44**$_{\pm 1.49}$	55.42$_{\pm 1.58}$
	MEMIT	95.86$_{\pm 0.34}$	66.10$_{\pm 2.56}$	76.37$_{\pm 2.80}$	21.18$_{\pm 0.92}$	61.16$_{\pm 0.92}$	57.04$_{\pm 1.99}$
	AlphaEdit	95.79$_{\pm 0.84}$	64.42$_{\pm 1.30}$	**78.29**$_{\pm 2.09}$	21.77$_{\pm 0.97}$	61.21$_{\pm 1.66}$	55.78$_{\pm 1.78}$
GPT-J-6B	Vanilla	75.00$_{\pm 0.00}$	42.07$_{\pm 0.00}$	**38.56**$_{\pm 0.00}$	**48.33**$_{\pm 0.00}$	50.00$_{\pm 0.00}$	18.75$_{\pm 0.00}$
	ROME	75.08$_{\pm 0.28}$	**43.04**$_{\pm 0.86}$	**38.56**$_{\pm 0.00}$	47.48$_{\pm 0.00}$	49.30$_{\pm 0.46}$	**19.74**$_{\pm 0.55}$
	MEMIT	**75.49**$_{\pm 0.64}$	35.84$_{\pm 3.74}$	36.26$_{\pm 1.18}$	43.97$_{\pm 2.71}$	**51.37**$_{\pm 1.86}$	18.51$_{\pm 1.33}$
	AlphaEdit	75.03$_{\pm 0.96}$	37.32$_{\pm 4.36}$	35.67$_{\pm 1.18}$	46.37$_{\pm 2.98}$	50.40$_{\pm 1.84}$	19.38$_{\pm 2.06}$

Table 9: Detailed evaluation of locality performance across individual text classification and language understanding benchmarks. Bold and underlined values indicate the best and second-best performance within each model group, respectively.

User	Edit Δ	Same-Topic Δ	Cross-Topic Δ
User 1	+0.617	+0.114	+0.042
User 2	+0.375	-0.125	+0.062
User 3	+0.567	+0.226	+0.020
User 4	+0.458	-0.067	+0.009
User 5	+0.417	-0.095	-0.016
User 6	+0.508	-0.040	+0.013
User 7	+0.625	+0.000	+0.029
User 8	+0.608	+0.111	+0.034
User 9	+0.300	+0.000	-0.021
User 10	+0.450	-0.176	+0.061

Table 10: Per-user editing performance changes on edited, same-topic, and cross-topic samples before and after editing.

JSON Field Definitions:

- `input_attribute`: A first-person user statement simulated based on the target option.
- `question`: The original survey question.
- `question_paraphrased`: A natural, conversational rephrasing of the question.
- `implicit_question`: Explore the underlying cues or concrete lived experiences that lead the respondent to arrive at this conclusion. Do not directly ask about the respondent's choice. Instead, ask about a specific situation or feeling that would allow the choice to be inferred indirectly.
- `target`: The exact option text from the list.

Question: `{question}`
Options: `{options}`

C Locality Results

This section provides detailed locality evaluation results across six benchmarks: SST, MRPC, CoLA, RTE, NLI, and MMLU. Table 9 presents the per-task performance for both LLaMA 3.1 8B Instruct and GPT-J-6B models, supplementing the summarized results in the main text.

D Same-Topic and Cross-Topic Editing Results

Table 10 reports detailed editing evaluation results on edited samples, same-topic samples, and cross-topic samples. Same-topic samples refer to questions belonging to the same topic category as the edited question, while cross-topic samples correspond to questions from different topic categories. The results are reported on a per-user basis.

E Significance Test Results

We report additional paired Wilcoxon signed-rank test (Woolson, 2007) results for user-level personalization performance across different retrieval settings and evaluation metrics. Specifically, we evaluate option accuracy, ordinal accuracy, macro-averaged MAE, and CEM under multiple personalization strategies, including model editing, persona augmentation, and retrieval-based few-shot prompting.

All statistical comparisons are conducted against the corresponding retrieval baseline. Tables 11 and 12 present the detailed significance test results for user-level option accuracy under BM25 and Contriever retrieval settings, respectively. Tables 13, 14, and 15 further report significance test results for ordinal accuracy, macro-averaged MAE, and CEM.

Overall, combining model editing with persona information and retrieval-based few-shot examples consistently achieves the strongest improvements across most evaluation metrics. The improvements are particularly notable for ordinal accuracy and macro-averaged MAE, where the combined setting demonstrates statistically significant gains over both BM25 and Contriever retrieval baselines.

We additionally compare each retrieval-based personalization baseline with its corresponding

Compared Method	Baseline Mean	Method Mean	Improvement	Wilcoxon p-value
PersonaEdit	0.4464	0.4467	+0.0003	0.940
PersonaEdit + persona	0.4464	0.4537	+0.0073	0.280
PersonaEdit + Few-shot$_{\text{BM25}}$	0.4464	0.4575	+0.0111	0.058
PersonaEdit + All info$_{\text{BM25}}$	0.4464	0.4702	+0.0238	0.00013

Table 11: User-level personalization accuracy across methods using BM25 retrieval. Baseline Mean and Method Mean represent the mean user-level accuracy averaged across respondents. All p-values are computed against the All info$_{\text{BM25}}$ baseline using paired Wilcoxon signed-rank tests.

Compared Method	Baseline Mean	Method Mean	Improvement	Wilcoxon p-value
PersonaEdit	0.4451	0.4467	+0.0016	0.657566
PersonaEdit + persona	0.4451	0.4537	+0.0086	0.036921
PersonaEdit + Few-shot$_{\text{BM25}}$	0.4451	0.4575	+0.0124	0.017897
PersonaEdit + All info$_{\text{BM25}}$	0.4451	0.4702	+0.0251	0.00000514

Table 12: User-level personalization accuracy across methods using Contriever retrieval. Baseline Mean and Method Mean represent the mean user-level accuracy averaged across respondents. All p-values are computed against the All info$_{\text{contriever}}$ baseline using paired Wilcoxon signed-rank tests.

model editing variant to evaluate whether model editing consistently improves personalization performance. Table 16 presents the paired Wilcoxon signed-rank test results for user-level option accuracy between retrieval-based personalization methods and their corresponding editing-enhanced variants.

Across all settings, integrating model editing leads to statistically significant improvements over the corresponding non-editing baselines. In particular, the largest improvement is observed when applying model editing on top of the Vanilla setting, while retrieval-enhanced and persona-enhanced settings also benefit consistently from model editing. These results suggest that model editing provides complementary gains beyond retrieval-based personalization alone.

F Editing Results with Respect to k and Dataset Size

Table 17 and Table 18 present the detailed editing and locality evaluation results under different hyperparameter settings.

Compared Method	Baseline Mean	Method Mean	Improvement	Wilcoxon p-value
All $\text{info}_{\text{BM25}}$ vs PersonaEdit	0.4254	0.4288	+0.0034	0.626346
All $\text{info}_{\text{BM25}}$ vs PersonaEdit + persona	0.4254	0.4271	+0.0017	0.597808
All $\text{info}_{\text{BM25}}$ vs PersonaEdit + Few-$\text{shot}_{\text{BM25}}$	0.4254	0.4368	+0.0114	0.145999
All $\text{info}_{\text{BM25}}$ vs PersonaEdit + All $\text{info}_{\text{BM25}}$	0.4254	0.4485	+0.0231	0.000555

Table 13: User-level ordinal accuracy comparison across methods. Baseline Mean and Method Mean represent the mean user-level accuracy averaged across respondents. All p-values are computed using paired Wilcoxon signed-rank tests against the corresponding retrieval baseline.

Compared Method	Baseline Mean	Method Mean	Improvement	Wilcoxon p-value
All $\text{info}_{\text{BM25}}$ vs PersonaEdit	0.2936	0.2831	+0.0105	0.073244
All $\text{info}_{\text{BM25}}$ vs PersonaEdit + persona	0.2936	0.2802	+0.0133	0.004338
All $\text{info}_{\text{BM25}}$ vs PersonaEdit + Few-$\text{shot}_{\text{BM25}}$	0.2936	0.2834	+0.0102	0.007612
All $\text{info}_{\text{BM25}}$ vs PersonaEdit + All $\text{info}_{\text{BM25}}$	0.2936	0.2764	+0.0172	4.97×10^{-5}

Table 14: User-level macro-averaged MAE comparison across methods. Baseline Mean and Method Mean represent the mean user-level accuracy averaged across respondents. All p-values are computed using paired Wilcoxon signed-rank tests against the corresponding retrieval baseline.

Compared Method	Baseline Mean	Method Mean	Improvement	Wilcoxon p-value
All $\text{info}_{\text{BM25}}$ vs PersonaEdit	0.5350	0.5383	+0.0033	0.013663
All $\text{info}_{\text{BM25}}$ vs PersonaEdit + persona	0.5350	0.5380	+0.0030	0.027741
All $\text{info}_{\text{BM25}}$ vs PersonaEdit + Few-$\text{shot}_{\text{BM25}}$	0.5350	0.5385	+0.0035	0.000111
All $\text{info}_{\text{BM25}}$ vs PersonaEdit + All $\text{info}_{\text{BM25}}$	0.5350	0.5371	+0.0021	0.004665

Table 15: User-level CEM comparison across methods. Baseline Mean and Method Mean represent the mean user-level accuracy averaged across respondents. All p-values are computed using paired Wilcoxon signed-rank tests against the corresponding retrieval baseline.

Compared Method	Baseline Mean	Method Mean	Improvement	Wilcoxon p-value
Vanilla vs PersonaEdit	0.3337	0.4467	+0.1131	1.73×10^{-6}
Few-$\text{shot}_{\text{BM25}}$ vs PersonaEdit + Few-$\text{shot}_{\text{BM25}}$	0.4304	0.4575	+0.0272	2.35×10^{-6}
Persona vs PersonaEdit + persona	0.4141	0.4537	+0.0396	5.14×10^{-6}
All $\text{info}_{\text{BM25}}$ vs PersonaEdit + All $\text{info}_{\text{BM25}}$	0.4464	0.4702	+0.0238	0.00013

Table 16: User-level option accuracy comparison between retrieval-based personalization baselines and their corresponding model editing variants. All p-values are computed using paired Wilcoxon signed-rank tests.

	20		**120**		**200**	
Split	Pre-edit	Edited	Pre-edit	Edited	Pre-edit	Edited
			K = 8			
Train	30.67%	85.67%	34.67%	88.75%	34.22%	89.45%
Test	33.37%	37.37%	33.37%	41.75%	33.37%	44.47%
			K = 13			
Train	28.83%	92.83%	33.11%	88.50%	34.57%	89.07%
Test	33.37%	38.40%	33.37%	42.67%	33.37%	44.12%
			K = 18			
Train	37.00%	86.50%	31.33%	88.61%	34.37%	88.37%
Test	33.37%	38.23%	33.37%	42.63%	33.37%	44.67%

Table 17: Analysis of the clustering hyperparameter K and the scale of edited samples (20, 120, and 200) in terms of accuracy.

Setting	Method	Efficacy (↑)	Paraphrase (↑)	Fluency (↑)	Locality (↑)					
					SST	MRPC	CoLA	RTE	NLI	MMLU
$k = 8, s = 20$	Vanilla	$40.17_{\pm 13.57}$	$42.33_{\pm 12.43}$	$638.76_{\pm 0.00}$	$95.99_{\pm 0.00}$	$69.81_{\pm 0.00}$	$69.38_{\pm 0.00}$	$22.77_{\pm 0.00}$	$61.62_{\pm 0.00}$	$57.52_{\pm 0.00}$
	AlphaEdit	$90.33_{\pm 7.52}$	$62.00_{\pm 11.15}$	$630.02_{\pm 2.95}$	$96.13_{\pm 0.43}$	$65.51_{\pm 1.91}$	$68.83_{\pm 1.02}$	$20.72_{\pm 0.66}$	$62.03_{\pm 0.69}$	$55.57_{\pm 1.41}$
$k = 8, s = 120$	Vanilla	$39.56_{\pm 6.95}$	$42.50_{\pm 7.39}$	$640.94_{\pm 0.00}$	$95.99_{\pm 0.00}$	$69.81_{\pm 0.00}$	$69.38_{\pm 0.00}$	$22.77_{\pm 0.00}$	$61.62_{\pm 0.00}$	$57.52_{\pm 0.00}$
	AlphaEdit	$89.56_{\pm 2.96}$	$64.25_{\pm 5.91}$	$632.24_{\pm 1.08}$	$96.29_{\pm 0.69}$	$63.54_{\pm 1.81}$	$78.13_{\pm 2.35}$	$20.97_{\pm 0.91}$	$60.38_{\pm 1.27}$	$54.54_{\pm 2.06}$
$k = 8, s = 200$	Vanilla	$39.88_{\pm 5.94}$	$40.32_{\pm 6.67}$	$640.98_{\pm 0.00}$	$95.99_{\pm 0.00}$	$69.81_{\pm 0.00}$	$69.38_{\pm 0.00}$	$22.77_{\pm 0.00}$	$61.62_{\pm 0.00}$	$57.52_{\pm 0.00}$
	AlphaEdit	$89.18_{\pm 1.99}$	$63.77_{\pm 5.14}$	$631.02_{\pm 1.08}$	$95.69_{\pm 0.64}$	$64.20_{\pm 1.66}$	$78.81_{\pm 2.27}$	$21.44_{\pm 1.05}$	$60.97_{\pm 1.46}$	$55.75_{\pm 1.52}$
$k = 13, s = 20$	Vanilla	$40.33_{\pm 12.78}$	$36.17_{\pm 12.82}$	$636.46_{\pm 0.00}$	$95.99_{\pm 0.00}$	$69.81_{\pm 0.00}$	$69.38_{\pm 0.00}$	$22.77_{\pm 0.00}$	$61.62_{\pm 0.00}$	$57.52_{\pm 0.00}$
	AlphaEdit	$94.17_{\pm 4.67}$	$62.67_{\pm 10.70}$	$632.68_{\pm 2.77}$	$96.06_{\pm 0.25}$	$65.12_{\pm 2.10}$	$68.96_{\pm 1.52}$	$20.98_{\pm 0.73}$	$62.06_{\pm 0.92}$	$55.73_{\pm 1.24}$
$k = 13, s = 120$	Vanilla	$37.56_{\pm 6.09}$	$38.97_{\pm 6.07}$	$642.59_{\pm 0.00}$	$95.99_{\pm 0.00}$	$69.81_{\pm 0.00}$	$69.38_{\pm 0.00}$	$22.77_{\pm 0.00}$	$61.62_{\pm 0.00}$	$57.52_{\pm 0.00}$
	AlphaEdit	$88.89_{\pm 3.01}$	$63.39_{\pm 5.65}$	$632.28_{\pm 1.38}$	$96.43_{\pm 0.62}$	$64.11_{\pm 1.49}$	$77.30_{\pm 2.80}$	$21.03_{\pm 1.14}$	$61.64_{\pm 1.32}$	$53.96_{\pm 1.66}$
$k = 13, s = 200$	Vanilla	$39.65_{\pm 5.77}$	$41.02_{\pm 6.25}$	$641.15_{\pm 0.00}$	$95.99_{\pm 0.00}$	$69.81_{\pm 0.00}$	$69.38_{\pm 0.00}$	$22.77_{\pm 0.00}$	$61.62_{\pm 0.00}$	$57.52_{\pm 0.00}$
	AlphaEdit	$89.47_{\pm 2.61}$	$66.33_{\pm 4.74}$	$630.93_{\pm 1.19}$	$95.79_{\pm 0.84}$	$64.42_{\pm 1.30}$	$78.29_{\pm 2.09}$	$21.77_{\pm 0.97}$	$61.21_{\pm 1.66}$	$55.78_{\pm 1.78}$
$k = 18, s = 20$	Vanilla	$46.00_{\pm 12.94}$	$36.50_{\pm 11.77}$	$647.53_{\pm 0.00}$	$95.99_{\pm 0.00}$	$69.81_{\pm 0.00}$	$69.38_{\pm 0.00}$	$22.77_{\pm 0.00}$	$61.62_{\pm 0.00}$	$57.52_{\pm 0.00}$
	AlphaEdit	$88.50_{\pm 6.21}$	$67.17_{\pm 9.46}$	$634.54_{\pm 2.21}$	$96.06_{\pm 0.36}$	$65.24_{\pm 1.75}$	$70.77_{\pm 1.72}$	$20.86_{\pm 0.83}$	$61.77_{\pm 1.04}$	$56.39_{\pm 1.03}$
$k = 18, s = 120$	Vanilla	$36.78_{\pm 6.62}$	$39.31_{\pm 7.08}$	$641.39_{\pm 0.00}$	$95.99_{\pm 0.00}$	$69.81_{\pm 0.00}$	$69.38_{\pm 0.00}$	$22.77_{\pm 0.00}$	$61.62_{\pm 0.00}$	$57.52_{\pm 0.00}$
	AlphaEdit	$89.14_{\pm 2.49}$	$63.22_{\pm 6.06}$	$632.32_{\pm 1.18}$	$96.60_{\pm 0.56}$	$63.16_{\pm 1.66}$	$77.43_{\pm 2.85}$	$21.39_{\pm 0.92}$	$61.43_{\pm 1.41}$	$54.21_{\pm 1.87}$
$k = 18, s = 200$	Vanilla	$39.18_{\pm 5.64}$	$39.95_{\pm 6.18}$	$641.11_{\pm 0.00}$	$95.99_{\pm 0.00}$	$69.81_{\pm 0.00}$	$69.38_{\pm 0.00}$	$22.77_{\pm 0.00}$	$61.62_{\pm 0.00}$	$57.52_{\pm 0.00}$
	AlphaEdit	$88.72_{\pm 2.50}$	$64.55_{\pm 4.72}$	$630.20_{\pm 1.21}$	$95.56_{\pm 0.85}$	$64.12_{\pm 1.83}$	$78.74_{\pm 2.03}$	$22.07_{\pm 0.91}$	$61.49_{\pm 1.86}$	$56.36_{\pm 2.30}$

Table 18: Editing results of AlphaEdit under various k and s hyperparameter configurations on the LLaMA 3.1 8B Instruct model across efficacy, generalization, fluency, and individual locality tasks.