Towards Effective Model Editing for LLM Personalization

Anonymous ACL submission

Abstract

Personalization is becoming indispensable for LLMs to align with individual user preferences and needs. Yet current approaches are often computationally expensive, data-intensive, susceptible to catastrophic forgetting, and prone to performance degradation in multi-turn interactions or when handling implicit queries. To address these challenges, we conceptualize personalization as a model editing task and introduce ***Personalization Editing***, a framework that applies localized edits guided by clustered preference representations. This design enables precise preference-aligned updates while preserving overall model capabilities. In addition, existing personalization benchmarks frequently rely on persona-based dialogs between LLMs rather than user-LLM interactions, or focus primarily on stylistic imitation while neglecting information-seeking tasks that require accurate recall of user-specific preferences. We introduce User Preference Question Answering (**UPQA**), a short-answer QA dataset constructed from in-situ user queries with varying levels of difficulty. Unlike prior benchmarks, UPQA directly evaluates a model's ability to recall and apply specific user preferences. Across experimental settings, Personalization Editing achieves higher editing accuracy and greater computational efficiency than fine-tuning, while outperforming prompting-based baselines in multi-turn conversations and implicit preference questions settings. [1]

1 Introduction

Large language models (LLMs) have demonstrated strong general-purpose capabilities, yet there is growing demand to tailor their behavior to individual users (Salemi et al., 2023). Personalization adjusts model outputs based on user-specific preferences, goals, and contextual signals derived from interaction history, thereby improving relevance and user satisfaction. For example, if a user's hobbies include running and reading, a personalized LLM responding to "Suggest some activities for the weekend" should recommend a local trail run or a nearby book club rather than generic options.

Despite its promise, effective personalization remains challenging. Fine-tuning–based methods require domain-specific data, are computationally expensive, and prone to catastrophic forgetting, which can lead to poor user experience and erode user trust (Laban et al., 2025; Zhang et al., 2024b). In-context methods such as prompt engineering avoid retraining but degrade substantially in **multi-turn conversations**, where relevant information becomes diluted in long prompts, causing the models to become increasingly unreliable (Bai et al., 2024). Moreover, these methods often struggle with **implicit-preference queries** that require reasoning beyond the explicit profile facts (Zhao et al., 2025). These limitations motivate more efficient and reliable personalization strategies.

Model editing, also known as knowledge editing, offers a parameter-efficient alternative by enabling localized changes to model behavior using minimal data and computation (Wang et al., 2024). We conceptualize personalization as a model-editing task: each user preference corresponds to a targeted update that overrides the original model's behavior while preserving unrelated behaviors. This approach avoids the costs of full fine-tuning helps mitigate forgetting commonly observed in multi-turn settings (Zhao et al., 2025).

We propose *Personalization Editing*, a framework that further strengthens editing-based personalization by incorporating clustering-based preference representations. Instead of binding each preference to a single fixed response, we represent preferences as clusters of semantically similar subjects and target responses. This richer representation enhances the model's robustness to varied

[1] Code, data, and additional resources are available at https://anonymous.4open.science/r/personalization-model-editing

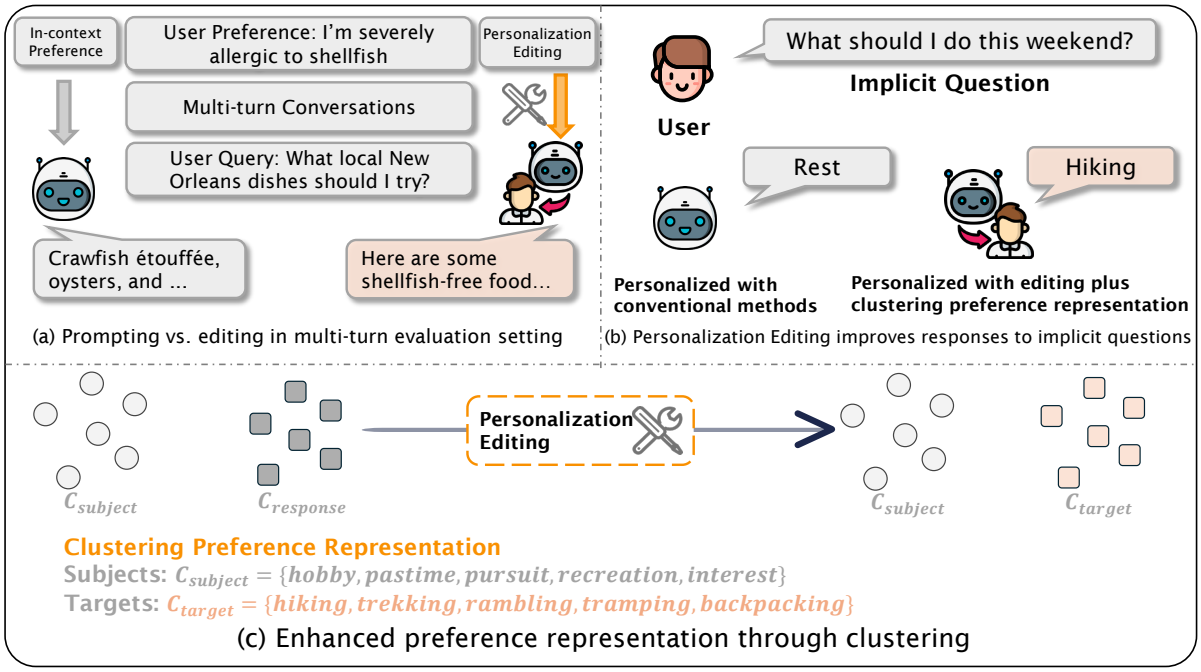

Figure 1: The proposed Personalization Editing framework outperforms prompting-based methods in multi-turn conversations and surpasses fine-tuning-based approaches in both accuracy and efficiency. Furthermore, its enhanced clustering-based preference representation enables the framework to recall user preferences even in challenging implicit queries, where existing methods often fail.

phrasing and contextual cues, enabling it to recognize when a preference applies even if it is not mentioned explicitly.

Model editing offers two central advantages for personalization. First, because editing modifies model parameters directly, it ensures consistent enforcement of user preferences across multi-turn conversations. For example, in Figure 1 (a), an in-context approach incorrectly recommends "crawfish étouffée and oysters" despite a declared shellfish allergy, whereas an edited model maintains shellfish-free recommendations throughout the dialog. Second, building on this stable editing foundation, our clustering-based preference representations enable accurate responses to implicit questions. As shown in Figure 1 (b), the model can correctly infer that a user who enjoys hiking would appreciate a weekend hiking suggestion, even when the preference is not explicitly restated.

A parallel challenge lies in evaluation. Existing personalization datasets focus on synthetic persona-based dialog between LLMs rather than realistic human-LLM interactions (Jandaghi et al., 2023). Other personalization benchmarks primarily emphasize content-generation tasks such as style-based email writing while overlooking information-seeking tasks that require accurate recall grounded in user-specific facts (Salemi et al., 2023). To address this gap, we introduce **UPQA** (User Preference Question Answering), a benchmark that evaluates whether models recall and apply user profile facts. UPQA includes structured questions for explicit preferences, implicit preferences, and practical scenarios such as product recommendations. By adopting a short-answer QA format with varying levels of difficulty, UPQA enables efficient and reliable assessment of personalization methods.

Through comprehensive experiments involving popular open-weight LLMs and diverse editing methodologies, our findings validate that the proposed framework improves editing accuracy, robustness in multi-turn conversations, and computational efficiency. Our key contributions are summarized as follows:

- **Model Editing for LLM Personalization.** We propose a novel conceptualization of personalization as a model editing problem, enabling precise, parameter-efficient updates that preserve general capabilities and outperform prompting-based techniques in multi-turn conversations.
- **Clustering-Based Preference Representation.** We design a clustering-based preference representation that augments existing model editing techniques, leading to superior robustness and accuracy when handling challenging implicit pref-

erence questions across diverse domains.

- **The UPQA Benchmark.** We introduce UPQA, a challenging dataset designed for the rigorous and standardized evaluation of personalization methods across diverse scenarios, specifically focusing on the accurate recall and application of user-specific facts.

2 Related Work

In this section, we review the areas relevant to our work: LLM Personalization and Model Editing.

2.1 LLM Personalization

LLM personalization adapts models to individual user preferences, enhancing satisfaction through more relevant interactions (Zhang et al., 2024b). In-context approaches, such as profile-augmented prompts (injecting a user's profile into the prompt) (Zhang et al., 2018) or RAG (Fan et al., 2024), which fetches user-specific information from an external memory, incorporate user data into the model's input without altering the model's weights. However, RAG depends on large, high-quality datasets, which are not consistently available to all users, particularly given the diverse and rapidly changing nature of user preferences (Zhang et al., 2025). Moreover, compressing extensive user history into a prompt can lead to information loss and is constrained by the model's limited context window (Liu et al., 2025). Fine-tuning approaches update the model's parameters on user-specific data. This includes training a personalized adapter within the model's layers (Zhong et al., 2021) or using reinforcement learning to align the model with preferences (Ouyang et al., 2022). However, these methods are often resource-intensive and prone to overfitting (Liu et al., 2025). Moreover, reward-alignment techniques such as (Bai et al., 2022; Rafailov et al., 2023) primarily optimize for global preferences, rather than adapting to individual, user-specific feedback preferences (Liu et al., 2025).

2.2 Model Editing

Model editing, also known as knowledge editing, enables efficient and precise modification in LLMs without full-parameter retraining while largely preserving non-targeted capabilities (Wang et al., 2024). Various editing techniques have been developed to update model knowledge efficiently. Some methods, such as ROME (Meng et al., 2022) and its multi-edit successor MEMIT (Meng et al., 2023), operate by directly locating and manipulating key factual associations within the model's internal representations. Other parameter-efficient techniques include Fine-Tuning with Masking (FT-M) (Rozner et al., 2024; Gangadhar and Stratos, 2024), Constrained fine-tuning (FT-L) (Meng et al., 2022), and LoRA (Hu et al., 2022). A key strategy for these PEFT-based editing methods is to first identify the most relevant layer, often a specific Feed-Forward Network layer, responsible for the target knowledge using diagnostic techniques like causal tracing, and then apply the minimal update only to that specific location. These techniques are are effective for updating factual knowledge, reducing hallucinations, and controlling ethical and safety behaviors without compromising the model's general capabilities (Chen et al., 2024; Huang et al., 2025a,b). Our proposed clustering-based preference representation method is designed to augment these existing model editing techniques, improving their performance on challenging implicit preference questions.

3 Problem Formulation

This section formally defines Personalization Editing as an objective to precisely map user preferences to model parameter updates while strictly preserving its general capabilities.

3.1 Personalization as Editing

Personalization Editing operates on a structure analogous to a knowledge tuple (s, r, o), where s represents a subject, r denotes a predicate, and o represents an object. The process of modifying model responses to align with specific user preferences is formalized as transforming an original tuple (s, r, o) into a new tuple (s, r, o^*) that reflects the personalized preference, where o^* represents the target response. Here, the user and predicate remain constant while the response adapts to user-specific preferences.

3.2 Input–Output Mapping

To probe and modify model responses for personalization, the subject s must be converted into a natural language question x, to which the model responds with an output y. This input-output pair is associated with a tuple (s, r, o). The input space corresponding to a personalization edit is denoted as $\mathcal{X}_e = I(s, r)$, where I, where I is a question-generation function that maps the subject and relation to a set of relevant input questions. The origi-

nal output space is defined as $\mathcal{Y}_e = O(s, r, o)$, and the desired personalized output space after editing is represented as $\mathcal{Y}_e^* = O^*(s, r, o^*)$. For a single edit e with input space $\mathcal{X}_e$, the objective of Personalization Editing is to transform the original output $\mathcal{Y}_e$ into the target output $\mathcal{Y}_e^*$.

When considering a set of personalization edits $\mathcal{E} = \{e_1, e_2, \ldots\}$, the combined input space is $\mathcal{X}_\mathcal{E} = \bigcup_{e \in \mathcal{E}} \mathcal{X}_e$, and the corresponding original and target output spaces are $\mathcal{Y}_\mathcal{E} = \bigcup_{e \in \mathcal{E}} \mathcal{Y}_e$ and $\mathcal{Y}_\mathcal{E}^* = \bigcup_{e \in \mathcal{E}} \mathcal{Y}_e^*$, respectively.

3.3 Objective

Let the original LLM be a function $f : \mathcal{X} \to \mathcal{Y}$. The goal of Personalization Editing is to produce a personalized model $f^* : \mathcal{X} \to \mathcal{Y}^*$, such that the edited model generates personalized outputs for inputs in $\mathcal{X}_\mathcal{E}$ while preserving its responses on all other inputs, preventing degradation of unrelated model behavior. The optimization aims to minimize the discrepancy between the personalized output $f^*(x)$ and the desired target output y^*, as measured by a loss function $\mathcal{L}$. Simultaneously, the editing must maintain consistency on all inputs outside the editing set, ensuring that $f^*(x) = f(x)$ for all $x \in \mathcal{X} \setminus \mathcal{X}_\mathcal{E}$. This yields the constrained optimization objective:

$$\begin{aligned} &\min \mathbb{E}_{e \in \mathcal{E}} \mathbb{E}_{x, y^* \in \mathcal{X}_e, \mathcal{Y}_e^*} \mathcal{L}(f^*(x), y^*) \\ &\text{s.t. } f^*(x) = f(x), \quad \forall x \in \mathcal{X} \setminus \mathcal{X}_\mathcal{E} \end{aligned}$$

4 Dataset Construction

To rigorously evaluate personalization editing. We first introduce UPQA, a short-answer QA benchmark built from in-situ user queries, specifically designed for the standardized and efficient evaluation of personalization editing. Second, we adapt PREFEVAL (Zhao et al., 2025), a multi-turn conversation benchmark, to better align with model editing. Together, these datasets provide a diverse and challenging testbed for assessing both the accuracy and robustness of personalization methods.

4.1 UPQA (User Preference Question Answering)

We curated **UPQA** by extracting user-profile features from the *Synthetic Persona Chat* (Jandaghi et al., 2023). We first aggregated all unique persona attributes, where each attribute encodes a specific user preference (e.g., "I enjoy hiking," "I have a dog," "I work as a teacher"). These serve as the foundation for personalization evaluation.

To transform persona attributes to structured evaluation queries, we employed `Claude-Sonnet-4`, selected for its strong performance on instruction-following benchmarks (Sharma et al., 2025). The model was prompted to analyze each persona attribute and generate a suite of *in-situ* user queries at different levels of difficulty. This process ensured that evaluation questions are systematically varied.

Since non-technical users often struggle to clearly articulate their intent, leading to underspecified or ineffective prompts (Bo et al., 2025), we design implicit questions as a more challenging variation of the original question. For each user preference, we also annotated an `attribute_type`, a high-level category of personal information such as *hobby, profession, family, pet, or location*. We designed four complementary query types:

1. `question`: Direct queries about the attribute using its exact term (e.g., "What's my hobby?" for a hiking-related persona).
2. `question_paraphrased`: Rephrasings of the direct query in more natural language.
3. `implicit_question`: Indirect queries that rely on prior knowledge (e.g., "What should I do this weekend?" for a hiking hobby).
4. `product_recommendation_question`: Preference-grounded recommendation requests that omit explicit attribute values (e.g., "Any gear I should buy for my hobby?").

We collected over 1,000 unique user preferences spanning a broad range of topics, including hobbies, family roles, professions, and personal attributes. To ensure fairness in evaluation, we sampled a balanced subset with equal representation across attribute types. This prevents bias toward overrepresented categories and supports standardized comparison across methods.

To support cluster-based editing, we augmented UPQA by generating clusters of nine semantically related synonyms for each editing subject and target. This augmentation captures lexical variability (e.g., "dog," "puppy," "canine") while preserving semantic meaning, ensuring robustness in both evaluation and model editing. Synonym clusters were again generated with `Claude-Sonnet-4`, followed by manual verification.

Below is an example from UPQA. More details, including attribute annotations, question types, and synonym clusters, are provided in Appendix E.

```
{
    "attribute_type": "food preference",
    "question": "What's my food preference?",
    "question_paraphrased": "What kind of food do I like best?",
    "implicit_question": "What should I have for dessert tonight?",
    "target": "Any products I should buy for my food preference?",
    "input_attribute": "My favorite food is ice cream."
}
```

4.2 PrefEval

In addition to UPQA, we also evaluate our framework on the PREFEVAL benchmark (Zhao et al., 2025), where user preferences are expressed directly in single-turn utterances, and subsequent queries test whether models can recall and apply these preferences after long, noisy contexts. PREFEVAL serves as a baseline for assessing preference following; however, it is primarily designed for prompting and retrieval-based methods rather than model editing.

To adapt PREFEVAL for our setting, we reformulate the benchmark into structured key-value pairs by extracting `subject` and `target`. This restructuring isolates the core preference signal and enables precise updates, facilitating efficient preference injection or correction without retraining on entire conversations. The augmentation details are provided in Appendix C. An example of the augmented data is given below.

```
{
    "topic": "travel_restaurant",
    "preference": "I have a severe peanut allergy, so I must avoid any foods containing peanuts or peanut products.",
    "question": "I'm visiting Thailand next month. What are some authentic Thai restaurants you would recommend?",
    "explanation": "Thai cuisine commonly utilizes peanuts and peanut-based sauces, so recommending authentic Thai restaurants presents a challenge given the user's peanut allergy.",
    "subject": "restaurants",
    "target": "peanut-free"
}
```

5 Experiments

5.1 Baseline Methods

- FT-L (Meng et al., 2022) Constrained fine-tuning that targets a specific FFN layer identified by causal tracing, maximizing likelihood of target sequences with parameter-space norm constraints to minimize interference with unmodified facts.
- FT-M (Zhang et al., 2024a) Fine-tuning with masking that uses cross-entropy loss on target answers while masking original text, providing more precise weight adjustments aligned with traditional fine-tuning objectives.
- LoRA (Hu et al., 2022) Low-rank adaptation that introduces trainable rank decomposition matrices into Transformer layers, freezing pretrained weights while optimizing low-rank matrices for parameter-efficient fine-tuning.
- ROME (Meng et al., 2022) Rank-one model editing that localizes factual associations in MLP modules through causal intervention then makes targeted rank-one parameter changes to alter factual associations with minimal disruption.
- GRACE (Hartvigsen et al., 2023) Sequential editing method that introduces layer adaptors with cached embeddings and codebook storage, enabling sequential edits while maintaining model stability through a deferral mechanism.
- Zero-shot (Zhao et al., 2025; Zheng et al., 2023) Zero-shot prompting that directly incorporates user preferences into the input context before presenting evaluation questions.

5.2 Evaluation

After constructing the UPQA benchmark, we design an evaluation pipeline to assess the effectiveness of model editing methods for personalization. Our evaluation primarily follows the established model editing paradigm and uses the *Efficacy Score* (%) as the main metric. This score measures whether the edited model can generate target answers that accurately reflect user preferences, and is equivalent to the success rate. To further examine whether a personalized LLM can robustly provide preference-aware responses across diverse question types, we introduce the *Generalization Score* (%), which evaluates the model's ability to handle paraphrased or implicit questions related to the same user preference. This metric captures the percentage of personalized responses produced under more challenging conditions.

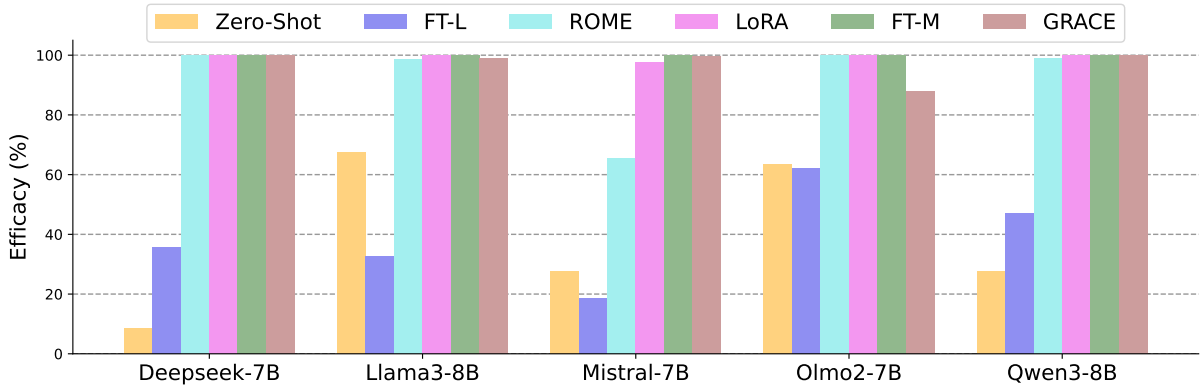

Figure 2: Efficacy score of various model editing methods across multiple LLMs on UPQA.

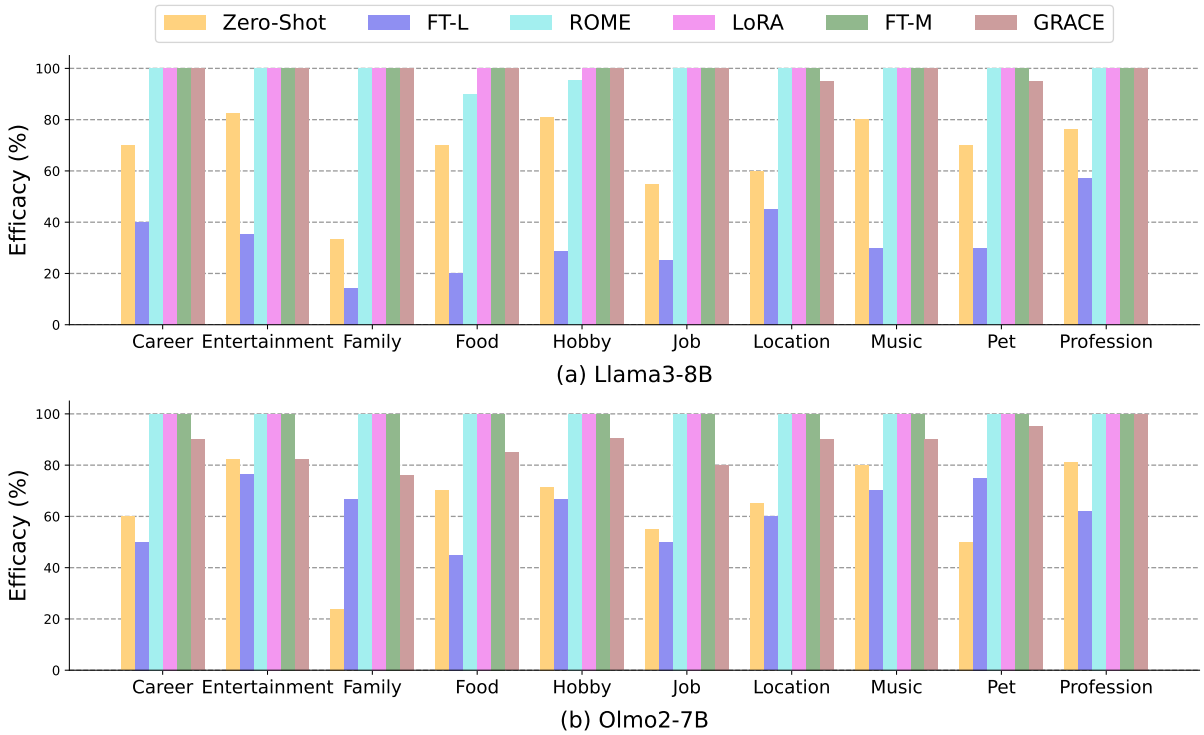

Figure 3: Evaluation results of various model-editing methods for Llama3-8B and Olmo2-7B on UPQA across 10 preference types. Efficacy scores (%) indicate the editing success rate on question–answer pairs.

For multi-turn conversation settings, we insert inter-turn dialog as distractions before the evaluation question. Following PrefEval (Zhao et al., 2025), we retrieve these inter-turn conversational turns from the Lmsys1M dataset (Zheng et al., 2024). However, unlike PrefEval, which is designed to evaluate prompting-based methods and thus explicitly inserts user preferences into the context (structured as user preference followed by inter-turn conversation and then the evaluation question), our evaluation does not include user preference information in the context. This design more accurately reflects the personalization editing setting, where user knowledge is embedded in the model itself rather than reintroduced through prompts.

We employ `Claude-Sonnet-4` as the automatic judge to assess whether model responses acknowledge, reflect, or demonstrate awareness of user preferences, following prior work that defines this metric as the *Acknowledge Rate* (%) (Zhao et al., 2025). The detailed evaluation prompts used for evaluation are provided in Appendix F.

5.3 Effectiveness of Personalization Editing

We first evaluate the effectiveness of Personalization Editing on the proposed UPQA data. Figure 2 and 3 show that Personalization Editing consistently achieves higher Efficacy Scores across all

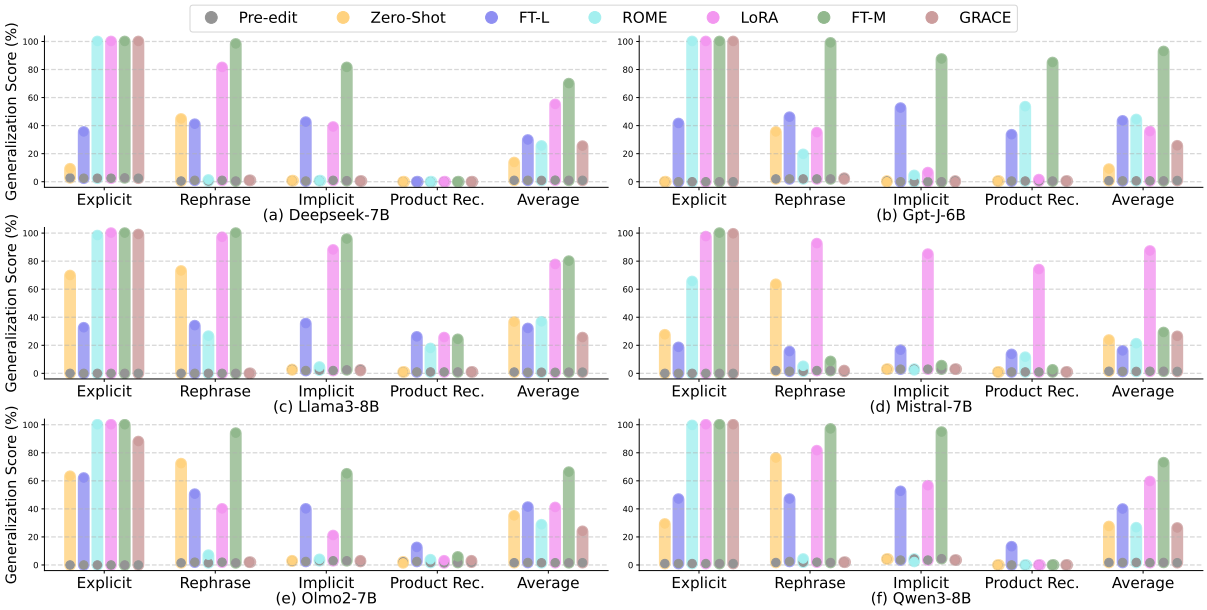

Figure 4: Generalization scores of various model editing methods across multiple LLMs on UPQA. Generalization Scores (%) are measured by accuracy on four types of Generalization Evaluation Questions including Questions ("Explicit"), Rephrased Questions ("rephrase"), Implicit Questions ("implicit"), Product-Recommendation Questions ("Product Rec."). The "Average" refers to averaged scores over four question types.

preference types, demonstrating its ability to robustly encode user-specific information. Moreover, Figure 4 highlights that Personalization Editing generalizes effectively across six different models.

While ROME exhibits strong efficacy on direct preference injection, it fails to generalize to rephrased questions, implicit references, and recommendation-style queries. In contrast, zero-shot prompting preserves some ability on rephrased questions but lags far behind editing-based methods in efficacy, underscoring that persistent and reliable personalization requires direct parameter updates rather than transient prompting. FT-M achieves competitive performance in generalization.

Finding 1

Personalization Editing is highly effective at encoding user-specific facts into LLMs, enabling them to provide personalized responses aligned with user preferences.

5.4 Sustaining Personalization Beyond the First Turn

To evaluate whether personalization persists across extended multi-tun interactions, we measure the *Acknowledgment Rate* in multi-turn dialogs on PREFEVAL (Zhao et al., 2025). As shown in Figure 5, Personalization Editing maintains a high acknowledgment of user preferences throughout 10 conversational turns, demonstrating robustness even as unrelated dialog content introduces distractions. In contrast, prompting-based methods degrade rapidly, falling below 20% by the 8th turn, as models fail to recall preferences without repeated explicit reminders. This gap highlights a key advantage of parameter-based editing: by modifying internal representations, the injected personalization becomes persistent and less susceptible to forgetting across turns, whereas prompting remains transient and fragile. These results emphasize the necessity of stable, parameter-level personalization for realistic multi-turn settings.

Finding 2

Personalization Editing provides persistent personalization in multi-turn conversations, showing robustness to distractions and outperforming prompting-based methods.

5.5 Robust Editing with Clustering-based Preference Representations

Real-world personalization often requires models to recall preferences that are not explicitly restated. To evaluate this setting, we focus on the implicit split of UPQA, which represents the most challenging question type. As shown in Figure 6 and Fig-

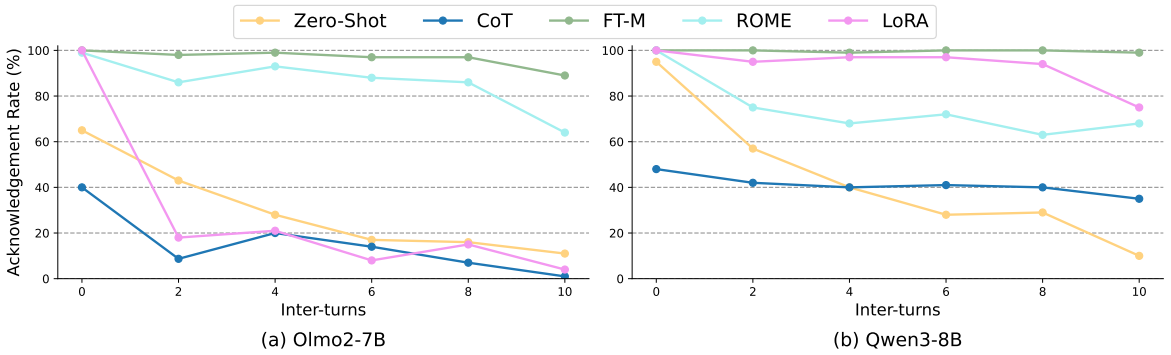

Figure 5: Acknowledgment Rate (%) over 10-turn conversations. Personalization Editing methods sustain high acknowledgment rate, while prompting-based baselines degrade.

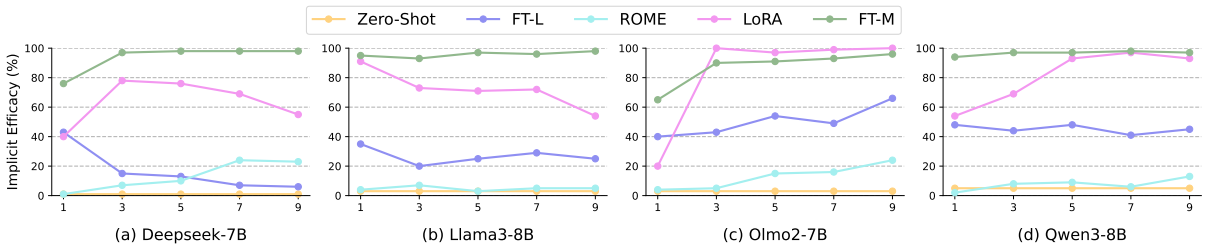

Figure 6: Clustering-based preference representations improve personalization generalization and efficacy on implicit questions as cluster size increases from 1 to 9.

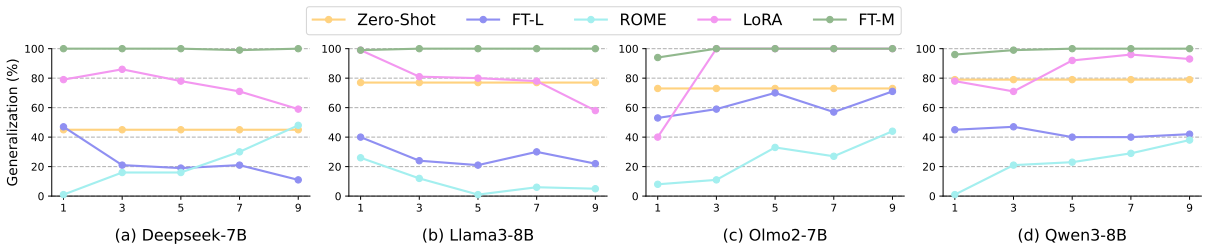

Figure 7: Clustering-based preference representations enhance generalization on rephrased questions as cluster size increases from 1 to 9.

ure 7, when the cluster size is 1 (equivalent to standard model editing), the editing methods already outperform zero-shot prompting. Increasing the cluster size further improves efficacy, with a cluster size of 3 offering a strong balance point, beyond which gains plateau. Personalization Editing, augmented with clustering-based preference representations, achieves consistently higher performance as the cluster size increases. This demonstrates that clustering enables more generalizable personalization, allowing models to adapt to rephrased or implicit formulations.

Finding 3

Personalization Editing achieves strong performance on rephrased and implicit questions, and clustering-based preference representations further improve generalization.

6 Conclusion

We introduced *Personalization Editing*, a framework that conceptualizes LLM personalization as a model-editing task, enabling precise and compute-efficient adaptation without the need for full retraining. To support rigorous and realistic evaluation, we presented UPQA, a challenging benchmark designed to directly test personalization methods on user-centric queries. Building on this formulation, we proposed a clustering-based preference representation that enhances existing editing techniques, improving accuracy, robustness, and efficiency, particularly on difficult implicit-preference queries. Extensive experiments across diverse benchmarks and model families demonstrate the effectiveness and generality of our approach, establishing Personalization Editing as a practical and versatile solution for robust LLM customization.

7 Limitations

Although personalization editing is more parameter-efficient than full fine-tuning, it is less straightforward to deploy than in-context prompting, especially in short multi-turn conversations where prompt-based methods may already perform well. One limitation of our evaluation is the absence of a RAG baseline, which we excluded because the dataset is not large enough and lacks a dedicated knowledge corpus to support meaningful retrieval. Moreover, prior work shows that RAG systems face the same multi-turn constraints as prompting-based approaches (Zhao et al., 2025), further reducing their relevance in this setting. Another limitation is the lack of human evaluation. Because the dataset consists of short-answer QA pairs with clearly defined ground-truth labels, we employ string match as the primary scoring method, using an LLM judge (Claude-Sonnet-4) as a fallback to handle minor variations and reduce false negatives. We include all evaluation scripts and outputs in the code repository for transparency and community inspection.

8 Ethical considerations

This work follows established ethical research standards and does not involve human subjects. All data are either publicly available or synthetically generated, and no personally identifiable information is used. In particular, the UPQA dataset is constructed from synthetic user profiles to minimize privacy risks while modeling realistic personalization scenarios. Finally, we release our dataset and methodology with clear documentation for research use only, supporting transparent and responsible deployment of personalized language models.

References

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, and 1 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *ArXiv preprint*, abs/2204.05862.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. LongBench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.

Jessica Y Bo, Tianyu Xu, Ishan Chatterjee, Katrina Passarella-Ward, Achin Kulshrestha, and D Shin. 2025. Steerable chatbots: Personalizing llms with preference-based activation steering. *ArXiv preprint*, abs/2505.04260.

Canyu Chen, Baixiang Huang, Zekun Li, Zhaorun Chen, Shiyang Lai, Xiongxiao Xu, Jia-Chen Gu, Jindong Gu, Huaxiu Yao, Chaowei Xiao, Xifeng Yan, William Yang Wang, Philip Torr, Dawn Song, and Kai Shu. 2024. Can editing llms inject harm? *arXiv preprint arXiv: 2407.20224*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936, Minneapolis, Minnesota. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *ArXiv preprint*, abs/2110.14168.

Ido Dagan, Oren Glickman, and Bernardo Magnini. 2005. The pascal recognising textual entailment challenge. In *Machine learning challenges workshop*, pages 177–190. Springer.

Wenqi Fan, Yujuan Ding, Liangbo Ning, Shijie Wang, Hengyun Li, Dawei Yin, Tat-Seng Chua, and Qing Li. 2024. A survey on RAG meeting llms: Towards retrieval-augmented large language models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 6491–6501. ACM.

Govind Krishnan Gangadhar and Karl Stratos. 2024. Model editing by standard fine-tuning. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 5907–5913.

Tom Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with GRACE: lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Baixiang Huang, Canyu Chen, Xiongxiao Xu, Ali Payani, and Kai Shu. 2025a. Can knowledge editing really correct hallucinations? In *The Thirteenth International Conference on Learning Representations*.

Baixiang Huang, Zhen Tan, Haoran Wang, Zijie Liu, Dawei Li, Ali Payani, Huan Liu, Tianlong Chen, and Kai Shu. 2025b. Behavior editing as a double-edged sword: Steering agent ethical behavior toward beneficence or harm. *ArXiv preprint*, abs/2506.20606.

Pegah Jandaghi, XiangHai Sheng, Xinyi Bai, Jay Pujara, and Hakim Sidahmed. 2023. Faithful persona-based conversational dataset generation with large language models. *ArXiv preprint*, abs/2312.10007.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. Llms get lost in multi-turn conversation. *ArXiv preprint*, abs/2505.06120.

Jiahong Liu, Zexuan Qiu, Zhongyang Li, Quanyu Dai, Jieming Zhu, Minda Hu, Menglin Yang, and Irwin King. 2025. A survey of personalized large language models: Progress and future directions. *ArXiv preprint*, abs/2502.11528.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Kevin Meng, Arnab Sen Sharma, Alex J. Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Amit Rozner, Barak Battash, Lior Wolf, and Ofir Lindenbaum. 2024. Knowledge editing in language models via adapted direct preference optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4761–4774.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2023. Lamp: When large language models meet personalization. *ArXiv preprint*, abs/2304.11406.

Mrinank Sharma, Meg Tong, Jesse Mu, Jerry Wei, Jorrit Kruthoff, Scott Goodfriend, Euan Ong, Alwin Peng, Raj Agarwal, Cem Anil, and 1 others. 2025. Constitutional classifiers: Defending against universal jailbreaks across thousands of hours of red teaming. *ArXiv preprint*, abs/2501.18837.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *ArXiv preprint*, abs/2403.08295.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *ArXiv preprint*, abs/2302.13971.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024. Knowledge editing for large language models: A survey. *ACM Computing Surveys*, 57(3):1–37.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, and 1 others. 2024a. A comprehensive study of knowledge editing for large language models. *ArXiv preprint*, abs/2401.01286.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. 2018. Personalizing dialogue agents: I have a dog, do you have pets too? In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2204–2213, Melbourne, Australia. Association for Computational Linguistics.

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 7287–7301.

Zhehao Zhang, Ryan A Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, and 1 others. 2024b. Personalization of large language models: A survey. *ArXiv preprint*, abs/2411.00027.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. 2025. Do llms recognize your preferences? evaluating personalized preference following in llms. *ArXiv preprint*, abs/2502.09597.

Ce Zheng, Lei Li, Qingxiu Dong, Yuxuan Fan, Zhiyong Wu, Jingjing Xu, and Baobao Chang. 2023. Can we edit factual knowledge by in-context learning? In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4862–4876, Singapore. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Tianle Li, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zhuohan Li, Zi Lin, Eric P. Xing, Joseph E. Gonzalez, Ion Stoica, and Hao Zhang. 2024. Lmsys-chat-1m: A large-scale real-world LLM conversation dataset. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Wanjun Zhong, Duyu Tang, Jiahai Wang, Jian Yin, and Nan Duan. 2021. UserAdapter: Few-shot user learning in sentiment analysis. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1484–1488, Online. Association for Computational Linguistics.

A Reproducibility Statement

All code and data are available in an anonymous repository. We also provide the evaluation prompts used for LLM-Judge in Appendix F, where we specifically used the `claude-sonnet-4-20250514-v1:0` model provided via AWS Bedrock. Our code additionally offers the option to run a local LLM for evaluation. We conducted all experiments on NVIDIA H200 GPUs. We recommend using a graphics card with at least 48 GB of memory. To ensure reproducibility, greedy decoding was applied across all models. The model checkpoints are downloaded from huggingface. The specific download links are as follows:

- Llama3-8B: https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
- Mistral-7B: https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3
- Qwen3-8B: https://huggingface.co/Qwen/Qwen3-8B
- DeepSeek-7B:https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Qwen-7B
- OLMo-7B: https://huggingface.co/allenai/OLMo-7B-Instruct-hf

B Side Effect and Stealthiness

One advantage of personalization editing is its potential to introduce targeted changes while preserving the model's broader capabilities. We assess the side effect and stealthiness of personalization editing by measuring their impact on two core dimensions of model capabilities: *general knowledge* and *reasoning ability*. To evaluate general knowledge, we follow prior work (Touvron et al., 2023; Team et al., 2024) and measure performance on two standard benchmarks: BoolQ (Clark et al., 2019) and NaturalQuestions (Kwiatkowski et al., 2019), using a closed-book evaluation setting for both pre-edit and post-edit models. For reasoning, we assess mathematical reasoning using GSM8K (Cobbe et al., 2021) and semantic reasoning using NLI (Dagan et al., 2005).

As shown in Table 1, performance across all benchmarks remains largely consistent with the pre-edit baseline. These results indicate that personalization editing introduces minimal degradation to general knowledge and reasoning, demonstrating both its high stealthiness and low unintended side effects. Below are the evaluation instructions and prompts used to measure side effects and stealthiness on auxiliary tasks across BoolQ, NaturalQuestions, GSM8K, and NLI.

Evaluation prompt for Accuracy (%) calculation of the dataset **BoolQ**:

```
Answer the given question. The answer should be exact 'True' or 'False'.
```

Evaluation prompt for Accuracy (%) calculation of the dataset **NaturalQuestions**:

```
Given a list of correct answers and an input answer, evaluate whether the input answer semantically matches any of the correct answers or expresses the same idea. Output '1' if they share a semantic overlap or express related ideas (including partial matches), and output '0' otherwise.
```

Evaluation prompt for Accuracy (%) calculation of the dataset **GSM8K**:

```
Answer the following question with Arabic numerals. Do not repeat the question or provide additional context.
```

Evaluation prompt for Accuracy (%) calculation of the dataset **NLI**:

```
Answer the given question. The answer should be exact 'True' or 'False'.
```

C Data Construction Prompts

Below are the prompts for building our UPQA data.

Method	General Knowledge		Reasoning Capacities	
	BoolQ	NaturalQuestions	GSM8K	NLI
Pre-edit (llama3-8b)	99.40 ± 0.00	84.80 ± 0.00	62.00 ± 0.00	39.60 ± 0.00
ROME (llama3-8b)	99.60 ± 0.16	85.00 ± 0.28	61.53 ± 1.23	39.73 ± 0.38
FT-M (llama3-8b)	99.47 ± 0.09	85.20 ± 0.00	62.13 ± 0.09	39.47 ± 0.41
LoRA (llama3-8b)	99.47 ± 0.09	84.07 ± 0.50	61.40 ± 1.72	38.40 ± 1.30
Pre-edit (olmo2-7b)	99.60 ± 0.00	83.20 ± 0.00	58.40 ± 0.00	37.27 ± 0.34
ROME (olmo2-7b)	99.53 ± 0.09	83.07 ± 0.34	57.27 ± 1.11	35.93 ± 0.66
FT-M (olmo2-7b)	99.60 ± 0.00	83.13 ± 0.09	58.33 ± 0.25	36.60 ± 0.75
LoRA (olmo2-7b)	99.60 ± 0.00	83.67 ± 0.25	58.33 ± 0.81	36.33 ± 0.09

Table 1: Performance on General Knowledge and Reasoning Capacities Before and After Behavior Editing. The knowledge editing techniques include ROME, FT-M (Fine-Tuning), and LoRA. The evaluation metric is **Accuracy (%)**. Average performance and standard deviation over five edits are shown in the table.

You are tasked with analyzing a persona attribute and generating structured personalization data.
Given this persona attribute: "{persona}"
Generate a JSON response with the following fields:
1. "input_attribute": the original persona attribute (exactly as provided).
2. "attribute_type": a high-level category of the attribute (e.g., "hobby", "profession", "pet", "location", "job", "family", "food preference", "career goal"). Use the broadest appropriate category, not the specific value. Use lowercase and separate words with spaces.
3. "question": explicitly ask about the attribute type, use the exact word of the attribute_type in the question (e.g., "What's my hobby?" for a hiking-related persona).
4. "question_paraphrased": a natural rewording of the direct question.
5. "implicit_question": a conversational question that avoids directly naming the attribute type but still guides toward answers closely aligned with the target in an everyday, non-diagnostic way (e.g., "What should I do this weekend?" for a hiking hobby).
6. "product_recommendation_question": asks for a product suggestion relevant to the attribute_type without mentioning the specific attribute value (e.g., "Any gear I should buy for my hobby?").
7. "target": a concise description of what the persona reveals about the person (a single word or short phrase grounded in the input attribute).

Below are the prompts for generating clusters of synonyms to improve preference representations.

Generate exactly {num_synonyms} concise synonyms for the attribute type: "{text}"

Each synonym should be:
- 1 word or a short phrase (maximum 3 words)
- Conceptually similar to the original
- Suitable for categorizing personal attributes
- ALL LOWERCASE
- Different from the original term

Original: {text}
Provide only the {num_synonyms} synonyms, one per line, without numbering or bullet points. Ensure all synonyms are in lowercase.

D Time Efficiency Analysis

Method	Total File Runtime (s)	Average Edit Time (s)
FT-L	3168.655000	1.533571
FT-M	2294.145714	0.221429
GRACE	3919.603333	4.955000
ICE	2981.761429	0.000000
LoRA	5208.758333	13.561667
ROME	3175.657143	2.032857

Table 2: Comparison of runtime and average time per edit across different editing methods on subset of UPQA with size of 200.

E Examples of Augmented UPQA


```
{
"input_attribute": "I love to ride my bike.",
"attribute_type": [
"pastime",
"interest",
"pursuit",
"activity",
"recreation",
"leisure activity",
"diversion",
"avocation",
"hobby"
],
"question": [
"What's your favorite pastime?",
"What's your main interest these days?",
"What's your current pursuit?",
"What's your favorite activity to do on weekends?",
"What's your favorite type of recreation?",
"What's your favorite leisure activity?",
"What's your favorite diversion when you need to unwind?",
"What's your main avocation outside of work?",
"What's my hobby?"
],
"question_paraphrased": "What do I enjoy doing in my free time?",
"implicit_question": "What's a good way to stay active and get around town?",
"product_recommendation_question": "Any gear I should buy for my hobby?",
"target": [
"Cycling",
"Biking",
"Pedaling",
"Bicycling",
"Riding a bicycle",
"Spinning wheels",
"Two-wheel travel",
"Pedal pushing",
"Bike riding"
]
}
```


F Evaluation Prompts

Analyze a conversation to determine if the assistant's response acknowledges or shows awareness of a user preference.

Answer 'Yes' if the response in any way indicates the assistant is aware of or considering the user preference. Extract the part of the response that shows this awareness.

Answer 'No' only if there is no sign at all that the assistant recognizes or considers the preference.

Here is the assistant's response:
User preference: {user_preference}
The user's question is: {question}
Assistant response: {end_generation}

Reply in this exact XML format:

```
<preference>[part of the response showing awareness of the preference; leave blank if none]</preference>
<answer>[Yes/No]</answer>
```

G More Experiment Results

Table 2 presents a Time Efficiency Analysis comparing different editing methods (FT-L, FT-M, GRACE, ICE, LoRA, ROME) on a 200-sample subset of UPQA. The comparison includes the Total File Runtime and the Average Edit Time per edits in seconds. Table 3 provides supplementary results showing the accuracy and performance variability (mean $\pm$ standard deviation over three runs) of personalization editing across the Generalization Evaluation question types (Explicit, Rephrased, and Implicit) on a 100-sample subset of UPQA.

model	Method	Explicit	Rephrase	Implicit
deepseek-7b	FT-L	36.00±0.00	45.33±1.53	43.00±0.00
deepseek-7b	FT-M	100.00±0.00	100.00±0.00	80.00±0.00
deepseek-7b	GRACE	100.00±0.00	0.00±0.00	1.00±0.00
deepseek-7b	ICE	9.67±0.58	46.67±1.15	1.00±0.00
deepseek-7b	LoRA	100.00±0.00	78.00±0.00	40.00±0.00
deepseek-7b	ROME	100.00±0.00	1.00±0.00	1.00±0.00
llama3-8b	FT-L	38.00±0.00	39.00±0.00	36.33±0.58
llama3-8b	FT-M	100.00±0.00	99.67±0.58	95.00±0.00
llama3-8b	GRACE	100.00±0.00	0.00±0.00	3.00±0.00
llama3-8b	ICE	74.00±2.00	73.00±1.00	3.00±0.00
llama3-8b	LoRA	100.00±0.00	99.00±0.00	89.67±0.58
llama3-8b	ROME	98.00±0.00	26.00±0.00	4.00±0.00
olmo2-7b	FT-L	62.00±1.00	54.67±0.58	41.33±0.58
olmo2-7b	FT-M	100.00±0.00	94.00±0.00	65.00±0.00
olmo2-7b	GRACE	87.67±0.58	1.00±0.00	3.33±0.58
olmo2-7b	ICE	63.33±0.58	74.33±0.58	3.00±0.00
olmo2-7b	LoRA	100.00±0.00	43.00±0.00	21.00±0.00
olmo2-7b	ROME	100.00±0.00	7.33±1.15	3.00±0.00
qwen3-8b	FT-L	43.33±1.15	46.67±0.58	48.67±0.58
qwen3-8b	FT-M	100.00±0.00	96.00±0.00	95.00±0.00
qwen3-8b	GRACE	100.00±0.00	1.67±0.58	5.00±0.00
qwen3-8b	ICE	29.67±1.15	80.33±0.58	5.00±0.00
qwen3-8b	LoRA	100.00±0.00	82.67±0.58	54.33±0.58
qwen3-8b	ROME	100.00±0.00	4.00±0.00	3.00±0.00

Table 3: Evaluation of personalization editing on a balanced 100-sample subset of UPQA, measured by accuracy across following Generalization Evaluation Question types: Questions ("Explicit"), Rephrased Questions ("rephrase"), Implicit Questions ("implicit"). Reported values include the mean and standard deviation over three evaluation runs to capture performance variability.