Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.062599

ARTICLE

HyTiFRec: Hybrid Time-Frequency Dual-Branch Transformer for Sequential Recommendation

Dawei Qiu[1], Peng Wu[1,*], Xiaoming Zhang[2,*] and Renjie Xu[3]

[1]School of Information Science and Engineering, Zhejiang Sci-Tech University, Hangzhou, 310018, China
[2]Department of Vehicle Engineering, Army Academy of Armored Forces, Beijing, 100072, China
[3]Performance and Training Center, Army Academy of Armored Forces, Beijing, 100072, China

*Corresponding Authors: Peng Wu. Email: wupeng@zstu.edu.cn; Xiaoming Zhang. Email: xhg.1999@tsinghua.org.cn

Received: 22 December 2024; Accepted: 08 February 2025; Published: 16 April 2025

ABSTRACT: Recently, many Sequential Recommendation methods adopt self-attention mechanisms to model user preferences. However, these methods tend to focus more on low-frequency information while neglecting high-frequency information, which makes them ineffective in balancing users' long- and short-term preferences. At the same time, many methods overlook the potential of frequency domain methods, ignoring their efficiency in processing frequency information. To overcome this limitation, we shift the focus to the combination of time and frequency domains and propose a novel Hybrid Time-Frequency Dual-Branch Transformer for Sequential Recommendation, namely HyTiFRec. Specifically, we design two hybrid filter modules: the learnable hybrid filter (LHF) and the window hybrid filter (WHF). We combine these with the Efficient Attention (EA) module to form the dual-branch structure to replace the self-attention components in Transformers. The EA module is used to extract sequential and global information. The LHF and WHF modules balance the proportion of different frequency bands, with LHF globally modulating the spectrum in the frequency domain and WHF retaining frequency components within specific local frequency bands. Furthermore, we use a time domain residual information addition operation in the hybrid filter module, which reduces information loss and further facilitates the hybrid of time-frequency methods. Extensive experiments on five widely-used real-world datasets show that our proposed method surpasses state-of-the-art methods.

KEYWORDS: Sequential recommendation; frequency domain; efficient attention

1 Introduction

In today's era of information overload, using recommendation algorithms to provide users with personalized recommendations from massive data sets has become a research priority. Traditional recommender systems [1,2] usually model user preferences in a static manner and capture only general interests. In contrast, sequential recommendation accounts for the temporal characteristics of user behavior and captures user's evolved and dynamic preferences.

Early Sequential recommendation methods mainly relied on Markov chains [3] to predict user behavior based on the transition probabilities between behaviors. With the development of deep learning, deep neural network-based sequential recommendation systems (SRSs) have gradually become the focus of research. Recurrent Neural Networks (RNNs) [4,5] effectively process sequential data through their recurrent structure, capturing long-term dependencies in user behavior sequences. Convolutional Neural Networks (CNNs) model [6,7] user preferences by extracting local features through convolutional operations. Additionally,

memory networks [8] and self-attention [9] mechanisms have also been introduced and have achieved notable success. In recent years, the Transformer architecture has demonstrated superior performance in Sequential recommendation tasks. Transformer-based approaches [10,11] leverage multi-head self-attention layers to model complex interactions in long sequences to improve recommendation accuracy and leverage its parallelization capabilities to significantly improve training efficiency.

However, recent studies [12,13] have shown that self-attention tends to preserve low-frequency signals while diminishing high-frequency signals. As shown in Fig. 1, in recommendation systems, high-frequency information typically involves items frequently purchased within short time intervals (e.g., milk or notebooks), reflecting users' immediate needs or short-term preference changes. In contrast, low-frequency information usually includes items purchased over long time intervals (e.g., cameras or smartphones), representing users' long-term preferences. This implies that sasrec [10] and its variants tend to ignore users' short-term preference changes. Jannach et al. [14] highlight the necessity of considering both short- and long-term preferences, and thus how to balance the two to overcome this limitation is insightful.

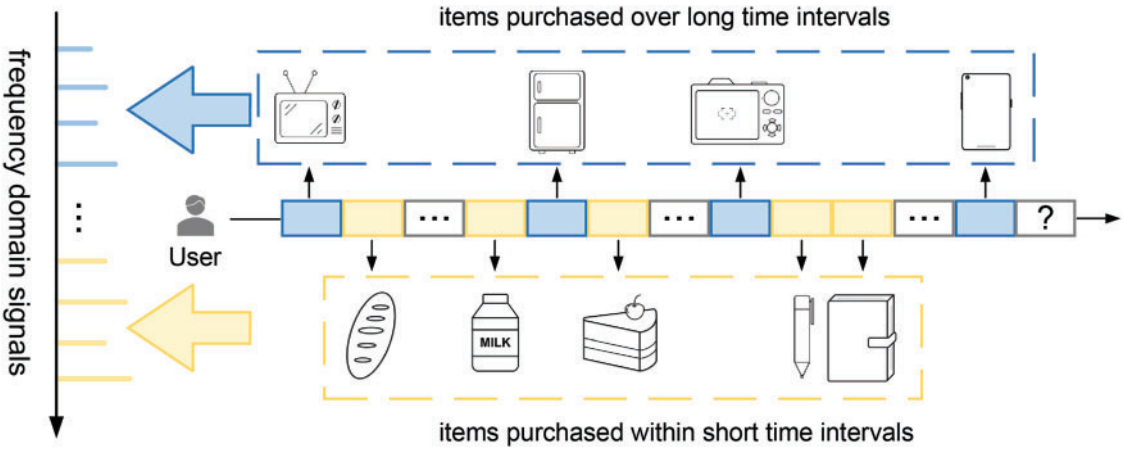

Figure 1: Illustration of high and low-frequency signals in SR

To achieve a better balance between users' short-term and long-term preferences, LOCKER [15] combines local encoder with global self-attention networks, enhancing its capability to capture recent user dynamics while preserving the extraction of long-term user preferences. MPAN [16] employs an innovative temporal attention mechanism to capture short-term preferences and utilizes a multi-head self-attention module to derive long-term preferences from diverse semantic perspectives. DIARec [17] combines the Intention Learning Module and the Preference Module to capture users' multifaceted preferences. FEARec [18] employs an adaptive frequency modulation mechanism to enhance self-attention, combining temporal attention with spectral attention based on an autocorrelation mechanism to extract both high and low frequency information as well as periodic features in user behavior.

We observe the convenience and efficiency of processing frequency band information in the frequency domain and the potential of hybrid methods to address this limitation. Thus, from the perspective of the synergy between time domain and frequency domain methods, we propose a Hybrid Time-Frequency Dual-Branch Transformer for Sequential Recommendation, named **HyTiFRec**. In HyTiFRec, we replaced the self-attention layer in the Transformer block with a dual-branch structure, where each branch consists of a time domain layer and a frequency domain layer. We define the time domain layer as a layer that directly processes time domain signals containing sequential information, and the frequency domain layer as a layer

that converts the time domain signals from the time domain to the frequency domain and processes it. The user's long-term interest reflects their overall preferences, providing a stable basis for recommendations. Therefore we employ the efficient attention mechanism [19] in the time domain layer to capture the temporal dependencies in the user's historical behavioral sequences. Efficient attention is mathematically equivalent to dot-product attention with scaling normalization, but it can better capture global information. In order to preserve high-frequency information, we use residual connection in the time domain layer to retain the original information. In the frequency layer, we design a learnable hybrid filter module and a window hybrid filter module. Specifically, the two modules are used to adjust and balance the frequency information by using the filtering method that modulates all frequency components of the spectrum using learnable filters and retains specific local frequency components, respectively. The combination of the time domain and frequency domain layers, along with the synergy between the two branches, enables HyTiFRec to effectively capture and balance users' global and recent preferences. This, to some extent, addresses the limitation of many attention-based methods in overlooking users' short-term interests. It is worth noting that, although many models [20,21] employ learnable filters, our proposed learnable hybrid filter module is unique in that it adaptively incorporates certain time domain information into the filtered output, reducing information loss and achieving a time-frequency fusion effect. We also apply this same design principle to the window hybrid filtering module.

We conduct comprehensive experiments on five real-world datasets from diverse scenarios to assess the effectiveness of our model. The results indicate that our approach surpasses seven baseline methods in recommendation performance.

2 Preliminary

This section outlines the problem statement and provides relevant background on the Fourier transform and efficient attention mechanism, which will facilitate a detailed explanation of the proposed HyTiFRec.

2.1 Problem Statement

The task of sequential recommendation is to estimate the probability distribution of the potential next interaction based on the user's chronological interaction history. Let $U = \left\{u_1, u_2, ..., u_{|U|}\right\}$ and $I = \left\{i_1, i_2, ..., i_{|I|}\right\}$ denote the sets of users and items, where $|U|$ and $|I|$ represent the number of users and items, and each u and i correspond to a user and an item, respectively. The interaction sequence of a user $u \in U$ with items, ordered chronologically, is represented as $S^u_{1:n} = \{i_1, i_2, ..., i_n\}$, where n is the sequence length. The task is to estimate the probability of the next interacted item given the historical interactions of the user, formulated as $p\left(i^u_{1:n+1} = i | S^u_{1:n}\right)$.

2.2 Fourier Transform

The Discrete Fourier Transform (DFT) [22,23] is a critical algorithm in the field of signal analysis and processing. This study utilize 1D DFT to project sequential feature signals from the time domain to the frequency domain. Given a digital sequence $\{x[n]\}$, $0 \le n < N$, the transformation extracts its frequency-domain representation as follows:

$$\hat{x}[k] = \sum_{n=0}^{N-1} x[n] e^{-i\frac{2\pi}{N}nk}, \ 0 \le k < N \tag{1}$$

where i is the imaginary unit, and e is the base of the natural logarithm. This transformation is denoted as $\hat{x} = F(x)$, with $\hat{x}[k]$ representing the frequency spectrum of $x[n]$ at frequency $\omega_k = \frac{2\pi k}{N}$. By applying

the DFT, signals are transformed from the time domain to the frequency domain, allowing us to study the spectral structure and variation patterns of the signal. The sequence $\hat{x}[k]$ obtained from the DFT can be restored to the original signal $x[n]$ using the Inverse Discrete Fourier Transform (IDFT):

$$x[n] = \frac{1}{N}\sum_{k=0}^{N-1}\hat{x}[k]\, e^{i\frac{2\pi}{N}nk},\ 0 \le n < N \tag{2}$$

The Fast Fourier Transform (FFT) [24,25] is an algorithm that efficiently computes the DFT and its inverse. Calculating the DFT for a sequence of length n has a computational complexity of $O\left(n^2\right)$, while the FFT reduces this to $O(n \log n)$. The IDFT, being an inverse operation of the DFT, can also be computed with the FFT, referred to as the Inverse Fast Fourier Transform (IFFT). In this paper, we employ the FFT to perform time-frequency domain transformations on the signal.

2.3 Efficient Attention

Efficient Attention (EA) is an optimized implementation of the attention mechanism. In EA, the feature matrix $X \in \mathbb{R}^{n\times d}$ is projected through three linear transformations to obtain the queries $Q \in \mathbb{R}^{n\times d_k}$, the keys $K \in \mathbb{R}^{n\times d_k}$ and the values $V \in \mathbb{R}^{n\times d_v}$. The specific implementation is as follows:

$$EA(Q,K,V) = f_q(Q)\, F_g = f_q(Q)\left(f_k(K)^T V\right) \tag{3}$$

where f_q applies softmax across the rows of Q, and f_k applies it across the columns of K. EA provides a new interpretation for the attention mechanism. Instead of gathering similarities for each position to generate attention maps, EA treats the keys K as d_k attention maps K_i^T. Each K_i^T represents a global attention map, reflecting one semantic aspect of the entire input. These d_k attention maps K_i^T are used to aggregate the values V through weighted summation, forming a global feature map F_g. The computational complexity decreases from $O\left(n^2\right)$ for $\left(QK^T\right)V$ to $O(d_k d_v)$ for $Q\left(K^T V\right)$, where n represents the input size, and d_k and d_v denote the dimensions of the key and value, respectively. As the sequence length n increases, the computational cost of dot-product attention grows quadratically. Therefore, Efficient Attention significantly alleviates the computational overhead for long sequences or large datasets.

3 Method

This section elaborates on the proposed method in three components, as illustrated in Fig. 2: Embedding Layer, Encoder, and Prediction Layer.

3.1 Embedding Layer

For a user interaction sequence $S_{1:N}^u = \{i_1, i_2, \ldots, i_N\}$, we create an item embedding matrix $M \in \mathbb{R}^{|I|\times d}$ and obtain the input embedding matrix $E^u \in \mathbb{R}^{n\times d}$ by retrieving the corresponding embeddings from M. For sequences with more than n interactions, only the first n items are considered, while sequences with fewer than n interactions are left-padded with a constant zero vector 0 to reach a length of n. We add a learnable positional embedding $P \in \mathbb{R}^{n\times d}$ to E^u to better utilize the order information between items. Additionally, layer normalization [26] and dropout [27] are implemented:

$$E^u = Dropout\left(LayerNorm\left(E^u + P\right)\right) \tag{4}$$

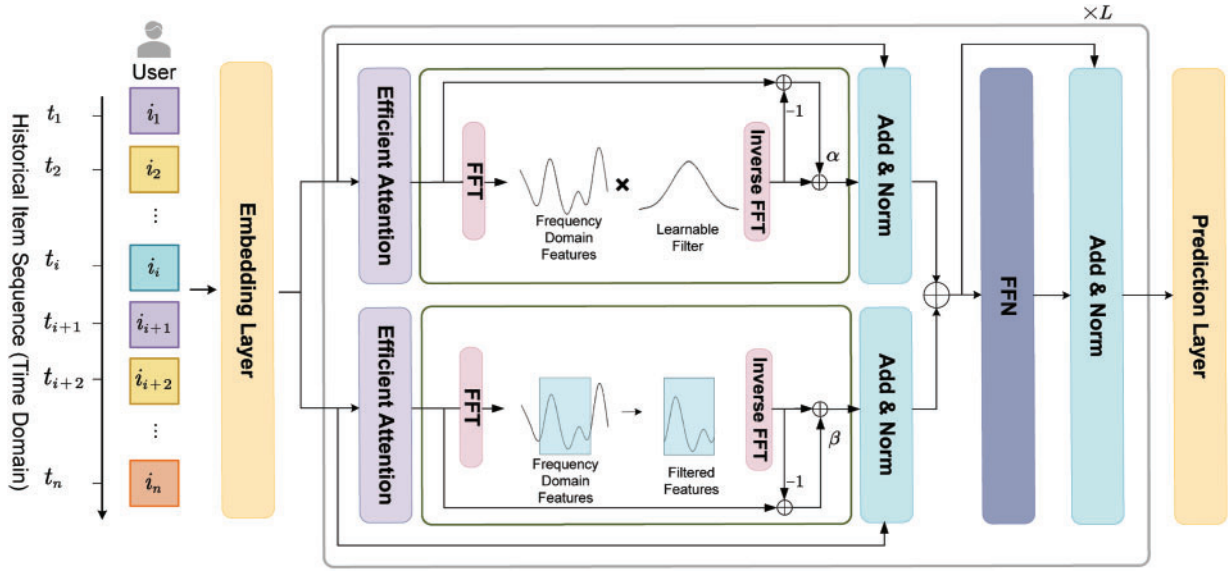

Figure 2: Illustration of HyTiFRec. HyTiFRec constructs item representations by combining item and position embeddings in the embedding layer, and then employs a proposed dual-branch structure based on time-frequency hybrid to replace the self-attention layer for extracting user preferences

3.2 Hybrid Time-Frequency Encoder

We construct the item encoder by stacking multiple Hybrid Time-Frequency (HyTiF) blocks. The HyTiF block generally consists of the dual-branch structure and the point-wise Feed-Forward Network (FFN). We employ the efficient attention layer as the time domain layer for the branches, and the proposed Learnable Hybrid Filter (LHF) and Window Hybrid Filter (WHF) layers as the frequency domain layers for the branches. Accordingly, we refer to these two branches as the LHF branch and the WHF branch.

3.2.1 Time Domain Layer

In the time domain layer, we use an attention mechanism to extract features. Considering that the feature map F_g of Efficient Attention (EA) explicitly aggregates global information, while classic self-attention primarily focuses on pair-wise relation, we adopt EA to better capture global information. Multi-head Efficient Attention (MEA) is the multi-head version of EA. In MEA, we execute the attention computations in parallel, generating d-dimensional outputs, which are concatenated and then projected to yield the final result. Let C^l denote the input to the l-th layer, where $C^0 = E^u$:

$$\hat{C}^l = MEA\left(C^l\right) = Concat\left(head_1, \ldots, head_h\right) W^O, head_i = EA\left(C^l W_i^Q, C^l W_i^K, C^l W_i^V\right) \tag{5}$$

where $W_i^Q \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{model} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{model} \times d_V}$ and $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ are the projection matrices. To mitigate the problem of vanishing gradients and unstable training, and also to preserve the original information, we perform residual connection, layer normalization, and dropout:

$$\hat{C}^l = LayerNorm\left(C^l + Dropout\left(\hat{C}^l\right)\right) \tag{6}$$

The outputs of the time domain layers in the LHF branch and the WHF branch are denoted as $\hat{C}_L^l \in \mathbb{R}^{n \times d}$ and $\hat{C}_W^l \in \mathbb{R}^{n \times d}$, respectively.

3.2.2 Learnable Hybrid Filter

For the frequency domain layer of the LHF branch, we propose a Learnable Hybrid Filter. LHF performs filtering operations on each dimension of the spectrum of $\hat{C}_L^l$, followed by residual connection and layer normalization. FFT is first applied along the item dimension to transform $\hat{C}_L^l$ into the frequency domain:

$$X_L^l = f\left(\hat{C}_L^l\right) \in \mathbb{C}^{n\times d} \tag{7}$$

where $f(\cdot)$ represents the 1D FFT. After applying FFT to $\hat{C}_L^l$, we obtain a complex tensor X_L^l, which represents the frequency spectrum of $\hat{C}_L^l$. Then, we set a learnable filter $K \in \mathbb{C}^{n\times d}$ to modulate X_L^l:

$$\tilde{X}_L^l = K \odot X_L^l \tag{8}$$

where $\odot$ denotes the element-wise multiplication. The learnable filter K essentially weights the frequency components, and its weights are updated during each training epoch using the Adam optimization algorithm, achieving adaptive filtering. The modulated frequency spectrum signal is then converted back to a time domain using the 1D IFFT:

$$\tilde{C}_L^l = f^{-1}\left(\tilde{X}_L^l\right) \in \mathbb{R}^{n\times d} \tag{9}$$

where $f^{-1}(\cdot)$ represents the 1D IFFT. The filter K is a learnable filter that can effectively preserve the user's complete information in the frequency domain. During the model optimization process, the learnable filter can adaptively represent any filter in the frequency domain. However, to prevent it from becoming overly biased towards representing a specific filter and thus ignoring some important frequency information, we obtain the residual information from the time domain and add it to the filtered output $\tilde{C}_L^l$ with a learnable scaling factor:

$$\tilde{C}_L^l = \tilde{C}_L^l + \alpha\left(\hat{C}_L^l - \tilde{C}_L^l\right) \tag{10}$$

where α is a learnable tensor. We acquire this residual information from the time domain instead of the frequency domain to achieve time-frequency synergy, which is why we term this frequency layer as the hybrid filter layer. The residual information is directly derived from the initial input of the filter, which helps mitigate the distortion caused by operations in the frequency domain and better preserves the original sequence information. We balance the global patterns in the frequency domain and the local correlations in the time domain through a weighted summation with the learnable weight α. Additionally, we apply the residual connection, layer normalization, and dropout.

3.2.3 Window Hybrid Filter

For the frequency domain layer of the WHF branch, we propose a Window Hybrid Filter. FFT is first applied along the item dimension to transform $\hat{C}_W^l$ into the frequency domain:

$$X_W^l = f\left(\hat{C}_W^l\right) \tag{11}$$

After obtaining the spectrum X_W^l of $\hat{C}_W^l$ through FFT, instead of performing frequency modulation across all dimensions like in the learnable filter, we only truncate the spectrum X_W^l along the item dimension to retain the frequency bands within a specified window:

$$\tilde{X}_W^l = W\left(X_W^l\right) \tag{12}$$

where $W(\cdot)$ represents the frequency band values retained within the spectral window (U, V), with the rest set to 0. U and V represent the lower and upper bounds of the frequency components to be preserved in the spectrum. We execute IFFT to convert the filtered frequency domain signal back to the time domain:

$$\tilde{C}_W^l = f^{-1}\left(\tilde{X}_W^l\right) \in \mathbb{R}^{n\times d} \tag{13}$$

We refer to this filtering method as the Window-Pass Filter (WPF). Depending on the selected strategy, WPF can represent a low-pass, high-pass, or band-pass filter in the frequency domain. WPF captures signals within a specified frequency band; however, since valuable information exists across all frequency bands, we aim to balance both low- and high-frequency signals and reduce information loss. To achieve this, we adopt residual information addition from a time-frequency hybrid perspective, the same as the approach used in the Learnable Hybrid Filter:

$$\tilde{C}_W^l = \tilde{C}_W^l + \beta\left(\hat{C}_W^l - \tilde{C}_W^l\right) \tag{14}$$

where β is a learnable tensor. Residual connection, layer normalization, and dropout are also applied. LHF uses a learnable filter K with dimensions matching those of the spectrum to modulate its global frequency components, while WHF primarily extracts local frequency components from the spectrum. We combine the outputs of LHF and WHF using the hyperparameter γ, where $0 < \gamma < 1$:

$$\tilde{C}_H^l = \gamma\tilde{C}_L^l + (1-\gamma)\,\tilde{C}_W^l \tag{15}$$

LHF leverages adaptive frequency-domain weighting to flexibly adjust to different preference patterns. It can effectively capture long-term preferences while also making adjustments to account for short-term fluctuations in user preferences to a certain extent. WHF, on the other hand, emphasizes specific frequency bands by adjusting the window position, thereby highlighting either long-term or short-term preferences. This proactive intervention, combined with its complementary role to LHF, enhances the model's flexibility and effectiveness in balancing short-term and long-term dependencies.

3.2.4 Point-Wise Feed-Forward Network

We use the GeLU activation function and linear transformations to form the point-wise feed-forward network, which introduces non-linearity to the model. This computation is defined as:

$$\tilde{C}^l = FFN\left(\tilde{C}_H^l\right) = \left(GeLU\left(\tilde{C}_H^l W_1 + b_1\right)\right)W_2 + b_2 \tag{16}$$

where W_1, W_2, b_1, b_2 are learnable parameters. Finally, we also apply residual connection, layer normalization, and dropout.

3.3 Prediction Layer

After extracting information from the user interaction sequence, we compute the relevance score of item i at step $(t+1)$:

$$\hat{y}_i = p\left(i_{t+1} = i | i_{1:t}\right) = e_i^T \tilde{C}_t^L \tag{17}$$

where e_i represents the embedding of item i in M_I, and $\tilde{C}_t^L$ represents the output of the L-layer HyTiF blocks at step t in the interaction history. We optimize the model parameters using the Cross-Entropy (CE) loss

function. The objective function of Sequential recommendation is formulated as:

$$L = -\log \frac{\exp\left(\hat{y}_{g\in|I|}\right)}{\sum_{i=1}^{|I|}\exp\left(\hat{y}_i\right)} \tag{18}$$

where $g \in |I|$ is the ground truth item, and $\hat{y}_{g\in|I|}$ is the model's predicted score for g.

4 Experiment

This section outlines our experimental setup and presents the empirical results. We design the experiments to investigate the following research questions:

RQ1: How does HyTiFRec perform compared to current state-of-the-art sequential recommendation methods?
RQ2: What is the influence of various component designs in HyTiFRec?
RQ3: How do different hyperparameters affect HyTiFRec's performance?
RQ4: Can the proposed method improve the model's ability to capture and balance users' long-term interests with short-term interest changes?

4.1 Experimental Setup

4.1.1 Dataset

We evaluate HyTiFRec on five real-world and widely used benchmark datasets. These datasets differ in scenario, size, and sparsity, and are commonly used for evaluating sequential recommendation methods. For all datasets, we exclude users and items with fewer than five interactions [28,29]. Table 1 presents the dataset statistics.

(a) **Beauty** and **Sports** are domain-specific datasets extracted from the Amazon Review dataset.
(b) **Yelp** is a business recommendation dataset from the Yelp open dataset.
(c) **ML-1M** is a movie rating dataset provided by MovieLens.
(d) **LastFM** is a dataset from the Last.fm music platform that contains user activity records.

Table 1: Statistics of the datasets after preprocessing

	# Users	# Items	# Interactions	Avg. length	Sparsity
Beauty	22,363	12,101	198,502	8.9	99.93%
Sports	25,598	18,357	296,337	8.3	99.95%
Yelp	30,431	20,033	316,354	10.4	99.95%
LastFM	1,090	3,646	52,551	48.2	98.68%
ML-1M	6,041	3,417	999,611	165.5	95.16%

4.1.2 Baselines

We compare HyTiFRec with the following baseline methods:

(a) GRU4Rec [4]: An RNN-based approach that leverages gated recurrent units (GRUs) for modeling user interaction sequences.
(b) SASRec [10]: A Transformer-based approach that employs self-attention to capture long-range dependencies within user behavior sequences.

(c) BERT4Rec [30]: A bidirectional Transformer-based method that leverages BERT [31] to capture bidirectional dependencies in user interaction sequences.
(d) FMLP-Rec [20]: A MLP-based method that uses filter-enhanced MLP to filter noise and learn user preferences.
(e) DuoRec [32]: A contrastive learning-based method that improves item embedding distribution through the uniformity property of contrastive learning.
(f) FEARec [18]: A Transformer-based method that combines attention mechanisms in the time and frequency domains to capture users' evolving preferences over time.
(g) BSARec [33]: A Transformer-based method that leverages the Fourier transform to capture inductive biases from frequency information and employs a frequency rescaler to address the issue of over-smoothing.

4.1.3 Implementation and Hyperparameter Settings

We implement all models using PyTorch. The other hyperparameters for each model are tuned based on recommendations from their respective papers. The experiments are conducted with the following hyperparameters: embedding size d is set to 64. Considering computational cost and the fact that users' excessively early preferences may not accurately reflect their current needs, the sequence length n is set to 50. For the upper limit V and lower limit U of the window in the WHF, U is set to 0, and V is selected from $\{1, 2, 3, 4, 5\}$, corresponding to the indices of the frequency components in the spectrum. To keep the model complexity and computational cost at an appropriate level, the number of HyTiF blocks L is tuned within $\{1, 2\}$. A lower learning rate significantly slows down the convergence speed, while a higher learning rate may lead to instability during the optimization process. Therefore, we select a moderate and commonly used range of learning rates, $\{5 \times 10^{-4}, 1 \times 10^{-3}\}$. For the number of heads in the multi-head attention mechanism, increasing the number of heads allows the model to extract diverse patterns from the input sequence. However, considering that small d is not suitable for decomposition into excessively small subspaces [10], we tune the number of heads h within $\{1, 2, 4\}$. The optimizer is the Adam optimizer, and the batch size is set to 256.

4.1.4 Evaluation Metrics

We use standard top-K ranking evaluation metrics, Hit Ratio@K (HR@K) and NDCG@K, to score the ranked lists of each method. HR@K is a recall-based metric that focuses on whether the items the user is interested in appear in the top K recommended items. NDCG@K is a ranking-based metric that focuses on the ranking of items in the recommendation list, providing a better evaluation of the ranking quality of the recommendation system. By default, K is set to 5, 10, and 20. As suggested by [34,35], we evaluate model performance using a full ranking approach, where rankings are computed across the entire item set without negative sampling.

4.2 Recommendation Performance (RQ1)

To demonstrate the sequential recommendation performance of HyTiFRec, we conduct a comparison with several state-of-the-art methods. Table 2 shows the results of different methods across five datasets. On all five datasets, Transformer-based methods outperformed CNN-based Caser and RNN-based GRU4Rec, demonstrating the efficiency of the Transformer architecture and self-attention mechanism in modeling user interaction sequences for sequential recommendation tasks. FMLP-Rec removes the self-attention component from Transformer and enhances MLP with a learnable filter, achieving better performance than SASRec across most datasets, indicating the research potential of frequency domain methods in sequential

recommendation. The performance of FEARec, which also utilizes frequency domain methods, further supports this conclusion. Furthermore, DuoRec and FEARec, which incorporate contrastive learning, significantly outperform methods that do not use contrastive learning, validating that contrastive learning is useful for improving performance. BSARec, which utilizes self-attention mechanisms and a frequency rescaler, outperforms other baselines in overall performance.

Table 2: Performance comparison of various methods across five datasets. The highest evaluation score in each row is highlighted in bold, and the second-best score is underlined. 'Improv.' indicates the relative improvement over the best baseline performance

Datasets	Metric	GRU4Rec	SASRec	BERT4Rec	FMLP-Rec	DuoRec	FEARec	BSARec	HyTiFRec	Improv.
Beauty	HR@5	0.0169	0.0283	0.0438	0.0342	0.0694	0.0669	0.0705	**0.0732**	3.82%
	NDCG@5	0.0103	0.0184	0.0285	0.0219	0.0498	0.0477	0.0504	**0.0517**	2.57%
	HR@10	0.0294	0.0456	0.0694	0.0561	0.0955	0.0963	0.0986	**0.1009**	2.33%
	NDCG@10	0.0144	0.0240	0.0368	0.0289	0.0582	0.0572	0.0595	**0.0606**	1.84%
	HR@20	0.0466	0.0733	0.1036	0.0897	0.1318	0.1338	0.1342	**0.1384**	3.12%
	NDCG@20	0.0187	0.0310	0.0454	0.0374	0.0673	0.0666	0.0684	**0.0701**	2.48%
Sports	HR@5	0.0105	0.0159	0.0265	0.0186	0.0388	0.0402	0.0404	**0.0421**	4.20%
	NDCG@5	0.0067	0.0108	0.0170	0.0120	0.0275	0.0276	0.0281	**0.0289**	2.84%
	HR@10	0.0181	0.0263	0.0441	0.0293	0.0558	0.0569	0.0575	**0.0597**	3.82%
	NDCG@10	0.0091	0.0142	0.0216	0.0155	0.0329	0.0330	0.0336	**0.0346**	2.97%
	HR@20	0.0291	0.0417	0.0641	0.0457	0.0798	0.0815	0.0821	**0.0841**	2.43%
	NDCG@20	0.0119	0.0180	0.0274	0.0196	0.0389	0.0392	0.0398	**0.0407**	2.26%
Yelp	HR@5	0.0133	0.0150	0.0234	0.0175	0.0260	0.0267	0.0255	**0.0272**	1.87%
	NDCG@5	0.0085	0.0092	0.0147	0.0108	0.0164	0.0169	0.0158	**0.0171**	1.18%
	HR@10	0.0221	0.0256	0.0396	0.0301	0.0438	0.0440	0.0436	**0.0462**	5.00%
	NDCG@10	0.0113	0.0126	0.0200	0.0149	0.0221	0.0224	0.0217	**0.0232**	3.57%
	HR@20	0.0382	0.0422	0.0656	0.0494	0.0714	0.0702	0.0716	**0.0750**	4.74%
	NDCG@20	0.0153	0.0168	0.0265	0.0197	0.0290	0.0290	0.0287	**0.0304**	4.82%
LastFM	HR@5	0.0229	0.0367	0.0330	0.0376	0.0440	0.0422	0.0560	**0.0587**	4.82%
	NDCG@5	0.0159	0.0258	0.0211	0.0256	0.0309	0.0284	0.0376	**0.0384**	2.12%
	HR@10	0.0349	0.0606	0.0541	0.0596	0.0606	0.0606	0.0761	**0.0771**	1.31%
	NDCG@10	0.0197	0.0334	0.0280	0.0326	0.0361	0.0344	0.0441	**0.0442**	0.02%
	HR@20	0.0495	0.0872	0.0835	0.0899	0.0890	0.0881	0.1073	**0.1193**	1.11%
	NDCG@20	0.0233	0.0401	0.0354	0.0402	0.0433	0.0413	0.0519	**0.0548**	5.58%
ML-1M	HR@5	0.1055	0.1414	0.1422	0.1288	0.1891	0.1816	0.1911	**0.2043**	6.90%
	NDCG@5	0.0674	0.0912	0.0935	0.0834	0.1264	0.1212	0.1268	**0.1382**	8.99%
	HR@10	0.1806	0.2192	0.2187	0.2086	0.2699	0.2651	0.2720	**0.2889**	6.21%
	NDCG@10	0.0917	0.1162	0.1180	0.1091	0.1524	0.1480	0.1527	**0.1654**	8.31%
	HR@20	0.2815	0.3182	0.3230	0.3257	0.3733	0.3677	0.3806	**0.3982**	4.62%
	NDCG@20	0.1168	0.1412	0.1443	0.1384	0.1785	0.1738	0.1802	**0.1930**	7.10%

Our proposed HyTiFRec is a novel transformer architecture that leverages the synergy of time domain and frequency domain methods. It outperforms SASRec, which uses only time domain attention; FMLP-Rec, which employs only frequency domain filters; and FEARec, which incorporates both time domain and frequency domain attention along with contrastive learning. Our method demonstrates relatively larger improvements compared to most baselines on datasets with lower sparsity and longer interaction sequence lengths (i.e., LastFM and ML-1M), indicating that HyTiFRec is more effective in capturing complex temporal patterns. On datasets with higher sparsity and shorter sequence lengths (i.e., Beauty, Sports, and Yelp), it shows certain improvements. Through the learnable frequency filtering, HyTiFRec can weigh information from different frequencies, effectively suppressing noise and enhancing the accuracy of user preference modeling.

4.3 Ablation Studies (RQ2)

To investigate the effectiveness of HyTiFRec's components, we conduct several ablation studies. The ablation studies examine HyTiFRec and its four variants across three datasets. The four variants are derived from HyTiFRec by removing the frequency domain layer LHF (w/o LHF), removing the frequency domain layer WHF (w/o WHF), removing the time domain layer EA (w/o EA), and removing the operation of adding time domain residual information in the frequency domain layers (w/o RIAdd).

Table 3 presents the evaluation results. When the time domain layer is removed, that is, when the EA component is removed, the model's performance decreases. This suggests that the EA is used to extract preliminary features and provide global information, which provides a solid foundation for subsequent filtering and integration of information using frequency domain methods. Removing either the LHF or WHF module also leads to performance degradation. The LHF globally modulates the spectrum, and the WHF retains local frequency components from the spectrum for integration. These two different frequency domain methods can effectively extract and integrate frequency domain information. Both the time domain and frequency domain layers are important, reflecting the potential of combining time domain and frequency domain methods. When the addition of time domain residual information is removed, the model's performance drops significantly. This suggests that adding time domain residual information to the output after frequency domain filtering helps reduce the loss of important information.

Table 3: Ablation studies of HyTiFRec with HR@5 and NDCG@5 metrics across ML-1M, Yelp and Beauty datasets. The best results are in bold

Methods	ML-1M		Yelp		Beauty	
	HR@5	NDCG@5	HR@5	NDCG@5	HR@5	NDCG@5
HyTiFRec	**0.2043**	**0.1382**	**0.0272**	**0.0171**	**0.0732**	**0.0517**
W/O EA	0.1904	0.1278	0.0262	0.0160	0.0716	0.0508
W/O LHF	0.1974	0.1343	0.0260	0.0161	0.0704	0.0495
W/O WHF	0.1962	0.1325	0.0255	0.0162	0.0695	0.0495
W/O RIAdd	0.1937	0.1296	0.0253	0.0160	0.0681	0.0482

4.4 Hyperparameter Studies (RQ3)

In this section, we investigate the effect of two key hyperparameters on HyTiFRec: the balance coefficient γ for the dual-branch structure and the upper truncation limit V in WHF. When analyzing each of these hyperparameters, all other hyperparameters are kept at their optimal settings.

4.4.1 Balance Coefficient γ

As shown in Fig. 3, HyTiFRec achieves the best performance on the ML-1M and Beauty datasets when $\gamma = 0.3$. On the Yelp dataset, HyTiFRec performs well at $\gamma = 0.5$ and $\gamma = 0.7$, but it receives the lowest evaluation score on the ML-1M dataset when $\gamma = 0.7$. These results suggest that the two branches in the HyTiF block contribute differently across datasets, highlighting their respective importance. Identifying the optimal balance between the two can enhance HyTiFRec's overall performance. Among the three datasets, Yelp is the most sparse, followed by Beauty and ML-1M. When assigning an appropriately higher weight to the LHF branch, the model performance slightly improves on Yelp, slightly declines on Beauty, and drops more significantly on ML-1M. We believe this is because the noise attenuation capability of the LHF branch is crucial for highly sparse datasets like Yelp. However, for less sparse datasets, the contribution of LHF

diminishes, and assigning it a greater weight may lead to performance degradation. To some extent, adjusting the weight balance between the LHF and WHF branches based on the sparsity of the dataset can effectively improve the overall performance of HyTiFRec.

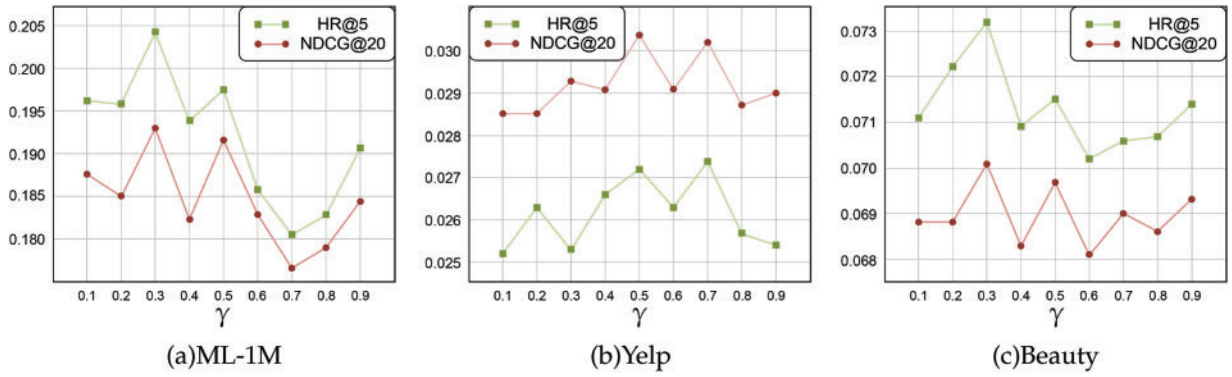

Figure 3: Performance of HyTiFRec on three datasets with different balance coefficients

4.4.2 Upper Truncation Limit V

As shown in Fig. 4, the models with $V = 1$ and $V = 5$ perform poorly across the three datasets, while the model with $V = 3$ achieves the best performance. This result indicates that the information retained by WHF in the frequency domain should be moderate; simply increasing the WHF window size to capture more frequency bands does not necessarily yield better performance. Instead, an excessively large window may cause the model to lose balance in focusing on users' long-term and short-term preferences, and similarly, retaining too little information has a similar effect on performance.

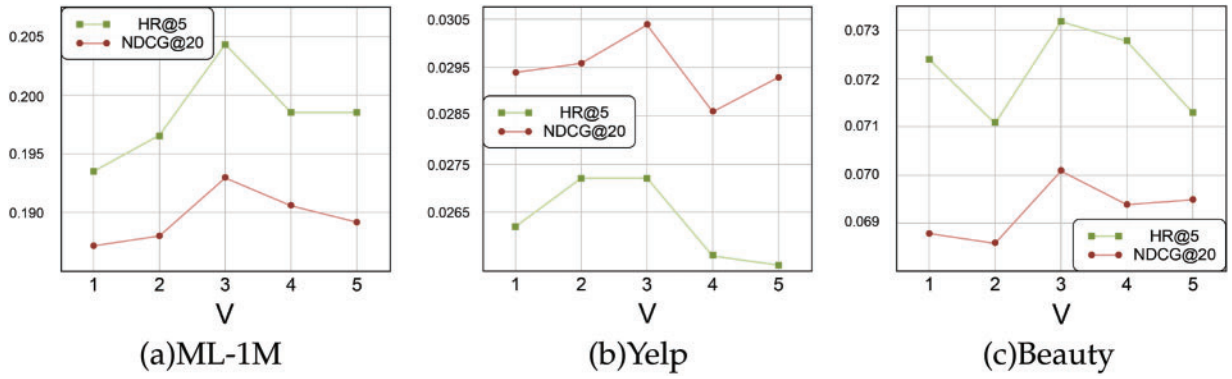

Figure 4: Performance of HyTiFRec w.r.t different upper truncation limit on three datasets

4.5 Performance Comparison Based on Transition Frequency (RQ4)

We evaluate HyTiFRec and SASRec on test samples with varying item transition frequencies using the NDCG@20 metric. The item transition frequency of each sample sequence is calculated based on the transition from the validation item (the second-to-last item) to the test item (the last item) in the sequence. Fig. 5 presents the evaluation results on two datasets. Whether on test samples with zero item transition frequency or those with high item transition frequency, HyTiFRec consistently outperforms SASRec. This suggests that HyTiFRec, compared to SASRec, has a deeper grasp of short-term changes in

user interests without reducing its ability to model long-term user preferences. By better balancing users' short-term and long-term preferences, HyTiFRec achieves higher prediction accuracy.

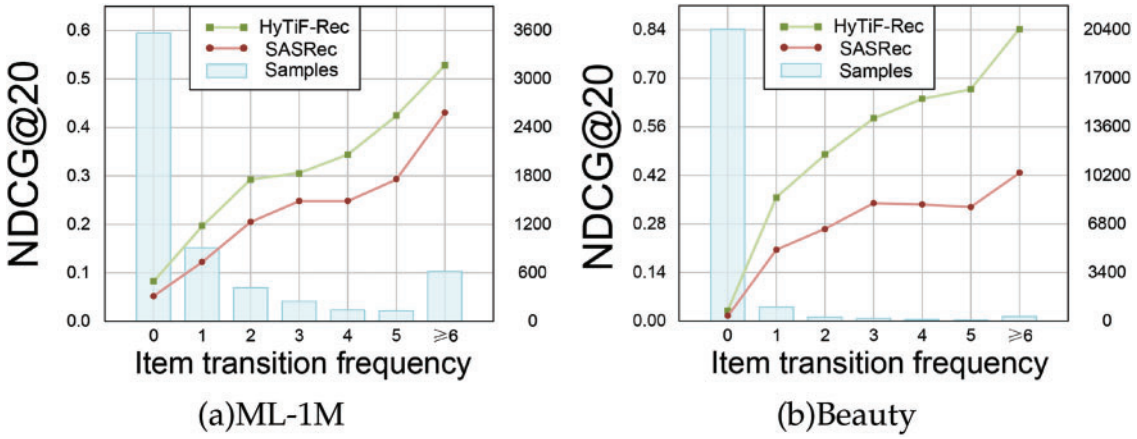

Figure 5: Performance of two methods w.r.t item transition frequency on two datasets

5 Related Work

5.1 Sequential Recommendation

sequential recommendation methods aim to predict users' future preferences based on their historical interaction sequences [36]. Early sequential recommendation approaches were primarily based on Markov Chains. With the development of deep learning, research in SRSs has increasingly focused on deep learning methods. GRU4Rec [4] employs Gated Recurrent Units (GRU). Caser [6] utilizes horizontal and vertical convolutional filters. Models like SASRec [10] and BERT4Rec [30] adopt unidirectional and bidirectional encoders, respectively, to capture contextual information from user interaction sequences. TiSASRec [28] uses a novel self-attention to explicitly capture the timestamps of items in a sequence. These self-attention models focus more on long-term preference modeling. HAFLS [37] combines a hierarchical attention structure with a multi-head self-attention mechanism and adopts a joint learning mechanism to integrate users' current interests and general preferences. DIARec [17] introduces a time-aware attention mechanism to model interest shifts and employs GRU to extract sequential selection patterns in short-term interaction items. However, these studies predominantly focus on the time domain, overlooking the exploration of frequency domain methods.

5.2 Frequency Domain Learning

There are many works related to frequency domain methods in the fields of natural language processing [38,39] and computer vision [40,41]. Recently, sequential recommendation has also started to recognize the effectiveness of frequency domain methods. FMLP-Rec [20] is the first to introduce frequency filter into sequential recommendation to address the noise issue in the data, proposing a fully MLP architecture with a learnable filter. To tackle the problem of self-attention underrate high frequency signals [18,42], leverage frequency domain methods and contrastive learning to improve the model's ability to capture high-frequency information and enhance overall performance. BSARec [33] injects inductive bias by considering fine-grained sequential patterns and addresses the oversmoothing issue by utilizing a frequency rescaler. Our proposed HyTiFRec replaces the self-attention layer in the Transformer module by serially combining

LHF and WHF with EA. It improves the model's flexibility and the ability to balance high-frequency and low-frequency signals through adaptive filtering and emphasizing specific frequency components.

6 Conclusion and Future Work

In this paper, to address the limitations of many sequential recommendation methods based on self-attention mechanisms in balancing users' long-term and short-term interests, we design a novel Transformer-based model, HyTiFRec. Instead of the self-attention component in the Transformer, we employ a hybrid time-frequency dual-branch structure. HyTiFRec utilizes efficient attention to extract initial features and global information. We propose a learnable hybrid filter module and a window hybrid filter module to perform frequency tuning for full-band frequency domain signals and to extract local frequency components from frequency domain signals, respectively, ensuring a balance between components across different frequency bands. After filtering in the filter module, time domain residual information is incorporated to mitigate information loss during the filtering process. Our model demonstrates superior sequential recommendation performance compared to seven baseline methods across five datasets, effectively improving the modeling of users' long-term preferences while enhancing sensitivity to short-term preference changes. Our method has certain limitations. The adjustable filter window size and position in WHF make the model more flexible but also introduce a significant tuning workload. Additionally, its filtering method is relatively simple and coarse, lacking a more refined and effective utilization of frequency components. In response to these limitations, in future work, we will conduct a more in-depth study on frequency-domain methods, particularly in aspects such as the partitioning and sampling of frequency components.

Acknowledgement: The authors appreciate the support of the Natural Science Foundation of Zhejiang Province.

Funding Statement: This work was supported by a grant from the Natural Science Foundation of Zhejiang Province under Grant LY21F010016.

Author Contributions: The authors confirm contribution to the paper as follows: Conceptualization, methodology and validation, Dawei Qiu; investigation and data curation, Xiaoming Zhang; writing—original draft preparation, Dawei Qiu; writing—review and editing, Peng Wu; supervision, Renjie Xu; project administration and funding acquisition, Peng Wu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data used in this work come mainly from public datasets. Specifically: Beauty and Sports datasets are accessible at https://cseweb.ucsd.edu/~jmcauley/datasets.html (accessed on 11 May 2024); The Yelp dataset can be downloaded from https://www.yelp.com/dataset (accessed on 11 May 2024); The ML-1M dataset is available through https://grouplens.org/datasets/movielens/1m/ (accessed on 11 May 2024); The LastFM dataset can be obtained from https://grouplens.org/datasets/hetrec-2011/ (accessed on 11 May 2024).

Ethics Approval: The user data used in this study were sourced from publicly available datasets and comply with relevant privacy protection regulations.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

References

1. Resnick P, Iacovou N, Suchak M, Bergstrom P, Riedl J. Grouplens: an open architecture for collaborative filtering of netnews. In: Proceedings of the 1994 ACM Conference on Computer Supported Cooperative Work; Chapel Hill, NC, USA. New York, NY, USA: Association for Computing Machinery; 1994. p. 175–86.
2. Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems. Computer. 2009;42(8):30–7. doi:10.1109/MC.2009.263.

3. Rendle S, Freudenthaler C, Schmidt-Thieme L. Factorizing personalized markov chains for next-basket recommendation. In: Proceedings of the 19th International Conference on World Wide Web; Raleigh, NC, USA. New York, NY, USA: Association for Computing Machinery; 2010. p. 811–20.
4. Hidasi B. Session-based recommendations with recurrent neural networks. arXiv:151106939. 2015.
5. Hidasi B, Karatzoglou A. Recurrent neural networks with top-k gains for session-based recommendations. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management; Torino, Italy. New York, NY, USA: Association for Computing Machinery; 2018. p. 843–52.
6. Tang J, Wang K. Personalized top-n sequential recommendation via convolutional sequence embedding. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining; Marina Del Rey, CA, USA. New York, NY, USA: Association for Computing Machinery; 2018. p. 565–73.
7. Yuan F, Karatzoglou A, Arapakis I, Jose JM, He X. A simple convolutional generative network for next item recommendation. In: Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining; Melbourne VIC, Australia. New York, NY, USA: Association for Computing Machinery; 2019. p. 582–90.
8. Chen X, Xu H, Zhang Y, Tang J, Cao Y, Qin Z, et al. Sequential recommendation with user memory networks. In: Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining; Marina Del Rey, CA, USA. New York, NY, USA: Association for Computing Machinery; 2018. p. 108–16.
9. Li J, Ren P, Chen Z, Ren Z, Lian T, Ma J. Neural attentive session-based recommendation. In: Proceedings of the 2017 ACM on Conference on Information and Knowledge Management; Singapore. New York, NY, USA: Association for Computing Machinery; 2017. p. 1419–28.
10. Kang WC, McAuley J. Self-attentive sequential recommendation. In: 2018 IEEE International Conference on Data Mining (ICDM); 2018; Singapore: IEEE. p. 197–206.
11. Zhou K, Wang H, Zhao WX, Zhu Y, Wang S, Zhang F, et al. S3-Rec: self-supervised learning for sequential recommendation with mutual information maximization. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management; Ireland. New York, NY, USA: Association for Computing Machinery; 2020. p. 1893–902.
12. Raghu M, Unterthiner T, Kornblith S, Zhang C, Dosovitskiy A. Do vision transformers see like convolutional neural networks? Adv Neural Inf Process Syst. 2021;34:12116–28.
13. Park N, Kim S. How do vision transformers work? arXiv:220206709. 2022.
14. Jannach D, Lerche L, Jugovac M. Adaptation and evaluation of recommendations for short-term shopping goals. In: Proceedings of the 9th ACM Conference on Recommender Systems; Vienna, Austria. New York, NY, USA: Association for Computing Machinery; 2015. p. 211–8.
15. He Z, Zhao H, Lin Z, Wang Z, Kale A, McAuley J. Locker: locally constrained self-attentive sequential recommendation. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management; Virtual Event, Queensland, Australia. New York, NY, USA: Association for Computing Machinery; 2021. p. 3088–92.
16. Zang T, Zhu Y, Zhu J, Xu Y, Liu H. MPAN: multi-parallel attention network for session-based recommendation. Neurocomputing. 2022;471(1):230–41. doi:10.1016/j.neucom.2021.11.030.
17. Vaghari H, Aghdam MH, Emami H. Diarec: dynamic intention-aware recommendation with attention-based context-aware item attributes modeling. J Artif Intell Soft Comput Res. 2024;14(2):171–89. doi:10.2478/jaiscr-2024-0010.
18. Du X, Yuan H, Zhao P, Qu J, Zhuang F, Liu G, et al. Frequency enhanced hybrid attention network for sequential recommendation. In: Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval; Taipei, Taiwan. New York, NY, USA: Association for Computing Machinery; 2023. p. 78–88.
19. Shen Z, Zhang M, Zhao H, Yi S, Li H. Efficient attention: attention with linear complexities. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision; Waikoloa, HI, USA; 2021. p. 3531–9.
20. Zhou K, Yu H, Zhao WX, Wen JR. Filter-enhanced MLP is all you need for sequential recommendation. In: Proceedings of the ACM Web Conference 2022; 2022. p. 2388–99.

21. Rao Y, Zhao W, Zhu Z, Lu J, Zhou J. Global filter networks for image classification. Adv Neural Inf Proc Syst. 2021;34:980–93.
22. Trick T. Theory and application of digital signal processing. IEEE Trans Acoust, Speech, Signal Process. 1975;23(4):394–5. doi:10.1109/TASSP.1975.1162708.
23. Soliman SS, Srinath MD. Continuous and discrete signals and systems. 2nd ed. Englewood Cliffs, NJ, USA: Pearson; 1990.
24. Cooley JW, Tukey JW. An algorithm for the machine calculation of complex Fourier series. Math Comput. 1965;19(90):297–301. doi:10.1090/S0025-5718-1965-0178586-1.
25. Frigo M, Johnson SG. The design and implementation of FFTW3. Proc IEEE. 2005;93(2):216–31. doi:10.1109/JPROC.2004.840301.
26. Ba JL. Layer normalization. arXiv:160706450. 2016.
27. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res. 2014;15(1):1929–58.
28. Li J, Wang Y, McAuley J. Time interval aware self-attention for sequential recommendation. In: Proceedings of the 13th International Conference on Web Search and Data Mining; Houston, TX, USA. New York, NY, USA: Association for Computing Machinery; 2020. p. 322–30.
29. Hidasi B, Quadrana M, Karatzoglou A, Tikk D. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In: Proceedings of the 10th ACM Conference on Recommender Systems; Boston, MA, USA. New York, NY, USA: Association for Computing Machinery; 2016. p. 241–8.
30. Sun F, Liu J, Wu J, Pei C, Lin X, Ou W, et al. BERT4Rec: sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management; Beijing, China. New York, NY, USA: Association for Computing Machinery; 2019. p. 1441–50.
31. Devlin J. Bert: pre-training of deep bidirectional transformers for language understanding. arXiv:181004805. 2018.
32. Qiu R, Huang Z, Yin H, Wang Z. Contrastive learning for representation degeneration problem in sequential recommendation. In: Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining; Virtual Event, AZ, USA. New York, NY, USA: Association for Computing Machinery; 2022. p. 813–23.
33. Shin Y, Choi J, Wi H, Park N. An attentive inductive bias for sequential recommendation beyond the self-attention. Proc AAAI Conf Artif Intell. 2024;38(8):8984–92. doi:10.1609/aaai.v38i8.28747.
34. Krichene W, Rendle S. On sampled metrics for item recommendation. In: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining; AZ, USA. New York, NY, USA: Association for Computing Machinery; 2020. p. 1748–57.
35. Dallmann A, Zoller D, Hotho A. A case study on sampling strategies for evaluating neural sequential item recommendation models. In: Proceedings of the 15th ACM Conference on Recommender Systems; Amsterdam, Netherlands. New York, NY, USA: Association for Computing Machinery; 2021. p. 505–14.
36. Wang S, Hu L, Wang Y, Cao L, Sheng QZ, Orgun M. Sequential recommender systems: challenges, progress and prospects. arXiv:200104830. 2019.
37. Du Y, Peng Z, Niu J, Yan J. A unified hierarchical attention framework for sequential recommendation by fusing long and short-term preferences. Expert Syst Appl. 2022;201(8):117102. doi:10.1016/j.eswa.2022.117102.
38. Tamkin A, Jurafsky D, Goodman N. Language through a prism: a spectral approach for multiscale language representations. In: Proceedings of the 34th International Conference on Neural Information Processing Systems; Vancouver, BC, Canada. Red Hook, NY, USA: Curran Associates Inc.; 2020. p. 5492–504.
39. Lee-Thorp J, Ainslie J, Eckstein I, Ontanon S. FNet: mixing tokens with fourier transforms. arXiv:210503824. 2021.
40. Xu K, Qin M, Sun F, Wang Y, Chen YK, Ren F. Learning in the frequency domain. arXiv:2002.12416. 2020.

41. Suvorov R, Logacheva E, Mashikhin A, Remizova A, Ashukha A, Silvestrov A, et al. Resolution-robust large mask inpainting with fourier convolutions. In: 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV); 2021. p. 3172–82. Available from: https://api.semanticscholar.org/CorpusID:237513361.
42. Du X, Yuan H, Zhao P, Fang J, Liu G, Liu Y, et al. Contrastive enhanced slide filter mixer for sequential recommendation. In: 2023 IEEE 39th International Conference on Data Engineering (ICDE); 2023, Anaheim, CA, USA: IEEE. p. 2673–85.