De-collapsing User Intent: Adaptive Diffusion Augmentation with Mixture-of-Experts for Sequential Recommendation

Xiaoxi Cui[1*], Chao Zhao[1*], Yurong Cheng[1†], Xiangmin Zhou[2]

[1]Beijing Institute of Technology, Beijing, China
[2]RMIT University, Melbourne, Australia
{3120215512, zhaochao, yrcheng}@bit.edu.cn, xiangmin.zhou@rmit.edu.au

Abstract

Sequential recommendation (SR) aims to predict users' next action based on their historical behavior, and is widely adopted by a number of platforms. The performance of SR models relies on rich interaction data. However, in real-world scenarios, many users only have a few historical interactions, leading to the problem of data sparsity. Data sparsity not only leads to model overfitting on sparse sequences, but also hinders the model's ability to capture the underlying hierarchy of user intents. This results in misinterpreting the user's true intents and recommending irrelevant items. Existing data augmentation methods attempt to mitigate overfitting by generating relevant and varied data. However, they overlook the problem of reconstructing the user's intent hierarchy, which is lost in sparse data. Consequently, the augmented data often fails to align with the user's true intents, potentially leading to misguided recommendations. To address this, we propose the Adaptive Diffusion Augmentation for Recommendation (ADARec) framework. Critically, instead of using a diffusion model as a black-box generator, we use its entire step-wise denoising trajectory to reconstruct a user's intent hierarchy from a single sparse sequence. To ensure both efficiency and effectiveness, our framework adaptively determines the required augmentation depth for each sequence and employs a specialized mixture-of-experts architecture to decouple coarse- and fine-grained intents. Experiments show ADARec outperforms state-of-the-art methods on standard benchmarks and on sparse sequences, demonstrating its ability to reconstruct hierarchical intent representations from sparse data.

Code — https://github.com/ZhaoChao52/ADARec

Introduction

Sequential recommendation (SR) aims to predict users' next action based on their historical behavior, which is widely adopted by many platforms in e-commerce, media streaming, and other online services. However, the quality of sequential recommendation relies on rich, high-quality interaction data. In reality, user activity in recommendation systems typically follows a long-tail distribution (Park and Tuzhilin 2008). This means many users have only a few historical interactions, leading to the data sparsity problem, which severely undermines the model's learning process, leading to the model overfitting on sparse interactions within user historical sequences (Wu et al. 2023; Wang et al. 2024).

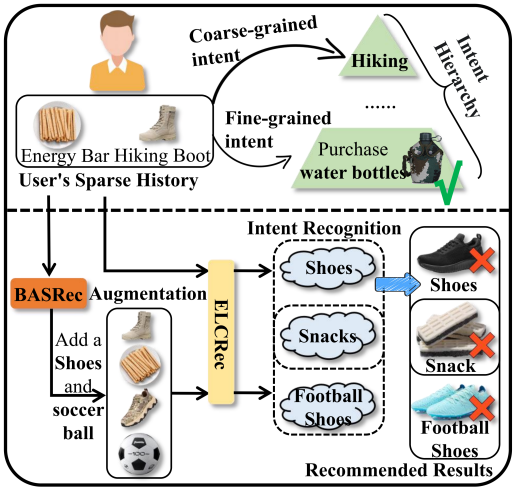


Figure 1: An illustration of how data sparsity hinders hierarchical intent modeling.

Data sparsity poses a challenge to modeling users' hierarchical intentions. As shown in Fig 1, the top part shows a user's short history of interactions. For example, the user only interacted with an energy bar and a hiking boot. These few interactions might imply that the user has a coarse-grained intent: hiking, and a more fine-grained intent: to buy a water bottle for that trip. However, as further illustrated in the lower section of Fig 1, existing intent-based recommendation methods, such as ELCRec (Liu et al. 2024), tend to overfit on these sparse items. They might simply categorize an energy bar as snacks and hiking boots as shoes. Consequently, the model could erroneously conclude the user's interests are limited to these broad categories, leading to irrelevant recommendations like other shoes or snacks.

Existing data augmentation methods (Dang et al. 2025; Ma et al. 2024; Luo, Li, and Lin 2025) attempt to mitigate

*These authors contributed equally.
†Corresponding author.

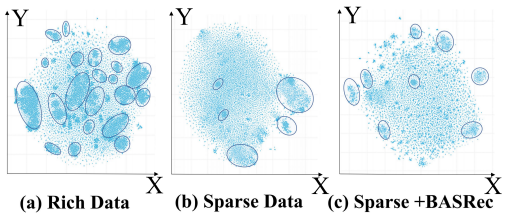


Figure 2: t-SNE visualization of user embeddings on sparse data. (a) With rich data, users form clear semantic clusters. (b) On sparse data (Sequence length < 5), clustering failure. (c) Even with a strong augmentation baseline like DiffDiv, failing to construct the hierarchy.

overfitting by generating both relevant and varied data. As shown in Fig 1, to maintain relevance, they might generate similar items, such as a different brand of shoes. To increase variety, they might generate items with low relevance, such as a soccer ball. Therefore, the model might wrongly infer that the user's intention is to purchase football shoes based on the shoes and the football.

We further validate the limited performance of existing intent-based models and data augmentation methods on sparse data through visualization experiments. As shown in Fig 2 (a) and (b), the user embedding clustering effect of the intent-based recommendation model on sparse datasets is worse than that on full datasets. This indicates that on sparse datasets, the model cannot effectively form clear semantic clusters, which makes it difficult for the model to effectively distinguish the intentions of different users. In addition, the existing data augmentation methods mainly rely on adding similar data and diverse data for augmentation. Specifically, addition of similar data will make the newly added data too close to the original data, resulting in limited discrimination of semantic clustering (as shown in Fig 2 (c)). Meanwhile, without the guidance of a user's high-level intent, the newly introduced data might deviate from the user's true intent, potentially leading to misguided recommendations.

Based on the above analysis, we argue that the key to data augmentation does not lie in blindly trading off similarity and diversity, but in constructing the hierarchy of users' intentions. Specifically, the deeper the level of a user's intent (e.g., from 'buying water bottles' to 'preparing for a hiking trip'), the wider the range of item categories covered by that intent, and the greater the diversity of the corresponding items. To realize this idea, we draw inspiration from the Information Bottleneck (IB) principle (Tishby, Pereira, and Bialek 2000). The core idea of the IB principle is that adding noise creates a bottleneck for information. To get the essential message through this bottleneck, the model must discard the less important, specific details and keep only the most fundamental and general pattern.

Based on this, we propose the Adaptive Diffusion Augmentation for Recommendation (ADARec) framework. Distinct from existing methods that only generate single data points (Dang et al. 2025; Ma et al. 2024; Luo, Li, and Lin 2025), ADARec leverages a diffusion model (Sohl-Dickstein et al. 2015; Ho, Jain, and Abbeel 2020) forward (noising) process to gradually remove fine-grained details from a user's sequence. Correspondingly, each step of the reverse (denoising) process compels the model to infer the user's intent at a progressively higher level of abstraction. The full denoising trajectory yields a rich intent hierarchy, spanning from fine-grained to coarse-grained intents. Capturing this hierarchy enables the reconstruction of a complete intent hierarchy, even from a single sparse sequence.

However, due to the high computational cost of diffusion models, applying this enhancement to every user sequence is not practical. To address this, our ADARec framework introduces three key innovations. First, an Adaptive Depth Controller (ADC) intelligently determines the optimal diffusion depth for each user sequence. Second, we introduce the Hierarchical Diffusion Augmentation (HDA) module, which leverages a diffusion process to generate a rich, hierarchical intent hierarchy for each user sequence. Finally, the intent hierarchy contains both coarse-grained, fine-grained intent representations. A single processing module would struggle to optimally handle both. Therefore, we design a specialized Hierarchical Parsing Mixture-of-Experts (HP-MoE) module to process representations at different levels of intent hierarchy. Ultimately, ADARec can effectively and efficiently capture a user's hierarchical intent representation, even from sparse sequences. Our main contributions are:

1. We propose the Adaptive Diffusion Augmentation for Recommendation (ADARec) framework to construct the intent hierarchy from sparse user sequences.
2. We introduce the Hierarchical Diffusion Augmentation (HDA) module, which leverages a diffusion process to generate a rich hierarchy of user intents, ranging from fine-grained to coarse-grained.
3. We propose two specialized modules, an Adaptive Depth Controller (ADC) and a Hierarchical Parsing Mixture-of-Experts (HP-MoE) module, to collaboratively enhance the framework's efficiency and effectiveness.
4. Extensive experiments on real-world datasets demonstrate that ADARec outperforms state-of-the-art baselines, validating the effectiveness of ADARec.

Related Work

SR has rapidly advanced from early methods like Markov Chains (He and McAuley 2016; Rendle, Freudenthaler, and Schmidt-Thieme 2010) to a variety of deep architectures, including those based on RNNs (Hidasi et al. 2015), Transformers (Kang and McAuley 2018; Sun et al. 2019; Zhou et al. 2024), GNNs (Chang et al. 2021; Li et al. 2023b), and MLPs (Li et al. 2022; Zhou et al. 2022). The self-supervised learning has also been widely adopted (Xie et al. 2022; Liu et al. 2021; Qiu et al. 2022; Yang et al. 2023). The intent learning in recommendation aims to capture multiple user intents by employing techniques such as multi-interest modeling (Li et al. 2019; Cen et al. 2020), disentanglement (Ma et al. 2020; Li et al. 2024), and more recently, end-to-end clustering (Chen et al. 2022; Liu et al. 2024).

To mitigate data sparsity, augmentation strategies have evolved from simple heuristics like masking and cropping

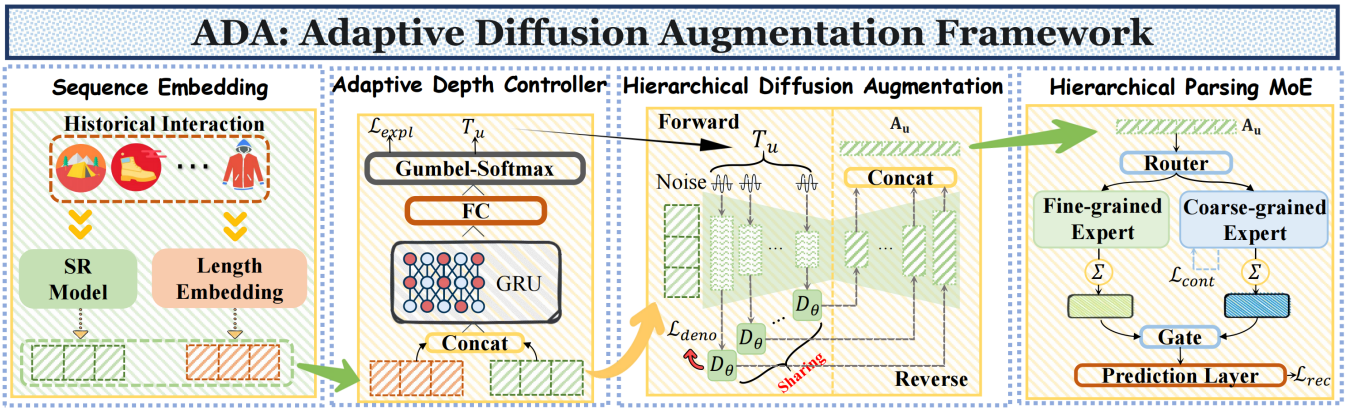


Figure 3: The overall architecture of the proposed ADARec framework. For a user sequence, (1) the ADC module determines the required augmentation depth T_u. (2) the HDA module process generates an intent hierarchy A_u. (3) The HP-MoE module parses the hierarchy, and its expert outputs are fused to produce the final user representation for recommendation.

(Xie et al. 2022; Liu et al. 2021) to more sophisticated approaches. These include leveraging diffusion models to generate generalized preferences (Ma et al. 2024; Luo, Li, and Lin 2025) or to denoise feature representations from sparse interactions (Cui et al. 2025a,b), and employing representation-level mixup to balance relevance and diversity (Dang et al. 2025).

However, their core strategy is to enrich the training set by generating a greater diversity of items. This designed diversity often lacks guidance from the user's high-level intent, causing the augmented data to potentially deviate from the user's true intents. This approach, by focusing on generating more data rather than modeling the structure of intent, ultimately fails to tackle the fundamental issue of the incomplete intent hierarchy representation which is a direct consequence of data sparsity.

Adaptive Diffusion Augmentation for Recommendation (ADARec) Framework

To construct a hierarchical user intent from sparse behavioral data, we propose a novel framework: Adaptive Diffusion Augmentation for Recommendation (ADARec). The framework operates through a sequential pipeline: (1) an Adaptive Depth Controller (ADC) first adaptively determines the optimal diffusion depth. (2) a Hierarchical Diffusion Augmentation (HDA) process then generates an intent hierarchy of the diffusion depth; and (3) finally, a Hierarchical Parsing Mixture-of-Experts (HP-MoE) network parses this hierarchy by dynamically routing representations to specialized experts. The overall architecture is shown in Fig 3.

Sequence Embedding

We first embed the user's historical sequence S_u using ELCRec (Liu et al. 2024) as our backbone encoder to obtain the sequence embedding E_u. We also encode the sequence length via an MLP. Both embeddings serve as input for our adaptive augmentation framework.

Adaptive Depth Controller (ADC)

To provide customized enhancement tailored to each user's needs, we introduce the Adaptive Depth Controller (ADC). The purpose is to determine the optimal enhancement depth $T_u$for the user sequence S_u.

The ADC takes the user's original sequence embedding $\mathbf{E}_u^0 \in \mathbb{R}^{L\times D}$ and an embedding vector of the normalized sequence length $\mathbf{v}_{l_u}$ as input. Due to GRU's proven effectiveness in capturing temporal dependencies within sequential data, these are concatenated and fed into a GRU network to generate a vector of logits over possible depths, ranging from 0 to a predefined maximum depth T_{max}:

$$\mathbf{h}_c = \text{Pool}(\text{GRU}([\mathbf{E}_u^0; \mathbf{v}_{l_u}])) \quad (1)$$

$$\mathbf{logits}_u = \mathbf{W}_\sigma \mathbf{h}_c + b_\sigma \quad (2)$$

where $\mathbf{W}_\sigma$ and b_σ are learnable parameters, and Pool is an aggregation function like mean pooling. The dimension of $\mathbf{logits}_u$ is $T_{max}+1$.

To select a discrete depth T_u in a differentiable manner, we use the Gumbel-Softmax estimator over the $T_{max}+1$ discrete choices.

$$\mathbf{p}_u = \text{Softmax}((\mathbf{logits}_u + \mathbf{g})/\tau_{gs}) \quad (3)$$

where $\mathbf{g}$ are i.i.d. samples from a Gumbel(0,1) distribution, and τ_{gs} is the softmax temperature. During the forward pass, we take the 'argmax' to get the one-hot vector representing the chosen depth T_u, while the Gumbel-Softmax probabilities $\mathbf{p}_u$ are used for the backward pass.

To prevent the ADC from collapsing to only selecting safe, shallow depths during training, especially in the early stages, we introduce an auxiliary exploration loss. This loss encourages the ADC to maintain a level of uncertainty in its choices, thereby promoting exploration across the full range of possible depths. We use the negative entropy of the ADC's probability distribution as the loss: $\mathcal{L}_{expl} = -\sum_{i=0}^{T_{max}} p_{u,i} \log p_{u,i}$. This ensures that the downstream HDA and HP-MoE modules receive inputs from diverse depths, facilitating more robust training.

Hierarchical Diffusion Augmentation (HDA)

To generate a user intent hierarchy, we propose the Hierarchical Diffusion Augmentation (HDA) process. Unlike existing diffusion-based augmentation methods (e.g., DiffDiv, GlobalDiff) that utilize only the final diffusion output for data augmentation, the core idea of the HDA module is to construct a hierarchy of user intents by leveraging the natural evolution of information granularity, from high to low, within the denoising trajectory of a diffusion model.

The Intuition behind Constructing Hierarchical Intent via Denoising Trajectory. During the diffusion model's denoising process ($E_k = \sqrt{\alpha_k}E_u + \sqrt{1-\alpha_k}\epsilon_k$), at high noise levels (k approaching T_u), the denoising network $\epsilon_\theta(E_k, k)$ is compelled to extract abstract information from the highly incomplete signal, forming a coarse-grained intent. The original information is significantly compressed ($I(\hat{E}_k; E_u) \ll H(E_u)$, where $H(E_u)$ denotes the information uncertainty of the original user sequence embedding E_u), yet predictive relevance is maintained (maximizing $I(\hat{E}_k; Y_{\text{next}})$, achieved by the $\mathcal{L}_{rec}$ in Eq. (13)).Here, $I(X;Y)$ represents the mutual information, quantifying how much knowing variable X reduces the uncertainty about variable Y. Therefore, $I(\hat{E}_k; E_u)$ quantifies how much information the denoised representation $\hat{E}_k$ retains from the original user sequence E_u. At low noise levels (k approaching 0), the network recovers more specific behavioral details, forming a fine-grained intent (I_{fine}), where information is almost fully preserved ($I(\hat{E}_k; E_u) \approx H(E_u)$). The entire denoising trajectory $\{\hat{E}_0, \hat{E}_1, \ldots, \hat{E}_{T_u}\}$ naturally forms a user intent hierarchy.

The process of HDA. Given a user u's sequence embedding $\mathbf{E}_u^0$ and ADC-predicted depth T_u, the diffusion's forward process progressively adds Gaussian noise to generate noisy representations $\{\mathbf{E}_u^k\}_{k=0}^{T_u}$ for the denoising network.

$$\mathbf{E}_u^k = \sqrt{\bar{\alpha}_k}\mathbf{E}_u^0 + \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}_k, \quad \boldsymbol{\epsilon}_k \sim \mathcal{N}(0, \mathbf{I}) \tag{4}$$

Here, $\bar{\alpha}_k$ is a predefined noise schedule parameter. We adopt the linear variance schedule from DDPM (Ho, Jain, and Abbeel 2020). The core of HDA lies in the reverse process executed by our trainable denoising network D_θ.

$$\hat{\mathbf{E}}_u^k = \frac{\mathbf{E}_u^k - \sqrt{1-\bar{\alpha}_k}\boldsymbol{\epsilon}_\theta(\mathbf{E}_u^k, k)}{\sqrt{\bar{\alpha}_k}} \tag{5}$$

To drive the recovery of information from coarse to fine granularity, the denoising network D_θ is optimized using a standard mean squared error (MSE) loss between the predicted and true noise: $\mathcal{L}_{deno} = ||\boldsymbol{\epsilon}_\theta(\mathbf{E}_u^k, k) - \boldsymbol{\epsilon}_k||^2$. For simplicity and efficiency, the denoising network D_θ is implemented as a simple Multi-Layer Perceptron (MLP).

This iterative process, from $\mathbf{E}_u^{T_u}$ down to $\mathbf{E}_u^0$, yields a high-quality intent hierarchy $A_u = \{\hat{\mathbf{E}}_u^{T_u}, \hat{\mathbf{E}}_u^{T_u-1}, \ldots, \hat{\mathbf{E}}_u^0\}$. In this hierarchy, high-k elements represent coarse-grained intents, while low-k elements capture fine-grained intents. The final output of our HDA module is not a single vector, but this entire structured set A_u.

Hierarchical Parsing MoE (HP-MoE)

To efficiently parse the generated intent hierarchy A_u, we design the Hierarchical Parsing Mixture-of-Experts (HP-MoE) module. This module is to intelligently decouple and process the fine-grained and coarse-grained intents.

Functionally Decoupled Experts. To functionally decouple the processing of different intent hierarchy, we employ two specialized experts: a fine-grained expert (E_{fin}) and a coarse-grained expert (E_{coar}). The fine-grained expert, E_{fin}, utilizes a Transformer (Vaswani et al. 2017) driven solely by the recommendation loss $\mathcal{L}_{rec}$ to capture fine-grained intents. The standard recommendation loss, $\mathcal{L}_{rec}$, which directly optimizes for next-item prediction, provides a sufficient and precise supervisory signal for this purpose. It naturally encourages E_{fin} to focus on the subtle, transient details within the user's sequence, thus fulfilling its role without the need for additional regularization. Conversely, the coarse-grained expert, E_{coar}, is designed to extract the user's coarse-grained intent. We apply an exclusive contrastive loss, $\mathcal{L}_{cont}$, to E_{coar}. This InfoNCE-based loss compels the expert to learn the user's invariant essence by pulling together representations from their coarse-grained hierarchy while pushing them apart from other users' intents:

$$\mathcal{L}_{cont} = \sum_{k=\lfloor T_u/2 \rfloor}^{T_u-1} -\log \frac{\exp(\text{sim}(\mathbf{z}_{u,k}, \mathbf{z}_{u,k+1})/\tau)}{\sum_{\mathbf{z}_j \in \mathcal{B}_k} \exp(\text{sim}(\mathbf{z}_{u,k}, \mathbf{z}_j)/\tau)} \tag{6}$$

where $\mathbf{z}_{u,k+1}$ is the positive sample, and $\mathcal{B}_k$ contains $\mathbf{z}_{u,k+1}$ and a set of negative samples consisting of coarse-grained representations from other users in the same batch.

Content-Aware Routing and Gated Fusion To synthesize a unified representation from the intent hierarchy, we design a two-level gated fusion mechanism.

First, at each hierarchy level k, we employ a content-aware routing to dynamically weigh the expert outputs. This gate's probability, g_k, considers both the noise level embedding $\mathbf{emb}(k)$ and the content representation $\text{Pool}(\hat{\mathbf{E}}_u^k)$:

$$g_k = \text{Sigmoid}(\text{MLP}([\text{Pool}(\hat{\mathbf{E}}_u^k); \mathbf{emb}(k)])) \tag{7}$$

Next, we aggregate the outputs from each expert across the entire hierarchy, weighted by their respective gate probabilities, to form a consolidated fine-grained representation $\mathbf{z}_{fin}$ and coarse-grained representation $\mathbf{z}_{coar}$:

$$\mathbf{z}_{fin} = \frac{\sum_{k=0}^{T_u} g_k \cdot E_{fin}(\hat{\mathbf{E}}_u^k)}{\sum_{k=0}^{T_u} g_k + \epsilon} \tag{8}$$

$$\mathbf{z}_{coar} = \frac{\sum_{k=0}^{T_u} (1-g_k) \cdot E_{coar}(\hat{\mathbf{E}}_u^k)}{\sum_{k=0}^{T_u} (1-g_k) + \epsilon} \tag{9}$$

Finally, to balance stable long-term themes against transient specifics, a final gating vector $\mathbf{gate}_u$ is computed from the aggregated coarse representation $\mathbf{z}_{coar}$. This gate adaptively combines the two aggregated representations to produce the final user representation $\mathbf{h}_u$:

$$\mathbf{gate}_u = \text{Sigmoid}(\mathbf{W}_g \mathbf{z}_{coar} + \mathbf{b}_g) \tag{10}$$

$$\mathbf{h}_u = \mathbf{gate}_u \odot \mathbf{z}_{fin} + (1 - \mathbf{gate}_u) \odot \mathbf{z}_{coar} \tag{11}$$

where $\odot$ denotes element-wise multiplication.

Prediction Layer. Finally, the fused user representation $\mathbf{h}_u$ is fed into a prediction layer to generate the final recommendation scores. It takes $\mathbf{h}_u$ as input and computes the probability scores $\mathbf{y}_u$ for all candidate items.

Training Objective

The ADARec framework is trained end-to-end with a composite objective function that synergistically combines the primary recommendation task with several auxiliary losses. The total loss is defined as:

$$\mathcal{L}_{tot} = \mathcal{L}_{rec} + \lambda_{deno}\mathcal{L}_{deno} + \lambda_{cont}\mathcal{L}_{cont} + \lambda_{expl}\mathcal{L}_{expl} \quad (12)$$

Here, The primary recommendation loss, $\mathcal{L}_{rec}$, is computed using the standard cross-entropy loss. For a given user u, the loss is calculated over all items in the item set $\mathcal{I}$, based on the predicted probability scores $\mathbf{y}_u$:

$$\mathcal{L}_{rec} = -\sum_{i \in \mathcal{I}} p(i) \log(\mathbf{y}_{u,i}) \quad (13)$$

. To structure the hierarchical learning, the denoising loss $\mathcal{L}_{deno}$ trains the generator for the intent hierarchy, while the contrastive loss $\mathcal{L}_{cont}$ acts as a key regularizer to enforce functional decoupling between the experts. Second, to ensure stability, the exploration loss $\mathcal{L}_{expl}$ uses an entropy-based regularizer to prevent the ADC from prematurely converging to shallow depths. The λ terms are hyperparameters that balance the contribution of each component.

Experiments

We conduct extensive experiments to evaluate our proposed ADARec framework, aiming to answer the following research questions:

- **RQ1:** Does ADARec outperform state-of-the-art SR models and can it effectively enhance the performance of various intent-based backbone models?
- **RQ2:** How effective are the key components of ADARec: the ADC, HDA, and HP-MoE?
- **RQ3:** How computationally efficient is ADARec?
- **RQ4:** How sensitive is ADARec's performance to its main hyperparameters?
- **RQ5:** Can ADARec effectively construct hierarchical user intents from sparse data?

Experimental Settings

Datasets We conduct our experiments on four public benchmark datasets: Beauty, Sports, Toys, and Yelp. To ensure fairness and reproducibility, we strictly adhere to the datasets and pipelines released by ELCRec (Liu et al. 2024) and utilizing ELCRec's publicly available datasets[1]. All datasets match those in ELCRec (Liu et al. 2024).

[1]https://github.com/yueliu1999/Awesome-Deep-Group-Recommendation

Baselines. We benchmark ADARec against eleven diverse baselines. Performance data for the initial nine models are adopted from ELCRec (Liu et al. 2024) for consistent comparison. These baselines cover the evolution of sequential recommendation, including early deep learning models (e.g., Caser (Tang and Wang 2018)), state-of-the-art Transformer architectures (e.g., SASRec (Kang and McAuley 2018), BERT4Rec (Sun et al. 2019)), recent self-supervised methods (e.g., intent-learning methods (IOCRec (Li et al. 2023a), ICLRec (Chen et al. 2022), ELCRec (Liu et al. 2024)), and powerful data augmentation frameworks, notably diffusion-based models like BASRec (Dang et al. 2025), PDRec (Ma et al. 2024), GlobalDiff (Luo, Li, and Lin 2025), and DiffDiv (Cai et al. 2025).

Implementation Details. We use the Adam optimizer with a weight decay of 1×10^{-4}. The maximum user sequence length is capped at 50, the Gumbel-Softmax temperature (τ_{gs}) is fixed at 0.5, and the InfoNCE temperature (τ) is set to 0.07. We tuned key model-specific hyperparameters via grid search. The maximum diffusion depth T_{max} is searched over $\{4, 6, 8, 10, 12\}$. Loss weights λ_{deno} and λ_{cont} are selected from $\{0.01, 0.03, 0.05, 0.1, 0.15\}$, while λ_{expl} ia searched in $\{0.001, 0.003, 0.01, 0.03, 0.1\}$. For fair comparison, BASRec, PDRec, GlobalDiff, and DiffDiv are implemented uniformly within our framework's pipeline. Our method is implemented on a server equipped with 20GB NVIDIA RTX 3090 GPUs.

Overall Performance (RQ1)

The main experimental results, presented in Table 1, demonstrate ADARec's consistent and significant outperformance across a comprehensive suite of baselines. This advantage is even more apparent when we focus on the most challenging cases: extremely short sequences (length ≤ 5), with results detailed in Table 2. On sparse datasets like Beauty and Sports, its relative improvement over the strongest competitor reaches over 15%. This demonstrates that ADARec's ability to generate a coarse-grained intent representation from only a few items prevents it from being misled by noise, enabling it to make far more accurate and generalized recommendations. We observed relatively modest performance improvements on the Yelp dataset. This is a common phenomenon in intent-based recommendation (Liu et al. 2024), largely because Yelp is a comprehensive dataset, making the clear disentanglement of user intents into distinct units more challenging. As shown in Table 3, we integrate our augmentation framework with several advanced intent-aware models. The results show that all baseline models achieve consistent performance improvements. This demonstrates that our method can effectively generate hierarchical user representations, which in turn provide richer semantic information for various intent-based models.

Ablation Study of Key Components (RQ2)

To show the contribution of each component, we conducted a thorough ablation study (Table 4), which confirms that all parts of ADARec are indispensable. (1) Removing the

Dataset	Metric	Caser (2018)	SASRec (2018)	BERT4Rec (2019)	IOCRec (2023)	ICLRec (2022)	ELCRec (2024)	PDRec (2024)	BASRec (2025)	GDiff (2025)	DDiv (2025)	**ADARec** (Ours)	Improv.
Beauty	H@5	0.0251	0.0374	0.0360	0.0408	0.0495	0.0529	0.0569	0.0551	0.0563	0.0575	**0.0600***	**+4.35%**
	N@5	0.0145	0.0241	0.0216	0.0245	0.0326	0.0355	0.0380	0.0385	0.0382	0.0375	**0.0403***	**+4.68%**
	H@20	0.0643	0.0901	0.0984	0.0916	0.1072	0.1079	0.1145	0.1135	0.1129	0.1152	**0.1187***	**+3.04%**
	N@20	0.0298	0.0387	0.0391	0.0444	0.0491	0.0509	0.0541	0.0539	0.0531	0.0548	**0.0568***	**+3.65%**
Sports	H@5	0.0154	0.0206	0.0217	0.0246	0.0263	0.0286	0.0325	0.0328	0.0321	0.0315	**0.0341***	**+3.96%**
	N@5	0.0114	0.0135	0.0143	0.0162	0.0173	0.0185	0.0218	0.0220	0.0215	0.0211	**0.0228***	**+3.64%**
	H@20	0.0399	0.0497	0.0604	0.0641	0.0630	0.0648	0.0701	0.0728	0.0719	0.0714	**0.0745***	**+2.33%**
	N@20	0.0178	0.0216	0.0251	0.0280	0.0276	0.0286	0.0328	0.0331	0.0325	0.0318	**0.0341***	**+3.02%**
Toys	H@5	0.0166	0.0463	0.0274	0.0311	0.0586	0.0585	0.0635	0.0648	0.0642	0.0645	**0.0691***	**+6.64%**
	N@5	0.0107	0.0306	0.0174	0.0197	0.0397	0.0403	0.0433	0.0431	0.0435	0.0428	**0.0474***	**+9.08%**
	H@20	0.0420	0.0941	0.0688	0.0781	0.1130	0.1138	0.1230	0.1245	0.1219	0.1225	**0.1298***	**+4.26%**
	N@20	0.0179	0.0441	0.0291	0.0330	0.0550	0.0560	0.0615	0.0618	0.0613	0.0605	**0.0646***	**+4.53%**
Yelp	H@5	0.0142	0.0160	0.0196	0.0222	0.0233	0.0248	0.0253	0.0255	0.0251	0.0258	**0.0264***	**+2.33%**
	N@5	0.0080	0.0101	0.0121	0.0137	0.0146	0.0153	0.0159	0.0153	0.0161	0.0155	**0.0167***	**+3.73%**
	H@20	0.0406	0.0443	0.0564	0.0640	0.0645	0.0667	0.0688	0.0681	0.0685	0.0691	**0.0712***	**+3.04%**
	N@20	0.0156	0.0179	0.0223	0.0263	0.0261	0.0275	0.0280	0.0275	0.0278	0.0283	**0.0291***	**+2.83%**

Table 1: Main experimental results on four datasets. The best results are in **bold**, and the second-best results are underlined.

Dataset	Metric	BASRec (2025)	GlobalDiff (2025)	DiffDiv (2025)	**ADARec** (Ours)	Improv.
Beauty	H@5	0.0441	0.0428	0.0455	**0.0531***	**+16.70%**
	N@5	0.0285	0.0272	0.0295	**0.0353***	**+19.66%**
	H@20	0.0910	0.0897	0.0925	**0.1066***	**+15.24%**
	N@20	0.0421	0.0409	0.0427	**0.0504***	**+17.48%**
Sports	H@5	0.0261	0.0256	0.0265	**0.0305***	**+15.09%**
	N@5	0.0180	0.0174	0.0179	**0.0199***	**+10.56%**
	H@20	0.0592	0.0585	0.0590	**0.0659***	**+11.32%**
	N@20	0.0270	0.0266	0.0269	**0.0298***	**+10.37%**
Toys	H@5	0.0557	0.0550	0.0560	**0.0618***	**+10.36%**
	N@5	0.0389	0.0387	0.0392	**0.0431***	**+9.95%**
	H@20	0.1051	0.1043	0.1050	**0.1169***	**+11.23%**
	N@20	0.0530	0.0518	0.0522	**0.0586***	**+10.57%**
Yelp	H@5	0.0194	0.0189	0.0190	**0.0200***	**+3.09%**
	N@5	0.0121	0.0112	0.0119	**0.0124***	**+2.48%**
	H@20	0.0528	0.0506	0.0522	**0.0559***	**+5.87%**
	N@20	0.0217	0.0204	0.0209	**0.0225***	**+3.69%**

Table 2: Performance comparison on extremely sparse sequences (user history length ≤ 5).

HDA module causes the most severe performance degradation, highlighting its foundational role in our framework. (2) Removing the ADC or the HP-MoE also leads to significant performance drops. This validates the necessity of both dynamically allocating augmentation resources (ADC) and using a specialized architecture for hierarchical parsing (HP-MoE). Notably, the variant with only ADC and HDA still outperforms strong baselines like ELCRec, demonstrating the strength of our augmentation strategy. (3) Furthermore, the variant using only HP-MoE performs comparably to ELCRec. This critically shows that our parser's effectiveness is contingent on the structured input provided by the HDA. (4) Finally, removing the Content-Aware Router or the contrastive loss ($\mathcal{L}_{cont}$) results in smaller but consistent performance drops, confirming their vital roles in nuanced routing and expert decoupling, respectively.

	Beauty		Toys	
Model	HR@20	NDCG@20	HR@20	NDCG@20
IOCRec (2023)	0.0916	0.0444	0.0781	0.0330
ADARec w/ IOCRec	0.1053	0.0501	0.1098	0.0515
ICLRec (2022)	0.1072	0.0491	0.1130	0.0550
ADARec w/ ICLRec	0.1165	0.0558	0.1271	0.0628
ELCRec (2024)	0.1079	0.0509	0.1138	0.0560
ADARec w/ ELCRec	0.1187	0.0568	0.1298	0.0646

Table 3: Performance of combining ADARec's augmentation with different intent-aware prediction layers.

Efficiency Analysis (RQ3)

We analyze the efficiency of ADARec in Table 5. While the training time is higher than the non-diffusion baselines (SASRec, ELCRec), this is a justifiable one-time offline cost that is overwhelmingly compensated by the significant improvements in recommendation quality. Notably, our model is more efficient than other diffusion-based methods like DiffDiv and GlobalDiff, due to our lightweight architecture.

Hyperparameter Sensitivity (RQ4)

Analysis of λ_{expl} and λ_{deno}: As shown in Figure 4, the model generally performs best when $\lambda_{expl} = 0.01$, and $\lambda_{deno} = 0.1$. We also observe that the performance curves for these auxiliary weights are relatively flat around their optimal points. This indicates that while optimal values exist, the model is robust to minor deviations, confirming its stability and ease of application.

ADC	HDA	HP-MoE	C-A Router	$\mathcal{L}_{cont}$	Beauty		Sports		Toys		Yelp	
					HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20	HR@20	NDCG@20
✓	✓	✓	✓	✓	**0.1187**	**0.0568**	**0.0745**	**0.0341**	**0.1298**	**0.0646**	**0.0712**	**0.0291**
✗	✓	✓	✓	✓	0.1141	0.0545	0.0715	0.0328	0.1246	0.0621	0.0690	0.0281
✓	✗	✓	✓	✓	0.1105	0.0528	0.0681	0.0310	0.1189	0.0592	0.0672	0.0276
✗	✗	✓	✓	✓	0.1075	0.0505	0.0645	0.0283	0.1132	0.0557	0.0661	0.0273
✓	✓	✗	✗	✓	0.1128	0.0539	0.0702	0.0321	0.1225	0.0610	0.0683	0.0279
✓	✓	✓	✗	✓	0.1172	0.0561	0.0736	0.0337	0.1281	0.0638	0.0705	0.0288
✓	✓	✓	✓	✗	0.1158	0.0553	0.0725	0.0332	0.1260	0.0627	0.0697	0.0284

Table 4: Ablation study results. We report HR@20 and NDCG@20. **C-A Router: Content-Aware Router**.

Units: Train (s), Infer (ms)

	Beauty		Sports		Toys		Yelp	
Model	Train	Infer	Train	Infer	Train	Infer	Train	Infer
SASRec	2.15	3.1	6.51	9.5	4.32	6.8	5.25	7.9
ELCRec	3.81	5.9	11.28	16.1	7.88	11.5	9.00	13.2
DiffDiv	7.55	14.8	20.10	31.5	15.60	22.4	18.23	26.1
GlobalDiff	8.90	21.2	23.50	45.8	18.95	33.6	21.75	38.5
Ours	6.21	10.5	17.95	24.9	13.52	18.7	15.88	19.3

Table 5: Efficiency comparison. We report training time (s/epoch) and inference latency (ms/user).

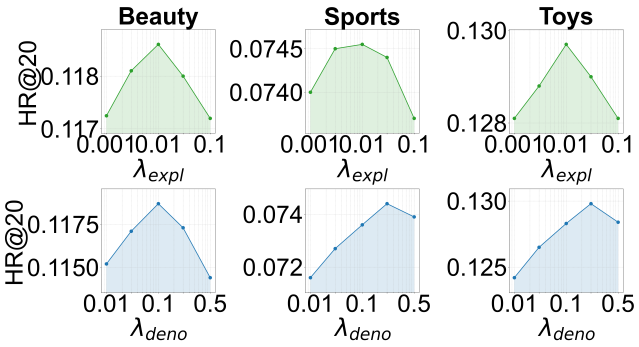


Figure 4: Stability analysis for auxiliary loss weights.

Analysis of Number of Experts in HP-MoE As Fig. 5 illustrates, expanding HP-MoE to 3 or 4 experts brings only marginal HR@20 gains but significantly increases training time. This result strongly supports ADARec's design choice of employing only two experts. This is primarily because the Adaptive Depth Controller (ADC) typically selects a relatively small actual diffusion depth T_u for sparse sequences, thereby limiting the number of potential intent hierarchies. Additional experts cannot uncover more independent intent levels from the the limited input information, resulting in redundancy and increased computational overhead without tangible benefits. The two-expert configuration optimally balances efficiency and effectiveness.

Visualization of Intent Representations (RO5)

We visualize the learned user intent representations on the sparse (sequence length ≤ 5) Yelp dataset using t-SNE (Figure 6). As shown in Figure 6(a), the strong baseline DiffDiv fails to form distinct clusters, with most intents collapsing into an indistinguishable central mass. In contrast, ADARec (Figure 6(b)) learns the clear, well-separated intent clusters. This shows that ADARec successfully identifies and reconstructs a user intent hierarchy even from sparse data.

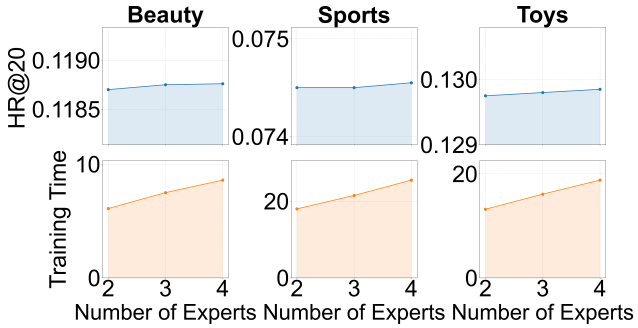
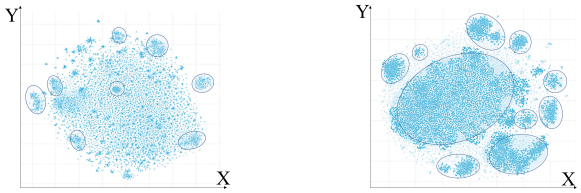


Figure 5: Analysis of Performance (HR@20) and Training Time Cost vs. Number of Experts.



(a) Yelp(DiffDiv) (b) Yelp(ADARec)

Figure 6: t-SNE visualization of the learned user intent representations on the Yelp dataset. (a) A strong baseline like DiffDiv struggles to form distinct clusters. (b) Our method, ADARec, successfully learns well-separated intent clusters.

Conclusion

To tackle data sparsity in hierarchical user intent learning, we propose ADARec. It leverages adaptive hierarchical diffusion for intent generation and a specialized MoE for parsing. By dynamically controlling augmentation depth and decoupling experts via an asymmetric contrastive objective, ADARec achieves SOTA performance.

Acknowledgments

Xiaoxi Cui and Yurong Cheng are supported by the NSFC (Grant No. 62472027, U21A20516,62225203, 62427808), are supported by the Beijing Natural Science Foundation (L241010). Xiangmin Zhou is supported by ARC Discovery Project (DP240100356).

References

Cai, Z.; Wang, S.; Chu, V. W.; and et al. 2025. Unleashing the Potential of Diffusion Models Towards Diversified Sequential Recommendations. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1476–1486. ACM.

Cen, Y.; Zhang, J.; Zou, X.; Zhou, C.; Yang, H.; and Tang, J. 2020. Controllable multi-interest framework for recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2942–2951.

Chang, J.; Gao, C.; Zheng, Y.; Hui, Y.; Niu, Y.; Song, Y.; Jin, D.; and Li, Y. 2021. Sequential recommendation with graph neural networks. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 378–387.

Chen, Y.; Liu, Z.; Li, J.; McAuley, J.; and Xiong, C. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, 2172–2182.

Cui, X.; Lu, W.; Tong, Y.; Li, Y.; and Zhao, Z. 2025a. Diffusion-based multi-modal synergy interest network for click-through rate prediction. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 581–591.

Cui, X.; Lu, W.; Tong, Y.; Li, Y.; and Zhao, Z. 2025b. Multi-Modal Multi-Behavior Sequential Recommendation with Conditional Diffusion-Based Feature Denoising. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1593–1602.

Dang, Y.; Zhang, J.; Liu, Y.; Yang, E.; Liang, Y.; Guo, G.; Zhao, J.; and Wang, X. 2025. Augmenting Sequential Recommendation with Balanced Relevance and Diversity. In *The Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)*, 11563–11571.

He, R.; and McAuley, J. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th International Conference on Data Mining (ICDM)*, 191–200. IEEE.

Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939*.

Ho, J.; Jain, A.; and Abbeel, P. 2020. Denoising Diffusion Probabilistic Models. In *Advances in Neural Information Processing Systems*, volume 33, 6840–6851.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*, 197–206. IEEE.

Li, C.; Liu, Z.; Wu, M.; Xu, Y.; Zhao, H.; Huang, P.; Kang, G.; Chen, Q.; Li, W.; and Lee, D. L. 2019. Multi-interest network with dynamic routing for recommendation at tmall. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2615–2623.

Li, M.; Zhao, X.; Lyu, C.; Zhao, M.; Wu, R.; and Guo, R. 2022. Mlp4rec: A pure mlp architecture for sequential recommendations. *arXiv preprint arXiv:2204.11510*.

Li, X.; Sun, A.; Zhao, M.; Yu, J.; Zhu, K.; Jin, D.; Yu, M.; and Yu, R. 2023a. Multi-Intention Oriented Contrastive Learning for Sequential Recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*, 411–419.

Li, Y.; Hao, Y.; Zhao, P.; Liu, G.; Liu, Y.; Sheng, V. S.; and Zhou, X. 2023b. Edge-enhanced global disentangled graph neural network for sequential recommendation. *ACM Transactions on Knowledge Discovery from Data*, 17(6): 1–22.

Li, Z.; Xie, Y.; Zhang, W. E.; Wang, P.; Zou, L.; Li, F.; Luo, X.; and Li, C. 2024. Disentangle interest trend and diversity for sequential recommendation. *Information Processing & Management*, 61(3): 103619.

Liu, Y.; Zhu, S.; Xia, J.; Ma, Y.; Ma, J.; Liu, X.; Yu, S.; Zhang, K.; and Zhong, W. 2024. End-to-end Learnable Clustering for Intent Learning in Recommendation. In *38th Conference on Neural Information Processing Systems (NeurIPS 2024)*.

Liu, Z.; Chen, Y.; Li, J.; Yu, P. S.; McAuley, J.; and Xiong, C. 2021. Contrastive self-supervised sequential recommendation with robust augmentation. *arXiv preprint arXiv:2108.06479*.

Luo, M.; Li, Y.; and Lin, C. 2025. Enhancing Sequential Recommendation with Global Diffusion. In *The Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI-25)*, 12309–12318.

Ma, H.; Xie, R.; Meng, L.; Chen, X.; Zhang, X.; Lin, L.; and Kang, Z. 2024. Plug-In Diffusion Model for Sequential Recommendation. In *The Thirty-Eighth AAAI Conference on Artificial Intelligence (AAAI-24)*, 8886–8894.

Ma, J.; Zhou, C.; Yang, H.; Cui, P.; Wang, X.; and Zhu, W. 2020. Disentangled self-supervision in sequential recommenders. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 483–491.

Park, Y.-J.; and Tuzhilin, A. 2008. The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM conference on Recommender systems*, 11–18.

Qiu, R.; Huang, Z.; Yin, H.; and Wang, Z. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, 813–823.

Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, 811–820.

Sohl-Dickstein, J.; Weiss, E.; Maheswaranathan, N.; and Ganguli, S. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, 2256–2265. pmlr.

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. 1441–1450.

Tang, J.; and Wang, K. 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.

Tishby, N.; Pereira, F. C.; and Bialek, W. 2000. The information bottleneck method. In *Proceedings of the 37th Annual Allerton Conference on Communication, Control, and Computing*, 368–377.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; and Kaiser, Ł. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 5998–6008. Curran Associates, Inc.

Wang, Y.; Piao, H.; Dong, D.; Yao, Q.; and Zhou, J. 2024. Warming up cold-start CTR prediction by learning item-specific feature interactions. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 3233–3244.

Wu, S.; Wang, Y.; Jing, Q.; Dong, D.; Dou, D.; and Yao, Q. 2023. ColdNAS: Search to modulate for user cold-start recommendation. In *Proceedings of the ACM Web Conference 2023*, 1021–1031.

Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Zhang, J.-G.; Ding, B.; and Cui, B. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 1259–1273. IEEE.

Yang, Y.; Huang, C.; Xia, L.; Huang, C.; Luo, D.; and Lin, K. 2023. Debiased contrastive learning for sequential recommendation. In *Proceedings of the ACM Web Conference 2023*, 1063–1073.

Zhou, K.; Yu, H.; Zhao, W. X.; and Wen, J.-R. 2022. Filter-enhanced mlp is all you need for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, 2388–2399.

Zhou, X.; Lumbantoruan, R.; Ren, Y.; Chen, L.; Yang, X.; and Shao, J. 2024. Dynamic bi-layer graph learning for context-aware sequential recommendation. *ACM Transactions on Recommender Systems*, 2(2): 1–23.