

Wavelet Enhanced Adaptive Frequency Filter for Sequential Recommendation

Huayang Xu^{1*}, Huanhuan Yuan^{1,2*}, Guanfeng Liu², Junhua Fang¹, Lei Zhao¹, Pengpeng Zhao^{1†}

¹School of Computer Science and Technology, Soochow University, China

²School of Computing, Macquarie University, Australia

hyxu2001@stu.suda.edu.cn, hhyuan@stu.suda.edu.cn, guanfeng.liu@mq.edu.au, {jhfang, zhaol, ppzhao}@suda.edu.cn

Abstract

Sequential recommendation has garnered significant attention for its ability to capture dynamic preferences by mining users' historical interaction data. Given that users' complex and intertwined periodic preferences are difficult to disentangle in the time domain, recent research is exploring frequency domain analysis to identify these hidden patterns. However, current frequency-domain-based methods suffer from two key limitations: (i) They primarily employ static filters with fixed characteristics, overlooking the personalized nature of behavioral patterns; (ii) While the global discrete Fourier transform excels at modeling long-range dependencies, it can blur non-stationary signals and short-term fluctuations. To overcome these limitations, we propose a novel method called Wavelet Enhanced Adaptive Frequency Filter for Sequential Recommendation (WEARec). Specifically, it consists of two vital modules: dynamic frequency-domain filtering and wavelet feature enhancement. The former is used to dynamically adjust filtering operations based on behavioral sequences to extract personalized global information, and the latter integrates wavelet transform to reconstruct sequences, enhancing blurred non-stationary signals and short-term fluctuations. Finally, these two modules work synergistically to achieve comprehensive performance and efficiency optimization in long sequential recommendation scenarios. Extensive experiments on four widely-used benchmark datasets demonstrate the superiority of WEARec.

Code — <https://github.com/xhy963319431/WEARec>

Introduction

Sequential Recommendation (SR) plays a crucial role in e-commerce applications by capturing users' dynamic interest shifts through their historical interaction data (Schedl et al. 2018; Hansen et al. 2020). The remarkable success of the transformer architecture in Natural Language Processing (NLP) (Vaswani et al. 2017) and Computer Vision (CV) (Dosovitskiy et al. 2020) has led to significant advancements in sequential recommendation (Kang and McAuley 2018; Zhang et al. 2019; Liu et al. 2021). This has directly inspired a multitude of sequential recommendation models based

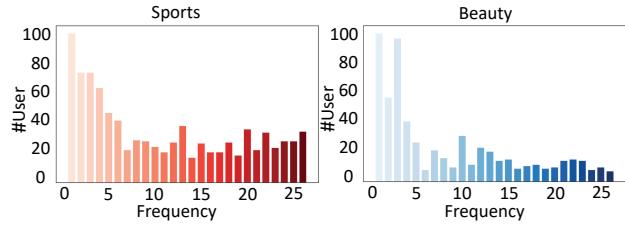


Figure 1: Number of users uniquely driven by each frequency component in the Sports and Beauty datasets.

on self-attention (Sun et al. 2019; Qiu et al. 2022; Zhou et al. 2020). However, items in user interactions are typically chronologically entangled and inherently noisy (Du et al. 2023a,b). Consequently, it is challenging for models to directly discern changes in behavioral preferences from raw sequences (Zhou et al. 2022).

To address this limitation, recent research has begun exploring frequency-domain approaches to replace self-attention mechanisms (Rao et al. 2021; Fein-Ashley, Kannan, and Prasanna 2025; Zhou et al. 2024). By decomposing user sequences into different frequency components (*e.g.*, high-frequency and low-frequency signals) using Fourier transform, periodic patterns that are difficult to identify in the time domain can be effectively captured (Zhou et al. 2022; Du et al. 2023a; Shin et al. 2024; Zhang et al. 2025). For example, FMLPRec (Zhou et al. 2022) pioneered frequency domain processing of sequential data, replacing the self-attention mechanism with learnable filters. SLIME4Rec (Du et al. 2023a) proposed a frequency ramp structure, which considers different frequency bands for each layer. BSARec (Shin et al. 2024) used a frequency domain retuning component as an inductive bias for self-attention.

However, despite their success in SR, existing frequency-domain sequential recommendation models have two key limitations. First, existing methods typically apply a static, fixed-pattern filter to all frequency components, which uniformly processes all user sequences, ignoring the personalized nature of behavioral patterns (Shin et al. 2024; Zhang et al. 2025). This unified filtering approach is susceptible to the influence of dominant users in the dataset and fails to consider the characteristics of individual users. In fact, users often exhibit diverse behavioral patterns: some user behav-

*These authors contributed equally.

†Corresponding author.

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

iors follow long-term preferences (*e.g.*, low-frequency signals), while other user behaviors show the opposite trend (Du et al. 2023a; Zhang et al. 2025). To illustrate this, we trained a classic sequential recommendation model (*i.e.*, FMLPRec (Zhou et al. 2022)) on the Beauty and Sports datasets, and replaced its learnable filters with specific band-pass filters. By statistically analyzing how many users could be correctly predicted by specific frequency components, we could identify which users were driven by particular frequency components. The results, shown in Figure 1, indicate that users exhibit diverse behavioral patterns, with each focusing on different frequencies. This emphasizes the importance of developing personalized filtering models to capture individualized user behavioral patterns.

The second limitation is related to the low-pass filtering characteristic of frequency-domain filters. The Discrete Fourier Transform (DFT) analyzes signal components globally, primarily serving as a global, rather than local, frequency extraction method. While the global DFT excels at capturing long-range dependencies in current frequency-domain recommendation models (Zhou et al. 2022; Du et al. 2023a), it struggles to capture the local temporal features of high-frequency interactions and short-term points of interest (Lu et al. 2025). For instance, FMLPRec has been shown to essentially act as a low-pass filter (Shin et al. 2024). Although SLIME4Rec attempts to balance high-frequency and low-frequency representations through its hierarchical learning mechanism, the model still tends to learn low-frequency components within the hierarchical frequency bands.

To overcome these challenges, we propose **Wavelet Enhanced Adaptive Frequency Filter for Sequential Recommendation (WEARec)**. WEARec consists of two key modules: Dynamic Frequency Filtering (DFF) and Wavelet Feature Enhancement (WFE) module. Specifically, the DFF module uses a simple Multi-Layer Perceptron (MLP) to enhance or suppress specific frequency bands based on context signals, ensuring effective global fusion. Furthermore, the WFE module reconstructs sequences via wavelet transform, amplifying obscure non-stationary signals and short-term fluctuations that are prone to being blurred by global DFT. Finally, to ensure all frequency components are considered and to best preserve meaningful periodic user features, we blend the DFF module with the WFE module. Moreover, our proposed model achieves better performance with lower computational costs, especially in long-sequence scenarios. The main contributions of this paper are summarized as follows:

- We propose a model that includes a dynamic frequency filtering module and wavelet feature enhancement module, which can efficiently fuse personalized global information with enhanced local information
- Our proposed sequential recommendation model demonstrates lower computational overhead and superior recommendation performance compared to state-of-the-art baselines in long-sequence scenarios.
- We conducted extensive experiments on four public datasets, demonstrating the advantages of WEARec over state-of-the-art baselines.

Preliminaries

Before elaborating on the proposed WEARec, we first introduce key mathematical foundations regarding the discrete Fourier transform and discrete wavelet transform.

Discrete Fourier Transform

Given a discrete sequence $\{x_m\}_{m=0}^{n-1}$ of length n , it can be transformed into frequency components via:

$$X_k = \sum_{m=0}^{n-1} x_m e^{-2\pi i m k / n}, 0 \leq k \leq n - 1 \quad (1)$$

where i denotes the imaginary unit, and X_k represents the complex value of the signal at frequency index k .

Simultaneously, $\{X_k\}_{k=0}^{n-1}$ can be transformed back to the time-domain feature representation via the Inverse DFT (IDFT), expressed as:

$$x_m = \frac{1}{n} \sum_{m=0}^{n-1} X_k e^{2\pi i m k / n} \quad (2)$$

In our paper, we convert sequential behaviors into the frequency domain via Fast Fourier Transform (FFT) and denote it by $\mathcal{F}()$. Similar to IDFT, the Inverse FFT (IFFT) (denoted by $\mathcal{F}^{-1}()$) is also used to efficiently transfer the frequency feature back to the time domain.

Discrete Wavelet Transform

Given a discrete sequence $\{x_m\}_{m=0}^{n-1}$ of length n , it can be decomposed into a set of high-frequency and low-frequency sub-signals through hierarchical decomposition. The j -th level decomposition is defined as:

$$A_{j+1}[m] = \sum_{k=0}^{K-1} L[k] A_j[2m - k] \quad (3)$$

$$D_{j+1}[m] = \sum_{k=0}^{K-1} H[k] A_j[2m - k] \quad (4)$$

The indexing $2m - k$ implements downsampling with stride 2, reducing output length by half. Therefore, in the equation, K is $n/2^j$. Where $A_j[m]$ denotes the approximation coefficients after level- j low-pass filtering L , containing the low-frequency components of the signal. When $j = 0$, we set $A_0[m] = x[m]$. $D_j[m]$ denotes the detail coefficients after level- j high-pass H filtering, containing the high-frequency components of the signal. Through wavelet decomposition, Discrete Wavelet Transform (DWT) can localize transient components in time-domain signals, thereby enabling the processing and analysis of non-stationary signals.

Moreover, the decomposed high-frequency and low-frequency sub-signals can be perfectly reconstructed into the original signal via the Inverse Discrete Wavelet Transform (IDWT). It reconstructs the signal stage-by-stage via iterative upsampling and filtering operations:

$$A_j[m] = \sum_{k=0}^{K-1} \tilde{L}[k] A_{j+1}[2m - k] + \sum_{k=0}^{K-1} \tilde{H}[k] D_{j+1}[2m - k] \quad (5)$$

Where \tilde{L} and \tilde{H} are reconstruction filters. In our paper, the forward DWT converts sequential behavior into high/low-frequency sub-signals and denote it by $\mathcal{W}()$. The IDWT (denoted by $\mathcal{W}^{-1}()$) reconstructs decomposed sub-signals into

the original signal. For more descriptions, interested readers should refer to Appendix A (Xu et al. 2025).

Proposed Method

In this section, we first present some necessary notations to formulate the sequential recommendation problem. Additionally, we provide a comprehensive explanation of the overall framework of WEARec, as shown in Figure 2.

Problem Statement

The goal of SR is to predict the next item a user will click based on the user's previous interactions. Given a set of users \mathcal{U} and items \mathcal{V} , where $u \in \mathcal{U}$ denotes a user and $v \in \mathcal{V}$ denotes an item. The numbers of users and items are denoted as $|\mathcal{U}|$ and $|\mathcal{V}|$, respectively. The set of user behavior can be represented as $S = \{s_1, s_2, \dots, s_{|\mathcal{U}|}\}$. In SR, the user's behavior sequence is usually in time order. This means that each user sequence is made up of (chronologically ordered) item interactions $s_u = [v_1^{(u)}, v_2^{(u)}, \dots, v_t^{(u)}, \dots, v_n^{(u)}]$, where $s_u \in S$, $v_t^{(u)} \in \mathcal{V}$ is the item with which user u interacts at step t , and n is the length of the sequence. Specifically, the recommendation model first divides the original sequence into multiple subsequences. After training, it generates a probability score for the candidate items in each subsequence, i.e., $\hat{y} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_{|\mathcal{V}|}\}$, where \hat{y}_i denotes the prediction score of item v_i . Given a user's historical interaction sequences and the maximum sequence length N , the sequence is first truncated by removing earliest item if $n > N$ or padded with 0s to get a fixed length sequence $s_u = [v_1^{(u)}, v_2^{(u)}, \dots, v_N^{(u)}]$. The SR task takes s_u as input to predict the top- K items at the timestamp $N + 1$.

Embedding Layer

Given a user behavior sequence s_u , we define the embedding representation of the sequence \mathbf{E}^u using the item embedding matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{V}| \times d}$, where d is the embedding size and $\mathbf{E}_i^u = \mathbf{M}_{s_i}$. Positional embeddings $\mathbf{P} \in \mathbb{R}^{N \times d}$ are used to add additional positional information while preserving the original embedding dimensionality of the items. Additionally, we perform layer normalization and dropout operations to stabilize the training process. Therefore, we generate the sequence representation $\mathbf{E}^u \in \mathbb{R}^{N \times d}$ as follows:

$$\mathbf{E}^u = \text{Dropout}(\text{LayerNorm}(\mathbf{E}^u + \mathbf{P})) \quad (6)$$

Dynamic Frequency-domain Filtering

Multi-Head Projection. To enhance the representation ability of the input item embedding \mathbf{E}^u in the frequency domain, we draw inspiration from the partitioning concept of the multi-head attention mechanism. Specifically, we decompose the input matrix $\mathbf{E}^u \in \mathbb{R}^{N \times d}$ along the embedding dimension into k parallel feature subspaces, each equipped with an adaptive filter tailored to its characteristics.

$$\mathbf{H}^0 = \mathbf{E}^u \quad (7)$$

$$\mathbf{H}^l = [\mathbf{B}_1, \mathbf{B}_2, \dots, \mathbf{B}_k] \quad (8)$$

where $\mathbf{H}^l \in \mathbb{R}^{N \times d}$ is the time feature of the l -th layer, and $\mathbf{B}_i \in \mathbb{R}^{N \times d/k}$ represents the i -th subspace.

For each subspace, the dynamic frequency-domain filtering layer first performs a fast Fourier transform along the item dimension:

$$\mathcal{F}(\mathbf{B}_i^l) \rightarrow \mathbf{F}_i^l \quad (9)$$

where $\mathbf{B}_i^l \in \mathbb{R}^{N \times d/k}$ is the i -th time domain subspaces feature of the l -th layer, and $\mathcal{F}(\cdot)$ denotes the 1D FFT. Note that $\mathbf{F}_i^l \in \mathbb{C}^{M \times d/k}$ is a complex tensor representing the i -th frequency domain subspace feature of the l -th layer. M is calculated as:

$$M = \lceil N/2 \rceil + 1 \quad (10)$$

To extract the overall information of the user context sequence, we perform mean processing on the input features in the time domain along the item dimension.

$$\mathbf{c}^l = \frac{1}{N} \sum_{i=1}^N \mathbf{H}_i^l \quad (11)$$

where $\mathbf{H}_i^l \in \mathbb{R}^{1 \times d}$ represents the i -th row of \mathbf{H}^l , and $\mathbf{c}^l \in \mathbb{R}^{1 \times d}$ denotes the overall representation of the user's historical interaction sequence at the l -th layer.

To enable dynamic adaptation of our frequency-domain filters to user-specific sequence contexts, we design two three-layer MLP networks that generate corresponding scaling factors and bias terms from captured user contextual features, thereby modulating personalized frequency-domain filters.

$$\Delta \mathbf{s}^l = \text{MLP}_1(\mathbf{c}^l) \quad (12)$$

$$\Delta \mathbf{b}^l = \text{MLP}_2(\mathbf{c}^l) \quad (13)$$

where $\Delta \mathbf{s}^l$ and $\Delta \mathbf{b}^l \in \mathbb{R}^{k \times M}$ denote the scaling factor and bias term of the l -th layer for dynamically adjusting the filter, respectively. The scaling factor shapes the filter's overall frequency response, while the bias term adjusts weights for specific frequency bands.

Given the base filter weights $\mathbf{W}^l \in \mathbb{R}^{k \times M}$ and bias $\mathbf{b}^l \in \mathbb{R}^{k \times M}$, with k representing the number of filters, the weights and bias of the personalized dynamic filter are obtained through the following operation using the personalization-generated scaling factor $\Delta \mathbf{s}^l$ and bias term $\Delta \mathbf{b}^l$:

$$\hat{\mathbf{W}}^l = \mathbf{W}^l \odot (1 + \Delta \mathbf{s}^l) \quad (14)$$

$$\hat{\mathbf{b}}^l = \mathbf{b}^l + \Delta \mathbf{b}^l \quad (15)$$

where $\hat{\mathbf{W}}^l$ and $\hat{\mathbf{b}}^l \in \mathbb{R}^{k \times M}$ denote the linearly modulated weights and bias of the dynamic filter at the l -th layer, respectively. The modulated filter adapts to the frequency-domain characteristics of different users.

Multiple Learnable Filter. By applying a linear transformation to the frequency-domain feature subspace using personalized filter weights and bias, we obtain personalized filtered frequency-domain information.

$$\tilde{\mathbf{F}}_i^l = \mathbf{F}_i^l \odot \hat{\mathbf{W}}^l + \hat{\mathbf{b}}^l \quad (16)$$

Finally, use IDFT to map the processed frequency-domain signal back to the time domain:

$$\mathbf{X}_i^l = \mathcal{F}^{-1}(\tilde{\mathbf{F}}_i^l) \quad (17)$$

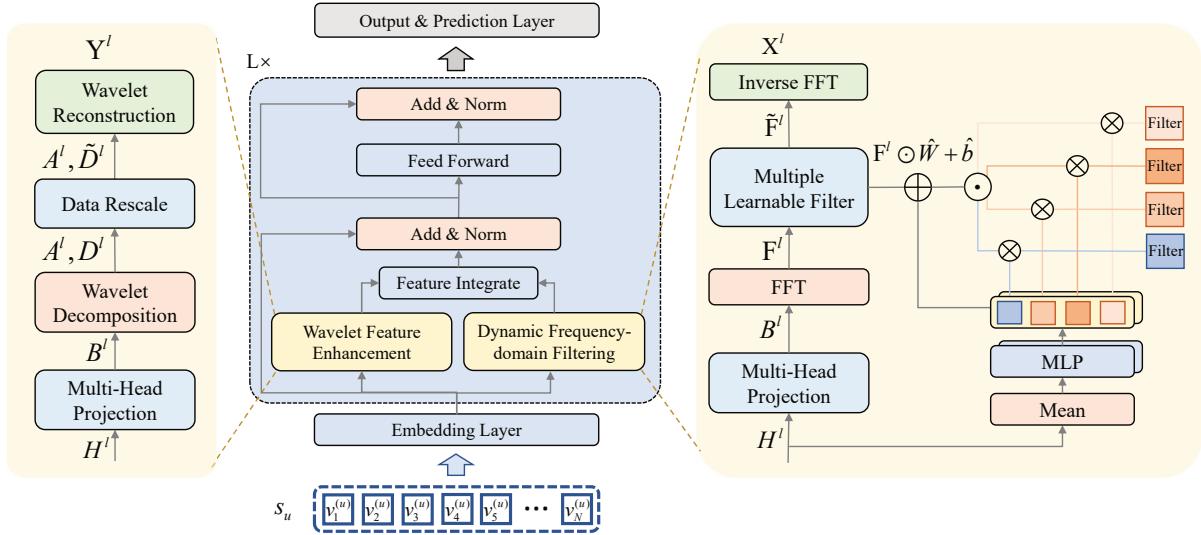


Figure 2: The model architecture of WEARec is similar to the transformer encoder. It first generates item embedding with positional embedding through the embedding layer , and then extracts user preference from the frequency domain by replacing the self-attention module with the wavelet feature enhancement module and dynamic frequency-domain filtering module. Their details are shown on both sides. Finally, a prediction layer computes a recommendation score for all candidate items.

Wavelet Feature Enhancement

This module captures fine-grained temporal patterns through differentiable wavelet transforms. Here, the Haar wavelet transform (Stanković and Falkowski 2003) was selected due to its simple structure, high computational efficiency, and the desirable property of perfect signal reconstruction.

Multi-Head Projection. To ensure alignment between the acquired fine-grained information and the spatial features obtained by dynamic frequency-domain filtering module, we extend the design philosophy of this module.

Wavelet Decomposition. To capture fine-grained temporal patterns in behavioral sequences and enhance non-stationary signals within them, we integrate DWT into the WEARec framework. We implement Haar wavelet transform along the item dimension to decompose temporal signals into low-frequency and high-frequency components.

$$\mathbf{A}_i^l, \mathbf{D}_i^l = \mathcal{W}(\mathbf{B}_i^l) \quad (18)$$

where $\mathbf{B}_i^l \in \mathbb{R}^{N \times d/k}$ is the i -th time domain subspaces feature of the l -th layer, and $\mathcal{W}(\cdot)$ denotes the 1D Haar wavelets transform. $\mathbf{A}_i^l \in \mathbb{R}^{K \times d/k}$ denotes the i -th subspaces' approximation coefficients representing the low-frequency components of the original signal at the l -th layer, while $\mathbf{D}_i^l \in \mathbb{R}^{K \times d/k}$ corresponds to the i -th subspaces' detail coefficients capturing its high-frequency components.

Data Rescale. To acquire the high-frequency information required by the model, we multiply different components of the high-frequency information by an adaptive learnable matrix, thereby adaptively enhancing or suppressing high-frequency signals in the sequence. Since low-frequency information records the original primary components of the sequence, we therefore avoid modifying it.

$$\tilde{\mathbf{D}}_i^l = \mathbf{D}_i^l \odot \mathbf{T}^l \quad (19)$$

where $\tilde{\mathbf{D}}_i^l \in \mathbb{R}^{K \times d/k}$ denotes the enhanced detail coefficients of the i -th subspaces at the l -th layer, enhancing the high-frequency components of the original signal, and $\mathbf{T}^l \in \mathbb{R}^{K \times d/k}$ denotes the adaptive high-frequency enhancer at the l -th layer.

Wavelet Reconstruction. Finally, we reconstruct the high-frequency enhanced time-domain signal by applying the inverse Haar wavelet transform to the processed coefficients.

$$\mathbf{Y}_i^l = \mathcal{W}^{-1}(\mathbf{A}_i^l, \tilde{\mathbf{D}}_i^l) \quad (20)$$

Feature Integrate

Finally, the global features extracted by the dynamic frequency filtering are mixed with the fine-grained features derived from the wavelet feature enhancement.

$$\widehat{\mathbf{H}}^l = \alpha \odot \mathbf{X}^l + (1 - \alpha) \odot \mathbf{Y}^l \quad (21)$$

where α is a hyperparameter designed to emphasize the fine-grained details enhanced by wavelet decomposition. Thus, our core design principle involves balancing wavelet-augmented local features and dynamically filtered global features.

To prevent gradient vanishing when the model gets deeper and to achieve a more stable training process with better generalization ability, typical techniques such as skip connection, dropout, and layer normalization are implemented.

$$\widehat{\mathbf{H}}^l = \text{LayerNorm}(\mathbf{H}^l + \text{Dropout}(\widehat{\mathbf{H}}^l)) \quad (22)$$

Point-wise Feed Forward Network

To endow the models with non-linearity characteristics between different dimensions in the time domain, we also add

a feed-forward network after each feature mixer, which consists of MLP with GELU activation. The process of the point-wise Feed-Forward Neural network (FFN) is defined as follows:

$$\tilde{\mathbf{H}}^l = \text{FFN}(\hat{\mathbf{H}}^l) = (\text{GELU}(\hat{\mathbf{H}}^l \mathbf{W}_1 + \mathbf{b}_1)) \mathbf{W}_2 + \mathbf{b}_2 \quad (23)$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_1, \mathbf{b}_2 \in \mathbb{R}^{1 \times d}$ are learnable parameters. In order to prevent overfitting, we add a dropout layer above each hidden layer and perform layer normalization procedures again using residual connection structure on the output \mathbf{H}^{l+1} , as below:

$$\mathbf{H}^{l+1} = \text{LayerNorm}(\hat{\mathbf{H}}^l + \text{Dropout}(\tilde{\mathbf{H}}^l)) \quad (24)$$

Prediction Layer

In the final layer of WEARec, we can compute the recommendation probability for each candidate item to predict how likely the user would adopt the item. Specifically, the corresponding predicted probability $\hat{\mathbf{y}}$ can be generated by:

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{h}^L(\mathbf{M})^\top) \quad (25)$$

where $\hat{\mathbf{y}} \in \mathbb{R}^{|\mathcal{V}|}$, and $\mathbf{h}^L \in \mathbb{R}^{1 \times d}$ is the output of the L -layer blocks at the final step. To optimize the model parameters, we therefore use the cross-entropy loss (Qiu et al. 2022; Du et al. 2023b,a; Shin et al. 2024). The objective function of SR can be formulated as:

$$\mathcal{L}_{Rec} = - \sum_{i=1}^{|\mathcal{V}|} y_i \log(\hat{y}_i) \quad (26)$$

where y_i is the i -th ground truth item, and \hat{y}_i denotes the preference score of v_i .

Experiments

In this section, we first briefly introduce the datasets used in our experiments, nine baselines, the evaluation metrics, and the implementation details in our experimental settings. Then, we compare our proposed model WEARec with state-of-the-art baseline methods. Specifically, to study the validity of WEARec, we conduct experiments to try to answer the following questions:

RQ1 Does WEARec perform better than the state-of-the-art baselines?

RQ2 How does WEARec perform and what is its computational overhead in long-sequence scenarios?

RQ3 How does each designed module in WEARec contribute to the performance?

RQ4 How do the hyper-parameters affect the effectiveness of WEARec?

Experimental Setup

Datasets. We conduct experiments on four public datasets collected from real-world platforms in order to thoroughly evaluate WEARec. i,ii) Beauty and Sports from Amazon (McAuley et al. 2015), iii) ML-1M (Harper and Konstan 2015), iv) LastFM. Following (Zhou et al. 2020, 2022), we also adopt the 5-core settings by filtering out users with less than 5 interactions. The detailed dataset statistics are presented in Appendix B.1 (Xu et al. 2025).

Evaluation Metrics. In our evaluation, we adopt the leave-one-out strategy for partitioning each user’s item sequence (Zhou et al. 2020). We rank the prediction scores throughout the entire item set without using negative sampling, as recommended by (Krichene and Rendle 2020). Performance is evaluated on a variety of evaluation metrics, including Hit Ratio at K (HR@ K) and Normalized Discounted Cumulative Gain at K (NDCG@ K , NG@ K) on all datasets. The K is set to 10 and 20.

Baseline Models. To demonstrate the effectiveness of the proposed model, we compare WEARec with the most widely used and state-of-the-art methods with two categories:

Time-domain SR models: GRU4Rec (Hidasi et al. 2015), Caser (Tang and Wang 2018), SASRec (Kang and McAuley 2018), and DuoRec (Qiu et al. 2022).

Frequency-domain SR models: FMLPRec (Zhou et al. 2022), FamouSRec (Zhang et al. 2025), FEARC (Du et al. 2023b), SLIME4Rec (Du et al. 2023a), BSARec (Shin et al. 2024).

Implementation Details. We implement our WEARec model in PyTorch. For the baseline models, we refer to their best hyper-parameters setups reported in the original papers and directly report their reimplementations results if available, since the datasets and evaluation metrics used in these works are strictly consistent with ours. Both the dimension of the feed-forward network and item embedding size are set to 64. The number of WEARec blocks L is set to 2, and the maximum sequence length N is set to 50. Batch size is set to 256. The model is optimized by Adam optimizer with a learning rate from {0.0005, 0.001}. The wavelet decomposition level is set to 1. The α is in {0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9}, and the k chosen from {1, 2, 4, 8}. We report the result of each model under its optimal hyper-parameter settings. The best hyperparameters are in Appendix B.3 (Xu et al. 2025) for reproducibility.

Recommendation Performance Comparison (RQ1)

The overall experimental results on four datasets are presented in Table 1. Based on these results, we can draw the following observations and conclusions. Firstly, traditional time-domain-based sequential recommendation methods, such as Caser, GRU4Rec, and SASRec, exhibit suboptimal performance. This is because they fail to adequately identify intertwined user periodic patterns, which are crucial for capturing users’ true interests. DuoRec validates the effectiveness of combining supervised and unsupervised contrastive learning through model and semantic augmentation. Secondly, among these models, methods leveraging the frequency domain (e.g., FMLPRec, FamouSRec, FEARC, SLIME4Rec, BSARec) generally demonstrate superior performance. FMLPRec, by utilizing an MLP structure to attenuate noise in the frequency domain, achieved nearly comparable or even better performance than SASRec on most datasets. FamouSRec, FEARC, and SLIME4Rec further advanced this direction by combining frequency-domain analysis with contrastive learning, achieving better performance. BSARec mitigated the insufficient inductive bias of the self-attention mechanism, enhanced the performance of the at-

Datasets	Metric	Caser	GRU4Rec	SASRec	DuoRec	FMLPRec	FamouSRec	FEARec	SLIME4Rec	BSARec	WEARec	Improv.
Beauty	HR@10	0.0225	0.0304	0.0531	0.0965	0.0559	0.0838	0.0982	0.1006	0.1008	0.1041	3.27%
	HR@20	0.0403	0.0527	0.0823	0.1313	0.0869	0.1146	0.1352	<u>0.1381</u>	0.1373	0.1391	1.31%
	NG@10	0.0108	0.0147	0.0283	0.0584	0.0291	0.0497	0.0601	0.0601	<u>0.0611</u>	0.0614	0.49%
	NG@20	0.0153	0.0203	0.0356	0.0671	0.0369	0.0575	0.0694	0.0696	0.0703	0.0703	0.00%
Sports	HR@10	0.0163	0.0187	0.0298	0.0569	0.0336	0.0424	0.0589	0.0611	<u>0.0612</u>	0.0631	3.10%
	HR@20	0.0260	0.0303	0.0459	0.0791	0.0525	0.0632	0.0836	<u>0.0869</u>	0.0858	0.0895	2.99%
	NG@10	0.0080	0.0101	0.0159	0.0331	0.0183	0.0244	0.0343	0.0357	<u>0.0360</u>	0.0367	1.94%
	NG@20	0.0104	0.0131	0.0200	0.0387	0.0231	0.0297	0.0405	0.0421	<u>0.0422</u>	0.0433	2.60%
LastFM	HR@10	0.0431	0.0404	0.0633	0.0624	0.0560	0.0569	0.0587	0.0633	<u>0.0807</u>	0.0899	11.40%
	HR@20	0.0642	0.0541	0.0927	0.0963	0.0826	0.0954	0.0826	0.0927	<u>0.1174</u>	0.1202	2.38%
	NG@10	0.0268	0.0245	0.0355	0.0361	0.0306	0.0318	0.0354	0.0359	<u>0.0435</u>	0.0465	6.89%
	NG@20	0.0321	0.0280	0.0429	0.0446	0.0372	0.0415	0.0414	0.0433	<u>0.0526</u>	0.0543	3.23%
ML-1M	HR@10	0.1556	0.1657	0.2137	0.2704	0.2065	0.2639	0.2705	<u>0.2891</u>	0.2757	0.2952	2.10%
	HR@20	0.2488	0.2664	0.3245	0.3738	0.3137	0.3717	0.3714	<u>0.3950</u>	0.3884	0.4031	2.05%
	NG@10	0.0795	0.0828	0.1116	0.1530	0.1087	0.1455	0.1516	<u>0.1673</u>	0.1568	0.1696	1.37%
	NG@20	0.1028	0.1081	0.1395	0.1790	0.1356	0.1727	0.1771	<u>0.1939</u>	0.1851	0.1968	1.49%

Table 1: Recommendation algorithms performance comparison on 4 datasets. The best results are in boldface and the second-best results are underlined. ‘Improv.’ indicates the relative improvement against the best baseline performance.

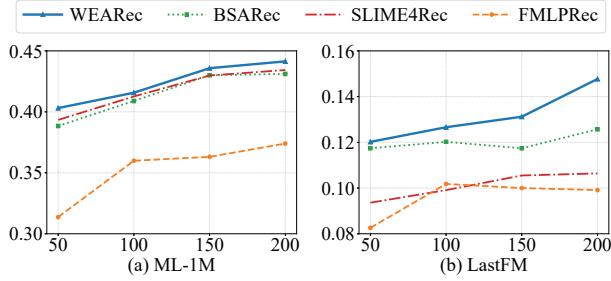


Figure 3: The HR@20 performance comparison of WEARec with FMLPRec, SLIME4Rec and BSARec at different sequence lengths N on ML-1M and LastFM.

tention mechanism, and alleviated over-smoothing through a frequency recalibrator. Finally, based on these results, WEARec achieved the best performance across all four datasets by combining a dynamic frequency-domain filtering module with wavelet feature enhancement module.

Model in Long Sequence Scenarios (RQ2)

Given the sparsity of most datasets in recommendation systems, the maximum sequence length N is often limited to 50 during the evaluation of sequential recommendation models. However, this setting is not appropriate for relatively dense datasets with frequent user interactions.

Model performance. To investigate the impact of long sequence scenarios on recommendation results, we varied the maximum sequence length N for FMLPRec, BSARec, SLIME4Rec and WEARec. We selected the LastFM and ML-1M datasets, which have longer average sequence lengths, for our experiments. Figure 3 presents the experimental results in terms of HR@20. We have obtained similar experimental results in terms of other metrics. We ob-

Methods	ML-1M		LastFM	
	# params	s/epoch	# params	s/epoch
WEARec	426,082	66.46	440,802	5.23
FMLPRec	324,160	36.93	338,880	4.91
BSARec	331,968	109.26	346,688	10.84
SLIME4Rec	375,872	120.43	390,592	13.77

Table 2: The number of parameters and training time (runtime per epoch) for $N = 200$ on ML-1M and LastFM. More results are in Appendix B (Xu et al. 2025).

served that almost all models achieved their best performance at $N = 200$, indicating that longer sequence information can more comprehensively represent user behavior patterns. Furthermore, while baseline models showed performance improvements in long-sequence scenarios, they were prone to overfitting, leading to performance convergence. Finally, WEARec consistently outperformed the baselines across all different maximum sequence length settings, and its improvement over baseline models was even more significant in long-sequence scenarios. For more descriptions, interested readers should refer to Appendix B.4 (Xu et al. 2025).

Model Complexity and Runtime Analyses. To evaluate the overhead of WEARec, we assessed the number of parameters and runtime per epoch during training at $N = 200$. The results are presented in Table 2. We can observe that WEARec exhibits a shorter training time compared to baseline models with competitive performance. Overall, WEARec’s total parameters are increased due to the use of a simple MLP. However, by not employing contrastive learning and self-attention mechanisms, WEARec actually

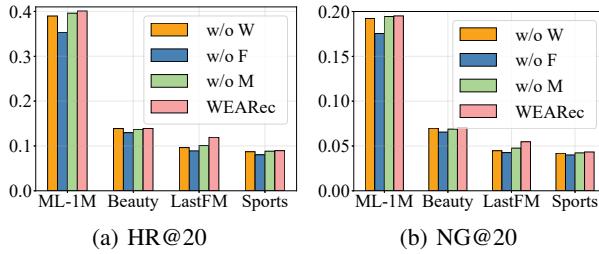


Figure 4: The HR@20 and NG@20 performance achieved by WEARec variants on four datasets.

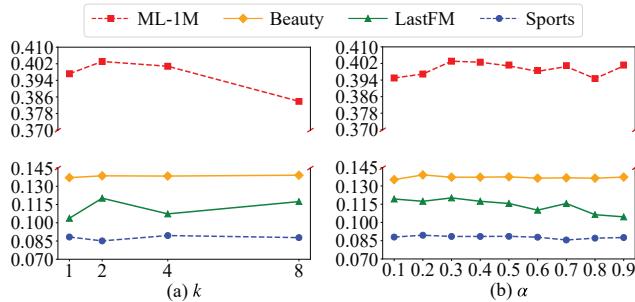


Figure 5: Performance of WEARec on HR@20 with varying hyperparameters..

trains faster than SLIME4Rec and BSARec. For more descriptions, interested readers should refer to Appendix B.5 (Xu et al. 2025).

In-depth Model Analysis (RQ3-RQ4)

Ablation study (RQ3). Figure 4 summarizes the HR@20 and NG@20 performance of WEARec and its variants across four datasets. In this figure, WEARec represents the full WEARec model, while $w/o W$, $w/o F$ and $w/o M$ represent variants where the WFE module, DFF module, and multi-head projection are removed, respectively, with all other components remaining unchanged. The results show that WEARec outperforms its variants on all datasets, indicating that all components are effective.

Hyper-parameter analysis (RQ4).

Sensitivity to k . Figure 5 shows the HR@20 by varying k . The results indicate that an optimal k value, neither too large nor too small, is critical for learning user interest preferences and consequently improving model performance.

Sensitivity to α . Figure 5 shows the HR@20 by varying α . These results suggest that optimal performance is more probable when α is approximately 0.3.

Visualization of the filters. Figure 6 presents the frequency and amplitude features learned by different types of filtering models. Due to static filter design, both FMLPRec and SLIME4Rec tend to learn low-frequency components within their respective frequency bands. Conversely, WEARec, benefiting from its dynamic frequency-domain filtering design, is capable of encompassing all frequency components.

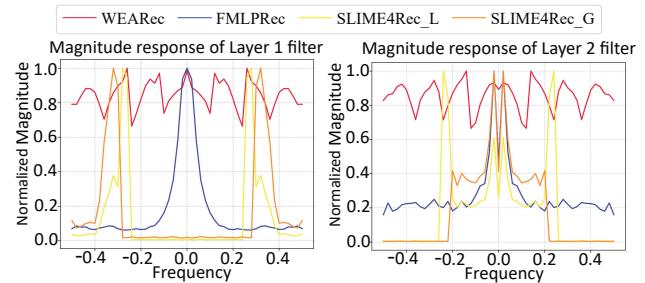


Figure 6: Visualization of spectral responses for different types of filter models across layers in Beauty. More in-depth model analysis in Appendix C (Xu et al. 2025).

Related Works

Time-domain SR Models

Early SR research often relied on Markov chain assumptions (Rendle, Freudenthaler, and Schmidt-Thieme 2010). With the widespread adoption of deep learning methods (He et al. 2017), numerous studies have employed neural network architectures as encoders. Caser (Tang and Wang 2018) utilizes convolutional operations to capture higher-order patterns. SASRec (Kang and McAuley 2018) leverages self-attention mechanisms to capture item-item relationships. Furthermore, recent studies have enhanced sequential embedding representations through contrastive learning (*e.g.*, CL4SRec (Xie et al. 2022) and DuoRec (Qiu et al. 2022)). However, these time-domain models still struggle to effectively capture users' underlying periodic behavioral patterns.

Frequency-domain SR Models

Recently, researchers have begun applying frequency-domain analysis to sequential recommendation. FMLPRec (Zhou et al. 2022) pioneered frequency-based MLP filtering to capture periodic patterns. SLIME4Rec (Du et al. 2023a) and FEAREC (Du et al. 2023b) further advanced this direction by proposing a layered frequency ramp structure integrated with contrastive learning. BSARec (Shin et al. 2024) sought to uncover fine-grained sequential patterns and inject them as inductive biases into the model. FamouSRec (Zhang et al. 2025) developed a MoE architecture for selecting specialized expert models tailored to users' specific frequency-based behavioral patterns. However, these models either lack user-specific adaptivity or incur substantial computational costs.

Conclusion

In this paper, we introduce WEARec, a more efficient model for handling long sequences in sequential recommendation tasks, designed to effectively capture diverse user behavioral patterns. Our method includes dynamic frequency-domain filtering and wavelet feature enhancement. The former dynamically adjusts filters based on user sequences to obtain personalized frequency-domain global distributions. The latter reconstructs sequences through wavelet transforms to enhance non-stationary signals. Extensive experiments on four public datasets validate the effectiveness of WEARec.

Acknowledgements

This research was partially supported by the NSFC (62376180, 62176175, 62572335), the National Key Research and Development Program of China (2023YFF0725002), Suzhou Science and Technology Development Program (SYG202328), and the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

- Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*.
- Du, X.; Yuan, H.; Zhao, P.; Fang, J.; Liu, G.; Liu, Y.; Sheng, V. S.; and Zhou, X. 2023a. Contrastive enhanced slide filter mixer for sequential recommendation. In *ICDE*, 2673–2685.
- Du, X.; Yuan, H.; Zhao, P.; Qu, J.; Zhuang, F.; Liu, G.; Liu, Y.; and Sheng, V. S. 2023b. Frequency enhanced hybrid attention network for sequential recommendation. In *SIGIR*, 78–88.
- Fein-Ashley, J.; Kannan, R.; and Prasanna, V. 2025. The fft strikes again: An efficient alternative to self-attention. *arXiv preprint arXiv:2502.18394*.
- Hansen, C.; Hansen, C.; Maystre, L.; Mehrotra, R.; Brost, B.; Tomasi, F.; and Lalmas, M. 2020. Contextual and sequential user embeddings for large-scale music recommendation. In *RecSys*, 53–62.
- Harper, F. M.; and Konstan, J. A. 2015. The movielens datasets: History and context. *ACM transactions on interactive intelligent systems*, 5(4): 1–19.
- He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *WWW*, 173–182.
- Hidasi, B.; Karatzoglou, A.; Baltrunas, L.; and Tikk, D. 2015. Session-based recommendations with recurrent neural networks. In *ICLR*.
- Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *ICDE*, 197–206.
- Krichene, W.; and Rendle, S. 2020. On sampled metrics for item recommendation. In *KDD*, 1748–1757.
- 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*.
- Lu, S.; Ge, M.; Zhang, J.; Zhu, W.; Li, G.; and Gu, F. 2025. Filtering with Time-frequency Analysis: An Adaptive and Lightweight Model for Sequential Recommender Systems Based on Discrete Wavelet Transform. *arXiv preprint arXiv:2503.23436*.
- McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *SIGIR*, 43–52.
- Qiu, R.; Huang, Z.; Yin, H.; and Wang, Z. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *WSDM*, 813–823.
- Rao, Y.; Zhao, W.; Zhu, Z.; Lu, J.; and Zhou, J. 2021. Global filter networks for image classification. *Advances in neural information processing systems*, 34: 980–993.
- Rendle, S.; Freudenthaler, C.; and Schmidt-Thieme, L. 2010. Factorizing personalized Markov chains for next-basket recommendation. In *WWW*, 811–820.
- Schedl, M.; Zamani, H.; Chen, C.-W.; Deldjoo, Y.; and Elahi, M. 2018. Current challenges and visions in music recommender systems research. *International journal of multimedia information retrieval*, 7: 95–116.
- Shin, Y.; Choi, J.; Wi, H.; and Park, N. 2024. An attentive inductive bias for sequential recommendation beyond the self-attention. In *AAAI*, 8984–8992.
- Stanković, R. S.; and Falkowski, B. J. 2003. The Haar wavelet transform: its status and achievements. *Computers & electrical engineering*, 29(1): 25–44.
- 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. In *CIKM*, 1441–1450.
- Tang, J.; and Wang, K. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *WSDM*, 565–573.
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.
- Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Zhang, J.; Ding, B.; and Cui, B. 2022. Contrastive learning for sequential recommendation. In *ICDE*, 1259–1273.
- Xu, H.; Yuan, H.; Liu, G.; Fang, J.; Zhao, L.; and Zhao, P. 2025. Wavelet Enhanced Adaptive Frequency Filter for Sequential Recommendation. *arXiv preprint arXiv:2511.07028*.
- Zhang, J.; Xie, R.; Lu, H.; Sun, W.; Zhao, W. X.; Chen, Y.; and Kang, Z. 2025. Frequency-Augmented Mixture-of-Heterogeneous-Experts Framework for Sequential Recommendation. In *WWW*, 2596–2605.
- Zhang, T.; Zhao, P.; Liu, Y.; Sheng, V. S.; Xu, J.; Wang, D.; Liu, G.; and Zhou, X. 2019. Feature-level deeper self-attention network for sequential recommendation. In *IJCAI*, 4320–4326.
- Zhou, K.; Wang, H.; Zhao, W. X.; Zhu, Y.; Wang, S.; Zhang, F.; Wang, Z.; and Wen, J.-R. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *CIKM*, 1893–1902.
- Zhou, K.; Yu, H.; Zhao, W. X.; and Wen, J.-R. 2022. Filter-enhanced MLP is all you need for sequential recommendation. In *WWW*, 2388–2399.
- Zhou, X.; Liu, Y.; Qi, L.; Xu, X.; Dou, W.; Zhang, X.; Zhang, Y.; and Zhou, X. 2024. GLFNet: Global and Local Frequency-domain Network for Long-term Time Series Forecasting. In *CIKM*, 3527–3536.

Appendix

A More Preliminaries

A.1 Discrete Fourier Transform

The DFT is a core mathematical tool in signal processing, used to convert discrete-time signals from the time domain to the frequency domain, revealing the frequency components and energy distribution of signals. The key to understanding DFT lies in its two core theorems: Parseval's theorem and the convolution theorem.

Parseval's theorem. It establishes the conservation of a signal's total energy before and after the transform. For an input sequence $\{x_m\}_{m=0}^{n-1}$ and its Fourier transform $\{X_k\}_{k=0}^{n-1}$, the total signal energy remains unchanged, apart from a constant scaling factor. That energy conservation is crucial for our method. It guarantees that adaptive filtering won't accidentally distort the intrinsic information of the input signal. This theorem can be expressed as:

$$\sum_{m=0}^{n-1} |x[m]|^2 = \frac{1}{n} \sum_{k=0}^{n-1} |X[k]|^2 \quad (27)$$

The convolution theorem. It reveals that time-domain convolution is equivalent to element-wise multiplication in the frequency domain. This property directly enables the design of filters that capture specific frequency components.

Given the input sequence signal $\{x_m\}_{m=0}^{n-1}$ and convolution parameters $\{h_m\}_{m=0}^{n-1}$, the discrete convolution operation yields:

$$h[m] * x[m] = \sum_{m=0}^{n-1} h[m] x_n[m - b] \quad (28)$$

Where $*$ denotes the circular convolution operator, and $x_n[\cdot]$ is the period extension of $\{x_m\}_{m=0}^{n-1}$. The convolution theorem proves that, given the transformed frequency feature $X[k]$, the above equation can be transformed into:

$$h[m] * x[m] = \mathcal{F}^{-1}(\mathcal{F}(h[m]) \odot X[k]) \quad (29)$$

Where $\mathcal{F}(h[m])$ is the learnable filter, and \odot denotes the hadamard product (element-wise multiplication). This formula can be simplified to:

$$\mathcal{F}(h * x) = \mathbf{W} \odot \mathbf{X} \quad (30)$$

Where $\mathbf{W} \in \mathbb{C}$ corresponds to the learnable complex-valued filtering matrix, and $\mathbf{X} \in \mathbb{C}$ represents the frequency-domain representation of the input signal.

A.2 Discrete Wavelet Transform

Discrete Wavelet Transform (DWT) is a multiresolution analysis-based signal processing tool that achieves time-frequency localized decomposition of signals via scaling and shifting operations on wavelet basis functions.

The properties of a wavelet transform depend on the choice of wavelet. For each wavelet type, there exists a father wavelet function and a mother wavelet function, used in wavelet transforms to extract the overall trend characteristics and detailed features of signals, respectively. In DWT, these

Specs.	LastFM	ML-1M	Beauty	Sports
# User	1,090	6,041	22,363	25,598
# Items	3,646	3,417	12,101	18357
# Interactions	52,551	999,611	198,502	296,337
# Avg.Length	48.2	165.5	8.9	8.3
Sparsity	98.68%	95.16%	99.93%	99.95%

Table 3: Statistics of the datasets after preprocessing.

Specs.	LastFM	ML-1M	Beauty	Sports
α	0.3	0.3	0.2	0.3
k	2	2	8	4
lr	0.001	0.0005	0.0005	0.001

Table 4: Best hyperparameters of WEARec on all datasets.

functions are replaced by a set of discrete filters L and H . L represents the low-pass filter, extracting low-frequency information (e.g., the signal's overall trend), corresponding to the father scaling function. H represents the high-pass filter, extracting high-frequency information (e.g., the signal's detailed features), corresponding to the mother wavelet function. They satisfy the relation: $H[n] = (-1)^n L[1 - n]$.

B Additional Details for Experiments

B.1 Details of Datasets

We provide 4 benchmark datasets used for our experiments. These datasets, which differ in scenarios, sizes, and sparsity, are frequently used in tests of sequential recommendation methods. The main statistics of four datasets after preprocessing are reported in Table 3. We elaborate on the descriptions of the individual dataset below.

- **LastFM** contains user interaction with music, such as artist listening records. It is used to recommend musicians to users in sequential recommendation with long sequence lengths.
- **MovieLens-1M** (Harper and Konstan 2015) is based on reviews of movies that were collected from the non-commercial movie recommendation website MovieLens. The interaction number in ML-1M is about 1 million.
- **Amazon Beauty, and Sports** (McAuley et al. 2015) These datasets contain user-item interactions from the Amazon review dataset. They consist of product reviews and ratings, providing a rich data source for evaluating sequential recommendation models in the e-commerce domain.

B.2 Details of Baselines

To demonstrate the effectiveness of the proposed model, we compare WEARec with the most widely used and state-of-the-art methods with two categories:

Method	ML-1M				LastFM				
	HR@10	NG@10	HR@20	NG@20	HR@10	NG@10	HR@20	NG@20	
$N = 50$	BSARec	0.2757	0.1568	0.3884	0.1851	0.0807	0.0435	0.1174	0.0526
	SLIME4Rec	0.2894	0.1675	0.3934	0.1937	0.0633	0.0376	0.0936	0.0453
	Ours	0.2952	0.1696	0.4031	0.1968	0.0899	0.0465	0.1202	0.0547
$N = 100$	BSARec	0.3073	0.1815	0.4089	0.2024	0.0798	0.0455	0.1202	0.0545
	SLIME4Rec	0.3147	0.1815	0.4126	0.2062	0.0679	0.0382	0.0991	0.0463
	Ours	0.3180	0.1819	0.4175	0.2069	0.0890	0.0494	0.1266	0.0589
$N = 150$	BSARec	0.3171	0.1826	0.4300	0.2111	0.0826	0.0476	0.1174	0.0564
	SLIME4Rec	0.3166	0.1820	0.4298	0.2127	0.0688	0.0387	0.1055	0.0479
	Ours	0.3215	0.1848	0.4338	0.2131	0.0927	0.0522	0.1312	0.0617
$N = 200$	BSARec	0.3161	0.1837	0.4311	0.2127	0.0862	0.0476	0.1257	0.0594
	SLIME4Rec	0.3166	0.1850	0.4343	0.2173	0.0679	0.0391	0.1064	0.0488
	Ours	0.3334	0.1904	0.4421	0.2179	0.0972	0.0556	0.1477	0.0682

Table 5: Performance comparison of WEARec with SLIME4Rec and BSARec at different sequence lengths N . HR is the abbreviation for Hit Ratio, and NG is the abbreviation for NDCG. The best performing part in each row is shown in bold.

Methods	ML-1M		Beauty		Sports		LastFM	
	# params	s/epoch	# params	s/epoch	# params	s/epoch	# params	s/epoch
WEARec	426,082	66.46	981,922	15.12	1,382,306	26.12	440,802	5.23
FMLPRec	324,160	36.93	880,000	10.11	1,280,384	22.78	338,880	4.91
BSARec	331,968	109.26	887,808	25.87	1,288,192	50.59	346,688	10.84
SLIME4Rec	375,872	120.43	931,712	31.44	1,332,096	68.74	390,592	13.77

Table 6: The number of parameters and training time (runtime per epoch) for $N = 200$ on all datasets.

- **Time-domain SR models:** **GRU4Rec** (Hidasi et al. 2015) is the first model to apply Gated Recurrent Unit (GRU) to model sequences of user behavior for sequential recommendation, **Caser** (Tang and Wang 2018) is a CNN-based method capturing local dynamic patterns of user activity by using horizontal and vertical convolutional filters over time, **SASRec** (Kang and McAuley 2018) captures relations between items in a sequence using the self-attention mechanism, and **DuoRec** (Qiu et al. 2022) uses unsupervised model-level augmentation and supervised semantic positive samples for contrastive learning.
- **Frequency-domain SR models:** **FMLPRec** (Zhou et al. 2022) is a all-MLP model using a learnable filter enhanced block to remove noise in the embedding matrix, **FamouSRec** (Zhang et al. 2025) uses a Mixture-of-Experts (MoE) approach, allowing the model to focus on different frequency ranges via heterogeneous encoder modules, **FEARec** (Du et al. 2023b) utilizes frequency domain information in attention computation and integrates information from both time and frequency domains, **SLIME4Rec** (Du et al. 2023a) utilizes a frequency ramp structure to consider different frequency bands for each layer with dynamic and static selection

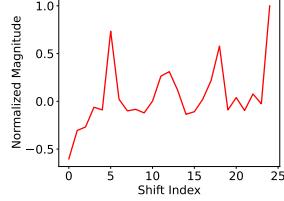
modules, **BSARec** (Shin et al. 2024) adjusts the influence on the high-frequency region to be learnable and utilizes it as an inductive bias of self-attention. It is the most recent and strong baseline for sequential recommendation.

B.3 Experimental Settings & Hyperparameters

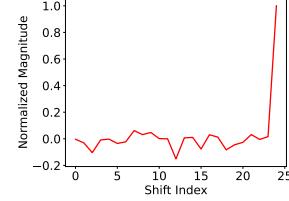
The following software and hardware environments were used for all experiments: PYTHON 3.9.7, PYTORCH 2.3.0, NUMPY 1.24.3, SCIPY 1.11.1, CUDA 12.2, and NVIDIA Driver 535.104.05, and Intel Xeon Gold 6248R CPU, and 40GB NVIDIA Tesla A100 GPU. For reproducibility, we introduce the best hyperparameter configurations for each dataset in Table 4. We conducted experiments under the following hyperparameters: the α is in $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, and the number of filters k is chosen from $\{1, 2, 4, 8\}$. For training, the Adam optimizer is optimized with learning rate in $\{0.0005, 0.001\}$. Moreover, to address the sparsity of Amazon datasets and LastFM dataset, a dropout rate of 0.5 is used, compared to 0.1 for MovieLens-1M.

B.4 Model Performance

To investigate the impact of long sequence scenarios on recommendation results, we varied the maximum se-



(a) WEARec,WFE,1 layer



(b) WEARec,WFE,2 layer

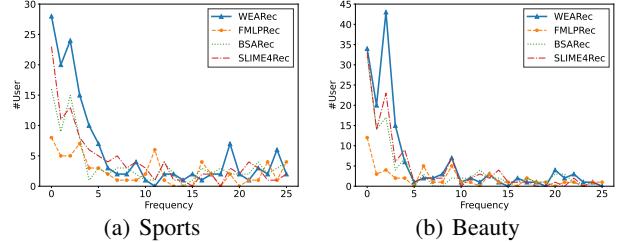
Figure 7: Visualization of learned \mathbf{T} in Beauty

quence length N for FMLPRec, BSARec, SLIME4Rec and WEARec. Tabel 5 presents the experimental results. We observed that almost all models achieved their best performance at $N = 200$, indicating that longer sequence information can more comprehensively represent user behavior patterns. Furthermore, while baseline models showed performance improvements in long-sequence scenarios, they were prone to overfitting, leading to performance convergence. Finally, WEARec consistently outperformed the baselines across all different maximum sequence length settings, and its improvement over baseline models was even more significant in long-sequence scenarios.

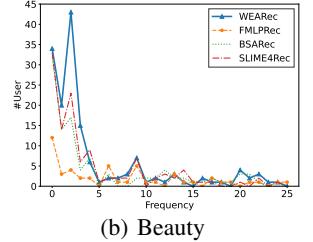
B.5 Model Complexity and Runtime Analyses

Different from transformer-based models that rely on the self-attention mechanism, WEARec is an attention-free architecture with dynamic frequency-domain filtering module and wavelet feature enhancement module. The computation complexity of traditional self-attention is $\mathcal{O}(n^2d + nd^2)$, where n is the sequence length of user and d is hidden size. In contrast, the time complexity of the feature mixing layer can be reduced to $\mathcal{O}(nd\log n + nd)$ with FFT and Haar wavelets transform. For the adaptive filtering, which includes two MLPs for context-based parameter generation, the time complexity is $\mathcal{O}(2nd^2)$, and the time cost of feed-forward networks is $\mathcal{O}(nd^2)$. Therefore, the total time complexity of WEARec is $\mathcal{O}((nd\log n + nd + 3nd^2))$ which is proportional to the log-linear complexity of the input sequence length n . Because $\mathcal{O}(nd\log n)$ typically dominates the cost of element-wise operations for large n , the total complexity remains $\mathcal{O}(nd\log n)$. This demonstrates that our model exhibits superior time complexity compared to models using self-attention mechanisms with $\mathcal{O}(n^2)$ overhead (*e.g.*, BSARec). Furthermore, as our model does not employ contrastive learning, it also achieves lower computational overhead than sequential recommendation models leveraging contrastive learning (*e.g.*, SLIME4Rec).

To evaluate the complexity and efficiency of WEARec, we evaluate the number of parameters and runtime per epoch. Results for all datasets are shown in Table 6. Overall, WEARec increases the total parameters. We show that WEARec has faster runtime times per epoch than FEARec and DuoRec across all datasets.



(a) Sports



(b) Beauty

Figure 8: Case Study in Sports and Beauty.

C More Visualization and Case Study

C.1 Visualization of Learned \mathbf{T}

In Figure 7, we visualize the learned wavelet feature enhancer \mathbf{T} in Equation 19 at each layer on the Beauty dataset. We can observe that the first-layer wavelet feature enhancement module learns multiple time points of non-stationary signals and assigns them higher weights. Moreover, for non-stationary signal points that might constitute noise information, negative weights are assigned to suppress them. In the second layer, the nearest non-stationary signal points are selected for enhancement. This indicates that the wavelet feature enhancer can capture the time points of non-stationary signals and adaptively adjust their weights.

C.2 Case Study

To evaluate whether the dynamic filter and wavelet enhancer can better capture a wider range of frequency domain features, we visualized the number of users correctly captured by different models, building upon the experiment shown in the paper’s introduction. This further explains the effectiveness of integrating these two modules. The results are presented in Figure 8. Based on our observations from Figure, we can draw the following conclusions: Firstly, FMLPRec captured the fewest users across multiple frequency bands. This is because its use of static filters prevents it from capturing diverse user behavior patterns. BSARec and SLIME4Rec achieved better results because they either utilize inductive bias to enhance self-attention, or use frequency ramp structures and contrastive learning to enhance user embedding representations. Finally, using dynamic filters combined with wavelet enhancers yielded even better results. Benefiting from the dynamic filter’s global information capture capability and the wavelet enhancer’s local detail extraction, the model achieved the best performance in low-frequency regions and certain high-frequency regions.