

# DHPRec: Bridging Immediate Intent and Denoised Historical Patterns for Sequential Recommendation

Anonymous Author(s)

## Abstract

Sequential Recommender Systems aim to capture dynamic user preferences by mining historical dependencies. However, existing methods typically adopt a truncation strategy that focuses mainly on the most recent interactions. From a cognitive psychology perspective, real-world user behavior is jointly driven by immediate intent and accumulated historical habits. The truncation strategy ignores information in the dimension of historical preferences and is limited entirely to recent information. A straightforward solution is to feed the user's entire historical sequence into the model. However, as the sequence length increases, the attention weights for each item are diluted. The dilution confronts two critical challenges: amplified noise interference and the inability to focus on specific regularities. To address these issues, we propose **DHPRec** (Denoised Historical Patterns for Sequential Recommendation), a novel framework designed to bridge immediate intent and specific long-term patterns. DHPRec introduces a Frequency-Domain Feature Refinement mechanism to filter high-frequency noise and extract stable long-term representations. Furthermore, we design a Pattern-Based Anchor-Guided Fusion mechanism, which extracts pattern units representing long-term regularities and utilizes an anchor to effectively extract the global specific historical preferences. Finally, a learnable gating mechanism is employed to balance the contribution of the immediate intent and the specific historical preferences. Extensive experiments on four real-world datasets demonstrate that DHPRec significantly outperforms state-of-the-art baselines, achieving an average relative improvement of 8.1% across all metrics, with gains reaching up to 14.6% in NDCG@10 on the Scientific dataset. Our code is available at <https://anonymous.4open.science/r/DHPRec-5F35>.

## CCS Concepts

• Information systems → Recommender systems.

## Keywords

Sequential Recommendation, Frequency-domain Denoising, Long-term User Preferences, User Intent Modeling

## 1 Introduction

Sequential Recommender Systems (SRS) aim to predict the next item of interest based on a user's time-ordered interaction records. The core advantage of sequential recommendation lies in its ability to model the dynamic evolution of human historical behavior. In recent years, with the explosive growth of user interaction data in online apps, user behavior sequences have become increasingly long and contain richer information. How to effectively utilize this growing historical behavior data has become a focal point in the field of sequential recommendation [4, 11, 16].

In the existing literature, current work in sequential recommendation can be mainly categorized into two dimensions. The first is

the representation learning of sequence items. Early research [5, 27] primarily relied on item identifiers (IDs) to model user behavior sequences, using historical ID sequences to predict the next item. Considering that simple ID representations lack the semantic information of the items themselves, current research [17, 19, 20, 33, 35] constructs rich semantic representations for sequence modeling to fully mine the deep user interests, employing encoders such as Pre-trained Language Models (PLMs) [3]. The second is the architecture design of sequence modeling. Obtaining high-quality representations, researchers have designed various architectures to capture dependencies in sequences from different angles, including Convolutional Neural Networks (CNN) [29, 32], Recurrent Neural Networks (RNN) [8, 28], and the currently mainstream Transformer [14, 27, 44] architecture.

Although these methods have made significant progress in modeling sequence dependencies, they typically adopt a *truncation strategy* when handling long sequences [18, 36, 43] (e.g., using only the recent 20 interactions). This strategy implicitly assumes that a user's next action is only influenced by their most recent interactions. However, from the perspective of cognitive psychology [13, 24, 26, 30], human decision-making behavior is typically driven jointly by two components: immediate significant information (*immediate intent*) and long-term accumulated stable regularities (*historical preferences*). The truncation strategy can effectively capture recent significant information, which is only one part of user decision-making. Only having immediate intent is not enough. For example, a user might have frequently purchased books during final exams. As the exams end and Christmas approaches, their recommendation results should largely depend on the immediate situation of the coming festival and the specific purchasing preferences formed during previous Christmas periods. They should not be driven by the recent exam-related consumption. Obviously, the truncation strategy ignores information in the dimension of historical preferences and is limited entirely to recent information.

A straightforward solution to gain historical preferences is to feed the user's entire historical sequence into the model. However, as the sequence length increases, the attention weights for each item are diluted across the vast history [1]. This dilution confronts two critical challenges: (1) The first is the *noise interference amplified by attention dilution*. Even if stochastic noise (e.g., accidental clicks) remains sparse across the entire sequence, as the sequence length grows, the dilution of attention amplifies the relative interference of noise. This amplification makes it challenging for the model to distinguish sparse positive signals from abundant random noise. (2) The second is the *inability to focus on specific regularities caused by attention dilution*. In long sequences, the attention weights are diluted, assigning very small weights to each item. This leads to an over-smoothed sequence representation, making the model unable to sharply focus on the specific historical regularities relevant to the current intent.

To address the two challenges, we propose **DHPRec** (**D**enoised **H**istorical **P**atterns for Sequential **R**ecommendation), a novel framework designed to bridge immediate intent and specific long-term patterns. (1) To address the first challenge, we design a *Frequency-Domain Feature Refinement mechanism*. We transform long sequences into the frequency domain to decompose them into high-frequency and low-frequency components. We subsequently design an *energy-aware filtering mechanism* to effectively reduce high-frequency noise while highlighting low-frequency signals that represent specific long-term regularities. (2) To address the second challenge, we design a *Pattern-Based Anchor-Guided Fusion mechanism*. We partition the sequence into slices based on time intervals, aggregating items within each slice and modeling the evolution between slices. This approach transforms the long and sparse item sequence into a short and dense sequence of *pattern units*, which represent long-term historical regularities. Subsequently, we treat the user’s most recent pattern as an *anchor*. This anchor aims to extract specific historical preferences from these pattern units. Through this method the global specific historical preferences match the current intent (the anchor). Finally, a learnable gating mechanism dynamically balances the contribution of the immediate intent and the specific historical preferences.

Extensive experiments on four real-world datasets demonstrate that DHPRec significantly outperforms state-of-the-art baselines

In summary, the main contributions of this paper are as follows:

- We are the first to propose a decision mechanism combining immediate intent with specific long-term regularities in SRS.

- We propose the Frequency-Domain Feature Refinement mechanism, which efficiently suppresses high-frequency noise and amplifies low-frequency regularities.

- We propose the Pattern-Based Anchor-Guided Fusion mechanism, which extracts pattern units representing long-term regularities and utilizes an anchor to effectively extract the global specific historical preferences.

- DHPRec achieves an average relative improvement of 8.1% across all metrics on four real-world datasets, with gains reaching up to 14.6% in NDCG@10 on the Scientific dataset, validating the effectiveness of the proposed method.

## 2 Preliminaries

This section formulates the sequential recommendation problem.

**Definition 1** (User, Item, and Attributes). We define the user set as  $\mathcal{U} = \{u_1, u_2, \dots, u_{|\mathcal{U}|}\}$  and the item set as  $\mathcal{V} = \{v_1, v_2, \dots, v_{|\mathcal{V}|}\}$ . Each item  $v \in \mathcal{V}$  is associated with a unique ID and a set of textual attributes (e.g., title, brand, category, etc.). We adopt an encoder to encode each item’s textual attributes into a representation vector  $z \in \mathbb{R}^d$ , where  $d$  denotes the embedding dimension. Consequently, the item representation set is defined as  $\mathcal{Z} = \{z_1, z_2, \dots, z_{|\mathcal{V}|}\}$ , where each item  $v_i \in \mathcal{V}$  is associated with a unique representation  $z_i$ . In the paper, we adopt a multi-view semantic encoder based on Vector Quantization [18].

**Definition 2** (Interaction Sequence). For a given user, the interaction sequence is defined as  $\mathcal{S} = \{s_1, \dots, s_l, \dots, s_L\}$ , where  $L$  represents the sequence length. Each interaction  $s_l = (z_l, t_l)$  consists of the item’s representation  $z_l$  and the interaction timestamp  $t_l$ .

**Definition 3** (Task). The goal of sequential recommendation is to predict the item  $v_{L+1}$  that the user is most likely to interact with at step  $L + 1$  given the history  $\mathcal{S}$ , formulated as  $\max P(v_{L+1} | \{s_1, \dots, s_l, \dots, s_L\})$

## 3 Methodology

In this section, we present the technical details of the DHPRec framework. As illustrated in Figure 1, the architecture consists of three bottom-up components, which are jointly designed to bridge immediate intent and long-term patterns by extracting stable regularities from noisy long sequences.

### 3.1 Frequency-Domain Feature Refinement

To address the *noise interference amplified by attention dilution*, we propose the Frequency-Domain Feature Refinement (FDFR) mechanism. This module is designed to filter out stochastic noise. It consists of three integral parts: Compact Spectral Transformation, Energy-Aware Filter Generation, and Frequency-Weighted Regularization.

**Compact Spectral Transformation.** Given the user’s historical interaction sequence  $\mathcal{S} = \{s_1, \dots, s_l, \dots, s_L\}$ , we obtain the sequence representation  $Z = [z_1, \dots, z_l, \dots, z_L] \in \mathbb{R}^{L \times d}$ . To explicitly separate noise, we transform the representation sequence  $Z$  into the frequency domain. Utilizing the Real-valued Fast Fourier Transform (rFFT, defined in Appendix A) [7, 31], we obtain the spectrum sequence denoted as  $\hat{Z} = [\hat{z}_0, \dots, \hat{z}_{l'}, \dots, \hat{z}_{L'}] \in \mathbb{C}^{(L'+1) \times d}$ , where  $L' = \lfloor L/2 \rfloor$ . The transformation is formally defined as:

$$[\hat{z}_0, \dots, \hat{z}_{l'}, \dots, \hat{z}_{L'}] = \text{rFFT}(Z). \quad (1)$$

To identify valuable parts hidden in the history, we quantify the importance of each frequency component. In the frequency domain, each component  $\hat{z}_{l'}$  is a complex vector composed of a real part and an imaginary part, formulated as  $\hat{z}_{l'} = \text{Re}(\hat{z}_{l'}) + i \cdot \text{Im}(\hat{z}_{l'})$ , where  $i$  is the imaginary unit. Its magnitude is computed as:

$$m_{l'} = \sqrt{\text{Re}(\hat{z}_{l'})^2 + \text{Im}(\hat{z}_{l'})^2}, \quad (2)$$

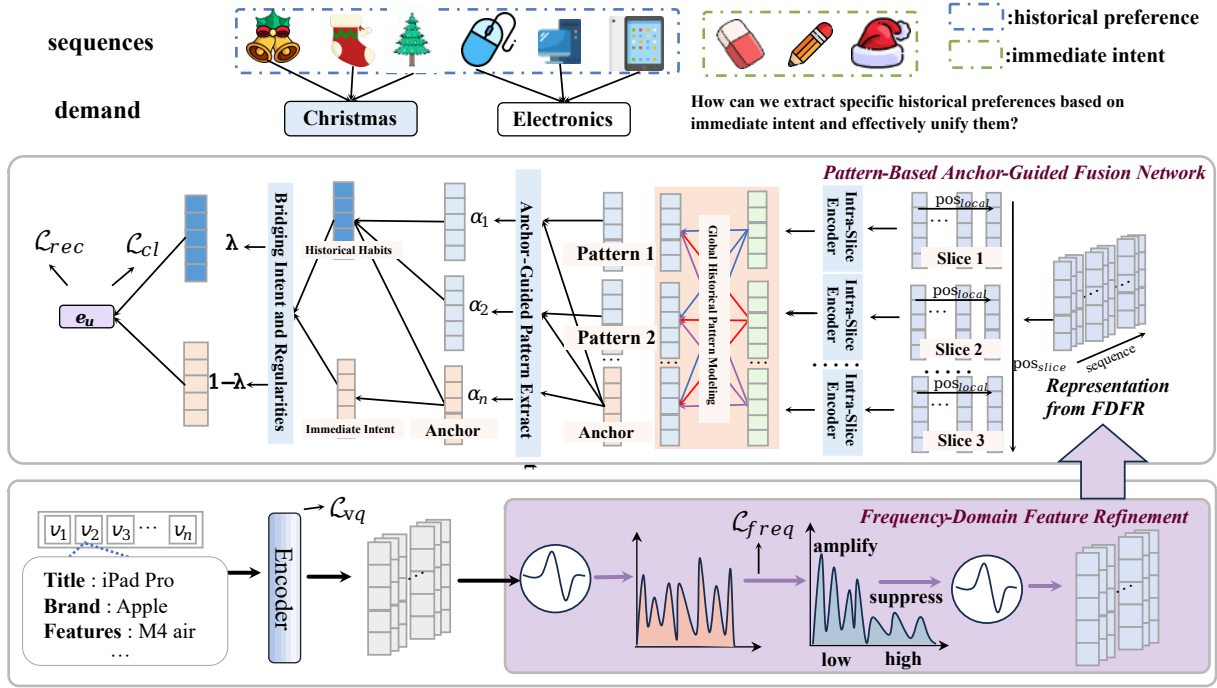
where  $M = [m_0, \dots, m_{l'}, \dots, m_{L'}] \in \mathbb{R}^{(L'+1) \times d}$ .

The decomposition separates the interaction history into distinct frequency components indexed from  $l' = 0$  to  $L'$ . In the context of user behavior, indices near  $l' = 0$  (low-frequency) correspond to stable, long-term preferences. Conversely, indices approaching the cutoff frequency  $L'$  (high-frequency) typically correspond to rapid, random noise.

**Energy-Aware Filter Generation.** Based on the global energy distribution  $M$ , we design an *energy-aware filtering mechanism* to assign higher attention weights to low-frequency components (representing stable long-term habits) while assigning lower weights to high-frequency components (representing stochastic noise). For the  $l'$ -th frequency component, the adaptive weight  $g_{l'}$  is computed as:

$$g_{l'} = \frac{\exp(W \cdot m_{l'} + b)}{\sum_{j=0}^{L'} \exp(W \cdot m_j + b)}, \quad (3)$$

where  $W \in \mathbb{R}^{d \times d}$  and  $b \in \mathbb{R}^d$  are learnable parameters shared across all frequencies.  $G = [g_0, \dots, g_{l'}, \dots, g_{L'}] \in \mathbb{R}^{(L'+1) \times d}$ . Through this



**Figure 1: The overall architecture of DHPRec. (Bottom) The Frequency-Domain Feature Refinement module constructs refined item representations via spectral denoising. (Middle) The Pattern-Based Anchor-Guided Fusion Network, which transforms the sequence into compact Pattern Units and utilizes the Anchor to extract relevant long-term regularities. (Top) A motivating example illustrating how the Immediate Intent (e.g., Christmas demand) bridges specific Historical Habits.**

mechanism, the model effectively distinguishes signal from noise, adaptively amplifying the informative low-frequency regularities while suppressing the interference of high-frequency fluctuations. Then We modulate the original signal with the generated weights via element-wise multiplication:

$$\hat{H} = \hat{Z} \odot G. \quad (4)$$

Through this operation, high-frequency noise components are adaptively suppressed (multiplied by small weights in  $G$ ), while low-frequency preference signals are amplified. Finally, we convert the refined spectrum back to the time domain using the irFFT (defined in Appendix A):

$$H = \text{irFFT}(\hat{H}). \quad (5)$$

**Frequency-Weighted Regularization.** To explicitly guide the model to focus on stable low-frequency regularities and suppress high-frequency noise, we further introduce a frequency-weighted regularization loss. This objective compels the model to prioritize low-frequency information by penalizing high attention weights:

$$\mathcal{L}_{freq} = \sum_{k=0}^{L'} \omega_k \cdot \frac{1}{d} \|g_k\|_1, \quad (6)$$

where  $\omega_k$  is a frequency-dependent penalty coefficient. We define  $\omega_k$  as a monotonically increasing function of the frequency index  $k$  (e.g.,  $\omega_k = k$  or  $\omega_k = k^2$ ), ensuring that the penalty cost increases as the frequency becomes higher. In the paper, we use  $\omega_k = k^2$ .

### 3.2 Pattern-Based Anchor-Guided Fusion

To address the inability to focus on specific regularities caused by attention dilution and bridge immediate intent with specific long-term patterns, we propose the *Pattern-Based Anchor-Guided Fusion mechanism*. Since historical sequences are often extremely long, directly modeling granular item-level interactions fails to effectively extract user preferences. User interests typically exhibit consistency within short periods. Therefore, we partition the long sequence into temporal slices based on a time interval and aggregate interactions within each slice. This process transforms the long item sequence into short *pattern units*, which represent long-term historical regularities (e.g., purchasing preferences for Christmas items during the holiday season can be extracted as a Christmas pattern). Then it utilizes an anchor to effectively extract relevant historical information matching the current intent. Finally, a learnable gating mechanism is employed to balance the contribution of the immediate intent and the specific historical preferences.

**3.2.1 Pattern Modeling.** To capture user's patterns, we partition the raw sequence  $\mathcal{S}$  into a series of temporal slices based on a time interval  $\Delta t$  (typically set to 2 days). Formally, the segmented sequence is represented as  $\mathcal{S} = \{\mathcal{S}_1, \dots, \mathcal{S}_n, \dots, \mathcal{S}_N\}$ . The  $n$ -th slice is defined as  $\mathcal{S}_n = \{s_{n,1}, s_{n,2}, \dots, s_{n,|\mathcal{S}_n|}\}$ . Here,  $s_{n,j}$  represents the  $j$ -th interaction within the  $n$ -th slice.  $s_{n,j} = (z_{n,j}, t_{n,j})$  consists of the representation  $z_{n,j}$  and the interaction timestamp  $t_{n,j}$ , where  $z_{n,j}$  is transformed into the refined representation  $h_{n,j}$  by the FDFR

module. For any pair of interactions  $s_{n,i}, s_{n,j} \in \mathcal{S}_n$ , the temporal constraint satisfies  $|t_{n,i} - t_{n,j}| \leq \Delta t$ .

**Intra-Slice Pattern Extraction.** we employ a Transformer encoder to capture intra-slice pattern representation. For each slice  $\mathcal{S}_n = \{s_{n,1}, s_{n,2}, \dots, s_{n,|\mathcal{S}_n|}\}$ , we obtain the refined representation  $H_n = [h_{n,1}, \dots, h_{n,j}, \dots, h_{n,|\mathcal{S}_n|}]$  (derived from Eq. 5). We introduce a compound position encoding method before the encoding layer. Specifically, we define two learnable embedding terms: (1) Global Slice-Level Position Embedding  $\text{pos}_{\text{slice}}^{(n)} \in \mathbb{R}^d$ : This captures the sequential order of the  $n$ -th slice within the entire user history. (2) Local Intra-Slice Position Embedding  $\text{pos}_{\text{local}}^{(j)} \in \mathbb{R}^d$ : This captures the relative temporal order of the  $j$ -th interaction within the current slice. By injecting these position signals into the semantic vector  $h_{n,j}$ , we obtain the position-enhanced input representation  $x_{n,j}$  for the  $j$ -th item in the  $n$ -th slice:

$$x_{n,j} = h_{n,j} + \text{pos}_{\text{slice}}^{(n)} + \text{pos}_{\text{local}}^{(j)}. \quad (7)$$

We obtain  $X_n = [x_{n,1}, \dots, x_{n,|\mathcal{S}_n|}]$ . Subsequently, to abstract the slice into a compact pattern, we feed  $X_n$  into the Transformer encoder followed by a Mean Pooling operation:

$$\hat{r}_n = \text{MeanPooling}(\text{TransformerEncoder}(X_n)), \quad (8)$$

where  $\hat{r}_n \in \mathbb{R}^d$  serves as the high-level representation of the  $n$ -th slice. Through this process, we transform the lengthy history into a sequence of semantically explicit pattern units, denoted as  $\{\hat{r}_1, \hat{r}_2, \dots, \hat{r}_N\}$ .

**Global Historical Pattern Modeling.** After extracting individual pattern units, we need to model their evolutionary trajectory to construct a complete *Global Historical Pattern* space. We adopt a Bidirectional LSTM (BiLSTM) to process the sequence of slice representations:

$$r_1, \dots, r_N = \text{BiLSTM}(\hat{r}_1, \dots, \hat{r}_N). \quad (9)$$

The resulting sequence  $R = \{r_1, \dots, r_N\}$  effectively captures both the aggregated user interests within patterns and the sequential evolution information between patterns. In summary,  $R$  serve as the final patterns representing the user's historical preferences.

**3.2.2 Anchor-Guided Pattern Extract.** User historical patterns are rich and diverse. In this module, we aim to extract specific historical preference patterns relevant to the immediate intent, and then capturing long-term regularities that align with the current context.

We define the user's most recent interaction slice  $r^* = r_N$  as the *Anchor*, representing the *Immediate Intent*. We firstly calculate the relevance score between each historical regularity pattern and the current intent (Anchor):

$$s_n = W_3^\top \tanh(W_1 r_n + W_2 r^* + b), \quad (10)$$

where  $W_1, W_2 \in \mathbb{R}^{d \times d}$ ,  $b \in \mathbb{R}^d$  and  $W_3 \in \mathbb{R}^d$ . Subsequently, we obtain the final weight for each slice:

$$\alpha_n = \frac{\exp(s_n)}{\sum_{j=1}^N \exp(s_j)}, \quad (11)$$

where  $\alpha_n$  represents the importance of the  $n$ -th historical pattern to the current decision. Based on the calculated weights, we aggregate the historical patterns to global long-term regularities:

$$c = \sum_{n=1}^N \alpha_n r_n, \quad (12)$$

where  $c \in \mathbb{R}^d$  represents the reconstructed Global Historical Context that best matches the current situation. This process achieves *precise extraction* of specific long-term regularities: if the current anchor reflects a Christmas intent, the model automatically assigns higher weights to historical slices corresponding to previous Christmas periods, effectively filtering out irrelevant recent behaviors (e.g., exam preparations) and other historical regularities.

**3.2.3 bridging Intent and Regularity.** Finally, to simulate the dynamic user decisions between immediate intent and historical regularities in real decision-making, we formulate the final user representation  $e_u$ :

$$e_u = \lambda \cdot r^* + (1 - \lambda) \cdot c, \quad (13)$$

where  $r^*$  denotes the user's last pattern (representing the *Immediate Intent*),  $c$  represents the *Global Historical Regularities*, and the term  $\lambda \in (0, 1)$  serves as a dynamic gating coefficient that balances the contribution of short-term and long-term factors.

Crucially,  $\lambda$  is dynamically generated by jointly considering the user's immediate intent and the global historical regularities. We compute it using a learnable gating network:

$$\lambda = \sigma(W_{g1} r^* + W_{g2} c + b_g), \quad (14)$$

where  $W_{g1}, W_{g2} \in \mathbb{R}^{d \times d}$  and  $b_g \in \mathbb{R}^d$ , and  $\sigma(\cdot)$  is the Sigmoid activation function.

### 3.3 Optimization Objectives

To achieve end-to-end training and comprehensively enhance representation quality, we adopt a multi-task learning strategy to jointly optimize the following four loss functions.

**Recommendation Loss.** As the core objective, we employ the standard Cross-Entropy loss to optimize the next-item prediction accuracy. Given the final user representation  $e_u$  derived from the model and the representation of the target item  $z_{\text{target}}$ , we maximize the prediction probability of the positive sample while suppressing the negative items  $j \in \mathcal{N}$ :

$$\mathcal{L}_{\text{rec}} = -\log \frac{\exp(e_u \cdot z_{\text{target}}/\tau)}{\exp(e_u \cdot z_{\text{target}}/\tau) + \sum_{j \in \mathcal{N}} \exp(e_u \cdot z_j/\tau)}, \quad (15)$$

where  $z$  denotes the item latent representation defined in Definition 1, and  $\tau$  is the temperature coefficient.

**Sequence Alignment Loss:** To further improve the robustness of long-sequence modeling and mitigate data sparsity, we adopt sequence-level contrastive learning [2, 21, 34, 38]. For a given input sequence, we construct two augmented views  $e_u^A$  and  $e_u^B$  via

random data augmentation. We utilize the symmetric InfoNCE loss to maximize the agreement between different views:

$$\mathcal{L}_{cl} = \frac{1}{2} \left( \ell(\mathbf{e}_u^A, \mathbf{e}_u^B) + \ell(\mathbf{e}_u^B, \mathbf{e}_u^A) \right), \quad (16)$$

$$\ell(\mathbf{u}, \mathbf{v}) = -\log \frac{\exp(\text{sim}(\mathbf{u}, \mathbf{v})/\tau)}{\sum_{\mathbf{k} \in \text{batch}} \exp(\text{sim}(\mathbf{u}, \mathbf{k})/\tau)}. \quad (17)$$

This symmetric objective helps the model learn more discriminative sequence representations.

**Multi-Task Learning:** Finally, we integrate the primary recommendation task with the auxiliary constraints. We adopt a multi-view semantic encoder based on Vector Quantization. To maintain the semantic consistency, we incorporate a standard Masked Code Modeling loss  $\mathcal{L}_{vq}$  [18] as a semantic regularizer. Furthermore, we include the frequency-weighted regularization loss  $\mathcal{L}_{freq}$  (defined in Eq. 6). The total objective function is formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_{rec} + \gamma_1 \mathcal{L}_{cl} + \gamma_2 \mathcal{L}_{freq} + \gamma_3 \mathcal{L}_{vq}, \quad (18)$$

where  $\gamma_1, \gamma_2, \gamma_3$  are hyperparameters controlling the contribution of contrastive learning, frequency regularization, and masked code modeling, respectively.

### 3.4 Discussion

**Comparison with Existing Work.** Table 1 compares DHPRec with representative baselines. Traditional methods like SASRec [14] and CCFRec [18] suffer from both noise interference and truncation-induced *short-sightedness*. While frequency-enhanced models like FMLP-Rec [43] and TedRec [36] utilize filters to suppress noise (Denoise.), they still rely on truncation strategies, failing to model the full interaction history (Long-Seq.). Moreover, they lack explicit mechanisms to extract historical patterns relevant to the current intent (Intent.). In contrast, DHPRec integrates spectral denoising with anchor-guided extraction, effectively utilizing the full sequence to bridge immediate intent with specific long-term habits.

**Complexity Analysis.** Computational efficiency is critical for modeling long user behaviors. Standard Transformer-based methods suffer from a quadratic complexity of  $O(L^2 \cdot d)$ , making them computationally prohibitive for extensive interaction histories. In contrast, DHPRec achieves quasi-linear complexity through its pattern design. First, the spectral operations in the FDFR module utilize the Fast Fourier Transform, which requires only  $O(d \cdot L \log L)$ . Second, for the Pattern-Based Anchor-Guided Fusion mechanism, we decompose the sequence of length  $L$  into  $N$  slices of length  $l$  (where  $L = N \times l$ ). This decomposition reduces the attention complexity from  $O(L^2 \cdot d)$  to  $O(N \cdot l^2 \cdot d)$ . Substituting  $N = L/l$ , this term becomes  $O(L \cdot l \cdot d)$ , which scales linearly with  $L$  since  $l$  is a small constant ( $l \ll L$ ). Consequently, the overall computational cost is dominated by the spectral transformation, resulting in a complexity of  $O(L \log L)$ . This efficiency allows DHPRec to explicitly model extremely long user histories without succumbing to the computational bottleneck.

**Table 1: Comparison of sequential recommendation models. Denoise.: Capability of reducing noise interference. Long-Seq.: Modeling full interaction history without truncation. Intent.: Extracting patterns relevant to current intent.**

Methods	Core Technique	Denoise.	Long-Seq.	Intent.
SASRec [14]	Self-Attention	✗	✗	✗
CCFRec [18]	Semantic Code	✗	✗	✗
FMLP-Rec [43]	FFT (Global Filter)	✓	✗	✗
TedRec [36]	FFT (Fusion)	✓	✗	✗
<b>DHPRec</b>	<b>Anchor-Spectral</b>	✓	✓	✓

**Table 2: Statistics of the preprocessed datasets.**

Dataset	#Users	#Items	#Actions	Sparsity
Baby	150,777	36,013	1,241,083	99.977%
Instrument	57,439	24,587	511,836	99.964%
Game	94,762	25,612	814,586	99.966%
Scientific	50,985	25,848	412,947	99.969%

## 4 Experiments

### 4.1 Experimental Setting

**4.1.1 Dataset Descriptions.** As shown in Table 2, we conduct experiments on four public benchmark datasets from the Amazon Review 2023 collection [10]: Baby Products (Baby), Musical Instruments (Instrument), Video Games (Game), and Industrial and Scientific (Scientific). We concatenate the fields of title, brand, features, categories and description as the textual feature. Following existing literature [36, 42, 43], we apply the 5-core strategy to filter inactive users and unpopular items. For each user, interaction records are sorted chronologically. Finally, we adopt the leave-one-out [23, 42] strategy for data splitting, where the last interaction is used for testing, the second-to-last for validation, and the rest for training.

**4.1.2 Baselines.** We compare DHPRec with a comprehensive set of state-of-the-art methods, categorized into three groups: (1) ID-based Methods, which rely solely on item identifiers (IDs) to model sequential patterns, including RNN-based **GRU4Rec** [8], Transformer-based **SASRec** [14] and **BERT4Rec** [27], as well as advanced representation learning models **DuoRec** [21] and **MAERec** [38]; (2) Semantic-Enhanced Methods, which leverage side information (e.g., text) to alleviate sparsity, such as **FDSA** [39], **S<sup>3</sup>Rec** [42], **UniSRec** [11], **VQRec** [9], **MMSR** [12], and **CCFRec** [18]; and (3) Frequency-Domain Methods, which utilize spectral analysis mechanisms to capture global dependencies for efficient modeling or fusion, specifically including representative models such as **FMLP-Rec** [43] and **TedRec** [36].

**4.1.3 Evaluation Metrics.** We evaluate the model performance on the test set. We adopt the full-ranking protocol, which ranks the ground-truth item against all other items in the candidate pool, and report the average results across all users. We utilize two metrics: Recall@K and NDCG@K, where K is set to 5 and 10. Detailed definitions are provided in Appendix B.

**Table 3: Performance comparisons among different methods. The columns are reordered as Baby, Instrument, Game, and Scientific. “Ours” denotes the results of DHPRec. Best results are highlighted in bold and second-best results are underlined.**

Methods	Baby				Instrument				Game				Scientific			
	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
BERT4Rec	0.0176	0.0305	0.0121	0.0149	0.0312	0.0480	0.0190	0.0257	0.0465	0.0730	0.0303	0.0381	0.0191	0.0291	0.0124	0.0150
GRU4Rec	0.0225	0.0349	0.0145	0.0179	0.0344	0.0535	0.0221	0.0286	0.0535	0.0815	0.0355	0.0438	0.0235	0.0369	0.0153	0.0189
SASRec	0.0216	0.0357	0.0140	0.0175	0.0328	0.0528	0.0218	0.0269	0.0540	0.0852	0.0336	0.0433	0.0264	0.0407	0.0155	0.0194
FMLPRec	0.0233	0.0362	0.0151	0.0185	0.0344	0.0541	0.0213	0.0277	0.0523	0.0862	0.0343	0.0439	0.0264	0.0417	0.0160	0.0199
FDSA	0.0238	0.0392	0.0163	0.0200	0.0374	0.0571	0.0245	0.0302	0.0549	0.0857	0.0356	0.0465	0.0268	0.0421	0.0178	0.0224
S <sup>3</sup> Rec	0.0221	0.0333	0.0131	0.0173	0.0322	0.0501	0.0204	0.0252	0.0490	0.0774	0.0320	0.0411	0.0258	0.0423	0.0176	0.0214
DuoRec	0.0206	0.0351	0.0144	0.0177	0.0368	0.0568	0.0251	0.0305	0.0564	0.0849	0.0373	0.0464	0.0250	0.0384	0.0161	0.0214
MAERec	0.0219	0.0375	0.0141	0.0188	0.0374	0.0572	0.0238	0.0315	0.0623	0.0931	0.0416	0.0508	0.0298	0.0457	0.0210	0.0248
UniSRec	0.0234	0.0381	0.0145	0.0185	0.0365	0.0603	0.0229	0.0313	0.0568	0.0916	0.0352	0.0464	0.0281	0.0462	0.0162	0.0209
VQRec	0.0265	0.0404	0.0171	0.0209	0.0374	0.0607	0.0232	0.0293	0.0586	0.0921	0.0360	0.0471	0.0298	0.0456	0.0175	0.0219
MMSR	0.0227	0.0394	0.0137	0.0190	0.0365	0.0564	0.0226	0.0305	0.0563	0.0876	0.0377	0.0456	0.0269	0.0422	0.0175	0.0203
TedRec	0.0251	0.0375	0.0160	0.0213	0.0369	0.0575	0.0254	0.0308	0.0629	0.0961	<u>0.0424</u>	0.0521	0.0277	0.0420	0.0190	0.0243
CCFRec	<u>0.0281</u>	<u>0.0460</u>	<u>0.0179</u>	<u>0.0243</u>	<u>0.0432</u>	<u>0.0677</u>	<u>0.0276</u>	<u>0.0366</u>	<u>0.0663</u>	<u>0.1037</u>	0.0408	<u>0.0541</u>	<u>0.0369</u>	<u>0.0550</u>	<u>0.0229</u>	<u>0.0280</u>
<b>Ours</b>	<b>0.0315</b>	<b>0.0504</b>	<b>0.0204</b>	<b>0.0265</b>	<b>0.0452</b>	<b>0.0712</b>	<b>0.0290</b>	<b>0.0370</b>	<b>0.0703</b>	<b>0.1114</b>	<b>0.0442</b>	<b>0.0578</b>	<b>0.0399</b>	<b>0.0607</b>	<b>0.0254</b>	<b>0.0321</b>
Improv.	+12.1%	+9.6%	+14.0%	+9.1%	+4.6%	+5.2%	+5.1%	+1.1%	+6.0%	+7.4%	+4.2%	+6.8%	+8.1%	+10.4%	+10.9%	+14.6%

**Table 4: Ablation study of our proposed method on three datasets. The best and second-best results are denoted in bold and underlined fonts, respectively.**

Variants	Baby				Game				Instrument			
	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
(0) DHPRec	<b>0.0315</b>	<b>0.0504</b>	<b>0.0204</b>	<b>0.0265</b>	<b>0.0703</b>	<b>0.1114</b>	<b>0.0442</b>	<b>0.0578</b>	<b>0.0452</b>	<b>0.0712</b>	<b>0.0290</b>	<b>0.0370</b>
(1) w/o FDFE	0.0302	0.0484	0.0193	0.0252	0.0675	0.1068	0.0427	0.0554	<u>0.0441</u>	<u>0.0705</u>	<u>0.0285</u>	0.0367
(2) w/o Pattern	0.0308	0.0490	0.0196	0.0255	<u>0.0682</u>	<u>0.1091</u>	<u>0.0432</u>	<u>0.0562</u>	0.0437	0.0701	0.0284	<u>0.0369</u>
(3) Fusion-Concat	<u>0.0309</u>	<u>0.0491</u>	<u>0.0199</u>	<u>0.0257</u>	0.0677	0.1073	0.0429	0.0559	0.0436	<u>0.0705</u>	0.0281	0.0367

**4.1.4 Implementation Details.** We implement all models based on the open-source benchmark library RecBole [37, 40, 41]. To ensure a fair comparison, we optimize models with the Adam optimizer and cross-entropy loss. For all models, the embedding dimension is fixed at 128. We also carefully tune the learning rate in {5e-4, 1e-3, 3.5e-3} for the optimal performance. For baselines, we search the hyperparameters following original papers. Furthermore, we provide our code, datasets, and logged results to improve reproducibility.

## 4.2 Overall Performance

Table 3 presents the performance comparison of DHPRec against various baselines across four datasets. Analyzing the experimental outcomes reveals several critical observations regarding the limitations of existing paradigms and the effectiveness of our proposed approach.

First, traditional ID-based Methods (e.g., SASRec, BERT4Rec) exhibit evident performance bottlenecks. Although these methods leverage Self-Attention to capture dynamic Immediate Intent within sequences, they largely succumb to the limitations of the truncation strategy. From a cognitive psychology perspective, this

design forces models to be *short-sighted*, effectively capturing recent signals while completely discarding the specific historical habits embedded in the early history. Consequently, their performance is strictly limited on datasets that rely on the accumulation of long-term preferences, such as Baby and Instrument.

Second, Semantic-Enhanced Methods (e.g., CCFRec, UniSRec) generally outperform pure ID-based baselines. Notably, CCFRec achieves strong results by bridging the semantic gap via vector quantization. However, these improvements are primarily confined to item representation rather than sequence modeling architecture. These methods typically still strictly follow the truncation-based paradigm (e.g., keeping only the recent 20 interactions). As a result, while they can better understand the semantics of current items, they fail to capture the complete evolutionary trajectory of user interests, leaving the challenge of long-term dependency unresolved.

Third, regarding Frequency-Domain Methods (e.g., FMLP-Rec, TedRec), while they effectively utilize spectral analysis to capture global sequence dependencies, they fail to distinguish which specific historical regularities are semantically consistent with the

user’s current decision-making moment, leading to suboptimal recommendation accuracy.

Finally, our proposed DHPRec consistently achieves the best performance across all datasets and metrics. DHPRec achieves average relative improvements of 8.1% across all metrics, with gains reaching up to 14.6% in NDCG@10 on the Scientific dataset. This significant leap validates the effectiveness of our framework, which stems from the organic unification of three strategic designs. The FDFR module first filters noise to provide a clean semantic environment. Building on this, the Pattern-Based mechanism transforms the lengthy interaction history into compact regularity units, allowing the model to capture the complete evolution of user habits without the information loss typical of truncation. Crucially, the Anchor-Guided Fusion utilizes the Immediate Intent to precisely extract specific regularity units that match the current context. By bridging current demands with specific historical patterns, DHPRec realizes a precise unification of intent and history.

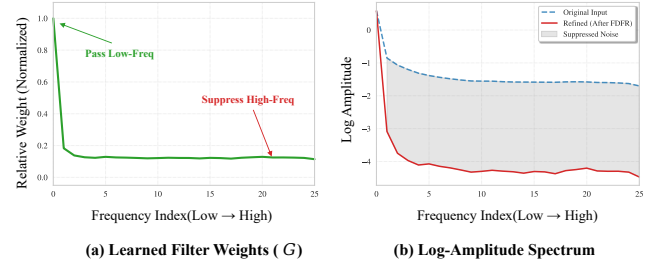
### 4.3 Ablation Study

To verify the contribution of each component in DHPRec, we conduct ablation studies on the Baby, Game, and Instrument datasets. The results are reported in Table 4. Specifically, we construct the following three variants for comparison: (1) **w/o FDFR** removes the Frequency-Domain Feature Refinement module and directly feeds the semantic-fused representations into the subsequent network, validating the necessity of the spectral energy-based denoising mechanism; (2) **w/o Pattern** replaces the Pattern-Based mechanism with a standard flat Transformer to model the entire long sequence item-by-item, verifying the effectiveness of abstracting long histories into compact pattern units; and (3) **Fusion-Concat** replaces the anchor-guided adaptive gating mechanism with simple concatenation to fuse immediate intents and specific long-term regularities, validating the effectiveness of our dynamic balancing strategy.

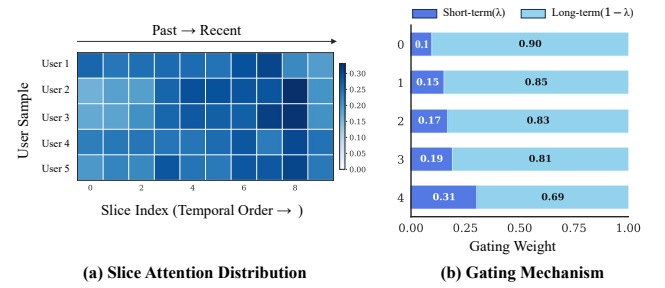
From the results in Table 4, we can observe that removing any core component leads to a significant degradation in model performance. First, the performance drop of **w/o FDFR** confirms that directly modeling raw interaction sequences with high-frequency noise harms recommendation accuracy. The FDFR module effectively enhances sequence robustness by filtering stochastic noise via spectral analysis. Second, **w/o Pattern** consistently underperforms the full model across all datasets, with a more pronounced drop on the Game dataset which features longer sequences. This indicates that flat models suffer from *attention dilution* when processing extensive histories. In contrast, our Pattern-Based design successfully transforms lengthy sequences into compact regularity units, enabling the model to effectively capture the evolution of user habits. Finally, the results of **Fusion-Concat** demonstrate that a user’s current decision is not always dominated by short-term history; the adaptive gating mechanism dynamically balances the immediate intent and the extracted long-term regularities based on context, enabling more precise predictions.

### 4.4 Further Analysis

**4.4.1 Visualization of Frequency Spectrum.** To gain insights into the inner workings of the FDFR module and verify its denoising



**Figure 2: Visualization of the learned adaptive  $G$  and spectral changes in the FDFR module. (a) The distribution of the learned weights, demonstrating a low-pass filtering characteristic. (b) Comparison of Log-Amplitude Spectrum before and after FDFR.**



**Figure 3: Visualization of the Pattern-Based Anchor-Guided Fusion. (a) pattern-level attention heatmap. (b) Adaptive gating weights distribution.**

efficacy, we randomly sample interaction sequences from the test set and visualize their spectral transformation, as shown in Figure 2. First, we examine the learned adaptive weights  $G$  in Figure 2(a). The curve exhibits a distinct low-pass filtering characteristic: the weights remain high in the low-frequency region (indices near 0), indicating that the model prioritizes preserving components representing long-term trends. Conversely, the weights decay rapidly in the high-frequency band. This confirms that our energy-aware filtering mechanism adaptively learns to focus on informative signals while neglecting high-frequency fluctuations. Second, the consequence of this filtering is illustrated in the log-amplitude spectrum comparison in Figure 2(b). The refined signal (red line) closely aligns with the original input (blue dashed line) in the low-frequency region, implying that essential user preference information is preserved. In contrast, amplitudes in the high-frequency region are significantly suppressed (highlighted by the shaded area), corresponding to the removal of random noise accumulated in the long sequence. Collectively, these visualizations provide intuitive evidence that FDFR acts as an effective semantic denoiser.

**4.4.2 Visualization of Pattern-Based Anchor-Guided Fusion Mechanism.** To verify whether DHPRec effectively unifies immediate intent with specific long-term patterns, we conducted a case study visualizing the pattern-level attention distribution and the adaptive gating weight  $\lambda$ . Figure 3 presents the results for five randomly

**Table 5: Performance sensitivity analysis of the temporal slice window  $\Delta t$  in DHPRec. We report Recall@K (R@K) and NDCG@K (N@K) on the Scientific and Baby datasets.**

Window $\Delta t$	Scientific				Baby			
	R@5	R@10	N@5	N@10	R@5	R@10	N@5	N@10
1 Hour	0.0385	0.0609	0.0246	0.0319	0.0302	0.0474	0.0193	0.0248
12 Hours	0.0388	0.0610	0.0247	0.0320	0.0308	0.0488	0.0196	0.0254
2 Days	<b>0.0399</b>	<b>0.0607</b>	<b>0.0254</b>	<b>0.0321</b>	<b>0.0315</b>	<b>0.0504</b>	<b>0.0204</b>	<b>0.0265</b>
1 Week	0.0387	0.0603	0.0243	0.0312	0.0310	0.0499	0.0199	0.0259
1 Month	0.0374	0.0588	0.0239	0.0308	0.0298	0.0478	0.0191	0.0249

sampled users. First, observing the attention heatmap in Figure 3(a), the model does not restrict itself to a *short-sighted* view (i.e., highlighting only the rightmost recent slices) but instead demonstrates the capability to extract relevant information across long temporal spans. Taking User 5 as an example, although the input anchor is located at the end of the sequence, the model assigns significant attention weights to extremely early historical patterns (e.g., around index 3). This strongly evidences that our Pattern-Based Anchor-Guided Fusion Mechanism functions as intended: the current interaction pattern (Anchor) serves as a potent cue, successfully awakening specific regularity patterns from the distant history that highly match the current intent. Simultaneously, the gating weights in Figure 3(b) reveal the model’s dynamic balance between trusting the moment and relying on history. For users with rich interactions and clear patterns (e.g., User 1), the model generates a lower  $\lambda$  (0.17), indicating that the decision is dominated by deep-seated historical habits; conversely, for users with sparser history (e.g., User 5), the model adaptively increases  $\lambda$  (0.26) to shift focus toward the immediate intent. This adaptivity confirms that DHPRec can intelligently coordinate the contributions of short-term signals and long-term regularities based on context uncertainty.

**4.4.3 Sensitivity Analysis of Temporal Slice Window.** To investigate the impact of temporal granularity on DHPRec’s ability to construct meaningful Pattern Units, we analyze the model performance with varying slice windows  $\Delta t$  (ranging from 1 hour to 1 month). The results are reported in Table 5. The performance exhibits a distinct “rise-then-fall” trend, peaking at  $\Delta t = 2$  Days. This observation validates that selecting an appropriate granularity is critical for generating independent Pattern Units. Specifically, an overly fine granularity (e.g., 1 Hour) wrongly splits a complete user habit into fragmented interactions. It cuts a continuous shopping intent into multiple incomplete slices, preventing the intra-slice modeling from capturing the full semantic meaning of the pattern. Conversely, an overly coarse granularity (e.g., 1 Month) mixes multiple different and distinct interests into a single slice. This packs unrelated behaviors (e.g., buying food and electronics) into one noisy unit, making it difficult for the Anchor (Immediate Intent) to accurately distinguish and find the specific matching regularity from the complex background. The optimal setting of  $\Delta t = 2$  Days strikes a necessary balance, ensuring that each slice forms a clear and independent pattern unit, thereby maximizing the overall retrieval effectiveness of the mechanism.

## 5 Related Work

### 5.1 Sequential Recommendation

Sequential Recommendation aims to capture the dynamic evolution of user preferences. Early works mostly relied on RNNs [8, 28], while Transformer-based models like SASRec [14] and BERT4Rec [27] established ID-based baselines using self-attention [5, 44]. To address data sparsity, recent trends incorporate PLM-based semantic information (e.g., FDSA [39], UnisRec [11]), with CCFRec [18] bridging the semantic gap via vector quantization. However, these methods typically employ truncation strategies (e.g., keeping only the recent 20 interactions) for efficiency [36, 43], failing to capture long-term patterns. Although hierarchical methods like DSIN [6] introduce a structured view to model long sequences, they are primarily designed for sparse ID sequences, ignoring the high-frequency noise inherent in dense semantic vectors. Moreover, they typically treat historical segments equally, failing to effectively extract specific patterns from the complex history that align with the user’s current intent.

### 5.2 Frequency-Domain Learning

Inspired by digital signal processing theories [22], frequency-domain learning has been widely explored in sequential recommendation. Existing works primarily utilize the Discrete Fourier Transform (DFT) to project interaction sequences into the frequency domain, categorized into two paradigms: efficiency optimization [15] and information enhancement [25]. For efficiency, FMLP-Rec [43] leverages the log-linear complexity  $O(N \log N)$  of the Fast Fourier Transform (FFT) to replace the quadratic complexity of traditional Self-Attention, achieving efficient global modeling. Regarding enhancement, FEHAN [4] and TedRec [36] utilize frequency-domain features as supplements to time-domain information, significantly enhancing the representation of complex interaction patterns. However, existing methods mostly adopt static filtering strategies, overlooking the inherent advantage of the frequency domain in distinguishing stable preferences (low-frequency) from random noise (high-frequency). In contrast, DHPRec focuses on *sequence denoising*, dynamically eliminating noise interference via adaptive spectral filtering to construct a pristine semantic environment for preference extraction.

## 6 Conclusion

In this paper, we address the limitations of truncation strategies from a cognitive perspective, highlighting the necessity of balancing Immediate Intent and Specific Historical Patterns. To achieve this unification, we propose DHPRec, a novel framework that tackles the critical challenges of noise interference and attention dilution in long-sequence modeling. Specifically, DHPRec integrates a Frequency-Domain Feature Refinement (FDFR) module to robustly filter stochastic noise and a Pattern-Based Anchor-Guided Fusion mechanism to precisely extract long-term regularities relevant to the current context. Extensive experiments on four benchmark datasets demonstrate DHPRec’s superiority over state-of-the-art baselines. In future work, we plan to explore more efficient spectral operators (e.g., Wavelet Transform) for ultra-long sequences and extend the framework to multi-modal recommendation scenarios.

## References

- [1] Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150* (2020).
- [2] Yongjun Chen, Zhiwei Liu, Jia Li, Julian McAuley, and Caiming Xiong. 2022. Intent contrastive learning for sequential recommendation. In *Proceedings of the ACM web conference 2022*. 2172–2182.
- [3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*. 4171–4186.
- [4] Xinyu Du, Huanhuan Yuan, Pengpeng Zhao, Jianfeng Qu, Fuzhen Zhuang, Guan-feng Liu, Yanchi Liu, and Victor S Sheng. 2023. Frequency enhanced hybrid attention network for sequential recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*. 78–88.
- [5] Xinyan Fan, Zheng Liu, Jianxun Lian, Wayne Xin Zhao, Xing Xie, and Ji-Rong Wen. 2021. Lighter and better: low-rank decomposed self-attention networks for next-item recommendation. In *Proceedings of the 44th international ACM SIGIR conference on research and development in information retrieval*. 1733–1737.
- [6] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep session interest network for click-through rate prediction. *arXiv preprint arXiv:1905.06482* (2019).
- [7] Michael Heideman, Don Johnson, and Charles Burrus. 1984. Gauss and the history of the fast Fourier transform. *IEEE Assp Magazine* 1, 4 (1984), 14–21.
- [8] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2015. Session-based recommendations with recurrent neural networks. *arXiv preprint arXiv:1511.06939* (2015).
- [9] Yupeng Hou, Zhankui He, Julian McAuley, and Wayne Xin Zhao. 2023. Learning vector-quantized item representation for transferable sequential recommenders. In *Proceedings of the ACM Web Conference 2023*. 1162–1171.
- [10] Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. 2024. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952* (2024).
- [11] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards universal sequence representation learning for recommender systems. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*. 585–593.
- [12] Hengchang Hu, Wei Guo, Yong Liu, and Min-Yen Kan. 2023. Adaptive multi-modalities fusion in sequential recommendation systems. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 843–853.
- [13] Daniel Kahneman and Amos Tversky. 1973. On the psychology of prediction. *Psychological review* 80, 4 (1973), 237.
- [14] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*. IEEE, 197–206.
- [15] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. 2022. Fnet: Mixing tokens with fourier transforms. In *Proceedings of the 2022 Conference of the north American chapter of the Association for Computational Linguistics: human language technologies*. 4296–4313.
- [16] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.
- [17] Chang Liu, Xiaoguang Li, Guohao Cai, Zhenhua Dong, Hong Zhu, and Lifeng Shang. 2021. Noninvasive self-attention for side information fusion in sequential recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 35. 4249–4256.
- [18] Enze Liu, Bowen Zheng, Wayne Xin Zhao, and Ji-Rong Wen. 2025. Bridging Textual-Collaborative Gap through Semantic Codes for Sequential Recommendation. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*. 1788–1798.
- [19] Haibo Liu, Zhixiang Deng, Liang Wang, Jinjia Peng, and Shi Feng. 2023. Distribution-based learnable filters with side information for sequential recommendation. In *Proceedings of the 17th ACM Conference on Recommender Systems*. 78–88.
- [20] Qidong Liu, Xian Wu, Wanyu Wang, Yejing Wang, Yuanshao Zhu, Xiangyu Zhao, Feng Tian, and Yefeng Zheng. 2025. Llmemb: Large language model can be a good embedding generator for sequential recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 12183–12191.
- [21] Ruihong Qiu, Zi Huang, Hongzhi Yin, and Zijian Wang. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*. 813–823.
- [22] Lawrence R Rabiner and Bernard Gold. 1975. *Theory and Application of Digital Signal Processing*: (by) Lawrence R. Rabiner (and) Bernard Gold. Prentice-Hall.
- [23] Shashank Rajput, Nikhil Mehta, Anima Singh, Raghunandan Hulikal Keshavan, Trung Vu, Lukasz Heldt, Lichan Hong, Yi Tay, Vinh Tran, Jonah Samost, et al. 2023. Recommender systems with generative retrieval. *Advances in Neural Information Processing Systems* 36 (2023), 10299–10315.
- [24] Valerie F Reyna and Charles J Brainerd. 1995. Fuzzy-trace theory: An interim synthesis. *Learning and individual Differences* 7, 1 (1995), 1–75.
- [25] Yehjin Shin, Jeongwhan Choi, Hyowon Wi, and Noseong Park. 2024. An attentive inductive bias for sequential recommendation beyond the self-attention. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 38. 8984–8992.
- [26] Keith E Stanovich and Richard F West. 2000. Advancing the rationality debate. *Behavioral and brain sciences* 23, 5 (2000), 701–717.
- [27] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.
- [28] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 17–22.
- [29] Jiayi Tang and Ke Wang. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 565–573.
- [30] Amos Tversky and Daniel Kahneman. 1973. Availability: A heuristic for judging frequency and probability. *Cognitive psychology* 5, 2 (1973), 207–232.
- [31] Charles Van Loan. 1992. *Computational frameworks for the fast Fourier transform*. SIAM.
- [32] Hao Wang, Jianxun Lian, Mingqi Wu, Haoxuan Li, Jiajun Fan, Wanyue Xu, Chaozhao Li, and Xing Xie. 2023. Conformer: Revisiting transformer for sequential user modeling. *arXiv preprint arXiv:2308.02925* (2023).
- [33] Yunjia Xi, Weiwen Liu, Jianghao Lin, Xiaoling Cai, Hong Zhu, Jieming Zhu, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Towards open-world recommendation with knowledge augmentation from large language models. In *Proceedings of the 18th ACM Conference on Recommender Systems*. 12–22.
- [34] Xu Xie, Fei Sun, Zhaoyang Liu, Shiwen Wu, Jinyang Gao, Jiandong Zhang, Bolin Ding, and Bin Cui. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*. IEEE, 1259–1273.
- [35] Yueqi Xie, Peilin Zhou, and Sunghun Kim. 2022. Decoupled side information fusion for sequential recommendation. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*. 1611–1621.
- [36] Lanling Xu, Zhen Tian, Bingqian Li, Junjie Zhang, Daoyuan Wang, Hongyu Wang, Jinpeng Wang, Sheng Chen, and Wayne Xin Zhao. 2024. Sequence-level semantic representation fusion for recommender systems. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*. 5015–5022.
- [37] Lanling Xu, Zhen Tian, Gaowei Zhang, Junjie Zhang, Lei Wang, Bowen Zheng, Yifan Li, Jiakai Tang, Zeyu Zhang, Yupeng Hou, et al. 2023. Towards a more user-friendly and easy-to-use benchmark library for recommender systems. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2837–2847.
- [38] Yaowen Ye, Lianhao Xia, and Chao Huang. 2023. Graph masked autoencoder for sequential recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*. 321–330.
- [39] Tingting Zhang, Pengpeng Zhao, Yanchi Liu, Victor S Sheng, Jiajie Xu, Deqing Wang, Guanfeng Liu, Xiaofang Zhou, et al. 2019. Feature-level deeper self-attention network for sequential recommendation. In *IJCAI*. 4320–4326.
- [40] Wayne Xin Zhao, Yupeng Hou, Xingyu Pan, Chen Yang, Zeyu Zhang, Zihan Lin, Jingsen Zhang, Shuqing Bian, Jiakai Tang, Wenqi Sun, et al. 2022. Recbole 2.0: Towards a more up-to-date recommendation library. In *Proceedings of the 31st ACM international conference on information & knowledge management*. 4722–4726.
- [41] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, et al. 2021. Recbole: Towards a unified, comprehensive and efficient framework for recommendation algorithms. In *proceedings of the 30th acm international conference on information & knowledge management*. 4653–4664.
- [42] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM international conference on information & knowledge management*. 1893–1902.
- [43] Kun Zhou, Hui Yu, Wayne Xin Zhao, and Ji-Rong Wen. 2022. Filter-enhanced MLP is all you need for sequential recommendation. In *Proceedings of the ACM web conference 2022*. 2388–2399.
- [44] Peilin Zhou, Qichen Ye, Yueqi Xie, Jingqi Gao, Shoujin Wang, Jae Boum Kim, Chenyu You, and Sunghun Kim. 2023. Attention calibration for transformer-based sequential recommendation. In *Proceedings of the 32nd ACM international conference on information and knowledge management*. 3595–3605.

## A Fourier Transform

The Discrete Fourier Transform (DFT) is one of the classical methods in the field of sequence signal processing, which converts the sampled signal from the *time domain* to the *frequency domain*. In this work, given that the input interaction sequence  $X \in \mathbb{R}^{L \times d}$  consists of real-valued vectors, we employ the Real-valued Fast Fourier Transform (rFFT). rFFT exploits the conjugate symmetry of real signals to map the sequence to a compact complex-valued frequency domain. For a sequence  $x \in \mathbb{R}^L$  in the  $d$ -th feature dimension, the forward transformation is formally defined as:

$$\text{rFFT} : X[k] = \sum_{n=0}^{L-1} x_n e^{-j \frac{2\pi}{L} nk}, \quad 0 \leq k \leq \lfloor L/2 \rfloor, \quad (19)$$

where  $j$  is the imaginary unit, and  $X[k]$  represents the complex spectrum at the  $k$ -th frequency component. Accordingly, given representations in the frequency domain, the following formula (Inverse rFFT) is used to losslessly recover signals to the time domain:

$$\text{irFFT} : x_n = \frac{1}{L} \sum_{k=0}^{L-1} X[k] e^{j \frac{2\pi}{L} nk}, \quad 0 \leq n \leq L-1. \quad (20)$$

This transformation allows us to efficiently capture long-term patterns utilizing the global receptive field of the frequency domain.

## B Evaluation Metrics Details

In our experiments, we adopt two widely used metrics, **Recall@K** and **NDCG@K** (Normalized Discounted Cumulative Gain), to evaluate the top- $K$  recommendation performance.

Let  $\mathcal{U}$  denote the set of users in the test set. For each user  $u \in \mathcal{U}$ , let  $v_{target}^{(u)}$  be the ground-truth target item (i.e., the next item the user actually interacted with) and  $\hat{R}_u = \{\hat{v}_1, \hat{v}_2, \dots, \hat{v}_K\}$  be the list of top- $K$  items recommended by the model, sorted by their predicted scores in descending order.

**Recall@K.** This metric measures the proportion of test cases where the ground-truth item is present in the top- $K$  recommendation list. In the context of leave-one-out evaluation where there is only one positive target item per user, Recall@K is mathematically defined as:

$$\text{Recall@K} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbb{I}(v_{target}^{(u)} \in \hat{R}_u), \quad (21)$$

where  $\mathbb{I}(\cdot)$  is the indicator function, which equals 1 if the condition is true and 0 otherwise. Essentially, it indicates whether the target item is successfully “hit” by the model within the top- $K$  cutoff.

**NDCG@K.** This metric accounts for the position of the hit in the recommendation list, assigning higher scores to hits at higher ranks (i.e., smaller indices). It is calculated as the mean of the NDCG scores across all users:

$$\text{NDCG@K} = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \frac{\text{DCG}_u@K}{\text{IDCG}_u@K}, \quad (22)$$

where  $\text{DCG}_u@K$  (Discounted Cumulative Gain) and  $\text{IDCG}_u@K$  (Ideal DCG) are defined as:

$$\text{DCG}_u@K = \sum_{i=1}^K \frac{\mathbb{I}(\hat{v}_i = v_{target}^{(u)})}{\log_2(i+1)}, \quad (23)$$

**Table 6: Performance comparison of different global evolution encoders (BiLSTM vs. Transformer).**

Dataset	Metric	Variants	
		(0) DHPRec	(1) Inter-Trans
Baby	R@5	<b>0.0315</b>	0.0302
	R@10	<b>0.0504</b>	0.0481
	N@5	<b>0.0204</b>	0.0193
	N@10	<b>0.0265</b>	0.0251
Game	R@5	<b>0.0703</b>	0.0664
	R@10	<b>0.1114</b>	0.1040
	N@5	<b>0.0442</b>	0.0408
	N@10	<b>0.0578</b>	0.0540
Inst.	R@5	<b>0.0452</b>	0.0427
	R@10	<b>0.0712</b>	0.0690
	N@5	<b>0.0290</b>	0.0274
	N@10	<b>0.0370</b>	0.0358

$$\text{IDCG}_u@K = \sum_{i=1}^{|\{v_{target}^{(u)}\}|} \frac{1}{\log_2(i+1)}. \quad (24)$$

Since there is only one ground-truth item in our test setting, the ideal list contains the target item at the first position. Thus, the normalization term  $\text{IDCG}_u@K$  is consistently equal to 1 (as  $\log_2(1+1) = 1$ ), making NDCG@K equivalent to calculating  $1/\log_2(\text{rank} + 1)$  if the target is found at position rank, and 0 otherwise.

## C Impact of Global Evolution Encoder

In the main framework of DHPRec, we employ a Bi-directional LSTM (BiLSTM) to model the global evolution of user interests across temporal slices (Section 3.3.2). To justify this architectural choice, we conduct an additional ablation study comparing BiLSTM with a Transformer encoder. We denote this variant as **Inter-Trans**.

Table 6 presents the performance comparison. We observe that **Inter-Trans** underperforms the proposed DHPRec (BiLSTM) across all datasets. For instance, on the Game dataset, which features long interaction sequences, the NDCG@10 drops from 0.0578 to 0.0540.

We attribute this to distinct inductive biases. At the inter-slice level, the input is a sequence of high-level abstractions (regularity units) where the chronological order represents a continuous “drift” of user preferences. The recurrent nature of BiLSTM possesses a strong sequential inductive bias, making it naturally adept at capturing such continuous evolutionary trajectories and state transitions. In contrast, while Transformer excels at capturing long-range dependencies via self-attention, it is permutation-invariant and relies heavily on position embeddings. In the context of macro-level evolution where the “flow” of time is critical, the Transformer structure may struggle to model the strict temporal causalities as effectively as RNN-based models. Therefore, BiLSTM serves as a more suitable backbone for the Global Base Rate Modeling module.