Empowering Denoising Sequential Recommendation with Large Language Model Embeddings

Tongzhou Wu
City University of Hong Kong
Hong Kong, China
tongzhowu3-c@my.cityu.edu.hk

Yuhao Wang
City University of Hong Kong
Hong Kong, China
yhwang25-c@my.cityu.edu.hk

Maolin Wang
City University of Hong Kong
Hong Kong, China
Morin.wang@my.cityu.edu.hk

Chi Zhang
Harbin Engineering University
Harbin, China
zhangchi20@hrbeu.edu.cn

Xiangyu Zhao✉
City University of Hong Kong
Hong Kong, China
xianzhao@cityu.edu.hk

Abstract

Sequential recommendation aims to capture user preferences by modeling sequential patterns in user-item interactions. However, these models are often influenced by noise such as accidental interactions, leading to suboptimal performance. Therefore, to reduce the effect of noise, some works propose explicitly identifying and removing noisy items. However, we find that simply relying on collaborative information may result in an over-denoising problem, especially for cold items. To overcome these limitations, we propose a novel framework: Interest Alignment for Denoising Sequential Recommendation (IADSR) which integrates both collaborative and semantic information. Specifically, IADSR is comprised of two stages: in the first stage, we obtain the collaborative and semantic embeddings of each item from a traditional sequential recommendation model and an LLM, respectively. In the second stage, we align the collaborative and semantic embeddings and then identify noise in the interaction sequence based on long-term and short-term interests captured in the collaborative and semantic modalities. Our extensive experiments on four public datasets validate the effectiveness of the proposed framework and its compatibility with different sequential recommendation systems. The code and data are released for reproducibility: https://github.com/Applied-Machine-Learning-Lab/IADSR.

CCS Concepts

• **Information systems → Recommender systems**.

Keywords

Denoising, Sequential Recommendation, Recommender System, Large Language Model, User Interest

---

✉Corresponding author.

1 Introduction

In recent years, recommender systems have become indispensable components of modern digital platforms, serving as essential tools to alleviate information overload and enhance user experience across diverse domains such as news [52], entertainment services [2] and social media [87]. Among various recommendation tasks, sequential recommendation has gained considerable attention since it aims to capture the temporal dynamics of user behavior and sequential dependencies in interaction histories [15, 45, 84], enabling more accurate and dynamic predictions of future interactions [72]. Recent advances in denoising sequential recommendation have explored diverse neural architectures, including Recurrent Neural Networks (RNNs) [13, 29], convolutional neural network (CNNs) [58, 68], Graph Neural Networks (GNNs) [14, 76], and Transformer-based models [30, 32, 44, 57]. However, these approaches primarily focus on improving recommendation performance through architectural modifications, without explicitly addressing the inherent noise present in interaction sequences.

Therefore, despite remarkable progress, sequential recommendation faces significant challenges in real-world applications due to noisy interactions [61]. Specifically, Such noise includes accidental clicks [74], exploratory behaviors, or interactions that do not reflect true user preferences [25, 49, 77]. This problem is exacerbated in sequential scenarios as noise propagates through the modeling process [8, 17, 19], potentially leading to misinterpreted user intentions [9, 76]. The presence of noisy interactions could severely distort the learned user preferences and sequential patterns, ultimately degrading recommendation quality [27, 40, 47].

To address this challenge, denoising sequential recommendation has emerged as a promising research direction [35]. Early approaches primarily focused on identifying and filtering noisy interactions based on collaborative signals derived from user-item interaction matrices [12, 37, 60] (e.g., directly removing identified noisy interactions [64, 71], or replace noisy items with alternative
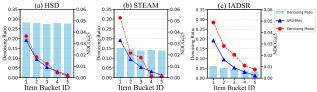
Figure 1: Performance of HSD, STEAM, and our proposed IADSR on Beauty dataset. Item bucket ID from 1 to 5 denotes hot to cold. The denoising ratio represents the percentage of items removed as noise, and the recommendation model used for comparison is GRU4Rec.

items that better align with the user's established preference patterns [40, 75, 76]). These tasks leverage patterns in collective user behaviors to distinguish real preferences from noise and aim to capture better user preferences and behavioral patterns embedded within historical interaction sequences, thereby enhancing recommendation accuracy and relevance [79]. However, relying solely on collaborative information presents inherent limitations, particularly for cold items with sparse interaction histories [55, 69, 81]. Without leveraging content features, these models lack contextual understanding of interactions, making it difficult to differentiate between real user preferences and random behaviors [11, 16, 28]. Consequently, it could lead to over-denoising issues, i.e., a large proportion of cold items would be identified as noise and removed, which potentially dampens the denoising performance.

To illustrate the potential over-denoising, we conducted a preliminary analysis of the denoising ratio and accuracy across all the items. The denoising ratio refers to the proportion of items in the original interaction sequence that are identified as noise by each method. As shown in Figure 1, we compared the denoising ratios and recommendation performance of HSD [75], STEAM [40], and IADSR on the Beauty dataset. We divided the original sequence into five equal-frequency buckets based on item popularity (from hot to cold items) and then evaluated the recommendation accuracy using NDCG@5 on each bucket. The blue bars represent the denoising ratio applied by each method, while the line graphs illustrate the recommendation performance. The blue curve represents the results tested with the GRU4Rec model. We can observe that across all methods, as item popularity decreases from bucket 1 (hottest) to bucket 5 (coldest), recommendation performance significantly declines. This uniform filtering process fails to account for the unique characteristics of cold items, potentially removing interactions that may seem unreasonable from a purely collaborative perspective but could actually reflect real user interests in less popular items [10, 31, 76]. However, IADSR performs better than baseline approaches, particularly for cold items. While the absolute improvement may appear larger on popular items due to their higher baseline performance, the relative improvement of IADSR is actually more pronounced for cold items. By preserving more useful information during the denoising process for these less popular items, IADSR achieves consistently better recommendation accuracy across all buckets, with the advantage becoming more pronounced for the coldest items in buckets 4 and 5.

Therefore, relying solely on collaborative signals may be insufficient [51, 59, 85], and we consider leveraging the textual information of items to address this limitation. Large Language Models

(LLMs), which have gained tremendous popularity in recent years, can effectively complement this aspect [1, 46, 82]. We can utilize LLMs to generate embeddings of textual information and combine them with traditional collaborative information for denoising [3]. By aligning embeddings from both modalities, we attempt to better capture users' genuine preferences.

Consequently, we present IADSR, a novel two-stage denoising framework that effectively integrates semantic information from LLMs with collaborative signals for enhanced denoising. Our framework operates through two distinct stages: (1) dual representation learning, where we independently obtain item embeddings from both LLMs and traditional sequential models; (2) cross-modal alignment and noise identification, where we leverage long-term and short-term user interests to detect and filter noisy interactions.

The main contributions of this paper are as follows.

- We propose IADSR, a novel denoising paradigm for sequential recommendation compatible with diverse backbone models and achieving performance enhancement.
- The proposed framework combines LLMs with sequential recommendation without fine-tuning.
- Experiments on four public datasets, i.e., Amazon Beauty, Sports, Toys, and MovieLens-100K, have demonstrated the effectiveness of our proposed method.

2 Preliminary

In this section, we introduce the basic notations and definitions used throughout this work.

2.1 Sequential Recommendation

Let $\mathcal{U} = \{u_1, u_2, ..., u_m\}$ denotes the set of users and $\mathcal{I} = \{i_1, i_2, ..., i_n\}$ denotes the set of items. For each user $u \in \mathcal{U}$ its interaction sequence $S^u = [i_1^u, i_2^u, ..., i_t^u]$ is sorted in ascending order by time, where $i_j^u \in \mathcal{I}$ represents the j-th item that user u has interacted with. The goal of sequential recommendation is to predict the next item i_{t+1}^u that user u is likely to interact with based on S^u.

Formally, let $\hat{\mathbf{y}}^u \in \mathbb{R}^{|I|}$ denote the prediction scores for all items in the item set I, where each element $\hat{y}_i^u$ represents the predicted probability that user u will interact with item i next. The ground truth is typically represented as a one-hot encoded vector $\mathbf{y}^u$, where $y_i^u = 1$ if item i is the actual next item in the sequence ($i = i_{t+1}^u$), and $y_i^u = 0$ otherwise.

The Cross-Entropy loss function is then defined as:

$$\mathcal{L}_{CE} = -\sum_{u \in \mathcal{U}} \sum_{i \in \mathcal{I}} y_i^u \log(\hat{y}_i^u) \tag{1}$$

This loss function encourages the model to assign a high probability to the correct next item. However, in the presence of noisy interactions in the sequence, optimizing solely based on this loss can lead the model to learn patterns from noise, potentially degrading recommendation performance.

2.2 Denoising Sequential Recommendation

In the denoising sequential recommendation task, for a user u with interaction sequence $S^u = [i_1^u, i_2^u, ..., i_t^u]$, we need to identify and remove the noise interactions:

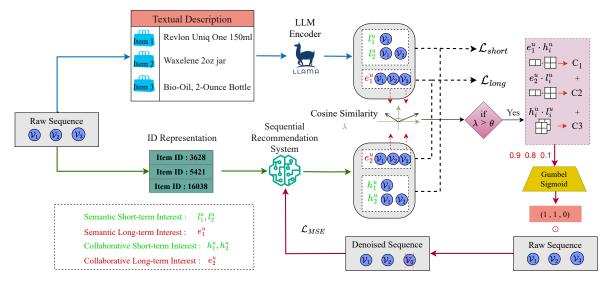$$S_{denoised}^u = S^u \setminus S_{noise}^u \tag{2}$$

Figure 2: Overview of the IADSR framework. The black arrow denotes the data flow. The blue and green color denotes semantic and collaborative modality, respectively.

where $S_{denoised}^u$ contains interactions that truly represent user u's preferences, and S_{noise}^u consists of noisy interactions that may mislead the recommendation model [56, 62].

3 Method

In this section, an overview of the proposed framework is first provided, followed by details of different modules.

3.1 Overview

In this section, we introduce the overall framework of IADSR, which employs a two-stage framework to enhance recommendation quality by identifying and removing noise from user interaction sequences. As depicted in Figure 1, in the first stage we construct item embeddings from two distinct sources by processing raw user sequences through parallel paths: (1) extracting semantic representations via LLM encoding of textual descriptions, generating comprehensive semantic understanding [20, 24, 43]; (2) learning collaborative patterns through traditional sequential models using item ID representations. These complementary embeddings capture both content semantics and user behavior patterns [42, 78]. In the second stage, we align the collaborative and semantic embeddings through cosine similarity measures to identify noise in user sequences [80]. We compute similarity scores between long-term and short-term interest representations from both embedding spaces, then apply a Gumbel-Sigmoid function to generate binary masks indicating noisy items.

3.2 Semantic Encoding

Previous studies have not focused on extracting semantic information, possibly due to high computational costs and insufficient world knowledge. Recently, Large Language Models (LLMs) have demonstrated remarkable capabilities in understanding text across diverse domains [36]. However, these decoder-only architectures exhibit limitations in encoding capabilities, as they are optimized

for generation rather than representation [7]. In recommender systems, while LLMs can extract semantic information from text, their architectural constraints may result in suboptimal embeddings compared to dedicated encoding frameworks [34].

For our approach, we employ LLM2Vec [5] to generate semantic embeddings for each item. Specifically, LLM2Vec addresses the encoding limitations of decoder-only Large Language Models by efficiently extracting high-quality semantic embeddings that capture nuanced user preferences from textual descriptions, enhancing recommender systems with encoder capabilities. This enables the extraction of high-quality semantic embeddings without additional fine-tuning, making it ideal for our recommendation scenario.

For each item $i \in \mathcal{I}$, we obtain its item name Z_i and process it through LLM2Vec to generate semantic embeddings:

$$\mathbf{e}_i^{LLM} = \text{LLM2Vec}(Z_i) \tag{3}$$

where $\mathbf{e}_i^{LLM} \in \mathbb{R}^d$ is the semantic embedding vector of item i with dimension d, and LLM2Vec$(\cdot)$ represents the encoding function that maps item names to dense vector embeddings.

These LLM-integrated embeddings capture semantic relationships between items based on their names, offering rich complementary information to collaborative signals. Even with just the item name, the semantic embeddings effectively capture product categories and attributes through the pre-trained knowledge embedded in the LLM. This approach is particularly valuable for cold items with limited interaction histories, as it allows the model to better infer item similarities from semantic meaning rather than relying solely on interaction patterns.

3.3 Interest Alignment

Despite differences between collaborative and semantic modalities in their representational spaces, we posit that for any given user, their fundamental interests should remain consistent regardless of which modality is used to represent them [83]. Both modalities ultimately attempt to capture the same underlying user interests.

In order to effectively leverage both semantic and collaborative information for optimal merging, we first organize user interests into long-term and short-term embeddings and then systematically align these interests across modalities.

3.3.1 Interest Representation For each user u with interaction sequence $S^u = \{i_1^u, i_2^u, \ldots, i_n^u\}$, we structure their interests as:

Long-term Interests: The long-term interest represents a comprehensive view of the user's preferences across their entire interaction history. Rather than encoding each item separately, we encode the complete sequence of user interactions directly:

$$\mathbf{e}_1^u = \text{LLM2Vec}(S^u) \tag{4}$$

where $\text{LLM2Vec}(S^u)$ processes the entire interaction sequence as a single input, capturing the holistic semantic meaning of the user's complete interaction history.

Short-term Interests: The short-term interests capture the evolving preferences at different time steps. For each time step t, we independently encode the partial sequence up to that point:

$$l_t^u = \text{LLM2Vec}(S_t^u), \quad \forall t \in \{1, 2, \ldots, n-1\} \tag{5}$$

where $S_t^u = \{i_1^u, i_2^u, \ldots, i_t^u\}$ is the complete subsequence of interactions up to time step t. This results in a total of $n-1$ distinct separate encodings, each representing the user's interests at a different point in their interaction history.

Similarly, for the collaborative interests derived from the sequential recommendation models:

$$\mathbf{e}_2^u = \text{SRS}(S^u) \tag{6}$$

$$\mathbf{h}_t^u = \text{SRS}(S_t^u), \quad \forall t \in \{1, 2, \ldots, n-1\} \tag{7}$$

where $\text{SRS}(\cdot)$ represents the encoding function of the sequential recommendation models.

In summary, we extract two types of interest embeddings:
- **Semantic Interests**: Long-term semantic interest $\mathbf{e}_1^u$ and corresponding short-term semantic interests $\{l_1^u, l_2^u, \ldots, l_{n-1}^u\}$ effectively derived from LLM embeddings.
- **Collaborative Interests**: Long-term collaborative interest $\mathbf{e}_2^u$ and short-term collaborative interests $\{\mathbf{h}_1^u, \mathbf{h}_2^u, \ldots, \mathbf{h}_{n-1}^u\}$ derived from the sequential model.

3.3.2 Cross-Modal Interest Alignment To effectively combine the strengths of both semantic and collaborative information, we align these interest embeddings using InfoNCE loss [21, 50]. This alignment maximizes the mutual information between corresponding interest embeddings from different modalities [54].

The InfoNCE loss for interest alignment is formulated as:

$$\mathcal{L}_{Info} = \mathcal{L}_{long} + \mathcal{L}_{short} \tag{8}$$

$$\mathcal{L}_{long} = -\frac{1}{N} \sum_{i=1}^{N} \log \frac{e^{sim(\mathbf{e}_2^i, \mathbf{e}_1^i)/\tau}}{\sum_{j=1}^{N} e^{sim(\mathbf{e}_2^i, \mathbf{e}_1^j)/\tau}} \tag{9}$$

$$\mathcal{L}_{short} = -\frac{1}{M} \sum_{i=1}^{M} \log \frac{e^{sim(\mathbf{h}_i, l_i)/\tau}}{\sum_{j=1}^{M} e^{sim(\mathbf{h}_i, l_j)/\tau}} \tag{10}$$

where N denotes the batch size, representing the number of users processed in each training iteration and M represents the total number of short-term interest embeddings in the batch, $sim(\cdot, \cdot)$ is the cosine similarity function and τ is a temperature parameter. Specifically, $\mathcal{L}_{long}$ represents the alignment loss between long-term

semantic interest $\mathbf{e}_1^u$ and long-term collaborative interest $\mathbf{e}_2^u$, while $\mathcal{L}_{short}$ represents the alignment loss between short-term semantic interests $\{l_1^u, l_2^u, \ldots, l_{n-1}^u\}$ and short-term collaborative interests $\{\mathbf{h}_1^u, \mathbf{h}_2^u, \ldots, \mathbf{h}_{n-1}^u\}$.

Through this alignment process, IADSR creates unified interest embeddings that leverage both the semantic understanding from LLMs and the collaborative patterns from sequential models, providing a more comprehensive basis for subsequent noise detection.

3.4 Sequence Denoising

After aligning interest embeddings across modalities, we proceed to identify and then filter out noise in user interaction sequences. This is particularly difficult since noise is often contextual and user-dependent without explicit labels. IADSR leverages the complementary nature of semantic and collaborative embeddings to distinguish real user preferences from noise that the traditional single-modal approaches might miss.

For each user u, we compute the cosine similarity between their long-term collaborative interest $\mathbf{e}_2^u$ and the corresponding long-term semantic interest $\mathbf{e}_1^u$:

$$\text{sim}_{long}(u) = \cos(\mathbf{e}_1^u, \mathbf{e}_2^u) \tag{11}$$

Users whose cross-modal similarity λ exceeds a threshold θ proceed to the detailed noise detection stage:

$$\text{qualified}(u) = \mathbf{1}[\text{sim}_{long}(u) \geq \theta] \tag{12}$$

where $\mathbf{1}[\cdot]$ is the indicator function. This preliminary filtering ensures that we only apply denoising to users whose embeddings show sufficient cross-modal consistency, avoiding potentially harmful modifications to sequences where modality alignment is poor.

3.4.1 Item-level Noise Detection For qualified users, we perform item-level noise detection by examining the consistency between different interest embeddings. For each time step t in the qualified user's sequence, we compute three similarity scores between the corresponding interest embeddings as (c_1, c_2, c_3):

$$c_1(t) = \cos(\mathbf{e}_1^u, \mathbf{h}_t^u) \tag{13}$$

$$c_2(t) = \cos(\mathbf{e}_2^u, l_t^u) \tag{14}$$

$$c_3(t) = \cos(\mathbf{h}_t^u, l_t^u) \tag{15}$$

These scores respectively measure: (1) the semantic long-term to collaborative short-term consistency, (2) the collaborative long-term to semantic short-term consistency, and (3) the overall short-term cross-modal consistency. This design ensures that potential noise can be detected from complementary perspectives across modalities and interest levels. We combine these scores to obtain a comprehensive noise indicator:

$$\text{score}(t) = c_1(t) + c_2(t) + c_3(t) \tag{16}$$

3.4.2 Mask Generation via Gumbel-Sigmoid To convert the continuous noise scores into binary denoising decisions, we employ the robust Gumbel-Sigmoid function [41]. The Gumbel-Sigmoid function enables differentiable binary sampling during training by adding Gumbel noise to logits and applying a temperature-controlled sigmoid function, allowing models to make discrete decisions (like masking noise) while effectively maintaining gradient

flow for end-to-end training [26, 48]:

$$m_t = \text{GumbelSigmoid}(\text{score}(t), \tau, \text{hard} = \text{True})$$

$$= \begin{cases} \mathbf{1}[y_t > 0.5], & \text{if hard} = \text{True} \\ y_t, & \text{if hard} = \text{False} \end{cases} \tag{17}$$

$$y_t = \delta\left(\frac{\text{score}(t) + g_t}{\tau}\right)$$

$$g_t = -\log(-\log(U_t + \epsilon) + \epsilon)$$

$$U_t \sim \text{Uniform}(0, 1)$$

Here, τ is the temperature parameter controlling the smoothness of the approximation, δ is the sigmoid function, U_t is a uniform random variable, and ϵ is a constant added for numerical stability. When hard = True, we discretize the output to binary values while preserving gradients through a straight-through estimator.

The resulting mask $m_t \in \{0, 1\}$ indicates whether each interaction should be preserved (1) or filtered out (0) as noise. By incorporating multiple similarity measures and maintaining differentiability, our model can make effective discrete denoising decisions while learning from its own denoising process through backpropagation.

The denoised sequence for user u is then obtained by applying the mask to the original sequence:

$$S_{denoised}^u = \{i_t^u \mid m_t = 1, t = 1, 2, \ldots, n\} \tag{18}$$

3.5 Sequence Reconstruction

To prevent the loss of critical signals from "over-denoising" (especially for cold items), we use a sequence reconstruction mechanism to balance noise removal with the preservation of user preferences.

3.5.1 Progressive Denoising Process
Our approach employs a progressive denoising strategy across training epochs. For each user, we apply the mask learned from the previous epoch to the original sequence embeddings, preserving the model's incremental learning process while always anchoring to the original data. Formally, for a user u at epoch e:

$$\mathbf{X}_u^{(e)} = \mathbf{X}_u^{original} \odot \mathbf{K}_u^{(e-1)} \tag{19}$$

where $\mathbf{X}_u^{(e)}$ represents the input sequence embedding at epoch e, $\mathbf{X}_u^{original}$ is the original unmodified sequence embedding, $\mathbf{K}_u^{(e-1)}$ is the binary mask generated from the previous epoch, and $\odot$ denotes element-wise multiplication. For the initial epoch ($e = 0$), we use the original sequence without masking.

3.5.2 Decoder-based Reconstruction
To ensure that the denoising process preserves essential information, we employ a decoder to reconstruct the original sequence from the denoised representation:

$$\hat{\mathbf{X}}_u = \text{Decoder}(\mathbf{H}_u \odot \mathbf{K}_u) \tag{20}$$

where $\hat{\mathbf{X}}_u$ represents the reconstructed sequence embedding for user u, $\mathbf{H}_u$ represents the model's hidden states (i.e., GRU output embeddings), $\mathbf{K}_u$ is the dynamically generated binary mask for the current epoch, and $\odot$ denotes element-wise multiplication. This process effectively transforms the denoised hidden states back into the original embedding space, allowing us to directly compare the reconstruction with the original input embeddings.

Table 1: Experimental data statistics.

Dataset	# Users	# Items	# Actions	Avg. len	Sparsity
Beauty	22,363	12,101	198,502	8.9	99.93%
Sports	35,598	18,357	296,337	8.3	99.95%
Toys	19,412	11,924	167,597	8.6	99.93%
ML-100K	943	1,682	100,000	106.0	93.70%

3.5.3 Reconstruction Loss
To ensure our denoising process preserves real user preferences while removing only truly noisy interactions, we introduce a reconstruction objective that:
- Encourages selective denoising by penalizing the removal of real preference signals.
- Provides additional training supervision that helps the model learn more robust embeddings.
- Anchors the denoised embeddings to the original data, preventing representation drift.

We implement this objective by systematically minimizing the mean squared error between the reconstructed sequence and the corresponding original sequence embeddings:

$$\mathcal{L}_{recon} = \frac{1}{|\mathcal{U}_{mask}|} \sum_{u \in \mathcal{U}_{mask}} ||\hat{\mathbf{X}}_u - \mathbf{X}_u^{original}||_2^2 \tag{21}$$

where $\mathcal{U}_{mask}$ represents the set of users who passed the initial cross-modal consistency check. This squared L2 distance effectively captures the overall reconstruction quality across all dimensions of the embedding space.

The total loss for our model combines the three components:

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \mathcal{L}_{Info} + \mathcal{L}_{recon} \tag{22}$$

where $\mathcal{L}_{CE}$ is the standard cross-entropy loss for next item prediction, $\mathcal{L}_{Info}$ is the interest alignment loss, and $\mathcal{L}_{recon}$ is the sequence reconstruction loss.

4 Experiments

In this section, we present the experiment results on four public datasets to validate the effectiveness of our methods. Our evaluation is guided by the following research questions:
- **RQ1:** How does IADSR perform compared with the state-of-the-art denoising baseline methods?
- **RQ2:** Is IADSR compatible with different sequential recommendation models?
- **RQ3:** What impact do the proposed loss functions have on the recommendation performance in IADSR?
- **RQ4:** How sensitive is IADSR to hyperparameters?
- **RQ5:** How do the introduced semantic embeddings contribute to denoising in IADSR?

4.1 Experimental Setting

4.1.1 Datasets and Pre-processing
We conduct experiments on three domains of Amazon datasets and the MovieLens-100K dataset. Their statistics are summarized in Table 1. Average length (Avg. len) represents the mean number of interactions per user, reflecting the length of user behavior sequences, ranging from approximately 8-9 interactions for Beauty, Sports, and Toys datasets to 106 interactions for ML-100K. Sparsity indicates the gap between actual user-item interactions and the theoretically maximum possible interactions

in the user-item matrix, demonstrating that all datasets are highly sparse, with Beauty, Sports, and Toys having approximately 99.9% sparsity, while ML-100K is relatively less sparse at 93.7%.

- **Amazon**: We utilize three categories from the Amazon review dataset: Beauty, Sports & Outdoors, and Toys & Games. For each category, we followed the previous studies [4, 67, 73, 86] and adopted the 5-core version where each user and item has at least five interactions, ensuring sufficient sequential patterns for modeling and reducing data sparsity. Each dataset contains user-item interaction records with timestamps, allowing us to construct highly meaningful temporal behavioral sequences. The product metadata in these datasets enables the extraction of semantic information through LLMs.
- **MovieLens-100K**: A widely-used benchmark dataset in the movie recommendation domain, containing 100,000 ratings from users on different movies. It offers a complementary domain to e-commerce and features more structured item attributes.

For pre-processing, we follow the standard practices in sequential recommendation as [57, 73, 76]. Specifically, each user's interaction sequence is sorted chronologically by timestamp to preserve the temporal order of user behaviors. Based on the observed average sequence lengths in the Amazon datasets, we set the maximum sequence length to 32 for these datasets, while for MovieLens we use a maximum length of 50 to ensure optimal performance.

4.1.2 Evaluation Metrics For evaluation, we adopt two highly widely used metrics in sequential recommendation: Hit Ratio (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K). HR@K accurately measures the proportion of test cases where the ground truth item appears in the top-K recommendation list, effectively reflecting the model's ability to recall relevant items [33, 70]. NDCG@K further considers the position of the ground truth item within the top-K list, assigning higher weights to higher positions, thus evaluating both precision and ranking quality [38, 65]. We report results for HR@$K \in \{5, 10, 20\}$, and NDCG@$K \in \{5, 10, 20\}$ to comprehensively evaluate recommendation performance at different levels of K. Following standard practice, we employ the leave-one-out strategy [6] for evaluation.

4.1.3 Backbones Since our method is compatible with different sequential recommendation models, we choose the following three representatives as the backbone [66].

- **GRU4Rec** [23]: One of the pioneering works in sequential recommendation that leverages Gated Recurrent Units to effectively capture temporal dynamics in user-item interaction sequences.
- **SASRec** [30]: This approach introduces self-attention mechanisms into sequential recommendation, enabling the model to adaptively focus on relevant historical interactions while maintaining computational efficiency.
- **Caser** [58]: By employing both horizontal and vertical convolutional filters, this CNN-based method captures local and global sequential patterns simultaneously to enhance accuracy.

4.1.4 Baselines We compare our approach with representative denoising sequential recommendation methods, including directly removing noise items and employing data augmentation:

- **STEAM** [40]: Self-correcting approach that modifies sequences through keep, delete, or insert operations using self-supervised learning to identify and fix misclicked items.
- **DCRec** [71]: Denoising contrastive framework that separates user conformity from genuine interests using a multi-channel weighting network and contrastive learning.
- **HSD** [75]: Hierarchical sequence denoising model that learns two-level item inconsistency signals to identify and remove noisy interactions without requiring explicit noise labels.
- **SSDRec** [76]: Framework that uses multi-relation graphs for cross-sequence patterns, injects global information at specific positions, and applies hierarchical denoising to identify noise in both enhanced and original sequences.

4.1.5 Implementation Details Following common practices in sequential recommendation, we set the embedding dimension to 64 and the hidden state dimension to 128 with 2 GRU layers. The model is trained with a batch size of 32 and the Adam optimizer with a learning rate of 1e-4. We use the Llama-3.1-8B-Instruct [18] model to generate semantic embeddings with llm2vec [5] and implement early stopping with a patience of 10 epochs to prevent overfitting.

4.2 Overall Performance (RQ1 and RQ2)

Tables 2 and 3 present the overall performance comparison of our proposed method against baseline and recent denoising models across four datasets. We evaluate IADSR on three backbones (GRU4Rec, Caser, SASRec) and compare it with HSD, SSDRec, as well as standalone frameworks DCRec and STEAM.

From the experimental results, we observe the following findings:

- **Effectiveness of Our Approach:** IADSR consistently outperforms the second-best approaches across different backbones and datasets. On Beauty with GRU4Rec, compared to the second-best method (HSD), IADSR shows average improvements of 24.6% across all metrics (31.0% on HR@5, 8.5% on HR@10, while HSD leads on HR@20, 51.1% on NDCG@5, 31.9% on NDCG@10, and 14.1% on NDCG@20). On Toys with Caser, the average gain reaches 36.3%. On ML-100K with SASRec, IADSR achieves a 7.8% average gain across metrics. These consistent gains highlight its robust denoising capability.
- **Performance Across Different Backbones:** Improvements hold regardless of the backbone. Even with SASRec, the strongest baseline, IADSR raises HR@20 on Beauty from 0.0554 to 0.0836 (+50.9
- **Comparison with Other Denoising Methods:** IADSR outperforms HSD and SSDRec on most metrics. While SSDRec occasionally excels (e.g., HR@20 with GRU4Rec on Beauty), IADSR provides more balanced improvements, achieving average HR@10 gains of 10.2%, 8.4%, and 6.7% on Beauty, Sports, and Toys, respectively.
- **Dataset-specific Observations:** Gains are most pronounced on Beauty and Sports, suggesting higher noise levels. On ML-100K, improvements are smaller, implying less or different noise patterns.

Our denoising framework consistently outperforms state-of-the-art methods across multiple backbones by integrating semantic and collaborative signals to effectively mitigate noise in sequential

Table 2: Overall performance comparison on Beauty, Sports and Toys dataset. Boldface denotes the best result and underline indicates the second-best results. '*' denotes significant improvement (i.e., two-sided t-test with p < 0.05).

Dataset	Model		HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20
Beauty	GRU4Rec	base	0.0174	0.0249	0.0351	0.0120	0.0144	0.0169
		HSD	<u>0.0229</u>	<u>0.0365</u>	**0.0520***	<u>0.0141</u>	<u>0.0185</u>	<u>0.0234</u>
		SSDRec	0.0218	0.0360	<u>0.0510</u>	0.0136	0.0181	0.0229
		IADSR	**0.0300***	**0.0396***	0.0486	**0.0213***	**0.0244***	**0.0267***
	Caser	base	0.0075	0.0130	0.0216	0.0043	0.0060	0.0082
		HSD	0.0137	0.0214	0.0327	<u>0.0086</u>	0.0111	0.0140
		SSDRec	<u>0.0142</u>	<u>0.0235</u>	<u>0.0363</u>	0.0085	<u>0.0115</u>	<u>0.0147</u>
		IADSR	**0.0178***	**0.0297***	**0.0469***	**0.0094***	**0.0137***	**0.0188***
	SASRec	base	0.0267	0.0385	0.0554	0.0184	0.0218	0.0261
		HSD	0.0245	0.0436	0.0668	0.0140	0.0201	0.0260
		SSDRec	**0.0342***	**0.0538***	<u>0.0782</u>	<u>0.0196</u>	<u>0.0259</u>	<u>0.0321</u>
		IADSR	<u>0.0323</u>	**0.0538***	**0.0836***	**0.0202***	**0.0272***	**0.0349***
	STEAM		0.0292	0.0395	0.0515	0.0201	0.0234	0.0264
	DCRec		0.0110	0.0202	0.0370	0.0066	0.0095	0.0137
Sports	GRU4Rec	base	0.0059	0.0940	0.0150	0.0037	0.0048	0.0060
		HSD	<u>0.0136</u>	0.0186	0.0303	0.0077	0.0099	0.0129
		SSDRec	0.0122	<u>0.0196</u>	<u>0.0318</u>	<u>0.0083</u>	<u>0.0107</u>	<u>0.0137</u>
		IADSR	**0.0155***	**0.0240***	**0.0340***	**0.0095***	**0.0122***	**0.0148***
	Caser	base	0.0059	0.0081	0.0118	0.0026	0.0031	0.0048
		HSD	<u>0.0063</u>	0.0119	0.0212	**0.0043***	0.0061	<u>0.0084</u>
		SSDRec	0.0060	<u>0.0123</u>	<u>0.0213</u>	<u>0.0042</u>	<u>0.0062</u>	<u>0.0084</u>
		IADSR	**0.0080***	**0.0152***	**0.0258***	0.0041	**0.0063***	**0.0089***
	SASRec	base	0.0112	0.0178	0.0269	0.0074	0.0093	0.0102
		HSD	0.0119	0.0202	0.0309	0.0078	0.0108	0.0127
		SSDRec	<u>0.0132</u>	<u>0.0212</u>	<u>0.0338</u>	<u>0.0099</u>	<u>0.0111</u>	<u>0.0131</u>
		IADSR	**0.0155***	**0.0260***	**0.0383***	**0.0101***	**0.0114***	**0.0139***
	STEAM		0.0149	0.0182	0.0250	0.0078	0.0101	0.0122
	DCRec		0.0080	0.0141	0.0288	0.0068	0.0088	0.0105
Toys	GRU4Rec	base	0.0110	0.0125	0.0133	0.0080	0.0085	0.0091
		HSD	<u>0.0167</u>	<u>0.0266</u>	**0.0413***	<u>0.0109</u>	<u>0.0142</u>	<u>0.0179</u>
		SSDRec	0.0140	0.0227	0.0354	0.0090	0.0118	0.0149
		IADSR	**0.0189***	**0.0281***	<u>0.0389</u>	**0.0130***	**0.0160***	**0.0186***
	Caser	base	0.0054	0.0089	0.0145	0.0035	0.0046	0.0060
		HSD	<u>0.0066</u>	<u>0.0124</u>	0.0192	0.0041	<u>0.0060</u>	0.0076
		SSDRec	0.0065	0.0116	<u>0.0198</u>	<u>0.0044</u>	<u>0.0060</u>	<u>0.0081</u>
		IADSR	**0.0098***	**0.0163***	**0.0224***	**0.0080***	**0.0108***	**0.0129***
	SASRec	base	0.0288	0.0394	0.0468	0.0162	0.0216	0.0254
		HSD	<u>0.0299</u>	0.0451	0.0649	**0.0180***	<u>0.0229</u>	0.0279
		SSDRec	**0.0303***	<u>0.0473</u>	<u>0.0689</u>	<u>0.0172</u>	0.0226	<u>0.0281</u>
		IADSR	0.0297	**0.0483***	**0.0697***	<u>0.0172</u>	**0.0230***	**0.0287***
	STEAM		0.0154	0.0330	0.0630	0.0087	0.0150	0.0214
	DCRec		0.0204	0.0379	0.0655	0.0123	0.0178	0.0247

recommendation, demonstrating robust and versatile performance across datasets.

4.3 Ablation Study (RQ3)

To validate the effectiveness of each component in our proposed framework, we conduct an ablation study on the Beauty dataset. Table 4 presents the results with different variants of our model. Specifically, we investigate the impact of different loss functions and interest embeddings:

- **w/o** *both*: Removing both InfoNCE loss and reconstruction loss, leaving only the basic cross-entropy loss.

- **w/o** $\mathcal{L}_{info}$: Removing the InfoNCE loss that aligns with semantic and collaborative embeddings.
- **w/o** $\mathcal{L}_{recon}$: Removing the sequence reconstruction loss.
- **Short-only**: Using only short-term interests for noise detection.
- **Long-only**: Using only long-term interests for noise detection.
- **Full Model**: Our full model with all components.

The results demonstrate that removing both losses leads to substantial drops (−42.0% HR@5, −38.8% NDCG@5), confirming their importance. $\mathcal{L}_{info}$ proves more critical than $\mathcal{L}_{recon}$, highlighting the necessity of aligning semantic and collaborative spaces.

Table 3: Overall performance comparison on Movielens-100k.

Dataset	Model		HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20
ML-100K	GRU4Rec	base	0.0180	0.0296	0.0607	0.0102	0.0152	0.0230
		HSD	0.0148	0.0339	0.0732*	0.0094	0.0163	0.0252
		SSDRec	0.0256	0.0511*	0.0721	0.0155	0.0217	0.0268
		IADSR	0.0286*	0.0455	0.0696	0.0176*	0.0223*	0.0286*
	Caser	base	0.0204	0.0361	0.0541	0.0104	0.0176	0.0210
		HSD	0.0255	0.0456	0.0721	0.0147*	0.0214*	0.0271
		SSDRec	0.0243	0.0424	0.0732	0.0142	0.0203	0.0278*
		IADSR	0.0265*	0.0467*	0.0742*	0.0142	0.0207	0.0276
	SASRec	base	0.0191	0.0350	0.0509	0.0114	0.0153	0.0200
		HSD	0.0223*	0.0403	0.0742	0.0143	0.0190	0.0256
		SSDRec	0.0223*	0.0435	0.0785	0.0140	0.0209*	0.0295
		IADSR	0.0212	0.0477*	0.0827*	0.0145*	0.0205	0.0311*
	STEAM		0.0207	0.0372	0.0563	0.0126	0.0178	0.0202
	DCRec		0.0215	0.0424	0.0710	0.0135	0.0202	0.0279

Table 4: Ablation study on the Amazon Beauty dataset.

Variants	HR@5	HR@10	HR@20	NDCG5	NDCG10	NDCG20
w/o *both*	0.0174	0.0249	0.0351	0.0120	0.0144	0.0169
w/o $\mathcal{L}_{info}$	0.0283	0.0346	0.0386	0.0202	0.0223	0.0233
w/o $\mathcal{L}_{recon}$	0.0213	0.0305	0.0416	0.0139	0.0168	0.0196
Short-only	0.0218	0.0329	0.0483	0.0144	0.0180	0.0219
Long-only	0.0225	0.0328	0.0456	0.0151	0.0181	0.0223
Full Model	**0.0300**	**0.0396**	**0.0486**	**0.0196**	**0.0259**	**0.0321**

Figure 3: (a) HR@5 and (b) NDCG@5 in hyper-parameter study on θ.

Regarding interest embeddings, we observe that long-term interests offer more stable signals than short-term ones, but the Full Model outperforms both (with improvements of up to 33.3% in HR@5 and 36.1% in NDCG@5 compared to Long-only), confirming our hypothesis that combining both time scales offers the most comprehensive view for identifying noise in user sequences.

4.4 Hyper-parameter Study (RQ4)

To understand the impact of the cross-modal consistency threshold θ on our framework's performance, we conducted experiments varying this parameter from -1.0 to 0.9. Figure 3 illustrates performance trends across different metrics. We observe that performance peaks around $\theta = -0.9$ (with HR@5=0.03 and NDCG@5=0.0213) and remains relatively stable across negative thresholds (-0.9 to -0.1). However, as θ exceeds 0.7, we observe significant performance degradation across all metrics.

These results indicate that moderate cross-modal alignment is sufficient for effective noise identification. Setting θ too high forces excessive agreement between semantic and collaborative signals, potentially ignoring complementary information. Based on these findings, we set $\theta = -0.9$ as the default value in our framework.

4.5 Case Study (RQ5)

To provide qualitative insights into our model's denoising capability, we randomly selected two users from the Beauty dataset and analyzed how our approach effectively identifies noisy interactions by leveraging both semantic and collaborative signals.

Table 5 illustrates the denoising results across different methods (IADSR, HSD, Steam) for these selected users. The blue text
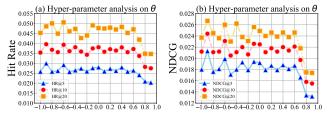
indicates cold items (representing the 20% of items with the lowest interaction counts), the red text represents hot items (the 20% with the highest interaction counts), and the black text denotes normal items, respectively. Furthermore, we display the user profile derived from their historical interaction patterns.

4.5.1 User Preference Analysis For User 22099, IADSR precisely filtered the irrelevant hair styling product while retaining body-contouring items, outperforming HSD and STEAM, which misclassified relevant products.

For User 19852, IADSR flagged only the "Beauty Without Cruelty" lotion as noise, while preserving other cold items consistent with the user's beauty interests. HSD overgeneralized by filtering all three natural products, including relevant ones such as the Essie base coat and Aloe Vera gel. STEAM performed worse, incorrectly marking an anti-wrinkle complex and eye makeup as noise despite the user's clear interest in anti-aging skincare and eye products. These cases demonstrate IADSR's ability to balance recommendation diversity with precise noise filtering, ensuring both relevance and coverage.

5 Related Work

In this section, we summarize the related works on sequential recommender systems and denoising sequential recommendation.

5.1 Sequential Recommender System

The sequential recommendation focuses on capturing temporal dynamics in user behaviors to predict future interactions. Early approaches used Markov Chain-based models [53], which were later superseded by deep learning methods [39], including GRU4Rec [23]

Table 5: A case study to demonstrate the effectiveness of IADSR in denoising cold items.

User ID	IADSR	HSD	Steam	User Profile
22099	Bedhead Hook Up 1/2 STRAIGHT	Bedhead Hook Up 1/2 STRAIGHT Skinny Cream Clinically Proven Cellulite Reduction, 6 Ounce	Slim Extreme 3d Super-concentrated Serum Shaping Buttocks, 200mL Palmers Cocoa Butter Bust Firming Cream 4.4oz	Based on the user's historical interactions, this user is primarily interested in body contouring, firming, and weight loss-related beauty products, with a particular focus on shaping creams and anti-cellulite products targeting specific body areas (buttocks, bust).
19852	Beauty Without Cruelty Fragrance Free Hand & Body Lotion, 100 % Vegetarian, 16 fl ozs	Beauty Without Cruelty Fragrance Free Hand & Body Lotion, 100 % Vegetarian, 16 fl ozs Essie Ridge Filler Base Coat, 0.46 oz Fruit Of The Earth 100% Aloe Vera 6oz. Gel Tube	Hydroxatone AM/PM Anti-Wrinkle Complex SPF 15 Blinc Kiss Me Mascara, Dark Brown Alkaline	Based on the user's historical interactions, this user has comprehensive beauty interests across multiple categories, showing particular focus on anti-aging skincare, natural/vegetarian beauty products, eye makeup, hair care tools, and various face and body treatments from both high-end and drugstore brands.

with RNNs and Caser [58] with CNNs. Recently, Transformer-based models like SASRec [30] have achieved consistently superior performance by leveraging self-attention mechanisms to model item relationships in user sequences.

However, these models lack noise mitigation mechanisms, making them vulnerable to accidental clicks or exploratory behaviors. IADSR addresses this by integrating LLM semantic knowledge with collaborative signals to distinguish preferences from noise.

5.2 Denoising Sequential Recommendation

Recent works on denoising sequential recommendation can be grouped into two categories. The first type relies solely on collaborative signals (ID-based interactions). For example, HSD [75] detects inconsistency signals to drop noisy items, while ADT [64] prunes high-loss interactions during training. Other approaches, such as STEAM [40], SSDRec [76], DCRec [71], and DCF [22], adjust or reweight noisy items through self-correction, graph modeling, or debiased contrastive learning. The second type incorporates additional modalities beyond IDs. LLM4DSR [59] leverages large language models to identify and replace noisy items, while LLaRD [63] extracts semantic preference patterns from textual contexts.

While ID-based methods risk overlooking the semantic richness of user behaviors, multimodal approaches struggle to align heterogeneous signals with collaborative sequences. Existing denoising strategies also face notable limitations: (1) reliance on collaborative signals makes them ineffective for cold items with sparse interactions; (2) LLM-based methods often demand costly fine-tuning;

and (3) many are tied to specific architectures, limiting generalizability. In contrast, our framework leverages LLM embeddings to improve denoising, remains compatible with diverse sequential recommenders, and is particularly effective for cold items where collaborative signals are insufficient.

6 Conclusion

In this paper, we proposed IADSR, a novel framework that integrates semantic knowledge from large language models with collaborative signals for denoising sequential recommendation. Through a two-stage process of cross-modal alignment, noise detection, and sequence reconstruction, IADSR effectively preserves real user preferences. Experiments on four public datasets show that it consistently outperforms state-of-the-art denoising methods across different sequential recommendation backbones.

GenAI Usage Disclosure

We provide full disclosure of our use of GenAI tools throughout this research and writing:

Data Processing: We utilized Claude to transform raw datasets into the required format during the data processing stage, ensuring data consistency and reducing manual processing errors.

Writing: Claude was employed solely for grammar checking and improving sentence clarity and expression.

References

[1] Arkadeep Acharya, Brijraj Singh, and Naoyuki Onoe. 2023. Llm based generation of item-description for recommendation system. In *Proceedings of the 17th ACM conference on recommender systems*. 1204–1207.

[2] Shilpi Aggarwal, Dipanjan Goswami, Madhurima Hooda, Amirta Chakravarty, Arpan Kar, and Vasudha. 2019. Recommendation systems for interactive multimedia entertainment. In *Data visualization and knowledge engineering: spotting data points with artificial intelligence*. Springer, 23–48.

[3] Malak Al-Hassan, Haiyan Lu, and Jie Lu. 2015. A semantic enhanced hybrid recommendation approach: A case study of e-Government tourism service recommendation system. *Decision Support Systems* 72 (2015), 97–109.

[4] Kumar Deep Barman, Bhaskar Bordoloi, Ansuman Kumar, and Anindya Halder. 2024. Review Rating Predictions Using Improved Deep Learning Architecture. In *2024 IEEE 16th International Conference on Computational Intelligence and Communication Networks (CICN)*. IEEE, 468–472.

[5] Parishad BehnamGhader, Vaibhav Adlakha, Marius Mosbach, Dzmitry Bahdanau, Nicolas Chapados, and Siva Reddy. 2024. Llm2vec: Large language models are secretly powerful text encoders. *arXiv preprint arXiv:2404.05961* (2024).

[6] Emily Black and Matt Fredrikson. 2021. Leave-one-out unfairness. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. 285–295.

[7] Artun Boz, Wouter Zorgdrager, Zoe Kotti, Jesse Harte, Panos Louridas, Vassilios Karakoidas, Dietmar Jannach, and Marios Fragkoulis. 2024. Improving sequential recommendations with llms. *ACM Transactions on Recommender Systems* (2024).

[8] Huiyuan Chen, Yusan Lin, Menghai Pan, Lan Wang, Chin-Chia Michael Yeh, Xiaoting Li, Yan Zheng, Fei Wang, and Hao Yang. 2022. Denoising self-attentive sequential recommendation. In *Proceedings of the 16th ACM conference on recommender systems*. 92–101.

[9] Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems* 41, 3 (2023), 1–39.

[10] Minmin Chen, Kilian Weinberger, Fei Sha, and Yoshua Bengio. 2014. Marginalized denoising auto-encoders for nonlinear representations. In *International conference on machine learning*. PMLR, 1476–1484.

[11] Wenjie Chen, Yi Zhang, Honghao Li, Lei Sang, and Yiwen Zhang. 2025. Dual-Domain Collaborative Denoising for Social Recommendation. *IEEE Transactions on Computational Social Systems* (2025).

[12] Mingyue Cheng, Hao Zhang, Qi Liu, Fajie Yuan, Zhi Li, Zhenya Huang, Enhong Chen, Jun Zhou, and Longfei Li. 2024. Empowering Sequential Recommendation from Collaborative Signals and Semantic Relatedness. In *International Conference on Database Systems for Advanced Applications*. Springer, 196–211.

[13] Byeongjin Choe, Taegwan Kang, and Kyomin Jung. 2021. Recommendation system with hierarchical recurrent neural network for long-term time series. *IEEE Access* 9 (2021), 72033–72039.

[14] Ziwei Fan, Ke Xu, Zhang Dong, Hao Peng, Jiawei Zhang, and Philip S Yu. 2023. Graph collaborative signals denoising and augmentation for recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*. 2037–2041.

[15] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. 2020. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)* 39, 1 (2020), 1–42.

[16] Zichuan Fu, Xiangyang Li, Chuhan Wu, Yichao Wang, Kuicai Dong, Xiangyu Zhao, Mengchen Zhao, Huifeng Guo, and Ruiming Tang. 2025. A unified framework for multi-domain ctr prediction via large language models. *ACM Transactions on Information Systems* 43, 5 (2025), 1–33.

[17] Jingtong Gao, Xiangyu Zhao, Muyang Li, Minghao Zhao, Runze Wu, Ruocheng Guo, Yiding Liu, and Dawei Yin. 2024. Smlp4rec: An efficient all-mlp architecture for sequential recommendations. *ACM Transactions on Information Systems* 42, 3 (2024), 1–23.

[18] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[19] Yongqiang Han, Hao Wang, Kefan Wang, Likang Wu, Zhi Li, Wei Guo, Yong Liu, Defu Lian, and Enhong Chen. 2024. End4rec: Efficient noise-decoupling

[20] for multi-behavior sequential recommendation. *arXiv preprint arXiv:2403.17603* (2024).

[20] Jesse Harte, Wouter Zorgdrager, Panos Louridas, Asterios Katsifodimos, Dietmar Jannach, and Marios Fragkoulis. 2023. Leveraging large language models for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 1096–1102.

[21] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9729–9738.

[22] Zhuangzhuang He, Yifan Wang, Yonghui Yang, Peijie Sun, Le Wu, Haoyue Bai, Jinqi Gong, Richang Hong, and Min Zhang. 2024. Double correction framework for denoising recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1062–1072.

[23] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).

[24] 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 *Companion Proceedings of the ACM Web Conference 2024*. 103–111.

[25] Kirti Jain and Rajni Jindal. 2023. Sampling and noise filtering methods for recommender systems: A literature review. *Engineering Applications of Artificial Intelligence* 122 (2023), 106129.

[26] Eric Jang, Shixiang Gu, and Ben Poole. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* (2016).

[27] Jongwon Jeong, Jeong Choi, Hyunsouk Cho, and Sehee Chung. 2022. FPAdaMetric: False-positive-aware adaptive metric learning for session-based recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 36. 4039–4047.

[28] Yogesh Jhamb, Travis Ebesu, and Yi Fang. 2018. Attentive contextual denoising autoencoder for recommendation. In *Proceedings of the 2018 ACM SIGIR international conference on theory of information retrieval*. 27–34.

[29] Miao Jiang, Ziyi Yang, and Chen Zhao. 2017. What to play next? A RNN-based music recommendation system. In *2017 51st Asilomar conference on signals, systems, and computers*. IEEE, 356–358.

[30] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.

[31] Lingchen Kong, Chuanqi Qi, and Hou-Duo Qi. 2019. Classical multidimensional scaling: A subspace perspective, over-denoising, and outlier detection. *IEEE Transactions on Signal Processing* 67, 14 (2019), 3842–3857.

[32] Chengxi Li, Yejing Wang, Qidong Liu, Xiangyu Zhao, Wanyu Wang, Yiqi Wang, Lixin Zou, Wenqi Fan, and Qing Li. 2023. STRec: Sparse transformer for sequential recommendations. In *Proceedings of the 17th ACM conference on recommender systems*. 101–111.

[33] Dong Li, Ruoming Jin, Jing Gao, and Zhi Liu. 2020. On sampling top-k recommendation evaluation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2114–2124.

[34] Lei Li, Yongfeng Zhang, and Li Chen. 2023. Prompt distillation for efficient llm-based recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1348–1357.

[35] Siqing Li, Liuyi Yao, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Tonglei Guo, Bolin Ding, and Ji-Rong Wen. 2021. Debiasing learning based cross-domain recommendation. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. 3190–3199.

[36] Xiaopeng Li, Fan Yan, Xiangyu Zhao, Yichao Wang, Bo Chen, Huifeng Guo, and Ruiming Tang. 2023. Hamur: Hyper adapter for multi-domain recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 1268–1277.

[37] Yang Li, Tong Chen, Yadan Luo, Hongzhi Yin, and Zi Huang. 2021. Discovering collaborative signals for next POI recommendation with iterative Seq2Graph augmentation. *arXiv preprint arXiv:2106.15814* (2021).

[38] Daryl Lim, Julian McAuley, and Gert Lanckriet. 2015. Top-n recommendation with missing implicit feedback. In *Proceedings of the 9th ACM Conference on Recommender Systems*. 309–312.

[39] Weilin Lin, Xiangyu Zhao, Yejing Wang, Yuanshao Zhu, and Wanyu Wang. 2023. Autodenoise: Automatic data instance denoising for recommendations. In *Proceedings of the ACM Web Conference 2023*. 1003–1011.

[40] Yujie Lin, Chenyang Wang, Zhumin Chen, Zhaochun Ren, Xin Xin, Qiang Yan, Maarten de Rijke, Xiuzhen Cheng, and Pengjie Ren. 2023. A self-correcting sequential recommender. In *Proceedings of the ACM Web Conference 2023*. 1283–1293.

[41] Pengbo Liu, Hailong Cao, and Tiejun Zhao. 2021. Gumbel-attention for multimodal machine translation. *arXiv preprint arXiv:2103.08862* (2021).

[42] Qiang Liu, Shu Wu, and Liang Wang. 2017. Multi-behavioral sequential prediction with recurrent log-bilinear model. *IEEE Transactions on Knowledge and Data Engineering* 29, 6 (2017), 1254–1267.

[43] Qidong Liu, Xian Wu, Yejing Wang, Zijian Zhang, Feng Tian, Yefeng Zheng, and Xiangyu Zhao. 2024. Llm-esr: Large language models enhancement for long-tailed sequential recommendation. *Advances in Neural Information Processing Systems* 37 (2024), 26701–26727.

[44] Qidong Liu, Xian Wu, Xiangyu Zhao, Yuanshao Zhu, Zijian Zhang, Feng Tian, and Yefeng Zheng. 2024. Large language model distilling medication recommendation model. *arXiv preprint arXiv:2402.02803* (2024).

[45] Shuchang Liu, Qingpeng Cai, Bowen Sun, Yuhao Wang, Ji Jiang, Dong Zheng, Peng Jiang, Kun Gai, Xiangyu Zhao, and Yongfeng Zhang. 2023. Exploration and regularization of the latent action space in recommendation. In *Proceedings of the ACM Web Conference 2023.* 833–844.

[46] Hanjia Lyu, Song Jiang, Hanqing Zeng, Yinglong Xia, Qifan Wang, Si Zhang, Ren Chen, Christopher Leung, Jiajie Tang, and Jiebo Luo. 2023. Llm-rec: Personalized recommendation via prompting large language models. *arXiv preprint arXiv:2307.15780* (2023).

[47] Luis Martínez, Jorge Castro, and Raciel Yera. 2016. Managing natural noise in recommender systems. In *Theory and Practice of Natural Computing: 5th International Conference, TPNC 2016, Sendai, Japan, December 12-13, 2016, Proceedings 5.* Springer, 3–17.

[48] Ignavier Ng, Shengyu Zhu, Zhuangyan Fang, Haoyang Li, Zhitang Chen, and Jun Wang. 2022. Masked gradient-based causal structure learning. In *Proceedings of the 2022 SIAM International Conference on Data Mining (SDM).* SIAM, 424–432.

[49] Michael P O'Mahony, Neil J Hurley, and Guénolé CM Silvestre. 2006. Detecting noise in recommender system databases. In *Proceedings of the 11th international conference on Intelligent user interfaces.* 109–115.

[50] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[51] Ahmad Bin Rabiah, Nafis Sadeq, and Julian McAuley. 2024. Bridging Conversational and Collaborative Signals for Conversational Recommendation. *arXiv preprint arXiv:2412.06949* (2024).

[52] Shaina Raza and Chen Ding. 2022. News recommender system: a review of recent progress, challenges, and opportunities. *Artificial Intelligence Review* (2022), 1–52.

[53] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web.* 811–820.

[54] Evgenia Rusak, Patrik Reizinger, Attila Juhos, Oliver Bringmann, Roland S Zimmermann, and Wieland Brendel. 2024. InfoNCE: Identifying the Gap Between Theory and Practice. *arXiv preprint arXiv:2407.00143* (2024).

[55] Andrew I Schein, Alexandrin Popescul, Lyle H Ungar, and David M Pennock. 2002. Methods and metrics for cold-start recommendations. In *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval.* 253–260.

[56] Harald Steck. 2013. Evaluation of recommendations: rating-prediction and ranking. In *Proceedings of the 7th ACM conference on Recommender systems.* 213–220.

[57] 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 *Proceedings of the 28th ACM international conference on information and knowledge management.* 1441–1450.

[58] Jiaxi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining.* 565–573.

[59] Bohao Wang, Feng Liu, Changwang Zhang, Jiawei Chen, Yudi Wu, Sheng Zhou, Xingyu Lou, Jun Wang, Yan Feng, Chun Chen, et al. 2024. Llm4dsr: Leveraging large language model for denoising sequential recommendation. *arXiv preprint arXiv:2408.08208* (2024).

[60] Chen Wang, Liangwei Yang, Zhiwei Liu, Xiaolong Liu, Mingdai Yang, Yueqing Liang, and Philip S Yu. 2024. Collaborative Alignment for Recommendation. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management.* 2315–2325.

[61] Shoujin Wang, Liang Hu, Yan Wang, Longbing Cao, Quan Z Sheng, and Mehmet Orgun. 2019. Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830* (2019).

[62] Shoujin Wang, Xiuzhen Zhang, Yan Wang, and Francesco Ricci. 2024. Trustworthy recommender systems. *ACM Transactions on Intelligent Systems and Technology* 15, 4 (2024), 1–20.

[63] Shuyao Wang, Zhi Zheng, Yongduo Sui, and Hui Xiong. 2025. Unleashing the Power of Large Language Model for Denoising Recommendation. *arXiv preprint arXiv:2502.09058* (2025).

[64] Wenjie Wang, Fuli Feng, Xiangnan He, Liqiang Nie, and Tat-Seng Chua. 2021. Denoising implicit feedback for recommendation. In *Proceedings of the 14th ACM international conference on web search and data mining.* 373–381.

[65] Yining Wang, Liwei Wang, Yuanzhi Li, Di He, and Tie-Yan Liu. 2013. A theoretical analysis of NDCG type ranking measures. In *Conference on learning theory.* PMLR, 25–54.

[66] Yuhao Wang, Xiangyu Zhao, Bo Chen, Qidong Liu, Huifeng Guo, Huanshuo Liu, Yichao Wang, Rui Zhang, and Ruiming Tang. 2023. PLATE: A prompt-enhanced paradigm for multi-scenario recommendations. In *Proceedings of the 46th international ACM SIGIR Conference on Research and Development in Information Retrieval.* 1498–1507.

[67] Shuyuan Xu, Wenyue Hua, and Yongfeng Zhang. 2023. Openp5: Benchmarking foundation models for recommendation. *arXiv preprint arXiv:2306.11134* (2023).

[68] An Yan, Shuo Cheng, Wang-Cheng Kang, Mengting Wan, and Julian McAuley. 2019. CosRec: 2D convolutional neural networks for sequential recommendation. In *Proceedings of the 28th ACM international conference on information and knowledge management.* 2173–2176.

[69] Xianmin Yang, Shuqiang Song, Xinshuo Zhao, and Shengquan Yu. 2018. Understanding user behavioral patterns in open knowledge communities. *Interactive Learning Environments* 26, 2 (2018), 245–255.

[70] Xiwang Yang, Harald Steck, Yang Guo, and Yong Liu. 2012. On top-k recommendation using social networks. In *Proceedings of the sixth ACM conference on Recommender systems.* 67–74.

[71] Yuhao Yang, Chao Huang, Lianghao Xia, Chunzhen Huang, Da Luo, and Kangyi Lin. 2023. Debiased contrastive learning for sequential recommendation. In *Proceedings of the ACM web conference 2023.* 1063–1073.

[72] Wenwen Ye, Shuaiqiang Wang, Xu Chen, Xuepeng Wang, Zheng Qin, and Dawei Yin. 2020. Time matters: Sequential recommendation with complex temporal information. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval.* 1459–1468.

[73] Jiahao Yuan, Zihan Song, Mingyou Sun, Xiaoling Wang, and Wayne Xin Zhao. 2021. Dual sparse attention network for session-based recommendation. In *Proceedings of the AAAI conference on artificial intelligence,* Vol. 35. 4635–4643.

[74] Chi Zhang, Rui Chen, Xiangyu Zhao, Qilong Han, and Li Li. 2023. Denoising and prompt-tuning for multi-behavior recommendation. In *Proceedings of the ACM web conference 2023.* 1355–1363.

[75] Chi Zhang, Yantong Du, Xiangyu Zhao, Qilong Han, Rui Chen, and Li Li. 2022. Hierarchical item inconsistency signal learning for sequence denoising in sequential recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management.* 2508–2518.

[76] Chi Zhang, Qilong Han, Rui Chen, Xiangyu Zhao, Peng Tang, and Hongtao Song. 2024. Ssdrec: self-augmented sequence denoising for sequential recommendation. In *2024 IEEE 40th International Conference on Data Engineering (ICDE).* IEEE, 803–815.

[77] D Zhang, H Wu, and F Yang. 2021. FSCR: A Deep Social Recommendation Model for Misleading Information. Information 2021, 12, 37.

[78] Jinghao Zhang, Yanqiao Zhu, Qiang Liu, Mengqi Zhang, Shu Wu, and Liang Wang. 2022. Latent structure mining with contrastive modality fusion for multimedia recommendation. *IEEE Transactions on Knowledge and Data Engineering* 35, 9 (2022), 9154–9167.

[79] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep learning based recommender system: A survey and new perspectives. *ACM computing surveys (CSUR)* 52, 1 (2019), 1–38.

[80] Zijian Zhang, Shuchang Liu, Jiaao Yu, Qingpeng Cai, Xiangyu Zhao, Chunxu Zhang, Ziru Liu, Qidong Liu, Hongwei Zhao, Lantao Hu, et al. 2024. M3oe: Multi-domain multi-task mixture-of experts recommendation framework. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval.* 893–902.

[81] Feng Zhao, Jie Liu, Juan Liu, Leonidas Guibas, and James Reich. 2003. Collaborative signal and information processing: an information-directed approach. *Proc. IEEE* 91, 8 (2003), 1199–1209.

[82] Jujia Zhao, Wenjie Wang, Chen Xu, Zhaochun Ren, See-Kiong Ng, and Tat-Seng Chua. 2024. Llm-based federated recommendation. *arXiv preprint arXiv:2402.09959* (2024).

[83] Kesen Zhao, Shuchang Liu, Qingpeng Cai, Xiangyu Zhao, Ziru Liu, Dong Zheng, Peng Jiang, and Kun Gai. 2023. KuaiSim: A comprehensive simulator for recommender systems. *Advances in Neural Information Processing Systems* 36 (2023), 44880–44897.

[84] Xiangyu Zhao, Long Xia, Liang Zhang, Zhuoye Ding, Dawei Yin, and Jiliang Tang. 2018. Deep reinforcement learning for page-wise recommendations. In *Proceedings of the 12th ACM conference on recommender systems.* 95–103.

[85] Xiangyu Zhao, Liang Zhang, Zhuoye Ding, Long Xia, Jiliang Tang, and Dawei Yin. 2018. Recommendations with negative feedback via pairwise deep reinforcement learning. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining.* 1040–1048.

[86] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is all you need for sequential recommendation. In *Proceedings of the ACM web conference 2022.* 2388–2399.

[87] Xiangmin Zhou, Dong Qin, Xiaolu Lu, Lei Chen, and Yanchun Zhang. 2019. Online social media recommendation over streams. In *2019 IEEE 35th International Conference on Data Engineering (ICDE).* IEEE, 938–949.