Hybrid Dual-Semantics Modeling for Enhancing Large Language Model Based Recommendation

Canyi Liu
Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University
Xiamen, Fujian, China
liucanyi01@stu.xmu.edu.cn

Tianyi Li
Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University
Xiamen, Fujian, China
litianyi@stu.xmu.edu.cn

Wei Li
School of Electronic and Computer Engineering, Peking University
Shenzhen, Guangzhou, China
liwei25@stu.pku.edu.cn

Youchen Zhang
University of California
Los Angeles, California, United States
vzhang996@g.ucla.edu

Xiaodong Li
Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University
Xiamen, Fujian, China
xdli@xmu.edu.cn

Hui Li*
Key Laboratory of Multimedia Trusted Perception and Efficient Computing, Ministry of Education of China, Xiamen University
Xiamen, Fujian, China
hui@xmu.edu.cn

Abstract

The blossoming of large language models (LLMs) has greatly shifted the paradigm of Sequential Recommender System (SRS). Numerous studies have attempted to integrate ID-based collaborative signals and text information for effectively capturing both ID semantics and text semantics to enhance LLM-based recommendation. However, existing fusion methods suffer from challenges like fusion noise and the semantic gap. To address these issues, we propose Hybrid Dual-Semantics Modeling for enhancing LLM-based Recommendation (HDRec), an effective hybrid fusion method based on a design of dual low-rank adaptation (LoRA). HDRec employs two LoRAs processes on a shared LLM decoder, with each process handling information from one of the two semantics. We further implement a dedicated fusion mechanism exclusively at the inference stage, allowing the robust textual representation to serve as the primary signal, which is adaptively enhanced by unique collaborative signals from ID semantics, ensuring stable and accurate final predictions. To mitigate gradient conflicts caused by the dual LoRA processes, we introduce the alternating training of dual low-rank adaptation strategy. This method effectively resolves gradient conflicts and enables successful optimization of HDRec. Extensive experiments show that HDRec outperforms existing non-LLM-based and LLM-based state-of-the-art methods. The implementation of HDRec is anonymously available at https://github.com/KDEGroup/HDRec.

CCS Concepts

• **Information systems → Recommender systems**.

*Corresponding author.

WSDM '26, Boise, ID, USA

ACM ISBN 979-8-4007-2292-9/2026/02
https://doi.org/10.1145/3773966.3777943

Keywords

sequential recommendation; recommender system

ACM Reference Format:
Canyi Liu, Tianyi Li, Wei Li, Youchen Zhang, Xiaodong Li, and Hui Li. 2026. Hybrid Dual-Semantics Modeling for Enhancing Large Language Model Based Recommendation. In *Proceedings of the Nineteenth ACM International Conference on Web Search and Data Mining (WSDM '26), February 22–26, 2026, Boise, ID, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3773966.3777943

1 Introduction

Sequential recommender system (SRS) suggests items that may interest users by modeling user historical interaction sequences. Aiming at helping users find items more efficiently and service providers better promote items, SRS has been applied in a wide variety of online applications including but not limited to online shopping (e.g., Amazon), music listening (e.g., Spotify) and news reading (e.g., Yahoo News) [2].

There has been a tremendous amount of work on SRS in recent decades [18, 24, 25]. In particular, deep learning techniques have greatly boosted the development of SRS [2]. Nevertheless, deep learning-based SRS still face various challenges. One representative issue is that, due to model scale and data size [35], deep learning-based SRS models are limited in their understanding of the context of the provided data (e.g., item descriptions), hindering their ability to leverage broad world knowledge.

Thanks to the rise of large language models (LLMs) [34], the aforementioned issue can be alleviated by introducing LLMs into SRS. Firstly, sequential recommendation data shares some similarities with natural language text in that they are both sequential and represented by individual items/tokens. It is natural to consider modeling sequential recommendation in a way similar to language modeling and leverage LLMs to capture complex sequential item relations. Moreover, items in contemporary SRS are typically accompanied by textual information (e.g., item titles, attributes and descriptions) that can be better understood by LLMs with strong

world knowledge beyond the context of SRS. Therefore, there is a surge of work on applying LLMs in SRS [8, 14, 15, 35].

Numerous studies have attempted to integrate ID-based collaborative signals and text information for effectively capturing both ID semantics and text semantics to enhance LLM-based recommendation [10, 13, 23, 26]. Early fusion methods that fuse ID and text embeddings before feeding them to learning modules and late fusion methods that fuse hidden representations of ID and text after feeding to learning modules are two representative semantics fusion paradigms.[1] However, in SRS, the semantic gap between ID-based collaborative signals and item text may introduce fusion noise during early fusion. While late fusion methods circumvent fusion noise, they face multiple challenges in cross-semantics modeling.

To address the aforementioned issues, in this paper, we propose Hybrid Dual-Semantics Modeling for enhancing LLM-based Recommendation (HDRec), an effective hybrid fusion method based on a design of dual low-rank adaptation (LoRA) [6] processes. HDRec not only achieves fine-grained cross-semantics alignment but also effectively handles the semantics gap problem. Specifically, we employ two LoRAs processes on a shared LLM Decoder, with each process handling information from one of the two semantics. Crucially, while our training process aligns ID and text representations by enabling each of them to optimize independently for the corresponding loss, we implement a dedicated fusion mechanism exclusively at the inference stage. This strategy allows the robust textual representation to serve as the primary signal, which is adaptively enhanced by unique collaborative signals from ID semantics, ensuring stable and accurate final predictions.

Nevertheless, the above novel design presents another challenge: simultaneously training two LoRA processes on a shared decoder can lead to gradient conflicts. To mitigate the negative impact on overall optimization, we further introduce alternating training of dual low-rank adaptation for our hybrid fusion approach. This method effectively resolves gradient conflicts and enables successful optimization of HDRec.

In summary, our main contributions of this work are:

- We propose a novel process for fusing ID and text semantics in LLM-based recommendation. This process achieves fine-grained cross-semantics alignment without requiring complex alignment procedures during training, while maintaining a relatively small number of fine-tuned parameters.
- We introduce alternating training of dual low-rank adaptation, a novel training strategy tailored to the effective fine-tuning of HDRec. It addresses the gradient conflicts arising from the simultaneous updates of the two LoRA processes on the shared decoder.
- We demonstrate a sophisticated inference-time fusion strategy that adaptively combines the strong textual understanding of LLMs with the precise collaborative signals from ID semantics, enhancing prediction stability and accuracy.

We have conducted extensive experiments on recommendation benchmarks. Experimental results show that HDRec significantly outperforms state-of-the-art non-LLM-based and LLM-based recommender systems.

[1] Details are discussed in Sec. 3.3.

2 Problem Definition

SRS comprises a set of users U and a set of items I. The user-item interaction history of a user $u \in U$ can be represented by $S_u = [i^{(1)}, \ldots, i^{(n)}]$, where each element i indicates an item in I and items in the sequence are sorted in chronological order. We can use a tuple $i = \langle i_{\text{id}}, i_{\text{text}} \rangle$ to represent an item i, where i_{id} serves as a unique identifier (ID) and i_{text} represents textual features, including but not limited to item titles and descriptors. The goal of SRS is to predict the next item $i^{(|S_u|+1)}$ that user u may interact with.

3 Our Method HDRec

3.1 Overview

Fig. 1 provides an overview of HDRec. In the following sections, we will first illustrate $\text{HDRec}_{\text{base}}$, the basic version of HDRec, that adapts LLM to sequential recommendation (Sec. 3.2). $\text{HDRec}_{\text{base}}$ consists of an LLM decoder and an item projection layer for mapping the LLM space to the item space. We design two adaptation tasks to endow $\text{HDRec}_{\text{base}}$ with the ability to adapt LLM for sequential recommendation and apply LoRA for parameter-efficient fine-tuning.

On top of $\text{HDRec}_{\text{base}}$, HDRec integrates the following enhancements to effectively incorporate collaborative filtering signals:

- In Sec. 3.3, we propose a hybrid dual-semantics modeling mechanism where HDRec integrates both collaborative filtering signal and textual attributes through two LoRA processes in a single LLM-based SRS, maintaining isolated parameter spaces for ID and textual data processing.
- In Sec. 3.4, we devise an interleaved optimization strategy that alternately updates the two LoRA processes through partitioned gradient propagation cycles, effectively preventing parameter interference during backpropagation.
- In Sec. 3.5, we implement a specialized inference-time fusion mechanism to combine the robust text-based predictions with calibrated ID-derived collaborative signals, ensuring stable and accurate final recommendations by leveraging their complementary strengths.

3.2 Adapt LLM for Sequential Recommendation

We first design two adaptation tasks for $\text{HDRec}_{\text{base}}$ and adapt LLM to sequential recommendation.

3.2.1 Task 1: Semantic Alignment. In SRS, Items are typically represented by IDs (e.g. "B00EFRN2KY"). However, these item IDs are rare in the training corpora of LLM. Hence, given the context of the recommendation, it is difficult for LLM to understand item IDs. To improve the understanding of LLM on item IDs, we implement an item semantic alignment task to align LLM's understanding of item IDs. Specifically, for each item i, we construct the input sequence as follows:

$$X_i = \{\text{Prompt}_{\text{align}},\ i_{\text{text}},\ [\text{ALIGN}]\} \tag{1}$$

where i_{text} denotes item's textual features and [ALIGN] is a learnable alignment token.

The hidden state $\mathbf{h}^i_{\text{ALIGN}}$ of the LLM's last layer corresponding to [ALIGN] is then projected through an item projection layer (IPL, a multilayer feedforward network) to produce item probability distributions. $\mathbf{h}^i_{\text{ALIGN}}$ is regarded as the learned representation of

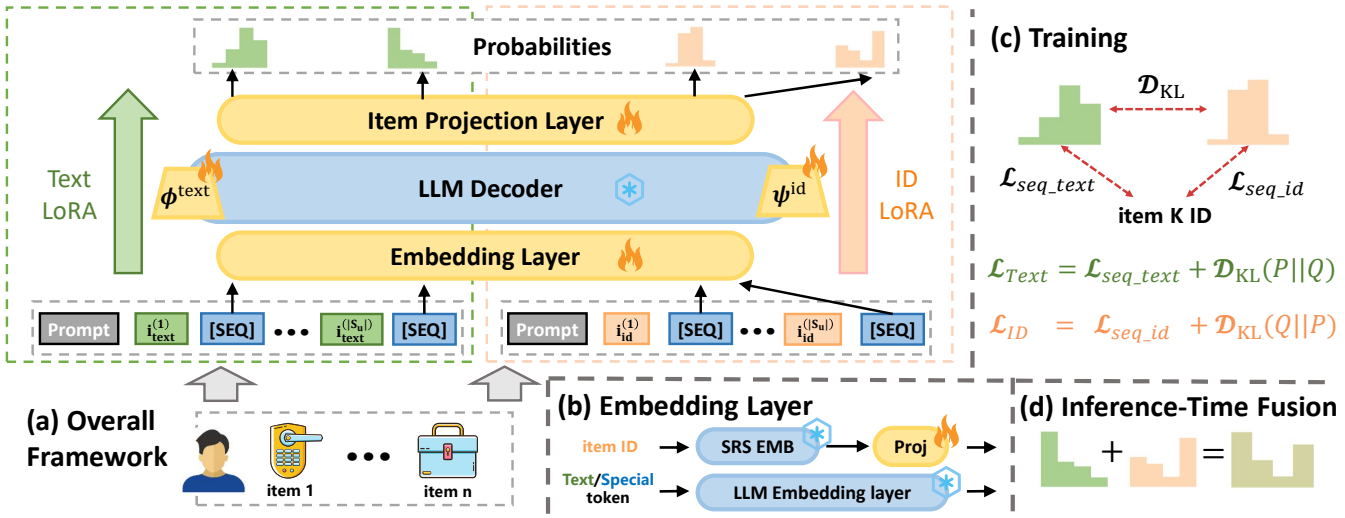

Figure 1: Overview of HDRec.

i, similar to using the representation of a special token [CLS] to represent a sentence when using a pre-trained model in the text classification task [1, 16].

During optimization, we use cross-entropy loss between the predicted probability distributions of item ID and actual item ID to align hidden states and corresponding item IDs:

$$\begin{aligned} \mathbf{h}^i_{\text{ALIGN}} &= \text{LLM}(\mathbf{e}^i_{\text{ALIGN}}), \\ \hat{\mathbf{y}}^i &= \text{Softmax}\left(\text{IPL}(\mathbf{h}^i_{\text{ALIGN}})\right), \\ \mathcal{L}_{\text{align}} &= -\sum_i \delta(i^i_{\text{id}}) \log \hat{\mathbf{y}}^i. \end{aligned} \tag{2}$$

where $\text{IPL}(\cdot)$ denotes the item projection layer, and $\delta(i^i_{\text{id}})$ denotes the one-hot encoding of item ID for item i.

Through semantic alignment, $\text{HDRec}_{\text{base}}$ can capture the relationship between item ID and the recommendation context where item ID appears, thus gaining a better understanding of the items.

Following is an example of the semantic alignment task:

Instruction:
Provide the item's hidden state for [ALIGN]. Item Description: title: The Sims 4 Premium Edition, brand: Electronic Arts [ALIGN]
Response:
<69>

3.2.2 Task 2: Link Next Item By Text. We further endow LLM with the ability to predict next items. Each item i in the historical interaction sequence S_u of the user u can be denoted by its item's textual representation i_{text}. We append a special token [SEQ] to the end of each i_{text} to separate items and construct the sequential textual data for u:

$$X^{\text{text}}_u = \left\{\text{Prompt}_{\text{seq}}, i^{(1)}_{\text{text}}, [\text{SEQ}], \ldots, i^{(k)}_{\text{text}}, [\text{SEQ}], \cdots, i^{(|S_u|)}_{\text{text}}, [\text{SEQ}]\right\} \tag{3}$$

X^{text}_u is then fed into LLM, and $\text{HDRec}_{\text{base}}$ uses IPL to map the hidden state $\mathbf{h}^{u,(k)}_{\text{SEQ-text}}$ from the last LLM layer that corresponds to the current item i to the next item ID.

We use cross-entropy loss between predicted probability distributions of next item ID and actual next-item ID during optimization:

$$\begin{aligned} \mathbf{h}^{u,(k)}_{\text{SEQ-text}} &= \text{LLM}\left(\mathbf{e}^u_{\text{SEQ-text}}; (k)\right), \\ \hat{\mathbf{z}}^{u,(k)}_{\text{text}} &= \text{Softmax}\left(\text{IPL}(\mathbf{h}^{u,(k)}_{\text{SEQ-text}})\right), \\ \mathcal{L}_{\text{seq\_text}} &= -\sum_u \sum_{k=1}^{|S_u|-1} \delta(i^{u,(k+1)}_{\text{id}}) \log \hat{\mathbf{z}}^{u,(k)}_{\text{text}}. \end{aligned} \tag{4}$$

where $\delta(i^{u,(k)}_{\text{id}})$ denotes the one-hot encoding of the item ID of k-th item in u's interaction sequence.

Following is an example of linking next item by text:

Instruction:
Provide the next item's hidden state for each [SEQ]. User's history: title: Syberia Collector's Edition 1 & 2, brand: Encore [SEQ] title: The Sims 4 Premium Edition, brand: Electronic Arts [SEQ]
Response:
<69> <0>

3.3 Hybrid Dual-Semantics Modeling

LLM serves as an effective sequential model in $\text{HDRec}_{\text{base}}$. However, $\text{HDRec}_{\text{base}}$ only captures sequential patterns by modeling textual features. Numerous studies have shown the importance of modeling ID semantics in SRS [10, 13, 36, 39]. Although LLMs excel at capturing sequential patterns from text, they may not adequately capture collaborative knowledge in SRS, especially from long input sequences, due to their inherent design for modeling textual semantics rather than ID-based collaborative signals [10]. This necessitates a tailored strategy for modeling ID-based collaborative signals together with textual features in LLM-based SRS.

Previous methods [10, 13] primarily employ early-fusion strategies (Fig. 2(c)) that fuse embeddings of item ID and text features before feeding them into SRS. However, in SRS, the semantic gap between ID-based collaborative signals and item text may introduce fusion noise during early fusion, a challenge that our empirical

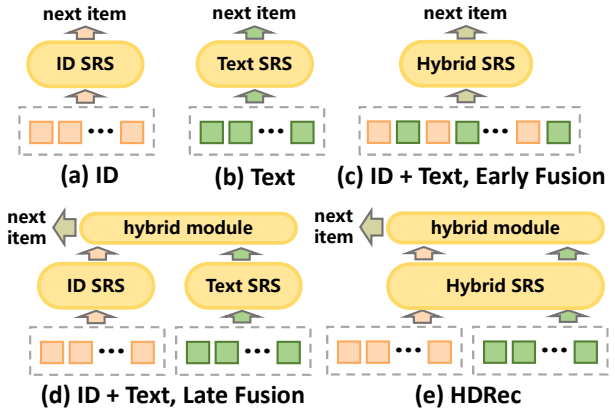

Figure 2: Comparisons among different fusion methods.

study in Sec. 4.3.2 also corroborates. To address this limitation, existing approaches typically require using sophisticated semantics alignment operations [10, 13]. Another direction is to fuse hidden states learned from two independent semantics-specific learning modules, the so-called late fusion, as shown in Fig. 2(d). While late fusion methods circumvent fusion noise, they face multiple challenges in cross-semantics modeling: (1) The discrepancy between ID modeling module and text modeling module leads to imbalanced learning, resulting in low quality of ID representations (evidenced by the empirical study in Tab. 4 of Sec. 4.3.2). (2) Employing an LLM architecture for both modeling modules increases the overall model size; (3) Logits-level fusion, which typically combines outputs at a coarse granularity, fails to preserve fine-grained cross-semantics correlations, thereby losing the benefits of early fusion.

To resolve these problems, we propose a hybrid dual-semantics modeling method (HDSM), as shown in Fig. 2(e). Specifically, we implement two adapters within a shared decoder and apply two LoRA process. Each LoRA process efficiently fine-tunes one adapter, achieving semantics-specific adaptation while maintaining a low update cost. Furthermore, through this parameter-sharing mechanism, HDRec enables deeper and more fine-grained interactions between ID and text semantics within the shared decoder. This allows the model to capture fine-grained cross-semantics correlations, similar to early fusion, because they are jointly processed in a common parameter space. Simultaneously, since the initial representations of ID and text are independently processed by their respective adapters, it avoids the semantics noise that early fusion may introduce, thereby also inheriting late fusion's advantage of circumventing fusion noise.

3.3.1 Task 3: Link Next Item By ID. To instruct LLM to capture ID-based collaborative signals together with textual features, we derive the Link Next Item By ID task by simply substituting textual representations in Task 2 of Sec. 3.2.2 with ID embeddings:

$$X_{u^{\text{id}}} = \left\{\text{Prompt}_{\text{seq}}, i_{\text{id}}^{(1)}, [\text{SEQ}], \ldots, i_{\text{id}}^{(k)}, [\text{SEQ}], \cdots, i_{\text{id}}^{(|S_u|)}, [\text{SEQ}]\right\},$$

$$\hat{\mathbf{z}}_{\text{id}}^{u,(k)} = \text{Softmax}\left(\text{IPL}(\mathbf{h}_{\text{SEQ-id}}^{u,(k)})\right), \tag{5}$$

$$\mathcal{L}_{\text{seq\_id}} = -\sum_{u}\sum_{k=1}^{|S_u|-1}\delta(i_{\text{id}}^{u,(k+1)})\log\hat{\mathbf{z}}_{\text{id}}^{u,(k)}.$$

Notably, the projection layer IPL$(\cdot)$ shares parameters with the one used in modeling text semantics (Eq. 2).

Following is a concrete example for linking next item by ID:

Instruction:
Provide the next item's hidden state for each [SEQ]. User's history:
<68> [SEQ] <69> [SEQ]
Response:
<69> <0>

3.3.2 Embeding Layer. The input of HDRec contains both text tokens and ID tokens. The embedding layer (Fig. 1(b)) in HDRec processes these tokens as follows:

- **Text Tokens**: For textual data, we utilize the pre-trained word embedding layer from the LLM to obtain token embeddings:

$$\mathbf{e}_{\text{text}} = \text{LLM}_{\text{EMB}}(token_{\text{text}}) \tag{6}$$

- **ID Tokens**: For ID data, we first retrieve the ID representations from a pre-trained SRS (e.g., SASRec [9]) to capture collaborative signals. To preserve the collaborative knowledge learned by the pre-trained SRS while aligning ID representations with the LLM's semantic space, we freeze the SRS embedding layer and employ a trainable projector (a multi-layer feed-forward network):

$$\mathbf{e}_{\text{id}} = \text{Proj}\left(\text{SRS}_{\text{EMB}}(token_{\text{id}})\right) \tag{7}$$

3.3.3 Cross-semantics Alignment. For the k-th interaction of user u, we generate semantics-specific representations $\hat{\mathbf{z}}_{\text{text}}^{u,(k)}$ and $\hat{\mathbf{z}}_{\text{id}}^{u,(k)}$ through a shared LLM with independent low-rank adaptation processes:

$$\begin{aligned}
\mathbf{h}_{\text{SEQ-text}}^{u,(k)} &= \text{LLM}\left(\mathbf{e}_{\text{SEQ-text}}^{u}; (k); \phi^{\text{text}}\right),\\
\hat{\mathbf{z}}_{\text{text}}^{u,(k)} &= \text{Softmax}\left(\text{IPL}(\mathbf{h}_{\text{SEQ-text}}^{u,(k)})\right);\\
\mathbf{h}_{\text{SEQ-id}}^{u,(k)} &= \text{LLM}\left(\mathbf{e}_{\text{SEQ-id}}^{u}; (k); \psi^{\text{id}}\right),\\
\hat{\mathbf{z}}_{\text{id}}^{u,(k)} &= \text{Softmax}\left(\text{IPL}(\mathbf{h}_{\text{SEQ-id}}^{u,(k)})\right).
\end{aligned} \tag{8}$$

where ϕ^{text} and ψ^{id} represent the LoRA introduced incremental weight matrices specifically trained for the text and ID semantics, respectively.

To achieve cross-semantics alignment, HDRec employs bidirectional Kullback–Leibler (KL) divergence during optimization (visualized in Fig. 1(c)). When optimizing ϕ^{text} (with ψ^{id} frozen):

$$\begin{aligned}
p_{k,i} &= (\hat{\mathbf{z}}_{\text{text}}^{u,(k)})_i, \quad q_{k,i} = (\hat{\mathbf{z}}_{\text{id}}^{u,(k)})_i\\
\mathcal{D}_{\text{KL}}(\hat{\mathbf{z}}_{\text{id}}^{u,(k)} \,\|\, \hat{\mathbf{z}}_{\text{text}}^{u,(k)}) &= \frac{1}{|S_u|}\sum_{k=1}^{|S_u|}\sum_{i=1}^{|I|} q_{k,i}(\log q_{k,i} - \log p_{k,i}),
\end{aligned} \tag{9}$$

This asymmetric alignment forces the text semantics $\hat{\mathbf{z}}_{\text{text}}$ to converge towards the ID semantics $\hat{\mathbf{z}}_{\text{id}}$ via backpropagation through ϕ^{text} only.

The complete training objective for ϕ^{text} integrates:

$$\mathcal{L}_{\text{Text}} = \mathcal{L}_{\text{seq\_text}} + \lambda \cdot \mathcal{D}_{\text{KL}}(\hat{\mathbf{z}}_{\text{id}}^{u,(k)} \,\|\, \hat{\mathbf{z}}_{\text{text}}^{u,(k)}) \tag{10}$$

In contrast, when updating ψ^{id} (with ϕ^{text} fixed):

$$\mathcal{D}_{\text{KL}}(\hat{\mathbf{z}}_{\text{text}}^{u,(k)} \,\|\, \hat{\mathbf{z}}_{\text{id}}^{u,(k)}) = \frac{1}{|S_u|}\sum_{k=1}^{|S_u|}\sum_{i=1}^{|I|} p_{k,i}(\log p_{k,i} - \log q_{k,i}) \tag{11}$$

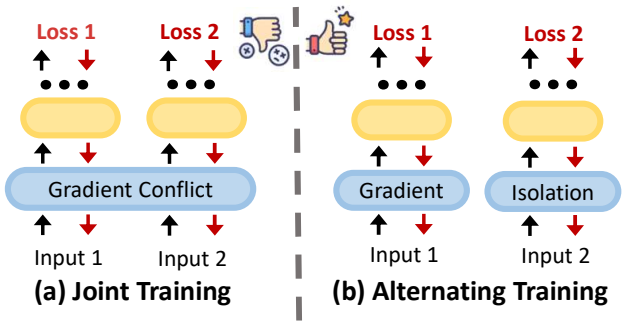

Figure 3: Comparison of (a) joint training with gradient conflict versus (b) our alternating training protocol.

yielding the ID-semantics loss:

$$\mathcal{L}_{\text{ID}} = \mathcal{L}_{\text{seq\_id}} + \lambda \cdot \mathcal{D}_{\text{KL}}(\hat{\mathbf{z}}_{\text{text}}^{u,(k)} \parallel \hat{\mathbf{z}}_{\text{id}}^{u,(k)}). \tag{12}$$

where λ is a hyper-parameter for balancing.

3.4 Alternating Training of Dual LoRA

HDRec conducts two independent low-rank adaptation processes on the shared decoder through loss-driven optimization. However, as visualized in Fig. 3(a), the two processes on the shared decoder bring concurrent gradient accumulation during simultaneous back-propagation, resulting in mutual interference between adapters. To resolve this conflict, we implement an alternating training protocol of dual low-rank adaption (DLA), as shown in Alg. 1.

In each alternation step, we first enter the **Text Representation Phase**: we fix the ID adapter ψ^{id}, activate the text adapter ϕ^{text}, and get the text representation $\hat{\mathbf{z}}_{\text{text}}$ through the shared decoder. After the first alternation step, we compute the text loss $\mathcal{L}_{\text{Text}}$ (which includes the sequence loss $\mathcal{L}_{\text{seq\_text}}$ and KL divergence with the ID representation) and update ϕ^{text}. Then, we detach $\hat{\mathbf{z}}_{\text{text}}$ to ensure gradient isolation. Immediately following, we enter the **ID Representation Phase**: we fix the text adapter ϕ^{text}, activate the ID adapter ψ^{id}, and compute the ID representation $\hat{\mathbf{z}}_{\text{id}}$ through the shared decoder. Before the last alternation step, we compute the ID loss $\mathcal{L}_{\text{ID}}$ and update ψ^{id}, similarly detaching $\hat{\mathbf{z}}_{\text{id}}$. The entire process is repeated for T alternation steps per batch of data.

This alternating training achieves dual optimization objectives: (1) Gradient isolation through sequential parameter updates, ensuring only one adapter is active per alternation cycle, and (2) Shared decoder parameters remain frozen to preserve LLM's world knowledge. Overall, the alternating training mechanism effectively eliminates concurrent gradient conflicts while maintaining computational efficiency.

3.5 Inference-Time Fusion

The LLM backbone, primarily pre-trained on text, excels at understanding textual semantics, capturing rich explicit knowledge and general patterns. In contrast, while our alternating training method DLA aligns ID and text representations, the nature of item IDs makes their learned semantic representations inherently focus on implicit collaborative signals and tend to exhibit less stable and sparser output distributions compared to the robust textual

Algorithm 1 Alternating Training of Dual Low-Rank Adaptation

Pre-trained LLM_θ ▷ Frozen base model
Initialize ϕ^{text}, ψ^{id} ▷ Task-specific adapters
for epoch = 1 to N **do**
  for batch $\mathcal{B} \in \mathbb{D}$ **do**
    for alternation step $t = 1$ to T **do**
      Text Representation Phase:
      Fix ψ^{id}, activate ϕ^{text}
      Compute $\hat{\mathbf{z}}_{\text{text}} \leftarrow \text{LLM}(\phi^{\text{text}})$
      if $t > 1$ **then**
        $\mathcal{L}_{\text{Text}} \leftarrow \mathcal{L}_{\text{seq\_text}} + \lambda \mathcal{D}_{\text{KL}}(\hat{\mathbf{z}}_{\text{id}} \parallel \tilde{\mathbf{z}}_{\text{text}})$
        Update ϕ with $\nabla_\phi \mathcal{L}_{\text{Text}}$
      end if
      $\tilde{\mathbf{z}}_{\text{text}} \leftarrow \text{detach}(\hat{\mathbf{z}}_{\text{text}})$ ▷ Gradient isolation
      ID Representation Phase:
      Fix ϕ^{text}, activate ψ^{id}
      Compute $\hat{\mathbf{z}}_{\text{id}} \leftarrow \text{LLM}(\psi^{\text{id}})$
      if $t < T$ **then**
        $\mathcal{L}_{\text{ID}} \leftarrow \mathcal{L}_{\text{seq\_id}} + \lambda \mathcal{D}_{\text{KL}}(\hat{\mathbf{z}}_{\text{text}} \parallel \tilde{\mathbf{z}}_{\text{id}})$
        Update ψ with $\nabla_\psi \mathcal{L}_{\text{ID}}$
      end if
      $\tilde{\mathbf{z}}_{\text{id}} \leftarrow \text{detach}(\hat{\mathbf{z}}_{\text{id}})$ ▷ Gradient isolation
    end for
  end for
end for

representations. Crucially, during training, we deliberately enable each semantics to optimize independently by focusing on the corresponding prediction loss. This ensures that, without mutual interference, the learned text semantics focuses on textual features, while the learned ID semantics precisely emphasizes user-item interaction patterns. To leverage the complementary strengths of both semantics and achieve a more stable and accurate final prediction, we implement a dedicated fusion mechanism exclusively at the inference stage. This approach prioritizes the robust textual representation as the primary signal, while adaptively enhancing it with the unique collaborative signals that require careful calibration before integration due to due to their sparse characteristics.

HDRec adaptively combines the text-based distribution $\hat{\mathbf{z}}_{\text{text}}$ with the ID-derived distribution $\hat{\mathbf{z}}_{\text{id}}$. This fusion mechanism consists of three sequential stages: feature normalization, confidence calibration, and adaptive enhancement.

First, for feature normalization, we compute the central tendency of ID features through dimensional averaging:

$$\boldsymbol{\mu}_{\text{id}} = \frac{1}{|I|} \sum_{i=1}^{|I|} (\hat{\mathbf{z}}_{\text{id}})_i \tag{13}$$

The mean value $\boldsymbol{\mu}_{\text{id}}$ serves as the reference point for subsequent transformations, standardizing the ID logits to a consistent scale.

Next, in confidence calibration, the ID features undergo sigmoid activation to produce confidence scores, effectively mapping them to a probability-like range:

$$\mathbf{S}_{\text{id}} = \sigma(\hat{\mathbf{z}}_{\text{id}} - \boldsymbol{\mu}_{\text{id}}) \tag{14}$$

Simultaneously, to prevent numerical instability from very small or negative ID logits, we identify the minimum activation value:

$$\mathbf{m}_{\text{id}} = \min_{1 \le i \le |I|} (\hat{\mathbf{z}}_{\text{id}})_i \tag{15}$$

Table 1: Statistics of the data after preprocessing. "Avg. Items" denotes the average number of items in user sequences.

Datasets	#Users	#Items	#Inters.	Avg. Items
Arts	55,490	22,303	472,676	8.52
Scientific	10,623	4,971	73,914	6.96
Instruments	27,321	10,360	224,680	8.22
Pantry	14,145	4,933	130,775	9.25
Games	54,721	16,835	463,840	8.48

to facilitate feature adjustment with numerical stability:

$$\hat{\mathbf{z}}_{\text{id}}^{\text{adj}} = \hat{\mathbf{z}}_{\text{id}} - \mathbf{m}_{\text{id}} + \boldsymbol{\epsilon} \tag{16}$$

where $\boldsymbol{\epsilon}$ denotes a small, random perturbation (e.g., 10^{-8}) to ensure all values are positive and non-zero.

Finally, for adaptive enhancement, the final fusion combines the dominant text-based representation with the calibrated ID confidence through element-wise enhancement:

$$\mathbf{E} = \alpha \cdot \hat{\mathbf{z}}_{\text{id}}^{\text{adj}} \odot \mathbf{S}_{\text{id}} \tag{17}$$

where $\alpha \in [0, 1]$ controls the contribution strength of the ID semantic. The enhanced representation, serving as the final prediction score for each item:

$$\hat{\mathbf{z}}_{\text{final}} = \hat{\mathbf{z}}_{\text{text}} + \mathbf{E} \tag{18}$$

After obtaining $\hat{\mathbf{z}}_{\text{final}}$ for all candidate items, we sort these scores in descending order. The top-k items with the highest scores are then recommended to the user. This process directly translates the model's predicted scores into a ranking list of suggestions.

4 Experiment

4.1 Experimental Settings

4.1.1 Datasets. We use five sub-categories of Amazon datasets[2] in our experiments, including "Arts, Crafts, and Sewing" (Arts), "Industrial and Scientific" (Scientific), "Musical Instruments" (Instruments), "Prime Pantry" (Pantry) and "Video Games" (Games). We filter out items without titles and those with fewer than five interactions. Tab. 1 presents the statistics of the preprocessed datasets.

4.1.2 Evaluation. We adopt the leave-one-out strategy for evaluating SRS [9]: the most recent item in each interaction sequence serves as the test item, the second most recent as the validation item, and the preceding items as the training item. We adopt Mean Reciprocal Rank (MRR), Recall@10 (R@10), and Normalized Discounted Cumulative Gain (N@10) as evaluation metrics. We rank the ground-truth item of each user sequence among all items and report the average scores across all user sequences on the test set.

4.1.3 Baselines. We compare HDRec with various baselines including (1) Non-LLM-Based SRS: GRU4Rec [5], SASRec [9] , FDSA [33] and FamouSRec[3] [32]. (2) LLM-Based SRS: SAID [7], P5[4] [3], RecFormer[5] [11], Tiger [19], LETTER[6] [26] and IDGenRec[7] [23]. For a fair comparison, we only consider baselines that perform full-ranking over all candidate items. Besides, we use the pre-trained RecFormer provided by its authors and continue to fine-tune it following its paper.

Table 2: Hyper-parameters for HDRec.

Parameter	Arts	Scientific	Instruments	Pantry	Games
Learning Rate	1.5e-4				
LR Scheduler	Cosine Scheduler				
LoRA rank	8				
LoRA alpha	32				
Batch Size	10				
CF Emb Size	128				
Epochs	10	10	8	10	12
λ	0.5	0.7	0.7	0.3	0.3
α	0.7	0.5	0.7	0.3	0.5

4.1.4 Implementation. We use RecBole[8] [28] to implement GRU4Rec, SASRec, and FDSA models, adhering to their original paper settings. For SAID, we use SASRec as the downstream sequential model. We use the implementation provided by the authors of LETTER for Tiger. For other baselines, we use the implementations provided by the original authors.

We conduct experiments on a machine with 4 NVIDIA A800 GPUs. We train HDRec with the AdamW optimizer [17]. We employ the pre-trained DeepSeek-R1-Distill-Llama-8B[9] as the backbone of HDRec. We use pre-trained SASRec to extract item ID embeddings in HDRec. For the alternating training, we set the number of alternation steps $T = 2$. The complete hyper-parameter setting after search is shown in Tab. 2.

4.2 Overall Performance

Tab. 3 reports the overall performance of all methods. Based on the results, we have the following observations:

- Overall, LLM-based methods generally demonstrate superior performance compared to traditional non-LLM methods, showing the significant benefits brought by leveraging LLMs in recommendation tasks. Note that some sophisticated non-LLM methods like FamouSRec, which is tailored to capture frequency patterns, still demonstrate competitive performance.
- Methods that combine LLM capabilities with traditional sequential recommendation techniques also present promising results. Taking SAID as an example, it first employs an LLM to generate item embeddings, which are then used to initialize the embedding layer of SASRec model. This approach highlights the potential of leveraging the strong language understanding ability of LLMs to augment established SRS.
- HDRec achieves the best performance across all datasets on all three metrics. Compared to the best baselines, HDRec shows an average improvement of 13.81% on N@10, 9.71% on R@10, and 16.86% on MRR. The overall results demonstrate the effectiveness of HDRec for sequential recommendation.

HDRec adopts LoRA for parameter-efficient fine-tuning. Hence, it demonstrates good training efficiency. Taking the Pantry dataset

[2]https://cseweb.ucsd.edu/~jmcauley/datasets/amazon_v2
[3]https://anonymous.4open.science/r/FamouSRec/
[4]https://github.com/jeykigung/P5
[5]https://github.com/JiachengLi1995/RecFormer
[6]https://github.com/HonghuiBao2000/LETTER
[7]https://github.com/agiresearch/IDGenRec
[8]https://github.com/RUCAIBox/RecBole
[9]https://huggingface.co/deepseek-ai/DeepSeek-R1-Distill-Llama-8B

Table 3: Performance comparison of different methods. The best performance is highlighted in bold, while the second-best performance is underlined. The last column indicates the improvements over the best baseline models. All improvements are significant at $p < 0.01$ compared to the best baseline under the paired t-test.

		Non-LLM-base				LLM-base							
Dataset	Metric	GRU4Rec	SASRec	FDSA	FamouSRec	SAID	P5	RecFormer	TIGER	LETTER	IDGenRec	HDRec	Improv.
Arts	N@10	0.0320	0.0890	0.0713	0.1290	0.1229	0.1286	0.1251	0.1166	0.1107	0.1184	**0.1412**	9.46%
	R@10	0.0601	0.1279	0.1131	0.1624	0.1500	0.1482	0.1580	0.1422	0.1345	0.1464	**0.1730**	6.53%
	MRR	0.0274	0.0807	0.0628	0.1228	0.1145	0.1234	0.1198	0.1166	0.1058	0.1097	**0.1368**	10.9%
Scientific	N@10	0.0255	0.0812	0.0845	0.1031	0.1069	0.1031	0.1040	0.0782	0.0829	0.0946	**0.1190**	11.3%
	R@10	0.0418	0.1280	0.1177	0.1316	0.1458	0.1257	0.1468	0.1074	0.1117	0.1177	**0.1568**	6.81%
	MRR	0.0238	0.0718	0.0788	0.0984	0.0949	0.0977	0.0966	0.0731	0.0782	0.0874	**0.1140**	15.9%
Instruments	N@10	0.0331	0.0637	0.0680	0.0922	0.0870	0.0847	0.0821	0.0725	0.0784	0.0870	**0.1023**	11.0%
	R@10	0.0547	0.0973	0.0928	0.1219	0.1114	0.1056	0.1030	0.0962	0.1044	0.1197	**0.1330**	9.11%
	MRR	0.0312	0.0580	0.0655	0.0874	0.0795	0.0801	0.0803	0.0689	0.0743	0.0848	**0.0989**	13.2%
Pantry	N@10	0.0182	0.0455	0.0438	0.0580	0.0571	0.0409	0.0566	0.0366	0.0357	0.0585	**0.0718**	22.7%
	R@10	0.0380	0.0716	0.0677	0.0841	0.0814	0.0529	0.0888	0.0587	0.0573	0.0855	**0.1034**	16.4%
	MRR	0.0171	0.0422	0.0414	0.0540	0.0497	0.0389	0.0513	0.0335	0.0326	0.0502	**0.0680**	25.9%
Games	N@10	0.0267	0.0545	0.0450	0.0816	0.0816	0.0568	0.0675	0.0615	0.065	0.0820	**0.0940**	14.6%
	R@10	0.0523	0.0871	0.0779	0.1291	0.1177	0.0729	0.1027	0.0968	0.0974	0.1178	**0.1416**	9.68%
	MRR	0.0257	0.0518	0.0424	0.0745	0.0706	0.0535	0.0513	0.0565	0.0609	0.0710	**0.0882**	18.4%

as an example, the training time of HDRec is 1.74 hours, whereas RecFormer requires 4.32 hours.

4.3 Contribution of Each Part in HDRec

4.3.1 Ablation Study. As shown in Fig. 4, we compare the complete HDRec against several variants: (1) HDRec without DLA, (2) HDRec without KL divergence alignment of output logits, and (3) HDRec w/o ID (only model text semantics, without KL alignment). The experimental results clearly show that the complete HDRec achieves the best performance across all three datasets, validating the effectiveness of each component.

Specifically, HDRec w/o DLA exhibits the poorest performance among all methods. While the dual-LoRA design enables fine-grained cross-semantics alignment and mitigates semantics discrepancy during early fusion, concurrently training two LoRA processes on a single decoder introduces gradient conflicts. The absence of the DLA mechanism results in unstable model optimization, leading to mutual interference between the learning processes of different semantics, which severely degrades the model's overall performance.

Removing the KL divergence alignment of output logits led to a performance degradation, suggesting that KL divergence, acting as a regularization technique, helps shape the model's output distribution and guide the alignment of representations from different semantics at the output layer, thereby enhancing recommendation accuracy.

When ID semantics is removed and only text semantics is modeled, the performance is also significantly affected. This finding underscores the importance of modeling ID semantics within our hybrid fusion framework. While only modeling text semantics can capture rich semantic information, combining it with ID semantics provides more direct and efficient user-item association information, which is indispensable for accurate recommendations.

4.3.2 Impacts of Fusion Designs. To understand the impacts of different fusion strategies on recommendation performance, we conducted experiments comparing three types of strategies: (1) capturing single semantics, (2) fusing ID and text semantics, and (3) hybrid fusion. Their differences are illustrated in Fig. 2. We report the results in Tab. 4.

Capturing Single Semantics. First, we evaluate two strategies that capture single semantics: Only ID (Fig. 2(a)) and Only Text (Fig. 2(b)). Both are based on $\text{HDRec}_{\text{base}}$. Tab. 4 shows that Only Text significantly outperforms Only ID across all datasets, indicating that solely considering textual features can provide richer information than capturing ID semantics alone.

Fusing ID and Text Semantics. Next, we examine three strategies that fuse both ID and text semantics. Early Fusion (Fig. 2(c)) concatenates tokens at the input. For Late Fusion (Fig. 2(d)), we test two specific strategies: Late Fusion and Late Fusion (Pretrained ID). In both strategies, the ID semantics is handled by an ID-side model (SASRec) and the text semantics is learned by $\text{HDRec}_{\text{base}}$, with their outputs fused late. Late Fusion (Pretrained ID) uses a pretrained SASRec model for the ID semantics, whereas Late Fusion trains the SASRec model from scratch. Results in Tab. 4 reveal that Early Fusion surpasses strategies that capture single semantics, yet both Late Fusion methods perform significantly worse. The stark performance gap confirms Late Fusion's inherent difficulty in achieving fine-grained cross-semantics associations. The discrepancy between the LLM-based module and the lightweight ID model (SASRec) leads to unbalanced learning. Simply using a pretrained ID model cannot effectively mitigate these inherent drawbacks.

Hybrid Fusion. We further investigate the performance of three Hybrid Fusion methods (Fig. 2(e)): Single LoRA Single IPL (Half Params.) configured with approximately half the trainable parameters of HDRec; Single LoRA Single IPL (Comparable Params.) with trainable parameters comparable to HDRec as it doubles the rank in LoRA; Dual LoRA Single IPL (HDRec); and Dual LoRA Dual IPL which employs a separate IPL for each semantics. Their main differences are the number of trainable parameters, whether two low-rank adaptation processes are employed, and whether two IPL modules are used. We can see that Dual LoRA Single IPL (HDRec)

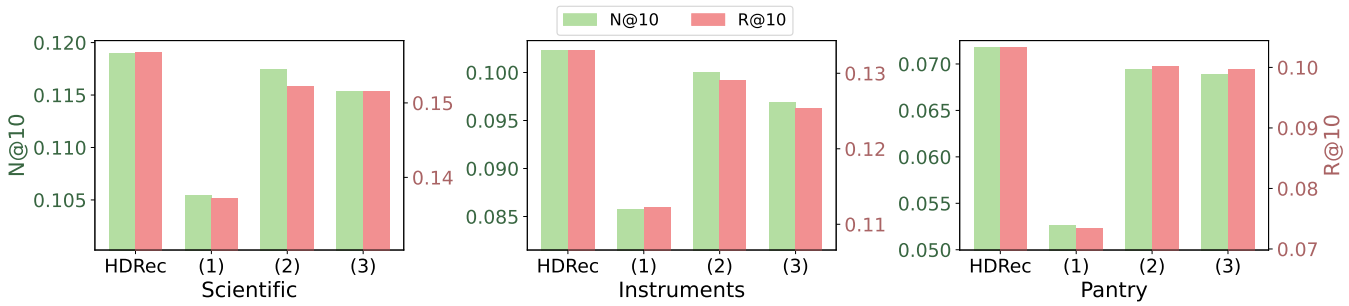

Figure 4: Comparison of HDRec and its variations: (1) HDRec w/o DLA, (2) HDRec w/o KL, (3) HDRec w/o ID+KL.

Table 4: Performance comparisons of different fusion designs. The best performance is highlighted in bold, while the second-best performance is underlined.

Fusion Design		Scientific			Instruments			Pantry		
		N@10	R@10	MRR	N@10	R@10	MRR	N@10	R@10	MRR
(1) Capturing Single Semantics	(a) Only ID	0.1103	0.1399	0.1062	0.0945	0.1211	0.0914	0.0592	0.0845	0.0568
	(b) Only Text	0.1154	0.1515	0.1105	0.0969	0.1254	0.0936	0.0689	0.0997	0.0653
(2) Fusing ID and Text Semantics	(a) Early Fusion	0.1155	0.1477	0.1111	0.0961	0.1212	0.0937	0.0686	0.0969	0.0655
	(b) Late Fusion	0.0097	0.0147	0.0103	0.0041	0.0077	0.0045	0.0038	0.0064	0.0047
	(c) Late Fusion (Pretrained ID)	0.0948	0.1366	0.0877	0.0808	0.1075	0.0779	0.0476	0.0682	0.0461
(3) Hybrid Fusion	(a) Single LoRA Single IPL (Half Params.)	0.1120	0.1482	0.0924	0.0958	0.1266	0.0924	0.0679	0.0972	0.0648
	(b) Single LoRA Single IPL (Comparable Params.)	0.1120	0.1501	0.0935	0.0970	0.1279	0.0935	0.0688	0.0971	0.0660
	(c) Dual LoRA Single IPL (HDRec)	**0.1190**	**0.1568**	**0.1140**	**0.1023**	**0.1330**	**0.0989**	**0.0718**	**0.1034**	**0.0680**
	(d) Dual LoRA Dual IPL	0.1140	0.1475	0.1097	0.0992	0.1290	0.0958	0.0653	0.0937	0.0620

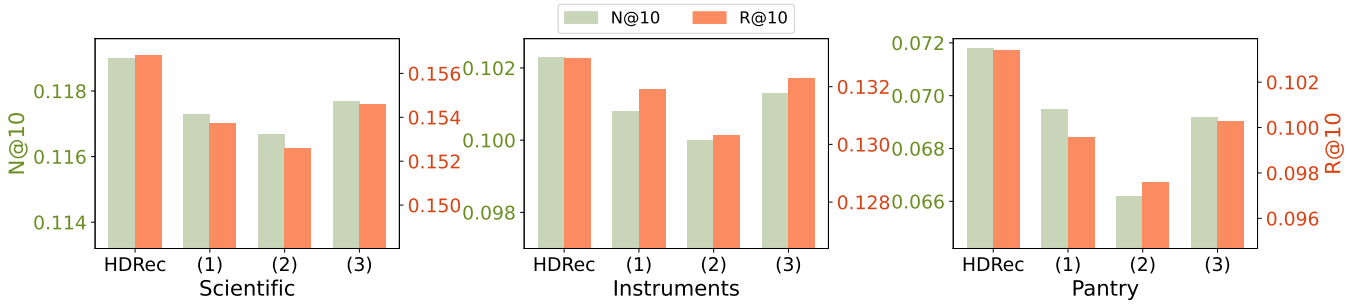

Figure 5: Ablation study on training loss and inference fusion strategies. Variants include (1) training using a composite loss derived from the error of the logits of fused semantics, (2) ID-dominant adaptive inference fusion, and (3) dual-semantics confidence weighted inference fusion. Our full method (HDRec) uses individual modality losses for training and a text-dominant adaptive fusion for inference.

achieved the best performance across all metrics and datasets. Compared to Single LoRA Single IPL, the dual LoRA processes cross-semantics information more effectively, suggesting that using independent low-rank adaptation for each semantics is beneficial. The slight improvement of Dual LoRA Single IPL (HDRec) over Dual LoRA Dual IPL indicates that parameter sharing via a single IPL further enhances fusion.

In summary, experimental results demonstrate that considering single semantics is insufficient, simple early fusion is limited by noise, and late fusion struggles with cross-semantics modeling. Our proposed Dual LoRA Single IPL (HDRec) method successfully integrates the strengths of early (fine-grained interaction) and late (noise avoidance) fusion, achieving better performance.

4.3.3 Impacts of Training Loss and Inference Fusion Strategies. To analyze the effectiveness of the proposed training loss and the inference fusion strategy employed in HDRec, we conduct ablation studies focusing on these two components. The results are presented in Fig. 5.

Training Loss. In HDRec, we only conduct fusion during inference time (Sec. 3.5). An alternative is to also conduct fusion during training, labeled as variant (1) in Fig. 5. By comparing their performance in Fig. 5, we can see that HDRec consistently outperforms variant (1), showing that fusion at inference is sufficient to provide high-quality recommendation. During training, separating two LoRA processes without fusion can improve stability.

Inference Fusion Strategy. We compare our proposed inference-time fusion (Sec. 3.5) with two alternative inference fusion mechanisms: ID-dominant adaptive inference fusion and dual-semantics confidence weighted inference fusion, which are represented as variants (2) and (3) in Fig. 5:

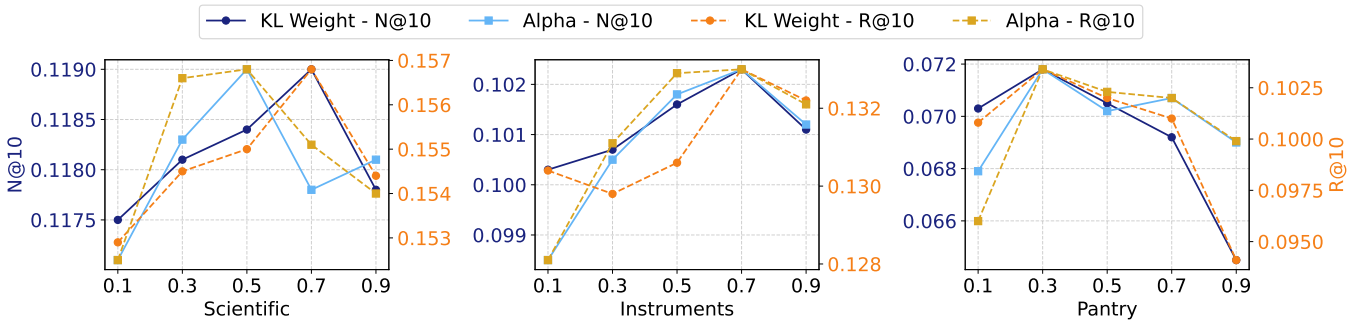

Figure 6: The impact of varying KL divergence weight (λ) and fusion parameter (α).

(1) **Variants (2):** Compared to HDRec, ID-dominant adaptive inference fusion applies the inverse logic, using text information to enhance the ID-derived logits, thereby prioritizing the ID semantics.
(2) **Variants (3):** Different to HDRec which only considers ID side in Eqs 13, 14, 15, 16 and 17, dual-semantics confidence weighted inference fusion also passes $\hat{\mathbf{z}}_{\text{text}}$ to Eqs 13, 14, 15, 16 and 17, and uses the result to replace original $\hat{\mathbf{z}}_{\text{text}}$ in Eq. 18.

Fig. 5 shows that HDRec achieves superior performance than the two variants. This highlights the advantage of our text-dominant approach, which effectively leverages the semantic richness of text as the primary signal while adaptively refining it with ID information, surpassing strategies that prioritize ID or rely on dual-semantics confidence weighting.

4.4 Analysis of Hyper-Parameter Settings

We analyze the impact of the KL divergence weight (used during training for aligning logits of text and ID semantics) and the fusion parameter α (control the contribution of ID semantics in the final distribution during inference). As shown in Fig. 6, the performance of HDRec initially increases with the rise of hyper-parameter and then begins to decrease after reaching a peak for both the KL divergence weight and α. Overall, the performance trends w.r.t. different hyper-parameters can be clearly observed, meaning that it is relatively easy to find a good hyper-parameter setting for HDRec.

5 Related Work

Emerging LLM-based SRS can be grouped into two paradigms: LLM-augmented SRS and LLM-centric SRS [7].

LLM-augmented SRS adopts LLM as the feature extractor to obtain item features that are further fed to SRS. Harte et al. [4] show that initializing BERT4Rec [21] with embeddings obtained from LLM can significantly improve sequential recommendation. LRD [30] is a LLM-based latent relation discovery framework. It uses language knowledge to discover latent relations that can further enhance SRS. SERALM underpins alignment training method to refine LLMs' generation using feedback from ID-based recommenders for better knowledge augmentation [20]. SAID [7] uses LLM to learn semantically aligned item ID embeddings that can be integrated with downstream SRS.

LLM-centric SRS leverages LLM as the SRS. RecFormer [11] describes items using item "sentences". It is trained to understand the "sentence" sequence and retrieve the next "sentence" for making sequential recommendations. E4SRec [12] takes ID sequences as inputs to LLM and ensures that the generated output falls within the candidate lists. LLM-TRSR [37] focuses on modeling over-length user behavior sequences. It segments behavior sequences and applies summarization techniques to form the inputs to LLM-based SRS. LLM4ISR [22], armed with prompt initialization, optimization, and selection modules, enables LLM to make intent-driven session recommendations. Re2LLM [27] guides LLM with self-reflection and a lightweight retrieval agent, enabling LLM to focus on specialized knowledge essential for more accurate sequential recommendations effectively and efficiently.

Besides, there is a surge of work on a new paradigm called generative SRS [19, 23, 26, 29, 31]. Instead of modeling pre-fixed item IDs, they generate tokens and construct item IDs using generative tokens. The generative IDs are then used in SRS. Compared to ID-based SRS, generative SRS can better leverage item content and encode item semantics into tokens that form item IDs. However, generative SRS also faces challenges such as the difficulty of training (e.g., producing identical token sequences for similar items) [38], and this direction is still under-explored.

6 Conclusion

In this paper, we present HDRec, an efficient hybrid dual-semantics modeling method for enhancing LLM-based Recommendation. HDRec employs two LoRAs processes on a shared LLM decoder, with each process handling information from one of the two semantics. We further implement an inference-time fusion strategy to allow for stable and accurate final predictions. To mitigate gradient conflicts caused by the dual LoRA processes, we introduce the alternating training of dual low-rank adaptation strategy. Extensive experiments have demonstrated that HDRec outperforms state-of-the-art SRS methods. In the future, we plan to extend HDRec to generative SRS, exploring the possibility of fusing different semantics in generative SRS.

7 Acknowledgments

Hui Li was supported by National Natural Science Foundation of China (No. 62572410, 42171456) and Natural Science Foundation of Xiamen, China (No. 3502Z202471028). Xiaodong Li was supported by Xiamen Science and Technology Project (No. 3502Z202571028) and the Fundamental Research Funds for the Central Universities, Xiamen University (No. 20720250171).

8 Ethical Considerations

This study aims to improve sequential recommendation systems by integrating collaborative and textual information via large language models. While this approach enhances recommendation quality, there is a risk of bias amplification. Our model relies on historical user data and LLM knowledge, both of which may contain inherent biases. Without careful bias mitigation, the system could amplify these biases, leading to unfair recommendations and reinforcing stereotypes. For example, female users may get frequent recommendations for baby products. To alleviate this issue, debiasing techniques such as causal Intervention (use counterfactual reasoning to simulate "what if" scenarios), training models with adversarial networks to remove gender influence, or balancing training data by oversampling non-stereotypical purchases can be applied.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT*. 4171–4186.

[2] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep Learning for Sequential Recommendation: Algorithms, Influential Factors, and Evaluations. *ACM Trans. Inf. Syst.* 39, 1 (2020), 10:1–10:42.

[3] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). In *RecSys*. 299–315.

[4] Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging Large Language Models for Sequential Recommendation. In *RecSys*. 1096–1102.

[5] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based Recommendations with Recurrent Neural Networks. In *ICLR*. https://openreview.net/pdf?id=yoffK5KZSgQ

[6] 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 *ICLR*. https://openreview.net/forum?id=nZeVKeeFYf9

[7] Jun Hu, Wenwen Xia, Xiaolu Zhang, Chilin Fu, Weichang Wu, Zhaoxin Huan, Ang Li, Zuoli Tang, and Jun Zhou. 2024. Enhancing Sequential Recommendation via LLM-based Semantic Embedding Learning. In *WWW*. 103–111.

[8] Chengkai Huang, Tong Yu, Kaige Xie, Shuai Zhang, Lina Yao, and Julian J. McAuley. 2024. Foundation Models for Recommender Systems: A Survey and New Perspectives. *arXiv Preprint* (2024). https://arxiv.org/abs/2402.11143

[9] Wang-Cheng Kang and Julian J. McAuley. 2018. Self-Attentive Sequential Recommendation. In *ICDM*. 197–206.

[10] Sein Kim, Hongseok Kang, Seungyoon Choi, Donghyun Kim, Min-Chul Yang, and Chanyoung Park. 2024. Large Language Models meet Collaborative Filtering: An Efficient All-round LLM-based Recommender System. In *KDD*.

[11] Jiacheng Li, Ming Wang, Jin Li, Jinmiao Fu, Xin Shen, Jingbo Shang, and Julian J. McAuley. 2023. Text Is All You Need: Learning Language Representations for Sequential Recommendation. In *KDD*. 1258–1267.

[12] Xinhang Li, Chong Chen, Xiangyu Zhao, Yong Zhang, and Chunxiao Xing. 2023. E4SRec: An Elegant Effective Efficient Extensible Solution of Large Language Models for Sequential Recommendation. *arXiv Preprint* (2023). https://arxiv.org/abs/2312.02443

[13] Jiayi Liao, Sihang Li, Zhengyi Yang, Jiancan Wu, Yancheng Yuan, Xiang Wang, and Xiangnan He. 2024. LLaRA: Large Language-Recommendation Assistant. In *SIGIR*. 1785–1795.

[14] Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and Weinan Zhang. 2023. How Can Recommender Systems Benefit from Large Language Models: A Survey. *arXiv Preprint* (2023). https://arxiv.org/abs/2306.05817

[15] Peng Liu, Lemei Zhang, and Jon Atle Gulla. 2023. Pre-train, Prompt and Recommendation: A Comprehensive Survey of Language Modelling Paradigm Adaptations in Recommender Systems. *arXiv Preprint* (2023). https://arxiv.org/abs/2302.03735

[16] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv Preprint* (2019). https://arxiv.org/abs/1907.11692

[17] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *ICLR*. https://openreview.net/forum?id=Bkg6RiCqY7

[18] Massimo Quadrana, Paolo Cremonesi, and Dietmar Jannach. 2018. Sequence-Aware Recommender Systems. *ACM Comput. Surv.* 51, 4 (2018), 66:1–66:36.

[19] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Q. Tran, Jonah Samost, Maciej Kula, Ed H. Chi, and Mahesh Sathiamoorthy. 2023. Recommender Systems with Generative Retrieval. In *NeurIPS*.

[20] Yankun Ren, Zhongde Chen, Xinxing Yang, Longfei Li, Cong Jiang, Lei Cheng, Bo Zhang, Linjian Mo, and Jun Zhou. 2024. Enhancing Sequential Recommenders with Augmented Knowledge from Aligned Large Language Models. In *SIGIR*. 345–354.

[21] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *CIKM*. 1441–1450.

[22] Zhu Sun, Hongyang Liu, Xinghua Qu, Kaidong Feng, Yan Wang, and Yew Soon Ong. 2024. Large Language Models for Intent-Driven Session Recommendations. In *SIGIR*. 324–334.

[23] Juntao Tan, Shuyuan Xu, Wenyue Hua, Yingqiang Ge, Zelong Li, and Yongfeng Zhang. 2024. IDGenRec: LLM-RecSys Alignment with Textual ID Learning. In *SIGIR*. 355–364.

[24] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z. Sheng, and Mehmet A. Orgun. 2019. Sequential Recommender Systems: Challenges, Progress and Prospects. In *IJCAI*. 6332–6338.

[25] Shoujin Wang, Qi Zhang, Liang Hu, Xiuzhen Zhang, Yan Wang, and Charu Aggarwal. 2022. Sequential/Session-based Recommendations: Challenges, Approaches, Applications and Opportunities. In *SIGIR*. 3425–3428.

[26] Wenjie Wang, Honghui Bao, Xinyu Lin, Jizhi Zhang, Yongqi Li, Fuli Feng, See-Kiong Ng, and Tat-Seng Chua. 2024. Learnable Item Tokenization for Generative Recommendation. In *CIKM*. 2400–2409.

[27] Ziyan Wang, Yingpeng Du, Zhu Sun, Haoyan Chua, Kaidong Feng, Wenya Wang, and Jie Zhang. 2024. Re2LLM: Reflective Reinforcement Large Language Model for Session-based Recommendation. *arXiv Preprint* (2024). https://arxiv.org/abs/2403.16427

[28] Lanling Xu, Zhen Tian, Gaowei Zhang, Junjie Zhang, Lei Wang, Bowen Zheng, Yifan Li, Jiakai Tang, Zeyu Zhang, Yupeng Hou, Xingyu Pan, Wayne Xin Zhao, Xu Chen, and Ji-Rong Wen. 2023. Towards a More User-Friendly and Easy-to-Use Benchmark Library for Recommender Systems. In *SIGIR*. 2837–2847.

[29] Kun Yang, Siyao Zheng, Tianyi Li, Xiaodong Li, and Hui Li. 2025. GENPLUGIN: A Plug-and-Play Framework for Long-Tail Generative Recommendation with Exposure Bias Mitigation. *arXiv Preprint* (2025). https://arxiv.org/abs/2507.03568

[30] Shenghao Yang, Weizhi Ma, Peijie Sun, Qingyao Ai, Yiqun Liu, Mingchen Cai, and Min Zhang. 2024. Sequential Recommendation with Latent Relations based on Large Language Model. In *SIGIR*. 335–344.

[31] Jianyang Zhai, Zi-Feng Mai, Chang-Dong Wang, Feidiao Yang, Xiawu Zheng, Hui Li, and Yonghong Tian. 2025. Multimodal Quantitative Language for Generative Recommendation. In *ICLR*. https://openreview.net/forum?id=v7YrIjpkTF

[32] Junjie Zhang, Ruobing Xie, Hongyu Lu, Wenqi Sun, Xin Zhao, Zhanhui Kang, et al. 2025. Frequency-Augmented Mixture-of-Heterogeneous-Experts Framework for Sequential Recommendation. In *WWW*.

[33] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S. Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, and Xiaofang Zhou. 2019. Feature-level Deeper Self-Attention Network for Sequential Recommendation. In *IJCAI*. 4320–4326.

[34] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. 2023. A Survey of Large Language Models. *arXiv Preprint* (2023). https://arxiv.org/abs/2303.18223

[35] Zihuai Zhao, Wenqi Fan, Jiatong Li, Yunqing Liu, Xiaowei Mei, Yiqi Wang, Zhen Wen, Fei Wang, Xiangyu Zhao, Jiliang Tang, and Qing Li. 2024. Recommender Systems in the Era of Large Language Models (LLMs). *IEEE Trans. Knowl. Data Eng.* 36, 11 (2024), 6889–6907.

[36] Bowen Zheng, Yupeng Hou, Hongyu Lu, Yu Chen, Wayne Xin Zhao, Ming Chen, and Ji-Rong Wen. 2024. Adapting large language models by integrating collaborative semantics for recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 1435–1448.

[37] Zhi Zheng, Wenshuo Chao, Zhaopeng Qiu, Hengshu Zhu, and Hui Xiong. 2024. Harnessing Large Language Models for Text-Rich Sequential Recommendation. In *WWW*. 3207–3216.

[38] Jieming Zhu, Mengqun Jin, Qijiong Liu, Zexuan Qiu, Zhenhua Dong, and Xiu Li. 2024. CoST: Contrastive Quantization based Semantic Tokenization for Generative Recommendation. In *RecSys*. 969–974.

[39] Yaochen Zhu, Liang Wu, Qi Guo, Liangjie Hong, and Jundong Li. 2024. Collaborative Large Language Model for Recommender Systems. In *WWW*. 3162–3172.