Dual-Perspective Modeling: Interest Trend-Detection and Diversity-Aware for Multi-Behavior Sequential Recommendation

Peixuan Sun[1†] [0009-0009-5347-7631], Zhida Zheng[2†] [0009-0005-9876-2103], Yuhan Yang[3][0009-0009-4785-0292], Wei Zhou[1][0000-0003-0839-8773], Junhao Wen[1*] [0000-0002-6561-560X]

[1] School of Big Data and Software Engineering, Chongqing University, Chongqing, 400044, China

[2] Simon Business School, University of Rochester, 500 Joseph C. Wilson Blvd. Rochester, NY 14627, USA

[3] School of Public Administration, Sichuan University, Chengdu, 610065, China

sunpeixuan@stu.cqu.edu.cn, {zhouwei,jhwen}@cqu.edu.cn, zzheng29@ur.rochester.edu, yyh@stu.scu.edu.cn

Abstract. In real-world recommendation systems, user interactions inherently include behavioral information (e.g., clicks, cart). Multi-behavior sequential recommendations focus on predicting the next item a user will interact with under the target behavior. Incorporating behavioral information enhances the diversity of user interaction data. However, existing multi-behavior recommendation methods perform unified modeling without distinguishing the interest categories reflected by different data. Hence, we propose a novel method called TDBSR (Trend-Detection and Diversity-Aware for Multi-Behavior Sequential Recommendation) which separately captures interest trend and dispersed diversity before performing a unified integration. We design a mask generation module that decouples the data, employing MLP-based modules: HITM (heterogeneous interest trend module) and ABIP (auxiliary behavior intent perception module) to extract user interest tendencies. Concurrently, we leverage MLP and max-pooling layers to extract user interest diversity. Experiments on two public datasets validate that our proposed TDBSR outperforms state-of-the-art methods.

Keywords: Multi-Behavior Sequential Recommendation, Multi-Interest Learning, Heterogeneous Behavioral Sequences.

1 Introduction

The rise of digital services has increased the information overload, making personalized recommendations essential. Sequential recommendation [1,2,3] predicts the next

* Corresponding author

† These authors contributed equally to this work.

desired item for a user from past behaviors. Real-world interactions include both target (e.g., purchase) and auxiliary (e.g., cart, click) behaviors, which offer richer preference signals to ease data sparsity and enhance modeling.

Current research in multi-behavior sequential recommendation often uses Transformer [1,2] or GNN [4,5,6,7] to individually model distinct user behaviors, uniformly integrating multi-behavior information into user representations to capture interest trends. However, as auxiliary data grows, the diversity of items (e.g., trends, low-frequency, noise) [8] increases, and ignoring this diversity leads to performance drops [4,9]. Research has split into multi-interest modeling and denoising methods, but we believe both still have drawbacks.

(1) Modeling both trends and diversity of user interests together leads to performance loss. A user's interests can be split into stable interest trends and occasional interest diversity. Trends represent primary focus areas in sequences, while diversity captures sporadic interests driven by external factors or exploration. Diversity adds novelty and variety to recommendations. When both are modeled together, diverse interaction data introduces noise. Existing multi-interest models [9,10] fail to decouple these aspects, resulting in suboptimal performance. (2) Treating interest diversity as noise leads to information loss. To reduce the impact of diverse data on trend extraction, current studies [3,8,11] often discard such data as noise, which improves performance. However, they remove all detected noise without considering if some "noise" represents users' diverse interests (e.g., low-frequency), potentially reducing recommendation diversity.

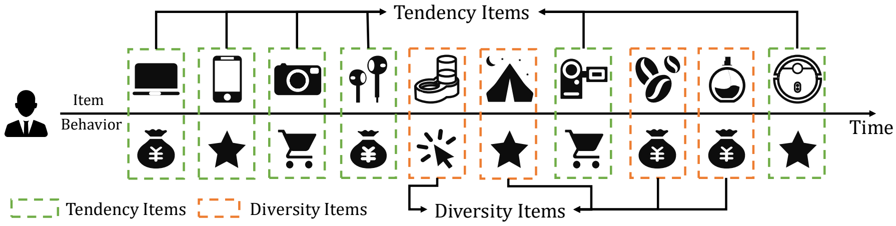

Fig. 1. Motivation Illustration: two components of interest.

As shown in Fig. 1, the user is a high-income young male. Green boxes represent interest trend items, and orange boxes represent interest diversity items. His main interests are reflected in frequent behaviors: buying premium digital products (e.g., laptops, headphones), collecting smartphones, and adding professional photography equipment to his cart. However, the sequence also includes low-frequency behaviors: occasional perfume and coffee bean purchases, camping gear promotions, and temporary clicks on pet smart feeder links. These behaviors reflect his diverse life scenarios—home relaxation, outdoor activities, and curiosity about smart home devices. Traditional multi-behavior models may misclassify low-frequency actions (e.g., pet feeder clicks) as noise, though they may still be linked to his main interests (e.g., tech preferences). For example, buying coffee beans may reflect household needs, showcasing his dispersed interests. However, existing models fail to separately model interest trends and diversity, leading to suboptimal performance.

To address these challenges, we introduce the TDBSR model, which uses a mask generator to separate user interactions into two components: interest trend subsequences and interest diversity items. This segmentation enables separate modeling for better handling of each component. TDBSR uses TIEB (temporal interaction encoding block) and FFEB (feature fusion encoding block) in HITM and ABIP, built on MLP, to extract interest trends by capturing sequential patterns and item features from interest trends and auxiliary behaviors. MLP with max-pooling layers uncovers hidden interests. The model also improves tail item embeddings with contextual information to better represent less frequent items. Experiments demonstrate that TDBSR outperforms current state-ofthe-art models. The key contributions of this research are as follows:

- TDBSR separates interest trends and diversity in behavior-aware sequential recommendation systems. This dual-perspective architecture solves issues in existing joint modeling approaches.
- We use a pure MLP architecture, where TIEB and FFEB capture sequential item information and behavior-aware features to identify core preferences. MLP with max-pooling layers effectively extracts dispersed interests. A context-aware tail item enhancement mechanism reduces long-tail bias through neighborhood representation learning.
- Our TDBSR model is efficient and improves across two datasets compared to the state-of-the-art baseline models.

2 Problem Statement

We let U be the set of users and $I = \{i_1, \dots, i_j, \dots, i_J\}$ be the set of items, divided into tail items I^{tail} and head items I^{head}. User-item interactions often involve multiple behaviors like clicks, cart, and purchase. We define the behavior set as $B = \{b_1, \dots, b_k, \dots, b_K\}$, where K is the total number of behavior types. We treat purchase as the target behavior and others as auxiliary.

For a user $u \in U$, we construct a heterogeneous interaction sequence $S_u = \{\langle i_{u,1}, b_{u,1}\rangle, \langle i_{u,2}, b_{u,2}\rangle, \dots, \langle i_{u,|S_u|}, b_{u,|S_u|}\rangle\}$, where $|S_u|$ is the sequence length. The goal is to extract latent information from the dynamic sequence to predict the top-K items u will interact with next, focusing on target behavior.

3 Methodology

We introduce the architecture of our proposed TDBSR model. As shown in Fig. 2, our framework first derives item embeddings from the enhanced table in the tail-item embedding enhancement module (TEEM). A mask generator then partitions user interaction sequences into two views: the interest trend subsequence T_u and the interest diversity set D_u. HITM extracts time-aware patterns and cross-behavior correlations for T_u, while ABIP infers purchase intentions from auxiliary behaviors, with both fused into a trend representation. D_u uses MLP and max-pooling to extract diversity features. Finally, the combined trend-diversity representation predicts user-item ratings.

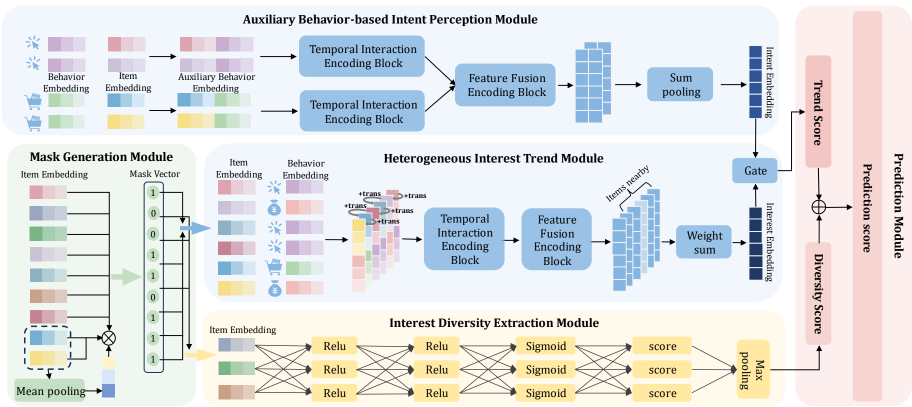

Fig. 2. The Overall Framework of TDBSR.

3.1 Tail-item Embedding Enhancement Module

Due to sparse interactions, traditional methods struggle to train tail items. We propose TEEM to refine tail embeddings using a matrix $V^g \in R^{\mathbb{1}\times d}$, which dynamically updates them with contextual neighbors in sequences. TEEM operates directly on the main embedding table V: initial tail embeddings are drawn from V, and updates are committed back to it, ensuring consistency and preventing bias from separate parameterization. The equation is as follows:

$$V^g = (1-a)\cdot V^g + \frac{a}{2s}\sum_{j=t-s}^{t+s} V_{i_j}, \tag{1}$$

where s specifies half of the window size, and $V_{i_j} \in R^{\mathbb{1}\times d}$ refers to the embedding of item i_j. $a \in (0,1)$ is a weighted parameter that controls the influence of the latest enhanced representation on the module. The enhanced embedding representations for tail items are obtained via Equation 1.

The final tail item representations are generated by fusing original embeddings with enhanced matrix V^g through a weighted mechanism, defined as:

$$V_{i_j} = \begin{cases} (1-\beta)\cdot V_{i_j} + \beta \cdot V_{i_j} V^g_{i_j}, if\ item\ i_j \in I^{\text{tail}}, \\ V_{i_j}, if\ item\ i_j \in I^{\text{head}}, \end{cases} \tag{2}$$

where the trade-off parameter $\beta \in (0,1)$ balances original and enhanced tail item embeddings. These refined embeddings are used across all modules.

3.2 Mask Generation Module

The mask generation module splits the user sequence S_u into an interest trend subsequence T_u and a diversity set D_u. We fix input length at $L = 50$, and use the recent f items' mean (after linear transformation) as a dynamic interest proxy, avoiding noise from older interactions. Semantic similarity between each item embedding and the proxy yields mask m_j, which determines whether item i_j is assigned to T_u or D_u, thus decoupling interests. The equations are:

$$p = \frac{1}{f}\sum_{j=L-f+1}^{L} W_b V_{i_j}, \tag{3}$$

where p is the trend proxy, $W_b \in R^{d\times d}$ is a learnable parameter. We compute the cosine similarity r_j between V_{i_j} and p. If $r_j \geq \gamma$ (threshold), mask $m_j = 1$ assigns i_j to T_u; else to D_u. The temporal order of items in T_u is maintained via position-wise masking on the original sequence.

3.3 Interest Trend Extraction Module

We propose two modules to extract user interest trends: 1) HITM processes heter-ogeneous sequences from the mask module to model interest trends; 2) ABIP ana-lyzes behavioral subsequences to capture recent intent.

Heterogeneous Interest Trend Module. User interaction sequences encode preferences via both items and behavior transitions. HITM captures interest evolution from heterogeneous sequences.Its input is $T_u = \{\langle i_{u,1}^m, b_{u,1}^m\rangle, \langle i_{u,2}^m, b_{u,2}^m\rangle, \dots, \langle i_{u,L}^m, b_{u,L}^m\rangle\}$ generated in section 4.2, with shorter sequences zero-padded.

For each item $i_{u,t}^m$, behavior embedding is computed as $C_{u,t} = B_{b_{u,t}^m} + \tau\left(b_{u,t}^m, b_{u,t+1}^m\right)$, where $B_{b_{u,t}^m} \in R^{1\times d}$ is the behavior embedding, and $\tau\left(b_{u,t}^m, b_{u,t+1}^m\right) \in R^{1\times d}$ is the transition embedding [12]. This is concatenated with the item embedding $V_{i_{u,t}^m}$ to form $x_{u,t} = \text{concat}\left(C_{u,t}, V_{i_{u,t}^m}\right) \in R^{1\times 2d}$, creating a high-dimensional user-item representation. The full sequence $X_u^{(0)} = \left[x_{u,1}; \dots; x_{u,L}\right] \in R^{L\times 2d}$ is then fed into TIEB and FFEB, where TIEB captures temporal-behavioral patterns and FFEB models cross-feature interactions.

TIEB uses an N-layer stack, where each layer refines hidden states via hierarchical propagation and dynamic updates, enabling adaptive temporal modeling, defined as $X_{\text{TIEB}}^{(n-1)} = X^{(n-1)} + \text{TIEB}\left(\tilde{X}^{(n-1)^T}\right)$. The details are:

$$\tilde{X}^{(n-1)} = \text{LayerNorm}\left(X^{(n-1)}\right), \tag{4}$$

$$\text{TIEB}\left(\tilde{X}^{(n-1)^T}\right) = \left[\text{GELU}\left(\tilde{X}^{(n-1)^T} W_{\text{TIEB}}^1\right)\right] W_{\text{TIEB}}^2, \tag{5}$$

where $W_{\text{TIEB}}^1 \in R^{1\times d_t}$ and $W_{\text{TIEB}}^2 \in R^{d_t\times L}$ are learnable matrices, d_t is TIEB's hidden dimension. $X^{(0)}$ is chronologically ordered to preserve temporal dependencies. Linear transformations and nonlinear activations capture sequential patterns and positional item interactions, helping the model learn complex sequence relations.

Inspired by multi-head attention, FFEB splits $X_{\text{TIEB}}^{(n-1)}$ into H subspaces along the feature dimension: $\hat{X}^{(h)} = X_{(h-1)\times\frac{2d}{H}+1:h\times\frac{2d}{H}}, h = 1,2,\dots,H$. Each is processed nonlinearly, then concatenated to form the global representation, producing the n-th FFEB layer output as:

$$X^{(n)} = X_{\mathrm{TIEB}}^{(n-1)} + \mathrm{FFEB}(\tilde{X}_{\mathrm{TIEB}}^{(n-1)}), \tag{6}$$

$$\mathrm{FFEB}(X) = \mathrm{concat}(\dots, \mathrm{Split}_h, \dots)W_{\mathrm{FFEB}}^{0}, \tag{7}$$

$$\mathrm{Split}_h = \sigma\big(\hat{X}^{(h)} W_{\mathrm{FFEB}}^{1}\big) W_{\mathrm{FFEB}}^{2}, \tag{8}$$

where $W_{\mathrm{FFEB}}^{0} \in R^{2d \times 2d}$, $W_{\mathrm{FFEB}}^{1} \in R^{\frac{2d}{H} \times d_c}$ and $W_{\mathrm{FFEB}}^{2} \in R^{d_c \times \frac{2d}{H}}$ are trainable matrices, d_c is FFEB's hidden dimension, and $\sigma(\cdot)$ is the sigmoid function. Finally, $X^{(N)} \in R^{L \times 2d}$ refines tail item embeddings using neighbor information (Equations 1 and 2), mitigating long-tail issues. Differential weights are then assigned to items as:

$$\alpha_t = \frac{\exp\big(x_t^{(N)} W_6\big)}{\sum_{j=1}^{L} \exp\big(x_j^{(N)} W_6\big)}, \tag{9}$$

where $W_6 \in R^{2d \times 1}$ is a trainable vector, and $\alpha_t \in R$ denotes the adaptive attention weight for the t-th item. Item embeddings are fused via weighted integration as follows:

$$e_1 = \mathrm{dropout}\big(\textstyle\sum_{t=1}^{L} \alpha_t x_t^{(N)}\big). \tag{10}$$

Auxiliary Behavior-based Intent Perception Module. We extract the latest L' auxiliary behavior items ($L' = 5$) form short subsequences for noise reduction. Each item's initial representation concatenates the behavior embedding and item embedding: $h_{b_{u,t}} = \mathrm{concat}\big(B_{b_{u,t}}, V_{i_{u,t}}\big)$. Inputs from all behaviors are merged into the input tensor $\mathcal{H}_u^{(0)} = \big[H_{b_u}^{1}; \dots; H_{b_u}^{k}\big] \in R^{L' \times k \times 2d}$, where k is the number of auxiliary behaviors. The N-th layer is defined as $\mathcal{H}^{(n)} = \mathcal{H}^{(n-1)} + \mathrm{FFEB}\big(\widetilde{H}_s^{(n-1)}\big)$. The details is:

$$\mathcal{H}_s^{(n-1)} = stack\left(TIEB\left({H_{b_u^1}^{(n-1)}}^{T}\right)^{T}, \dots, TIEB\left({H_{b_u^m}^{(n-1)}}^{T}\right)^{T}\right). \tag{11}$$

We extract terminal items from each behavioral sequence for cross-behavior aggregation, generating target behavior-oriented interest representations: $e_2 = \mathrm{mean}\big(\mathcal{H}_{\mathcal{L}'}^{(N)}\big)$, where $e_2 \in R^{1 \times 2d}$ and $\mathcal{H}_{\mathcal{L}'}^{(N)} \in R^{m \times 2d}$ denote the terminal item representation of auxiliary behaviors.

Finally, we integrate HITM and ABIP via a gating mechanism with adaptive weights to generate target behavior-oriented user interest trends are:

$$gate = \sigma\big(e_1 W_1 + e_2 W_2 + b_{gate}\big), \tag{12}$$

$$Z_{trend} = gate \otimes e_1 + (1 - gate) \otimes e_2, \tag{13}$$

where $W_1 \in R^{2d \times 2d}, W_2 \in R^{2d \times 2d}$ and $b_{gate} \in R^{1 \times 2d}$ are learnable weights and biases. $\otimes$ is the element-wise product. So, we can obtain the interest trend representation $\boldsymbol{Z}_{trend}$. We use Equation 14 to make the prediction:

$$\hat{y}_{trend} = \mathrm{softmax}(\boldsymbol{Z}_{trend} W_{\mathrm{trend}} + b_{trend}), \tag{14}$$

where $W_{\text{trend}} \in R^{2d\times|I|}$ is learnable matrix and $b_{trend} \in R^{1\times|I|}$ is learnable bias. $\hat{y}_{trend} \in R^{1\times|I|}$ contains the predicted score on each item.

3.4 Interest Diversity Extraction Layer

We model users' diverse interests using the set D_u, which contains both diversity signals and noise due to high inter-item variance. To extract diversity and suppress noise, each item in D_u is passed through two MLP layers with ReLU activation, then aggregated via max pooling. The computation is:

$$z_j = \mathrm{ReLU}\big(\mathrm{ReLU}\big(x_{u,j}W_3^T + b_3^T\big)W_4^T + b_4^T\big), \tag{15}$$

where $W_3, W_4 \in R^{2d\times 2d}$ and $b_3, b_4 \in R^{1\times 2d}$ are learnable parameters. Features z_j are extracted via MLPs. A linear transformation followed by max pooling yields the diversity-based prediction:

$$\hat{y}_{diversity} = \max\Big(\sigma\big(Z_{diversity}W_5^T + b_5^T\big)\Big), \tag{16}$$

where $Z_{diversity} = [z_1; \dots; z_{n-1}; z_n]$ and n is the size of D_u. $W_5 \in R^{2d\times|I|}$ and $b_5 \in R^{|I|}$ are learnable parameters. $\max(\cdot)$ returns row-wise maximums.

3.5 Prediction aggregation and optimization

The final prediction score $\hat{y}$ is a linear combination of trend-based $\hat{y}_{trend}$ and diversity-based $\hat{y}_{diversity}$ predictions, controlled by a coupling factor $\lambda \in [0,1]$:

$$\hat{y} = \mathrm{softmax}\big(\lambda * \hat{y}_{trend} + (1-\lambda) * \hat{y}_{diversity}\big). \tag{17}$$

The model is trained using the following loss:

$$\mathcal{L} = -\sum_{s\in\mathcal{S}}\sum_{j\in\mathcal{I}} y_{sj}\log\big(\hat{y}_j\big) + \big(1-y_{sj}\big)\log\big(1-\hat{y}_j\big), \tag{18}$$

where $y_{sj} = 1$ if item j is the true next item in sequence s; otherwise $y_{sj} = 0$.

4 Experiments

4.1 Experimental Settings

This section provides a concise overview of the experimental configurations, encompassing the datasets employed, baseline methods for comparison, performance evaluation metrics, and specifics of the experimental parameters.

Datasets. We use two public real-world datasets. Users with fewer than 5 target behavior interactions are removed, and the bottom 5% of items by frequency are labeled as tail items. **UB** from Alibaba, items with fewer than 10 interactions are removed. **Rec15** from the RecSys 2015 Challenge, items with fewer than 5 interactions are removed. Following the paper[13], use the latest target behavior item for testing, the second latest for validation, and auxiliary behaviors between them for evaluation.

Baselines. We compare our model across four categories: single-behavior sequential recommendation [1,2], multi-behavior sequential recommendation [3,4,5,10,14], denoising [3,14] and multi-interest modeling [4,6]. **SASRec** [1] first utilizes attention mechanisms to model the sequence, significantly improving the performance. **BERT4Rec** [2] first introduces BERT and masked item prediction to learn item-sequence dependencies. **MGNM** [6] combines GNN and CapsNet to extract multi-level interests. **FMLP-Rec**: [3] applies fast Fourier transform (FFT) and inverse FFT to suppress noise and enhance sequence representation. **END4Rec**: [14] uses hard/soft noise filtering and contrastive learning to improve representation. **GHTID** [4] learns global and short-term interests via GNN. **MSR** [5] Uses GNN for item relations and GRU for heterogeneous behavior modeling. **BMLP**: [10] a pure MLP framework modeling short- and long-term preferences via non-linear feature interactions.

Evaluation Metrics. We evaluate TDBSR and baselines using Hit Rate (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K) with $K = \{10,20\}$. HR@K checks if the ground-truth item appears in top-K results, while NDCG@K considers its rank via logarithmic discounting. For evaluation, each user's most recent purchase is used as the test item. We rank items by predicted relevance from historical data and check whether the target appears in the top-K list and its position.

Experimental Parameters. Baseline parameters follow their original papers. TDBSR uses Adam optimizer (learning rate: 0.001), with batch size 512. Hidden dimensions are selected from {64, 128, 256}, block number is 3, and attention heads from {1, 2, 4, 8}. Recent item count f from {1, 2, 2nd, 3, 3rd}; threshold γ from {0, 0.2, 0.4, 0.6, 0.8}. Dropout is chosen from {0.2, 0.3, 0.4, 0.5} and regularization from {0.0001, 0.001, 0.01, 0.1}. Early stopping is applied (no improvement for 20 epochs). All experiments run on an NVIDIA RTX 4090 GPU.

4.2 Overall Performance and Analysis

As shown in Table 1, TDBSR outperforms baselines on both datasets. The gains arise from two key designs: 1) the mask generator decouples interest trends from diversity signals, enabling fine-grained modeling and context-aware representations; 2) the joint optimization of HITM, ABIP, and IDEM effectively captures both target-driven trends and diverse interest patterns. We further summarize the following observations:

Table 1. Comparative Analysis of Overall Performance on Datasets.

Dataset	UB				Rec15			
Metric	HR@10	NDCG@10	HR@20	NDCG @20	HR@10	NDCG @20	HR@10	NDCG @20
SASRec	0.181	0.106	0.238	0.121	0.437	0.248	0.566	0.283
BERT4Rec	0.098	0.055	0.138	0.063	0.389	0.196	0.535	0.233
MGNM	0.164	0.116	0.205	0.115	0.600	0.320	0.673	0.373
FMLP-Rec	0.158	0.103	0.192	0.113	0.541	0.255	0.672	0.286
END4Rec	<u>0.210</u>	0.092	0.242	0.124	0.605	0.327	0.690	0.413
GHTID	0.150	0.082	0.190	0.091	0.508	0.359	0.597	0.383
MSR	0.172	0.099	0.225	0.117	0.605	0.378	0.715	0.408

BMLP	0.200	0.114	0.254	0.130	0.618	0.407	0.726	0.434
TDBSR	**0.228**	**0.134**	**0.267**	**0.147**	**0.637**	**0.425**	**0.757**	**0.453**
Imp	**8.57%**	**17.54%**	**5.1%**	**13.08%**	**3.07%**	**4.42%**	**4.27%**	**4.38%**

Impact of Multi-Behavior Modeling: experimental results show that incorporating multi-behavior improves model performance over single-behavior methods, confirming the value of modeling behavioral dynamics for user preference prediction. The advantage highlights the importance of multi-behavior dependencies in capturing complex user preferences. **Performance of MLP Methods**: in multi-behavior scenarios, the graph-based MSR performs better on Rec15 but worse on the other dataset due to its suitability for short sequences. In contrast, the MLP-based BMLP outperforms MSR and GHTID, and our model also performs well, indicating that MLP architectures can achieve strong results. **Performance of Denoising and Multi-Interest Models**: FMLP-Rec suggests dense data aids interest modeling but requires fine-grained learning, as excessive filtering risks losing useful signals. END4REC overlooks sporadic interests when filtering noise. MGNM and GHTID capture diversity but ignore interaction noise, weakening their modeling. Our TDBSR, however, decouples sequences to suppress noise while preserving low-frequency interests.

4.3 Ablation Study

This section proposes seven model variants to investigate each component's effectiveness. We conduct relevant experiments and studies on two datasets, and the descriptions of the variants of TDBSR are as follows:

- **TDBSR w/o HITM.** This variant removes the module that uses heterogeneous interaction sequences to extract user target behavior interest trends and only uses auxiliary behavior for interest trend extraction.
- **TDBSR w/o ABIP.** This variant removes the module for extracting user target behavior interest trends from auxiliary behavior subsequences.
- **TDBSR w/o IDEM.** This variant removes the interest diversity extraction module, and the recommendation is solely based on interest trends.
- **TDBSR w/o TEEM.** This variant eliminates the module for enhancing tail items and does not consider the enhancement of tail item representations.

Table 2. Results of the ablation studies.

	UB		Rec15	
	HR@10	NDCG@10	HR@10	NDCG@10
w/o HITM	0.191	0.112	0.466	0.239
w/o ABIP	0.194	0.115	0.566	0.328
w/o IDEM	0.199	0.127	0.534	0.290
w/o TEEM	0.199	0.127	0.577	0.329
TDBSR	**0.228**	**0.134**	**0.637**	**0.425**

Table 2 presents the performance comparison of our model and its variants across two datasets. From the results, we can draw the following conclusions:

Comparison of model and variant performance. Removing any core module leads to a clear performance drop, confirming each component's contribution. **Interest Trend Extraction Module.** Using only auxiliary behaviors or full sequences hurts performance, showing that combining both better captures user interests. **Interest Diversity Extraction Module.** Removing this leads to notable decline, proving the value of separating interest diversity and noise from trend modeling. **Tail-item Embedding Enhancement Module.** Its removal reduces performance, indicating that tail item learning is insufficient without enhancement.

4.4 Hyperparameter Experiments

In section 4.4, we systematically investigate the impact of two critical hyperparameters on model performance: (1) the number of heads and (2) interest diversity weight coefficient λ.

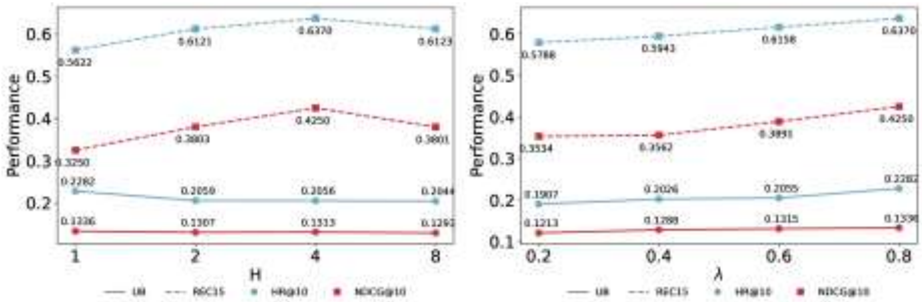

Fig. 3. Hyperparameter experiments results of TDBSR.

(1) We test head numbers $H \in \{1, 2, 4, 8\}$. As shown in Fig. 3(left), Rec15 performs best at $H = 4$, likely due to higher complexity and diversity, where moderate heads capture patterns effectively. UB performs best at $H = 1$, suggesting its simpler nature makes more heads unnecessary, risking overfitting. Larger H reduces per-head dimension; if too small, it weakens subspace representation. Thus, a proper H balances subspace count and feature dimension.

(2) As discussed, we decouple and jointly model interest trends and diversity using a coupling factor λ to balance their contributions. We test λ values in $\{0.2, 0.4, 0.6, 0.8\}$, finding that larger values improve performance, while smaller ones harm it—implying that excessive focus on diversity weakens critical trend signals. When $\lambda = 0$, the model only considers trends; when λ approaches 1, it depends mostly on diversity. The results show that too small values of λ leads to homogenized recommendations, while a moderate λ achieves a good balance between capturing temporal patterns and diverse preferences. Thus, tuning λ is key for optimizing recommendation quality.

5 Conclusion and future work

This work proposes TDBSR, an MLP-based framework that decouples and models interest trends and diversity for multi-behavior sequential recommendation. Experiments on two public datasets show that TDBSR outperforms strong baselines, and ablation studies confirm the effectiveness of its design. Case studies reveal that the trend module captures stable buying patterns, while the diversity module detects exploratory behavior. In future work, we aim to enhance behavior discrimination and better model behavior transitions to improve interpretability and robustness.

Acknowledgments. This work was funded by the National Natural Science Foundation of China (Grant No. 72074036, 62072060) and supported by the Natural Science Foundation of Chongqing, China, No. CSTB2024NSCQ-MSX0701.

References

1. Kang, W.C., McAuley, J.: Self-attentive sequential recommendation. In: 2018 IEEE international conference on data mining (ICDM). pp. 197–206. IEEE (2018)
2. Sun, F., Liu, J., Wu, J., Pei, C., Lin, X., Ou, W., Jiang, P.: Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In: Proceedings of the 28th ACM international conference on information and knowl-edge management. pp. 1441–1450 (2019)
3. Zhou, K., Yu, H., Zhao, W.X., Wen, J.R.: Filter-enhanced mlp is all you need for sequential recommendation. In: Proceedings of the ACM web conference 2022. pp. 2388–2399 (2022)
4. Li, X., Chen, H., Yu, J., Zhao, M., Xu, T., Zhang, W., Yu, M.: Global heterogeneous graph and target interest denoising for multi-behavior sequential recommendation. In: Proceedings of the 17th ACM International Conference on Web Search and Data Mining. pp. 387–395 (2024)
5. Meng, W., Yang, D., Xiao, Y.: Incorporating user micro-behaviors and item knowledge into multi-task learning for session-based recommendation. In: Proceedings of the 43rd international ACM SIGIR conference on research and development in Information Retrieval. pp. 1091–1100 (2020)
6. Tian, Y., Chang, J., Niu, Y., Song, Y., Li, C.: When multi-level meets multi-interest: A multi-grained neural model for sequential recommendation. In: Proceed-ings of the 45th international ACM SIGIR conference on research and development in information retrieval. pp. 1632–1641 (2022)
7. Li, Q., Ma, H., Jin, W., Ji, Y., Li, Z.: Hypergraph-enhanced multi-interest learning for multi-behavior sequential recommendation. Expert Systems with Applications 255, 124497 (2024)
8. Zhang, Y., Chen, W., Cai, F., Shi, Z., Qi, F.: Dmr: disentangled and denoised learning for multi-behavior recommendation. Complex & Intelligent Systems 11(2), 1–23 (2025)
9. Wu, B., Cheng, Y., Yuan, H., Ma, Q.: When multi-behavior meets multi-interest: Multi-behavior sequential recommendation with multi-interest self-supervised learning. In: 2024 IEEE 40th International Conference on Data Engineering (ICDE). pp. 845–858. IEEE (2024)
10. Li, W., Wu, Y., Liu, Y., Pan, W., Ming, Z.: Bmlp: behavior-aware mlp for heterogeneous sequential recommendation. arXiv preprint arXiv:2402.12733 (2024)

11. Lin, K., Wang, Z., Shen, S., Wang, Z., Chen, B., Chen, X.: Sequential recommendation with decomposed item feature routing. In: Proceedings of the ACM Web Conference 2022. pp. 2288–2297 (2022)
12. Zhan, Z., He, M., Pan, W., Ming, Z.: Transrec++: Translation-based sequential recommendation with heterogeneous feedback. Frontiers Comput. Sci. 16(2), 162615 (2022)
13. Chen, W., He, M., Ni, Y., Pan, W., Chen, L., Ming, Z.: Global and personalized graphs for heterogeneous sequential recommendation by learning behavior transitions and user intentions. In: Proceedings of the 16th ACM Conference on Recommender Systems. pp. 268–277 (2022)
14. Han, Y., Wang, H., Wang, K., Wu, L., Li, Z., Guo, W., Liu, Y., Lian, D., Chen, E.: Efficient noise-decoupling for multi-behavior sequential recommendation. In: Proceedings of the ACM on Web Conference 2024. pp. 3297–3306 (2024)