Latent Inter-User Difference Modeling for LLM Personalization

Anonymous ACL submission

Abstract

Large language models (LLMs) are increasingly integrated into users' daily lives, leading to a growing demand for personalized outputs. Previous work focuses on leveraging a user's own history, overlooking inter-user differences that are crucial for effective personalization. While recent work has attempted to model such differences, the reliance on language-based prompts often hampers the effective extraction of meaningful distinctions. To address these issues, we propose Difference-aware Embedding-based Personalization (DEP), a framework that models inter-user differences in the latent space instead of relying on language prompts. DEP constructs soft prompts by contrasting a user's embedding with those of peers who engaged with similar content, highlighting relative behavioral signals. A sparse autoencoder then filters and compresses both user-specific and difference-aware embeddings, preserving only task-relevant features before injecting them into a frozen LLM. Experiments on personalized review generation show that DEP consistently outperforms baseline methods across multiple metrics. Our code is available on an Anonymous GitHub.

1 Introduction

With continuous advancements in general-purpose intelligence, large language models (LLMs) (Achiam et al., 2023; Grattafiori et al., 2024; Yang et al., 2024; Guo et al., 2025) are increasingly integrated into everyday life, assisting users in making decisions (Yao et al., 2023; Deng et al., 2023), retrieving information (Gao et al., 2023; Asai et al., 2024), and task management (Shen et al., 2024). This growing presence has raised expectations for LLMs to go beyond generic, one-size-fits-all responses and instead produce responses that align with individual users' unique preferences. To meet these heightened expectations, there appears the interests of *LLM personalization* (Zhang et al., 2024; Xu et al., 2025; Liu et al., 2025), which aims at tailoring model outputs based on user-specific information.

Widely used methods generally follow the memory-retrieval framework, where user history is stored in memory, and key information is then retrieved as a steering prompt to guide model generation. Previous works focused solely on retrieving information about the user themselves for personalization. However, recent work such as DPL (Qiu et al., 2025) argues that effective personalization should also capture how a user differs from others. This view is grounded in insights from psychology and behavioral science (Snyder and Fromkin, 1977, 2012; Irmak et al., 2010), which highlight that inter-user variability determines individuality and shapes users' distinct preferences. Accordingly, DPL incorporates inter-user comparison in the retrieval history, formulating the comparison as a language task performed by the LLM.

Despite DPL's demonstrated effectiveness, we argue that its language-based inter-user comparison paradigm using LLMs is structurally ill-suited for accurately extracting inter-user differences. On one hand, controlling the extraction of differences using an LLM is challenging; although providing extraction criteria can help, some aspects of distinction may be missed due to the difficulty of defining comprehensive standards. On the other hand, including other users' raw data for comparison in LLMs can result in verbose prompts that strain the model's context window, ultimately hindering the extraction of meaningful inter-user differences.

To address these limitations, we propose shifting to latent-space difference modeling, where task-relevant differences between users are structurally represented and compared in the latent embedding space (Doddapaneni et al., 2024; Liu et al., 2024; Zeldes et al., 2025). Compared to natural language, latent embeddings offer two key advantages: (1) they encode fine-grained, context-dependent behav-

ioral patterns in a compact form; and (2) they inherently support inter-user comparison through vector operations, enabling direct integration of comparison signals. Together, these properties make latent embeddings a more suitable medium for modeling inter-user differences within LLMs.

Building on this idea, we propose a new method called *Difference-aware Embedding-based Personalization* (DEP), which models task-relevant inter-user differences in the latent space and injects them into LLMs as soft prompts for personalization. DEP extracts a difference-aware embedding as a soft prompt by subtracting and aggregating the user's embedding against those of other users who engaged with similar items. At the same time, the original user-specific embedding is provided as a reference to supply contextual information. Both embeddings are essential: the user-specific embedding defines the behavioral context, while the difference-aware embedding captures deviations from that context. Together, they form a contextualized inter-user signal that reflects both individualized preferences and relative differences.

Taking a step further, latent differences can be redundant, as not all aspects are task-relevant—some may simply constitute noise for the task. To extract essential information while filtering out irrelevant signals, we process both user-specific and difference-aware embeddings using a sparse autoencoder (SAE) (Huben et al., 2024), which enforces sparsity to retain only key features. The resulting compressed representations are then injected into a frozen LLM as soft prompts. The SAE is fine-tuned to align these representations with the LLM's internal understanding, allowing the model to effectively leverage inter-user differences for improved personalization. We conduct extensive experiments on one representative task, review generation (Ni et al., 2019), where DEP achieves state-of-the-art performance across multiple evaluation metrics.

Our main contributions are as follows:

- We propose modeling inter-user differences in the latent space to enable more comprehensive and flexible extraction of preference signals for LLM personalization.
- We propose a novel method, DEP, to achieve latent difference modeling, equipped with a sparse autoencoder to extract task-relevant differences while filtering out noise.
- Extensive experiments show that our DEP consistently outperforms baseline methods with significant improvements.

2 Preliminary

Problem Formulation. This work studies the task of LLM personalization, where the goal is to produce user-aligned output that reflects the individual preferences of a given user. We assume that each user has a set of historical texts. These historical texts are utilized to help the LLM infer the user's interests and generate personalized content. Formally, let D denote the collection of historical records from all users. Each record in D is represented as (u, i, y_u^i), where u is a user, i is an item (or object) the user has focused on, and y_u^i denotes the text written or preferred by user u for item i. When the target user u' submits a request to generate text for a target item i', the LLM is expected to produce an output that aligns with the preference of u' based on D.

Without loss of generality, this work focuses on review generation, a representative personalization task. The goal is to generate reviews tailored to a user's style and preferences, ensuring the output aligns with how the user typically expresses opinions on items such as movies or products.

Memory-retrieval framework. A common approach to enabling LLMs to perform personalized generation is to store users' history and retrieve relevant signals at inference. Following DPL (Qiu et al., 2025), effective personalization should capture both a user's own behavioral patterns and how they differ from others. This involves extracting key preference signals from two sources: the user's own history, which reflects individual tendencies, and other users' behaviors, which provide materials for modeling inter-user differences. Formally, given a target user u' and a target item i', the personalized generation process can be formulated as:

$$\hat{y}_{u'}^{i'} = \mathrm{LLM}(u', i', \phi(D_{u'}; D)), \tag{1}$$

where $\hat{y}_{u'}^{i'}$ denotes the generated text, $D_{u'}$ denotes the history of the target user u', and $\phi(D_{u'}; D)$ denotes the process that extracts user-specific and difference-aware preference signals from D_u and D. This memory retrieval framework supports lifelong user modeling without requiring LLM retraining (Zhuang et al., 2024), making it both adaptable and resource-efficient for real-world personalization scenarios.

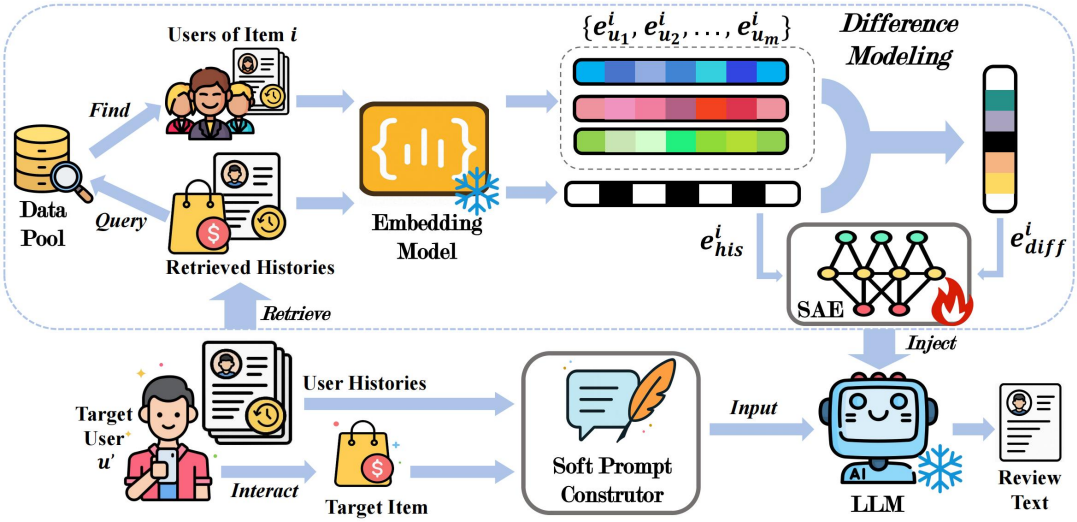


Figure 1: Overview of the proposed DEP method, which introduces user-specific and difference-aware embeddings to capture both individual preferences and inter-user differences. A sparse autoencoder (SAE) refines these representations, which are then injected into a frozen LLM as soft prompts to guide personalized text generation.

3 Methodology

This section introduces our proposed *Difference-aware Embedding-based Personalization* (**DEP**). We begin with its motivation and an overview of the framework, followed by detailed descriptions of each key steps.

3.1 Overview

Personalization modeling requires capturing not only a user's own behavioral patterns, but also how this user differs from others. In modeling inter-user differences, existing work (Qiu et al., 2025) relies on LLMs to summarize inter-user comparisons in natural language, which may miss some key aspects of distinctions during the summarization. To address this limitation, we propose the DEP method, which aims to model inter-user differences in the latent space. DEP has three main parts: (1) constructing two representations to capture difference-aware preference: a user-specific embedding to model the behavioral context, and a difference-aware embedding to model how the user deviates from others within that context; (2) distilling the representations with a sparse autoencoder to retain informative preference signals; and (3) injecting the compressed representation into a frozen LLM as soft prompts and fine-tuning the autoencoder to align this representation with the LLM's internal understanding. Figure 1 provides an overview of our proposed DEP. Next, we elaborate the three parts in detail.

3.2 Difference-aware Embedding-based Personalization (DEP)

In this section, we introduce three key steps of DEP: constructing difference-aware representations, distilling them via a sparse autoencoder, and injecting them into an LLM for personalization.

3.2.1 Latent-space Difference-aware Representation Modeling

The core of DEP is to model inter-user differences in the latent space through contrastive signals grounded in shared item contexts. To achieve this, following the memory-retrieval paradigm (Salemi et al., 2024; Kumar et al., 2024; Qiu et al., 2025), DEP first retrieves a set of representative interactions from the user's history, which serve as anchors for inter-user comparison. For a given user u', we assume a subset of N key interactions, denoted as $D^*_{u'}$, can be obtained via retrieval (Zhang et al., 2024) from $D_{u'}$. Then, for each retrieved interaction $(u', i, y^i_{u'}) \in D^*_{u'}$, we aim to compare it with reviews written by other users for the same item i, which provides a natural basis for inter-user comparison. To this end, we first encode the user's own review $y^i_{u'}$ using a frozen text embedding model $f_{\text{emb}}(\cdot)$ to obtain the user-specific embedding:

$$e^i_{\text{his}} = f_{\text{emb}}(y^i_{u'}), \tag{2}$$

where e^i_{his} denotes the user-specific embedding that reflects the preference pattern of user u' on item y. Next, to construct inter-user embeddings, we identify the set of peer users who also interacted

with item i, excluding u', as $\{u_1, u_2, \ldots, u_m\}$, where u_j denotes the j-th peer user of item i. Each peer user u_j provides a review $y^i_{u_j}$, which is encoded into an embedding:

$$e^i_{u_j} = f_{\text{emb}}(y^i_{u_j})). \tag{3}$$

Then we compute the difference-aware embedding by aggregating the vector differences between the target user and each peer:

$$e^i_{\text{diff}} = \frac{1}{m}\sum_{j=1}^{m}(e^i_{\text{his}} - e^i_{u_j}), \tag{4}$$

where e^i_{diff} denotes the difference-aware embedding. These two embeddings capture complementary perspectives: the user-specific embedding e^i_{his} represents the behavior pattern of the target user and serves as a reference of context, while the difference-aware embedding e^i_{diff} models how this behavior pattern relatively deviates from others under the context. Together, they form a structured representative to capture the inter-user differences.

3.2.2 Sparse Representation Distillation

While the user-specific and difference-aware embeddings capture rich semantic and contrastive signals, they may contain redundant or irrelevant information that hinders efficient personalization. To address this, we apply a sparse autoencoder (SAE) (Huben et al., 2024) to compress the high-dimensional embeddings into informative representations. The SAE adopts an encoder-decoder architecture with an ℓ_1-based sparsity constraint on the latent space, encouraging the model to retain only the most salient features. Given a history embedding e^i_{his} and a difference-aware embedding e^i_{diff}, the encoder produces their respective low-dimensional latent vectors, z^i_{his} and z^i_{diff}, formally:

$$\begin{aligned} z^i_{\text{his}} &= f_{\text{enc}}(e^i_{\text{his}}), \qquad \hat{e}^i_{\text{his}} = f_{\text{dec}}(z^i_{\text{his}}), \\ z^i_{\text{diff}} &= f_{\text{enc}}(e^i_{\text{diff}}), \qquad \hat{e}^i_{\text{diff}} = f_{\text{dec}}(z^i_{\text{diff}}), \end{aligned} \tag{5}$$

where $f_{\text{enc}}(\cdot)$ and $f_{\text{dec}}(\cdot)$ denote the encoder and decoder networks, respectively. The encoder outputs z^i_{his} and z^i_{diff} are used as sparse preference representations for downstream soft prompt construction.

3.2.3 Representation Injection

After obtaining the distilled latent representations from the sparse autoencoder, we aim to integrate personalized signals into the generation process of a frozen LLM. To achieve this, we adopt a soft prompt injection mechanism, where the compressed user-specific and difference-aware embeddings are projected into the input space of the LLM and used to condition its output without updating model parameters.

Soft Prompt Construction and Injection. For each retrieved history (u', i, y), we obtain z^i_{his} and z^i_{diff} from the SAE encoder, corresponding to the user-specific and difference-aware embeddings. These representations are projected into the LLM input space via a lightweight projection network $\mathcal{M}_p(\cdot)$, which aligns their dimensionality with that of the LLM's embedding layer:

$$p^i_{\text{his}} = \mathcal{M}_p(z^i_{\text{his}}), \quad p^i_{\text{diff}} = \mathcal{M}_p(z^i_{\text{diff}}), \tag{6}$$

where p^i_{his} and p^i_{diff} are resulting soft prompt vectors, which are injected into the input sequence at designated positions. Then, the personalized generation process given the target user u' and the target item i' is performed as:

$$\hat{y}^{i'}_{u'} = LLM\left(\mathcal{S}(i', \{i, p^i_{\text{his}}, p^i_{\text{diff}}\}_{i \in I^*_{u'}})\right), \tag{7}$$

where $I^*_{u'}$ denotes the top-N retrieved items from the target user's interacted history, and $\mathcal{S}(i', \{i, p^i_{\text{his}}, p^i_{\text{diff}}\}_{i \in I^*_{u'}})$ is a textual prompt constructed from both the target item i' and the soft prompts to model inter-user differences, and the original user's original review history to model user's own writing patterns. The template can be found in Figure 6 in Appendix F.

Training Objectives. To guide the SAE learning informative representation for LLM personalization and make the soft prompts align with the LLM internal understanding, we jointly optimize two components: the SAE for latent representation learning and the LLM for personalized generation. The LLM is trained using a standard generation loss, denoted as $\mathcal{L}_{\text{gen}}$, computed based on the generated output and ground-truth personalized text. The SAE is trained with two standard objectives: a reconstruction loss to ensure information preservation, and a sparsity loss to promote selective preference encoding. For the reconstruction loss, we adopt the Smooth L1 loss, which is formulated as follows:

$$\mathcal{L}_{\text{recon}} = \text{SmoothL1}(e^i_{\text{his}}, \hat{e}^i_{\text{his}}) + \text{SmoothL1}(e^i_{\text{diff}}, \hat{e}^i_{\text{diff}}). \tag{8}$$

The sparsity loss is applied to the distilled latent vector $z^i_{\text{his}} \in \mathbb{R}^{d'}$ and $z^i_{\text{diff}} \in \mathbb{R}^{d'}$, encouraging the preservation of the most informative signals. For

each, we compute the average activation $\hat{\rho}_{\text{his}}$ and $\hat{\rho}_{\text{diff}}$ as:

$$\hat{\rho}_{\text{his}} = \frac{1}{N}\sum_{i=1}^{N} z^i_{\text{his}}, \quad \hat{\rho}_{\text{diff}} = \frac{1}{N}\sum_{i=1}^{N} z^i_{\text{diff}}. \tag{9}$$

We then compute the sparsity loss by applying KL divergence between each of $\hat{\rho}_{\text{his}}$ and $\hat{\rho}_{\text{diff}}$ and a predefined sparsity target ρ.

$$\mathcal{L}_{\text{sparse}} = \frac{1}{d'}\sum_{j=1}^{d'} KL(\rho||\hat{\rho}^j_{\text{his}}) + \frac{1}{d'}\sum_{j=1}^{d'} KL(\rho||\hat{\rho}^j_{\text{diff}}). \tag{10}$$

The final training objective combines the generation loss from the LLM and the SAE loss, including both reconstruction and sparsity terms:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{gen}} + \lambda \cdot (\mathcal{L}_{\text{recon}} + \gamma \cdot \mathcal{L}_{\text{sparse}}). \tag{11}$$

where λ and γ balance the contributions of the SAE loss and the sparsity constraint, respectively.

4 Experiment

We conduct experiments in real-world datasets to answer the following research questions:

- **RQ1**: How does DEP compare with baseline methods on the personalized text generation task?
- **RQ2**: What is the contribution of each individual component of DEP to its overall effectiveness?
- **RQ3**: What is the impact of the number of retrieved histories on the performance of DEP?
- **RQ4**: How does DEP perform under different levels of user uniqueness compared to DPL?

4.1 Experimental Setup

Datasets. Building upon prior work, we focus on the representative task of item review generation for LLM personalization (Ni et al., 2019; Peng et al., 2024; Kumar et al., 2024; Au et al., 2025). Specifically, we adopt the Amazon Reviews 2023 dataset[1] (Hou et al., 2024) preprocessed by DPL[2] (Qiu et al., 2025), which covers three categories: *Books*, *Movies & TV*, and *CDs & Vinyl*. To maximize data utilization, we follow the setting of REST-PG (Salemi et al., 2025) to train a unified model across categories. For training, we retain each user's most recent interaction per category. For validation, we randomly select 512 instances from the merged validation set across all three categories, while for testing, we follow the original test splits provided by DPL. More details about the dataset are provided in Appendix A.

Baselines. We compare our proposed DEP with the following baseline methods. Further implementation details of all baselines can be found in Appendix B.

- **Non-Perso**: A non-personalized baseline that generates reviews using only item information, along with the review's title and rating.
- **RAG** (Salemi et al., 2024): A retrieval-based method that incorporates the user's history records to provide contextual personalization.
- **PAG** (Richardson et al., 2023): An extension of RAG that summarizes the user's history records into a compact profile and combines it with retrieved content for higher-level personalization.
- **DPL** (Qiu et al., 2025): A prompt-based method that enhances personalization by explicitly comparing a user's recent behavior with representative peers and summarizing the differences into a profile integrated into the LLM input.
- **PPlug** (Liu et al., 2024): A plug-and-play approach that encodes user history into a dense embedding, which is projected into the LLM's input space to guide generation.

Evaluation Metrics. Following previous works on personalized text generation (Salemi et al., 2024; Kumar et al., 2024; Zhang et al., 2025; Au et al., 2025), we evaluate all methods using ROUGE-1 (Lin, 2004), METEOR (Banerjee and Lavie, 2005), and BLEU[3] (Papineni et al., 2002).

Implementation Details. We use the `Qwen2.5 Instruct`[4] (Yang et al., 2024) series models (7B and 32B) as backbone LLMs for all baseline methods and DEP. To retrieve user histories, we adopt a recency-based strategy, selecting the most recent history for each user. Additionally, we employ `bge-m3`[5] (Chen et al., 2024a) as the embedding model to map user reviews into vector representations. We train DEP for 5 epochs and select the checkpoint with the highest METEOR score on the validation set for testing. For more details, please refer to Appendix C.

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

[2] https://huggingface.co/datasets/SnowCharmQ/DPL-main & https://huggingface.co/datasets/SnowCharmQ/DPL-meta

[3] We use the standard SacreBLEU (Post, 2018) library to calculate the BLEU score: https://github.com/mjpost/sacrebleu.

[4] https://huggingface.co/Qwen

[5] https://huggingface.co/BAAI/bge-m3

Table 1: Performance comparison between the baselines and our DEP across the three datasets. *7B* and *32B* represent the size of base LLMs. The best results are highlighted in **bold**, and the second-best results are underlined. "R-1", "MET.", and "BL." respectively denote ROUGE-1, METEOR, and BLEU. Higher values indicate better performance across all metrics.

Datasets (→)		Books			Movies & TV			CDs & Vinyl		
Methods (↓)		R-1	MET.	BL.	R-1	MET.	BL.	R-1	MET.	BL.
32B	Non-Perso	0.3025	0.1949	2.6728	0.2608	0.1666	1.1226	0.2765	0.1767	1.6597
	RAG	0.3404	0.2735	6.8178	0.2983	0.2142	2.8680	0.3092	0.2177	3.1588
	PAG	0.3276	0.2830	6.8920	0.2816	0.2130	2.7751	0.2971	0.2215	3.2164
	DPL	0.3392	0.3003	7.7423	0.2967	0.2238	3.2965	0.3119	0.2337	3.8271
7B	Non-Perso	0.2907	0.1735	1.9766	0.2469	0.1503	0.7242	0.2604	0.1561	1.0997
	RAG	0.3149	0.2101	3.6874	0.2693	0.1701	1.3021	0.2796	0.1733	1.6129
	PAG	0.3136	0.2378	4.6762	0.2761	0.1905	1.9360	0.2882	0.1979	2.4740
	DPL	0.3194	0.2459	5.6623	0.2845	0.1958	2.2451	0.2952	0.2003	2.6943
	PPlug	0.3033	0.2234	7.0469	0.2530	0.1724	3.2291	0.2619	0.1711	3.0753
	DEP (ours)	**0.3745**	**0.3156**	**13.5300**	**0.3092**	**0.2381**	**6.6835**	**0.3165**	**0.2364**	**6.5166**

4.2 Main Results (RQ1)

We first evaluate the overall performance of all compared methods. Table 1 presents the main experimental results across three datasets, from which we draw the following observations:

- **Incorporating user context significantly improves the model's capability for personalized text generation.** Methods like RAG and PAG leverage retrieved user information to condition generation, significantly outperforming the Non-Perso baseline. DPL further improves upon these by explicitly modeling inter-user differences, achieving the relatively best performance among all ICL-based methods. This shows that capturing user differences yields better personalization than simple relevance or summarization.
- **Scaling up the model size leads to stronger performance across different personalization methods.** For methods where both 7B and 32B models are evaluated, we observe consistent improvements across three metrics. This trend highlights the capacity of larger models to capture more nuanced personalization patterns.
- **Using a single soft prompt for user history, PPlug lacks informative signals and overlooks inter-user differences.** Although PPlug outperforms the Non-Perso baseline by introducing lightweight user modeling through the soft prompt, its gains remain limited. This limitation motivates our design of a more effective soft prompt strategy.
- **DEP consistently outperforms all baselines across datasets and metrics.** Despite operating on a much smaller model scale, DEP not only significantly outperforms all 7B-based methods, but also surpasses all baselines under the 32B backbone. Notably, averaged across three datasets, DEP yields relative improvements of 5.05% in ROUGE-1, 4.21% in METEOR, and 82.59% in BLEU compared to the strongest baseline. This substantial performance gain is primarily attributed to the integration of implicit modeling of user history and inter-user differences, which provides more informative and discriminative signals for personalization.

4.3 Ablation Studies (RQ2)

To better understand the contribution of different components in our personalization framework, we conduct extensive ablation studies from two perspectives: user embedding configuration and representation refinement.

We report METEOR scores on all three datasets here, and leave results for the other two metrics in Appendix D.

4.3.1 User Embedding Configuration

To assess the effectiveness of incorporating different types of user embeddings, we conduct a detailed study comparing various configurations of personalized signals. Specifically, we consider two types of embeddings: (1) user-specific embeddings (*his_emb*), which represent the user's past interactions, and (2) difference-aware embeddings (*diff_emb*), which encode inter-user differences by contrasting the target user's review history with those of other users. We examine these embedding configurations individually and in combination, un-

Table 2: Ablation study on different configurations of user embeddings. *his_emb* and *diff_emb* denote user history and difference-aware embeddings. *w/o text* and *w/ text* refer to the exclusion or inclusion of retrieved review texts.

	Datasets (→) Methods (↓)	Books	Movies & TV	CDs & Vinyl
	Non-Perso-7B	0.1735	0.1503	0.1561
w/o text	his_emb	0.1718	0.1625	0.1711
	diff_emb	0.1839	0.1546	0.1616
	his_emb + diff_emb	0.2227	0.1871	0.1853
w/ text	his_emb	0.3110	0.2332	0.2268
	diff_emb	0.2781	0.2128	0.2108
	his_emb + diff_emb (ours)	**0.3156**	**0.2381**	**0.2364**

Table 3: Ablation study on representation refinement. *w/o DR* uses raw embeddings, *w/ AE* uses a standard autoencoder, and *w/ SAE* is our implementation.

Datasets (→) Methods (↓)	Books	Movies & TV	CDs & Vinyl
w/o DR	0.3016	0.2325	0.2283
w/ AE	0.2994	0.2350	0.2355
w/ SAE (ours)	**0.3156**	**0.2381**	**0.2364**

der two settings: with retrieved review text (*w/ text*) and without it (*w/o text*).

Results in Table 2 show that both *his_emb* and *diff_emb* individually outperform the non-personalized baseline, demonstrating the effectiveness of modeling both user history and inter-user differences. Combining the two leads to further improvements, suggesting that user-specific embedding and difference-aware embedding capture complementary aspects of personalization. Additionally, incorporating retrieved texts (*w/ text*) consistently enhances all configurations, highlighting the benefit of contextual grounding.

4.3.2 Representation Refinement

We further evaluate the impact of different strategies for refining user embeddings before soft prompt injection. Specifically, we compare three variants: (1) *w/o DR*, where raw high-dimensional embeddings are directly projected without dimensionality reduction, (2) *w/ AE*, which uses a standard autoencoder for compression without sparsity, and (3) *w/ SAE*, which applies our sparse autoencoder to introduce the sparsity constraint.

Table 3 shows that removing dimensionality reduction (*w/o DR*) generally results in weaker performance. While the standard autoencoder (*w/ AE*) brings partial improvements on *Movies & TV* and *CDs & Vinyl* datasets, it does not consistently outperform the raw embedding variant, suggesting that compression alone is insufficient. In contrast, we introduce a sparse autoencoder (*w/ SAE*), achieving the best results across all datasets, highlighting the effectiveness of sparsity constraint in enhancing representation quality for personalization.

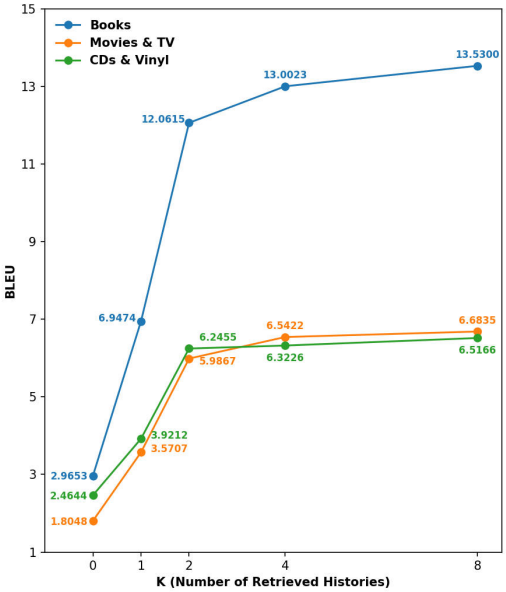


Figure 2: Effect of the number of retrieved user histories (K) on BLEU performance across datasets.

4.4 In-Depth Analysis

We conduct additional experiments to further study the design and effectiveness of our approach.

4.4.1 Impact of History Number (RQ3)

Figure 2 shows how the number of retrieved user histories (K) affects the performance on BLEU across datasets. A key observation is the substantial jump in performance from $K = 0$ to $K = 1$, which marks the transition from the non-personalized setting to the personalized framework of DEP. This single-step increase highlights the substantial benefit of incorporating even one user-specific history with both the user-specific and difference-aware embeddings, demonstrating the effectiveness of our method once personalization is engaged. As K increases further, performance continues to improve, though with diminishing returns.

For a more comprehensive view, we provide the full results across all evaluation metrics and datasets in Appendix D.3.

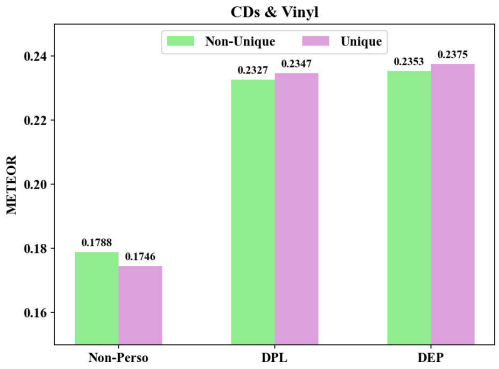


Figure 3: Results of the performance of DEP across different levels of uniqueness. The experiments are conducted on *CDs & Vinyl* and evaluated in METEOR.

4.4.2 Impact of User Uniqueness (RQ4)

Following the procedure in DPL (Qiu et al., 2025), we further investigate how user uniqueness affects personalization performance. Similarly, we adopt a grouping strategy based on the user embedding derived from historical reviews. Specifically, we compute the Euclidean distance between each user's review embedding and the global average embedding across all users, and divide users into two groups: the top 50% as *Unique* users and the bottom 50% as *Non-Unique* users.

As shown in Figure 3, both DPL and DEP outperform the non-personalized baseline across user groups. DEP consistently achieves the best results and maintains stable improvements for both *Unique* and *Non-Unique* users. Similar to DPL, larger gains are observed in the *Unique* group, highlighting the importance of modeling user distinctiveness. Unlike DPL, which relies on prompt-level representations, DEP models inter-user differences in the latent space, enabling more compact and robust personalization, leading to better performance.

5 Related Work

The personalization of LLMs has become a critical research direction, aiming to adapt general-purpose models to individual user preferences (Chen et al., 2024b; Li et al., 2025; Chen et al., 2025; Zhao et al., 2025). Among various approaches, the memory-retrieval framework is widely adopted for its interpretability and scalability. It retrieves user-specific signals from interaction history to guide the model without changing its parameters. Methods under this framework generally fall into two types: retrieval-augmented generation (RAG) and profile-augmented generation (PAG). RAG-based approaches retrieve relevant past interactions to construct a personalized prompt. For example, HYDRA (Zhuang et al., 2024) employs a personalized reranker to refine retrieval quality, while PERAL (Mysore et al., 2024) trains a retriever with a scale-calibrated objective to select useful information. In contrast, PAG-based methods summarize the user's behavior into a condensed profile, which is then integrated into the prompt to guide generation (Richardson et al., 2023).

Beyond retrieving individual histories, recent studies have explored incorporating other users' information as auxiliary signals to enhance individual personalization. CFRAG (Shi et al., 2025), Persona-DB (Sun et al., 2025), and AP-Bots (Yazan et al., 2025) borrow the concept of collaborative filtering (He et al., 2017; Wang et al., 2019) to retrieve similar users' histories and incorporate them into the prompt to guide the generation. DPL (Qiu et al., 2025) further highlights that individual uniqueness lies in the differences from others and proposes to model such differences by formulating inter-user comparison as a language modeling task performed directly by the LLM. While this method has shown promising results, modeling inter-user differences through prompt engineering poses challenges. In contrast, our method shifts this process to the latent embedding space, which avoids prompt-length constraints and enables more structured and nuanced modeling of user differences.

6 Conclusion

In this work, we propose DEP, a novel personalization framework that models inter-user differences in the latent embedding space to guide LLMs for personalized text generation. Unlike prior approaches that rely only on prompt-level construction to integrate user histories and inter-user contrastive signals, our method jointly encodes both user-specific and difference-aware embeddings, and refines them through a sparse autoencoder to retain only task-relevant personalization cues. These embeddings are then injected into a frozen LLM via soft prompts, enabling efficient personalization. Experimental results across multiple domains show that DEP achieves state-of-the-art performance, especially for users with distinctive behavior patterns, confirming the effectiveness of latent inter-user difference modeling. For future work, we plan to explore privacy-preserving inter-user comparison, real-time embedding updates, and extensions to tasks such as conversational agents.

Limitations

While our proposed method DEP demonstrates strong performance in personalized text generation, it also introduces several limitations. First, the method relies on sufficient user history to construct meaningful embeddings; in cold-start or data-sparse settings, its effectiveness may degrade. Second, although more efficient than language-based comparison methods, the computation of difference-aware embeddings and sparse autoencoding introduces additional overhead compared to standard prompting pipelines. Lastly, our evaluation is centered on review generation, where preferences are explicit; adapting the approach to broader tasks like dialogue or recommendation requires further study.

Ethical Statements

This work explores user-level personalization through the use of retrieved historical data and inter-user relational modeling. While effective for improving generation quality, such approaches raise important ethical considerations. In particular, accessing and processing users' historical interactions requires careful attention to data privacy, consent, and security. Moreover, modeling inter-user differences may inadvertently expose sensitive behavioral patterns or amplify existing biases.

To mitigate these concerns, any real-world deployment of our method should incorporate privacy-preserving techniques such as anonymization, encryption, and transparent consent protocols. Special care should be taken to avoid unintended inferences or misuse of user-level representations.

All experiments are conducted on publicly available datasets that have been preprocessed and released by prior work. The original raw data is open-source and distributed under the MIT license. We ensure that our use of the data adheres to established ethical standards and respects the original data usage guidelines.

References

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

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. Self-rag: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Steven Au, Cameron J Dimacali, Ojasmitha Pedirappagari, Namyong Park, Franck Dernoncourt, Yu Wang, Nikos Kanakaris, Hanieh Deilamsalehy, Ryan A Rossi, and Nesreen K Ahmed. 2025. Personalized graph-based retrieval for large language models. *arXiv preprint arXiv:2501.02157*.

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, pages 65–72.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*.

Jin Chen, Zheng Liu, Xu Huang, Chenwang Wu, Qi Liu, Gangwei Jiang, Yuanhao Pu, Yuxuan Lei, Xiaolong Chen, Xingmei Wang, and 1 others. 2024b. When large language models meet personalization: Perspectives of challenges and opportunities. *World Wide Web*, 27(4):42.

Ruizhe Chen, Xiaotian Zhang, Meng Luo, Wenhao Chai, and Zuozhu Liu. 2025. PAD: personalized alignment of llms at decoding-time. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. *Advances in Neural Information Processing Systems*, 36:28091–28114.

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. Association for Computational Linguistics.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2:1.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*,

pages 249–256. JMLR Workshop and Conference Proceedings.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*.

Robert Huben, Hoagy Cunningham, Logan Riggs, Aidan Ewart, and Lee Sharkey. 2024. Sparse autoencoders find highly interpretable features in language models. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*.

Caglar Irmak, Beth Vallen, and Sankar Sen. 2010. You like what i like, but i don't like what you like: Uniqueness motivations in product preferences. *Journal of Consumer Research*, 37(3):443–455.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022.

Ishita Kumar, Snigdha Viswanathan, Sushrita Yerra, Alireza Salemi, Ryan A Rossi, Franck Dernoncourt, Hanieh Deilamsalehy, Xiang Chen, Ruiyi Zhang, Shubham Agarwal, and 1 others. 2024. Longlamp: A benchmark for personalized long-form text generation. *arXiv preprint arXiv:2407.11016*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Xiaopeng Li, Pengyue Jia, Derong Xu, Yi Wen, Yingyi Zhang, Wenlin Zhang, Wanyu Wang, Yichao Wang, Zhaocheng Du, Xiangyang Li, and 1 others. 2025. A survey of personalization: From rag to agent. *arXiv preprint arXiv:2504.10147*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

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 arXiv:2502.11528*.

Jiongnan Liu, Yutao Zhu, Shuting Wang, Xiaochi Wei, Erxue Min, Yu Lu, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. 2024. Llms+ persona-plug= personalized llms. *arXiv preprint arXiv:2409.11901*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

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

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 188–197.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and 1 others. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Qiyao Peng, Hongtao Liu, Hongyan Xu, Qing Yang, Minglai Shao, and Wenjun Wang. 2024. Llm: Harnessing large language models for personalized review generation. *arXiv preprint arXiv:2407.07487*.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191. Association for Computational Linguistics.

Yilun Qiu, Xiaoyan Zhao, Yang Zhang, Yimeng Bai, Wenjie Wang, Hong Cheng, Fuli Feng, and Tat-Seng Chua. 2025. Measuring what makes you unique: Difference-aware user modeling for enhancing llm personalization. *arXiv preprint arXiv:2503.02450*.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 3505–3506.

Chris Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. 2023. Integrating summarization and retrieval for enhanced personalization via large language models. *arXiv preprint arXiv:2310.20081*.

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

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

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392. Association for Computational Linguistics.

Yongliang Shen, Kaitao Song, Xu Tan, Wenqi Zhang, Kan Ren, Siyu Yuan, Weiming Lu, Dongsheng Li, and Yueting Zhuang. 2024. Taskbench: Benchmarking large language models for task automation. In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Teng Shi, Jun Xu, Xiao Zhang, Xiaoxue Zang, Kai Zheng, Yang Song, and Han Li. 2025. Retrieval augmented generation with collaborative filtering for personalized text generation. *arXiv preprint arXiv:2504.05731*.

Charles R Snyder and Howard L Fromkin. 1977. Abnormality as a positive characteristic: The development and validation of a scale measuring need for uniqueness. *Journal of Abnormal Psychology*, 86(5):518.

Charles R Snyder and Howard L Fromkin. 2012. *Uniqueness: The human pursuit of difference*. Springer Science & Business Media.

Chenkai Sun, Ke Yang, Revanth Gangi Reddy, Yi Fung, Hou Pong Chan, Kevin Small, ChengXiang Zhai, and Heng Ji. 2025. Persona-DB: Efficient large language model personalization for response prediction with collaborative data refinement. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 281–296. Association for Computational Linguistics.

Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and 1 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Yiyan Xu, Jinghao Zhang, Alireza Salemi, Xinting Hu, Wenjie Wang, Fuli Feng, Hamed Zamani, Xiangnan He, and Tat-Seng Chua. 2025. Personalized generation in large model era: A survey. *arXiv preprint arXiv:2503.02614*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, and 1 others. 2024. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Mert Yazan, Suzan Verberne, and Frederik Situmeang. 2025. Improving rag for personalization with author features and contrastive examples. In *European Conference on Information Retrieval*, pages 408–416. Springer.

Yoel Zeldes, Amir Zait, Ilia Labzovsky, Danny Karmon, and Efrat Farkash. 2025. Commer: a framework for compressing and merging user data for personalization. *arXiv preprint arXiv:2501.03276*.

Jinghao Zhang, Yuting Liu, Wenjie Wang, Qiang Liu, Shu Wu, Liang Wang, and Tat-Seng Chua. 2025. Personalized text generation with contrastive activation steering. *arXiv preprint arXiv:2503.05213*.

Zhehao Zhang, Ryan A Rossi, Branislav Kveton, Yijia Shao, Diyi Yang, Hamed Zamani, Franck Dernoncourt, Joe Barrow, Tong Yu, Sungchul Kim, and 1 others. 2024. Personalization of large language models: A survey. *arXiv preprint arXiv: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 arXiv:2502.09597*.

Yuchen Zhuang, Haotian Sun, Yue Yu, Rushi Qiang, Qifan Wang, Chao Zhang, and Bo Dai. 2024. Hydra: Model factorization framework for black-box llm personalization. In *Advances in Neural Information Processing Systems*, volume 37, pages 100783–100815. Curran Associates, Inc.

A Dataset Details

In this paper, we focus on the task of review generation. Specifically, we adopt the Amazon (Hou et al., 2024) dataset preprocessed by DPL (Qiu et al., 2025). We select each user's most recent interaction from the training sets of the three categories and merge them into a unified training dataset, which is used to train the model. For validation, we also aggregate the three categories and randomly sample 512 instances. For testing, we directly use the test splits preprocessed by DPL. During data preprocessing, we construct complete prompts as model inputs by concatenating the target item title, target item description, output review title, output review rating, and the retrieved user's past reviews. For clarity, we provide an example of the dataset preprocessed by DPL as shown in Figure 4, and dataset statistics after processing are summarized in Table 4.

B Baseline Details

We compare our proposed DEP with several baseline methods. The comparison between different baselines and our method is shown in Table 5. In this section, we further introduce each baseline method in detail:

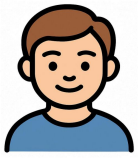
- **Non-Perso**: This method generates reviews without leveraging any user-specific information. The input to the model includes only the item's title and description, along with the output review's rating and title.

- **RAG** (Salemi et al., 2024): This method uses a simple recency-based retrieval strategy to select the most recent reviews from the user's history. The retrieved reviews are then directly formatted and incorporated into the LLM's input to provide contextual personalization.

- **PAG** (Richardson et al., 2023): Building upon RAG, this method first summarizes the most recent reviews from the user's history into a compact profile. The generated profile, along with the retrieved records, is included in the input to the LLM, allowing it to generate personalized reviews guided by a higher-level understanding of the user.

- **DPL** (Qiu et al., 2025): The method prompts the LLM to find inter-user differences by comparing the target user's most recent interactions with representative users selected via clustering from predefined dimensions (*e.g.,* writing, emitional tone, and semantics), and summarizes them with the user's history to form a user profile. This profile, along with recent reviews, is incorporated into the model input to enhance generation. To select representative users, DPL employs an embedding model; in our implementation, we use


```
{
  "user_id": "AHFGIFWFRGG4UKGJVMU5XJQH36RA",
  "profile": [
    {
      "asin": "B00005RDSN",
      "title": Flawed, but entertainig flick from Disney.
      "timestamp": 1645076985230,
      "rating": 4.0,
      "text": "It seems as though Disney always takes hits whenever it
    tries to do something daring and different..."
    },
    ...
  ],
  "data": {
    "asin": "B000NQRE1E",
    "title": "Actually lives up to the hype.",
    "timestamp": 1645076985230,
    "rating": 5.0,
    "text": "James Cameron\'s TITANIC was one of those movies that
   could have been a major disaster, what with all the bad word of
   mouth about its production process..."
  }
}
```


```
{
  "asin": "B00005RDSN",
  "title": "Titanic (Four-Disc Combo:
Blu-ray / DVD / Digital Copy)",
  "description": "Experience Academy
Award-winning director James
Cameron' s epic masterpiece Titanic
like never before..."
}
```

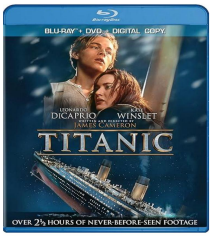



Figure 4: An example of the user review from the main dataset (above) and the corresponding item from the meta dataset (below).

Table 4: Overview of dataset statistics across the three benchmark categories.

Categories (↓)		#data	Profile Size	Output Length
Training Dataset		3996	37.47±33.53	1608.82±1476.99
Validation Dataset		512	39.14±36.01	1557.29±1378.43
Test Dataset	**Books**	317	34.84±22.55	1194.90±802.44
	Movies & TV	1925	41.11±35.90	1704.61±1752.44
	CDs & Vinyl	1754	38.50±32.37	1600.04±1419.89

the same embedding model as in our method.

- **PPlug** (Liu et al., 2024): A plug-and-play personalization method that encodes a user's history into a dense user-specific embedding through a lightweight user embedder. This embedding is constructed via input-aware attention over user histories. The resulting embedding, along with an instruction embedding, are projected into the LLM input space via a trainable projector and prepended to the input to guide a frozen LLM. In our implementation of PPlug, we adopt the same user embedder as used in our proposed method.

C Implementation Details

C.1 Running Environments

We implement all baseline methods and DEP with `Python 3.11.11`, `PyTorch`[6] (Paszke et al., 2019), `transformers`[7] (Wolf et al., 2020), and `vLLM`[8] (Kwon et al., 2023). To train the model, we utilize the `transformers` library. Besides, we employ the `vLLM` library as the inference engine for both validation and testing, and adapt our model accordingly to ensure compatibility.

C.2 Hyperparameter Configurations

C.2.1 Method Parameters

In our implementation, the SAE model is implemented as a two-layer feed-forward network, consisting of an encoder that projects input embeddings from dimension $d = 1024$ to a lower-dimensional latent space of size $d' = 512$, and a decoder that reconstructs the input. For the sparsity parameter ρ, we set it to 0.05. To align the SAE output with the LLM input space, we employ two independent projection networks $\mathcal{M}_{\text{his}}$ and $\mathcal{M}_{\text{diff}}$, each implemented as a two-layer MLP with GELU activations, mapping the latent representation z to the LLM embedding space. Additionally, we use $\lambda = 100$ and $\gamma = 1\text{e}{-3}$ to balance the reconstruction and sparsity losses during training.

A maximum of 8 user history entries are retrieved for each instance. If the input exceeds the context length limit, excess histories are discarded to ensure compatibility.

[6] https://pytorch.org/
[7] https://huggingface.co/
[8] https://github.com/vllm-project/vllm

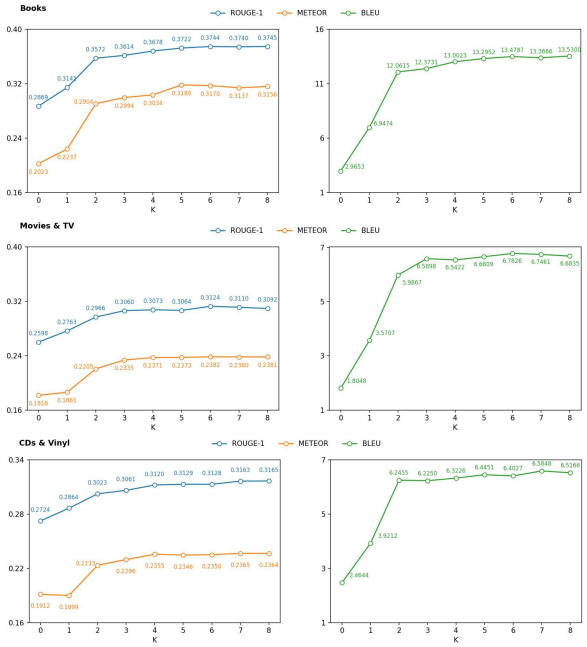


Figure 5: Detailed evaluation results across all three datasets (Books, Movies & TV, CDs & Vinyl) with varying numbers of retrieved user histories (K). The left figure shows ROUGE-1 and METEOR scores, and the right figure demonstrates BLEU scores.

C.2.2 Training Settings

Before training, we initialize the model parameters using Xavier uniform initialization (Glorot and Bengio, 2010). We train the model using the `AdamW` (Loshchilov and Hutter, 2019) optimizer for a maximum of 8 epochs. The learning rate is set to 1e-5 with a weight decay of 0.025. We apply a warmup ratio of 0.01 at the beginning of training. The batch size per device is 1, and the gradient accumulation steps are 16 to achieve an effective batch size of 16. We also enable `bfloat16` mixed precision and incorporate flash attention (Dao, 2023). Additionally, the training is conducted using DeepSpeed[9] (Rajbhandari et al., 2020; Rasley et al., 2020) ZeRO Stage 1 optimization.

C.2.3 Inference Settings

We configure the model with a maximum length of 2048 tokens for both input and output. During inference for both validation and test, the temperature is set to 0.8, and the parameter top_p is 0.95.

Table 5: We provide a comparison between the different baseline methods and our proposed DEP, focusing on the following aspects: (1) retrieval augmentation, (2) embedded representation, and (3) inter-user difference.

Methods (↓)	Retrieval Augmentation	Embedded Representation	Inter-User Difference
Non-Perso	✗	✗	✗
RAG	✓	✗	✗
PAG	✓	✗	✗
DPL	✓	✗	✓
PPlug	✗	✓	✗
DEP	✓	✓	✓

Table 6: Complete ablation study on different configurations of user embeddings.

	Datasets (→)	**Books**			**Movies & TV**			**CDs & Vinyl**		
	Methods (↓)	R-1	MET.	BL.	R-1	MET.	BL.	R-1	MET.	BL.
	Non-Perso-7B	0.2907	0.1735	1.9766	0.2469	0.1503	0.7242	0.2604	0.1561	1.0997
w/o text	his_emb	0.2912	0.1718	2.4364	0.2545	0.1625	1.7048	0.2726	0.1711	2.1962
	diff_emb	0.3022	0.1839	2.6648	0.2542	0.1546	0.8574	0.2690	0.1616	1.2601
	his_emb + diff_emb	0.2970	0.2227	5.5622	0.2586	0.1871	3.5629	0.2713	0.1853	3.3092
w/ text	his_emb	0.3722	0.3110	12.9361	0.3026	0.2332	6.0120	0.3051	0.2268	5.3390
	diff_emb	0.3596	0.2781	10.6435	0.2964	0.2128	5.1985	0.3049	0.2108	4.9141
	his_emb + diff_emb (ours)	**0.3745**	**0.3156**	**13.5300**	**0.3092**	**0.2381**	**6.6835**	**0.3165**	**0.2364**	**6.5166**

Table 7: Complete ablation study on representation refinement.

Datasets (→)	**Books**			**Movies & TV**			**CDs & Vinyl**		
Methods (↓)	R-1	MET.	BL.	R-1	MET.	BL.	R-1	MET.	BL.
w/o DR	0.3704	0.3016	13.3651	0.3091	0.2325	6.5149	0.3039	0.2283	5.6812
w/ AE	0.3691	0.2994	12.5453	0.3084	0.2350	6.5949	0.3167	0.2355	6.4352
w/ SAE	**0.3745**	**0.3156**	**13.5300**	**0.3092**	**0.2381**	**6.6835**	**0.3165**	**0.2364**	**6.5166**

D Complete Ablation Studies & In-Depth Analysis

D.1 User Embedding Configuration

In this section, we provide the complete results for different user embedding configurations evaluated in our ablation study. While the main paper only reports METEOR scores in Table 2, we include here the full results for all three metrics (ROUGE-1, METEOR, and BLEU) across all datasets. The results in Table 6 offer a more comprehensive view of how different embedding types (*his_emb*, *diff_emb*) and the presence or absence of retrieved text affect personalization performance.

[9] https://github.com/deepspeedai/DeepSpeed

D.2 Representation Refinement

This section presents the complete results for the different representation refinement strategies discussed in our ablation study. Table 7 reports ROUGE-1, METEOR, and BLEU scores for the *w/o DR*, *w/ AE*, and *w/ SAE* settings across all datasets, providing a more detailed understanding of their relative effectiveness.

D.3 Impact of History Number

We provide the full results across all evaluation metrics in Figure 5. As shown in the figure, all three evaluation metrics (ROUGE-1, METEOR, and BLEU) exhibit a consistent upward trend across the three datasets as the number of retrieved histories (K) increases. This improvement can be attributed to the additional contextual information provided

Table 8: Performance comparison between different retrieval strategies across the three datasets.

Datasets (→)	**Books**			**Movies & TV**			**CDs & Vinyl**		
Methods (↓)	R-1	MET.	BL.	R-1	MET.	BL.	R-1	MET.	BL.
Random	0.3287	0.2573	5.4657	0.2955	0.2125	2.6946	0.3064	0.2138	2.9218
BM25	0.3325	0.2650	5.9851	0.2953	0.2123	2.7802	0.3066	0.2148	2.9832
Contriever	0.3325	0.2608	5.7479	0.2958	0.2128	2.7584	0.3077	0.2160	3.0204
Recency	**0.3404**	**0.2735**	**6.8178**	**0.2983**	**0.2142**	**2.8680**	**0.3092**	**0.2177**	**3.1588**

Table 9: Performance comparison with and without system prompt guidance.

Datasets (→)	**Books**			**Movies & TV**			**CDs & Vinyl**		
Methods (↓)	R-1	MET.	BL.	R-1	MET.	BL.	R-1	MET.	BL.
w/o Guidance	0.3704	0.3016	13.3651	0.3091	0.2325	6.5149	0.3039	0.2283	5.6812
w/ Guidance	**0.3745**	**0.3156**	**13.5300**	**0.3092**	**0.2381**	**6.6835**	**0.3165**	**0.2364**	**6.5166**
+Improvement	*0.0041*	*0.0140*	*0.1649*	*0.0001*	*0.0056*	*0.1686*	*0.0126*	*0.0081*	*0.8354*

by retrieved histories, along with our injected user-specific embedding and difference-aware embedding. Notably, the most significant gains occur when K increases from 0 to 3, especially for the BLEU metric. Beyond this range, the performance tends to plateau, with only marginal improvements or slight fluctuations. A slight dip is observed in METEOR on the *CDs & Vinyl* dataset when K increases from 0 to 1, which may result from noise or limited informativeness in the single retrieved history. As more histories are incorporated, the signal becomes more stable and representative, leading to consistent improvements.

Overall, these results demonstrate that our method substantially enhances the RAG pipeline. The retrieve-and-inject paradigm we adopt proves to be a strong and effective framework for personalization.

E Additional Experiment & Analysis

E.1 Retrieval Method

To investigate the impact of different retrieval strategies and identify the most effective one for use in both the baselines and our method, we evaluate four retrieval approaches: random, BM25 (Robertson et al., 2009), Contriever (Izacard et al., 2022), and recency (the most recent). Experiments are conducted using the `Qwen2.5-32B-Instruct` model, and the results are presented in Table 8.

As shown in Table 8, the choice of retrieval strategy has a notable impact on generation performance. The random retrieval baseline yields the lowest performance, indicating the importance of relevant context in guiding generation. BM25 and Contriever perform comparably, with slight advantages in different metrics. Among the four methods evaluated, the recency-based retrieval consistently outperforms the others across all metrics. Based on these results, we adopt the recency retrieval strategy in all subsequent experiments.

E.2 System Prompt Guidance

As described in Section 6, we incorporate additional information into the system prompt to help the model better understand the injected personalization prompts. To assess its effectiveness, we conduct experiments to analyze the impact of this guidance. Table 9 reports the results across all datasets and evaluation metrics. We observe that incorporating system prompt guidance consistently improves performance across the board. Hence, we adopt the system prompt guidance by default in all experiments.

F Overview of Templates & Prompts

In this section, we illustrate the prompt design used in our framework. As shown in Figure 6, the upper part depicts the system prompt, which defines the model's global behavior and task instruction. The lower part shows an example of the input prompt, including retrieved user histories and object descriptions, which are fed into the model for generation. This prompt structure follows the retrieve-and-inject paradigm, where both user-specific and difference-aware embeddings are embedded via soft

Template

Given the title and description of an item, along with the user's past reviews (including item title, item description, review rating, review title, review text, review embedding, review difference embedding), and the output review rating and review title, generate a personalized item review for the user.
Note: [Review Embedding] denotes a soft prompt of the review text and [Review Difference Embedding] denotes a soft prompt showing the difference between the review text and other reviews on the same item. [Review Embedding] and [Review Difference Embedding] should serve as hints for personalized review text generation.

Input

[Item Title]: <*item title*>
[Item Description]: <*item description*>
[User' s Past Reviews]:

- [Review <*i*>]:
  - [Item Title]: <*item title*>
  - [Item Description]: <*item description*>
  - [Review Rating]: <*review rating*>
  - [Review Title]: <*review title*>
  - [Review Text]: <*review text*>
  - [Review Embedding]:
<his_token_start>[HIS_TOKEN_<*i*>]<his_token_end>
  - [Review Difference Embedding]:
<diff_token_start>[DIFF_TOKEN_<*i*>]<diff_token_end>

(K blocks are concatenated by the new line symbol)

[Output Review Rating]: <*output review rating*>
[Output Review Title]: <*output review title*>

Figure 6: The system prompt template and input template for the base LLM.

prompts *[HIS_TOKEN_i]* and *[DIFF_TOKEN_i]* to guide the generation. The four special tokens <*his_token_start*>, <*his_token_end*>, <*diff_token_start*>, and <*diff_token_end*> are introduced to explicitly mark the boundaries of user-specific and difference-aware embeddings in the input sequence.

G Case Study

In this section, we present a case study to illustrate the output generated by our framework as shown in Figure 7.

In this example, the review generated by DEP closely aligns with the user's real review in both content and sentiment. Both reviews highlight the central observation that *Avengers: Age of Ultron* feels very similar to the first Avengers movie, with the main difference being the villain. Moreover, DEP incorporates additional signals such as the user's familiarity with the franchise and a moderately positive tone that matches the provided 3-star rating. This case demonstrates that DEP can generate reviews that are not only coherent but also well-aligned with the user's original opinion, supporting the effectiveness of difference-aware modeling in the embedding space for personalization.

[Item Title]: Marvel's Avengers: Age of Ultron
[Review Rating]: 3.0
[Review Title]: Same as Avengers One with different bad guys

DEP Generated Review:
I went to see this movie because of the previous reviews I read here. I liked the first one, but I didn't think it was a great movie, but I'm really glad I saw it. I loved this one. I thought it was a much better movie. I've seen all the Avengers movies so I was already familiar with the characters, so that helped me get into the movie more. It's basically the same story as the first one, but with a different bad guy. The same thing happens. Avengers team together, save the day. In this case the bad guy is a terrible AI robot, who has been programmed to think it can save the world. It doesn't work of course, but the Avengers do save the day. This movie was very good. I would recommend it. 3 stars.

User’s Real Review:
If you saw the first Avengers and most people have, the second will look very, very familiar. The first movie made over $1.5 Billion at the box office. Not being fools, Disney and Marvel chose to stay very close to the same formula for the second. Once again the good guys battle each other, then realize the importance of team work before finally being able to combat and destroy the enemy. In the first movie it was Loki and the Chitauri. In the second it's Ultron and the Ultron clones. The clones by the way are like metal piñatas. They blow up and explode very easily when hit. One terminator would be more challenging than 100 clones. But Avengers 2 also has the usual back and forth one lines and joking banter between the team members, the usual "end of the world" threat and the usual deep sigh "boy that was a close one" ending. I'm sure Avengers 3 will also be very similar to the first two. Between one and two, I preferred one. But the CGI is good and the popcorn was OK.

Figure 7: A case study which compares the DEP-generated review and the user’s real review for the item *Avengers: Age of Ultron*.