BLADE: A Behavior-Level Data Augmentation Framework with Dual Fusion Modeling for Multi-Behavior Sequential Recommendation

Yupeng Li[1], **Mingyue Cheng**[1*], **Yucong Luo**[1], **Yitong Zhou**[1], **Qingyang Mao**[1], **Shijin Wang**[2]

[1]State Key Laboratory of Cognitive Intelligence, University of Science and Technology of China
[2]iFLYTEK AI Research (Central China), iFLYTEK Co., Ltd.
{liyupeng,prime666,yitong.zhou,maoqy0503}@mail.ustc.edu.cn, mycheng@ustc.edu.cn, sjwang3@iflytek.com

Abstract

Multi-behavior sequential recommendation aims to capture users' dynamic interests by modeling diverse types of user interactions over time. Although several studies have explored this setting, the recommendation performance remains suboptimal, mainly due to two fundamental challenges: the heterogeneity of user behaviors and data sparsity. To address these challenges, we propose BLADE, a framework that enhances multi-behavior modeling while mitigating data sparsity. Specifically, to handle behavior heterogeneity, we introduce a dual item-behavior fusion architecture that incorporates behavior information at both the input and intermediate levels, enabling preference modeling from multiple perspectives. To mitigate data sparsity, we design three behavior-level data augmentation methods that operate directly on behavior sequences rather than core item sequences. These methods generate diverse augmented views while preserving the semantic consistency of item sequences. These augmented views further enhance representation learning and generalization via contrastive learning. Experiments on three real-world datasets demonstrate the effectiveness of our approach.

Code — https://github.com/WindSighiii/BLADE

1 Introduction

Recommender systems have become essential components of online platforms such as e-commerce and social media, enabling personalized content delivery and alleviating information overload (He et al. 2017; Lee, Park, and Lee 2025; He et al. 2023; Cheng et al. 2025). Among various recommendation paradigms, sequential recommendation (SR) has received substantial attention due to its capability to model the temporal dependencies within user-item interactions and predict subsequent user actions (Fang et al. 2020; Wang et al. 2019; Rendle, Freudenthaler, and Schmidt-Thieme 2010; Cheng et al. 2021, 2022; Luo et al. 2025; Wang, Cheng, and Liu 2025). To better capture complex user intentions, recent studies have extended SR to multi-behavior sequential recommendation (MBSR), which incorporates diverse user behaviors (e.g., clicks, favorites, and cart additions) to assist purchase predictions in e-commerce contexts (Luo et al. 2022; Gong et al. 2025; Zhang et al. 2025). Further extending this concept, EIDP (Chen, Pan, and Ming 2024) introduced Behavior Set-informed Sequential Recommendation (BSSR), where each interaction is represented as a set of concurrent behaviors (e.g., likes, favorites, and shares) on the same item. This formulation enables more expressive and fine-grained modeling of user preferences, making it particularly suitable for scenarios like social media where the rich co-occurrence of behaviors is common.

Despite its enhanced modeling capacity, BSSR still faces two key challenges. First, different behavior types have distinct semantics and complex dependencies, and the behavior set format further increases the difficulty, making behavior heterogeneity a greater challenge in modeling user preferences. To handle such heterogeneity, existing methods have proposed several item-behavior fusion strategies: (1) Early fusion integrates item and behavior representations at the input level, enriching local semantics but potentially introducing semantic interference due to differences in representation spaces (Zhou et al. 2018; Li et al. 2018a; He, Pan, and Ming 2022); (2) Intermediate fusion introduces behavioral information in intermediate layers to mitigate interference, but the final user representation remains item-dominated, missing the collaborative effect brought by direct item-behavior fusion (Yuan et al. 2022a); (3) Late fusion models each behavior separately as sub-sequences before aggregation, preserving independence but failing to effectively model cross-interactions among items and behaviors (Cho et al. 2023). Second, the inherent data sparsity limits the model's ability to learn reliable representations. Although self-supervised learning and data augmentation have proven effective in mitigating data sparsity in SR (Xie et al. 2022; Qiu et al. 2022; Chen et al. 2022b) and MBSR (Xiao, Pan, and Ming 2024), these methods are primarily designed for item sequences. However, item sequences encode the core semantics of user preferences. Operating on item sequences directly may distort the core semantics, causing the contrastive views to deviate from users' true preferences and ultimately weakening the learned representations.

To address these challenges, we propose BLADE, a **B**ehavior-**L**evel data **A**ugmentation framework with **D**ual fusion mod**E**ling, which enhances multi-behavior modeling and simultaneously mitigates data sparsity. Specifically, we first introduce a dual item-behavior fusion architecture to

*Corresponding Author.

handle behavior heterogeneity by combining early and intermediate fusion, capturing both static and dynamic item-behavior relations to enhance multi-behavior semantic modeling. Then, we design three behavior-level data augmentation methods to mitigate data sparsity and diversify behavioral patterns. These strategies also help reduce the dominance of frequent behaviors and improve learning from long-tail ones: (1) Co-occurrence behavior addition adds potential co-occurring behaviors based on global statistics to simulate realistic behavior combinations; (2) Frequency-based behavior masking masks frequent behaviors, which encourages the model to focus on more informative rare behaviors; (3) Auxiliary behavior flipping randomly perturbs high-frequency auxiliary behaviors (e.g., clicks), reducing dependence on dominant behaviors and enhancing generalization to diverse behavior combinations. Distinct from item-based augmentations, these methods diversify training views without compromising the semantics of the item sequence. Furthermore, we introduce a behavior richness-based loss weighting, assigning higher loss weights to prediction steps with richer behavior sets to enhance the model's ability to learn from complex supervisory signals.

We summarize our main contributions as follows.

- We design a dual item-behavior fusion architecture that collaboratively leverages early and intermediate fusion strategies to improve semantic modeling capabilities for multiple behaviors.
- We propose three behavior-level augmentation methods for BSSR that operate on behavior sequences to generate diverse yet semantically consistent views, alleviating both data sparsity and behavior imbalance.
- Extensive experiments on three real-world datasets are conducted to verify the effectiveness of BLADE.

2 Related Work

Single-Behavior Sequential Recommendation. Early work on SR employed Markov chains (Rendle, Freudenthaler, and Schmidt-Thieme 2010; He and McAuley 2016) to model item transitions. With the advancement of deep neural networks, numerous architectures have been explored, including RNN-based (Li et al. 2017), CNN-based (Tang and Wang 2018), and attention-based models (Kang and McAuley 2018; Sun et al. 2019; Wang et al. 2025), which laid the foundation for a range of follow-up improvements. Despite their success, these methods typically model only one type of behavior (e.g., purchases), overlooking auxiliary behaviors such as clicks and favorites. Therefore, they may not adapt well to real-world recommendation scenarios.

Multi-Behavior Sequential Recommendation. Most existing studies on MBSR are deep learning-based algorithms, including RNN-based models (Cho et al. 2023; Li et al. 2018b; Liu, Wu, and Wang 2017), GNN-based models (Chen et al. 2022a; Wang et al. 2020), Transformer-based models (Zhan et al. 2022; Yuan et al. 2022a; Luo et al. 2022) and hybrid techniques-based models (Xia et al. 2022, 2021; Meng, Yang, and Xiao 2020). DyMuS (Cho et al. 2023), a recent RNN-based framework, models each type of user behavior with an individual GRU, capturing behavior-specific dynamics. It then employs a dynamic routing mechanism to adaptively combine the resulting behavior-aware representations. MBHT (Yang et al. 2022) combines multi-scale Transformers and multi-behavior hypergraph learning to model short-term dynamics at different temporal granularities and long-range dependencies across behavior types, effectively enhancing user preference modeling. MB-STR (Yuan et al. 2022a) configures the weights in the classic multi-head self-attention layer to be behavior-specific. It also introduces a relative positional encoding scheme and uses MMoE to integrate behavior representations. To better represent social media behavior patterns, EIDP extends MBSR to the BSSR setting by modeling each interaction as a behavior set (Chen, Pan, and Ming 2024), allowing one item to be linked to multiple behaviors. While this richer representation increases expressiveness, BSSR still faces long-standing challenges such as data sparsity and behavior heterogeneity.

3 Method

3.1 Problem Formulation

We use $\mathcal{U} = \{u\}$, $\mathcal{V} = \{v\}$ and $\mathcal{B}$ to denote a set of users, a set of items and a set of user behaviors, respectively. $|\mathcal{U}|$, $|\mathcal{V}|$ and $|\mathcal{B}|$ are the numbers of users, items and behavior types, respectively. The interaction sequence with behavior sets of a specific user $u \in \mathcal{U}$ can be represented as: $\mathcal{S}_u = \{(v_u^1, \mathbf{b}_u^1), \ldots, (v_u^l, \mathbf{b}_u^l), \ldots, (v_u^L, \mathbf{b}_u^L)\}$. We represent the general form of a behavior set as a multi-hot vector, $\mathbf{b}_u^l = (b_{u,1}^l, \ldots, b_{u,k}^l, \ldots, b_{u,|\mathcal{B}|}^l) \in \mathbb{R}^{|\mathcal{B}|}$, where $b_{u,k}^l = 1$ if user u has interacted with item v_u^l with the k-th behavior at the l-th step, and $b_{u,k}^l = 0$ otherwise. L indicates the length of the user sequence.

The goal is to predict the next item $v \in \mathcal{V} \setminus \mathcal{S}_u$ that is likely to be interacted with under a target behavior set by user u at the $L+1$ step. The target behavior set is a vector containing the target behaviors.

3.2 Overview of BLADE

To address behavior heterogeneity and data sparsity in BSSR, we propose BLADE, which comprises two components: (1) dual item-behavior fusion: We integrate item and behavior information via early and intermediate fusion to enhance semantic modeling of behaviors and capture multi-granularity behavioral patterns; (2) behavior-level data augmentation: We design three augmentation methods—Co-occurrence addition, Frequency-based masking, and Auxiliary behavior flipping—that operate at the behavior level while preserving item-sequence semantics.

3.3 Dual Item-Behavior Fusion

In our dual item-behavior fusion architecture, we integrate item and behavior information at both early and intermediate stages, enhancing modeling capacity across multiple semantic levels. Figure 1 illustrates the overall architecture.

Embedding Layer. An embedding matrix $\boldsymbol{V} \in \mathbb{R}^{|\mathcal{V}| \times d}$ maps items to d-dimensional vectors. Given a sequence $\mathcal{S}_u$, we obtain its item embeddings via lookup and stack them as $\boldsymbol{E} = [\boldsymbol{e}_{v_u^1}; \boldsymbol{e}_{v_u^2}; \ldots; \boldsymbol{e}_{v_u^L}] \in \mathbb{R}^{L \times d}$. To encode temporal

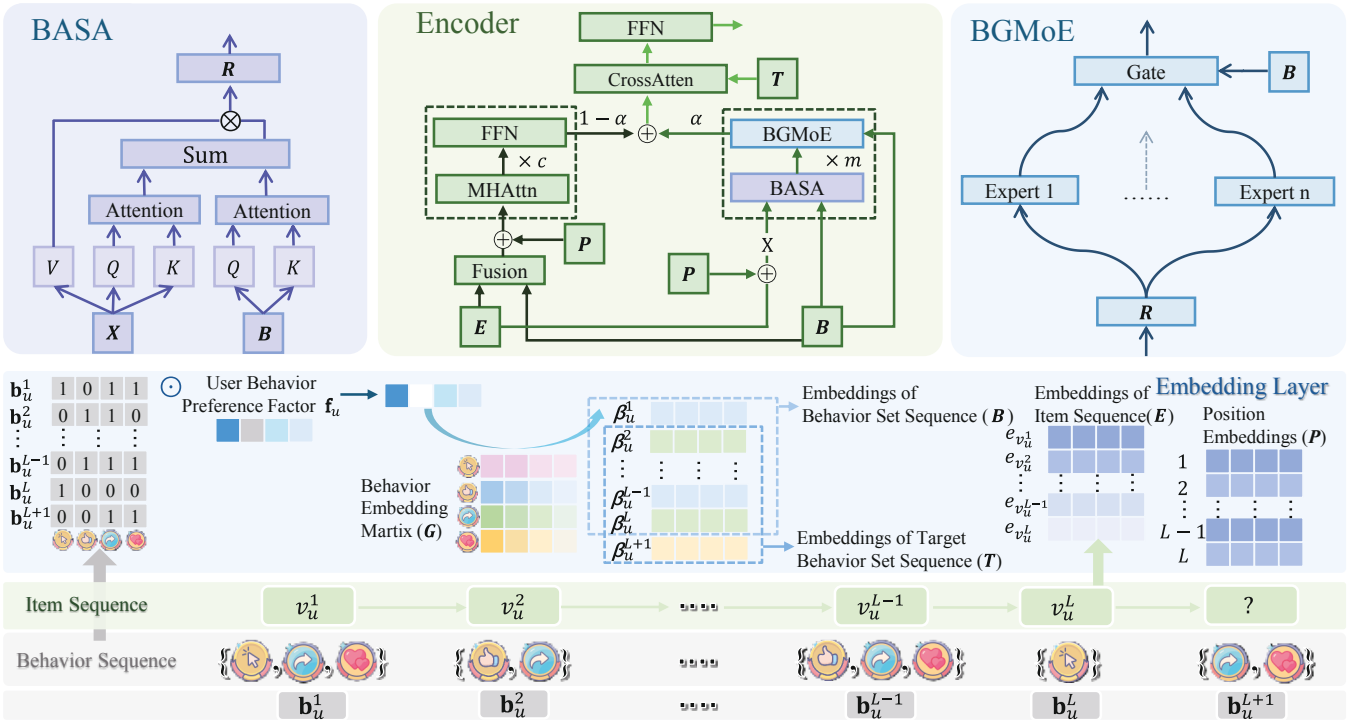

Figure 1: Overview of the proposed dual item-behavior fusion architecture, which integrates behavior information at both the input and intermediate layers. For clarity, residual connections and normalization layers are omitted in the illustration.

order, a learnable positional encoding $\boldsymbol{P} \in \mathbb{R}^{L\times d}$ is constructed. For behavior set encoding, following the design in EIDP (Chen, Pan, and Ming 2024), we distinguish different behavior types by introducing a behavior embedding matrix $\boldsymbol{G} \in \mathbb{R}^{|\mathcal{B}|\times d}$. Considering that different users may express varying preference intensities towards different behaviors, a factor matrix $\boldsymbol{F} \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{B}|}$ is used to model user behavioral preferences. For a given behavior set $\mathbf{b}_u^l$, the personalized behavior set embedding can be computed as:

$$\boldsymbol{\beta}_u^l = \text{softmax}(\mathbf{f}_u \odot \mathbf{b}_u^l) \cdot \boldsymbol{G}, \tag{1}$$

where $\odot$ is the element-wise product, $\mathbf{f}_u \in \mathbb{R}^{|\mathcal{B}|}$ denotes the behavioral preference factor of user u, and $\boldsymbol{\beta}_u^l \in \mathbb{R}^d$ can be viewed as the embedding corresponding to the behavior set $\mathbf{b}_u^l$. Note that, we will use the uppercase symbols $\boldsymbol{B}$ and $\mathfrak{B}$ to represent the matrix forms of $\boldsymbol{\beta}$ and $\mathbf{b}$, respectively. For brevity, we omit the user index u in the following derivations when there is no ambiguity.

Early Item-Behavior Fusion. We integrate the item embeddings $\boldsymbol{E}$ and the behavior set embeddings $\boldsymbol{B}$ at the input layer to enable rich interactions between items and behaviors, as follows:

$$\boldsymbol{E}' = f(\boldsymbol{E}, \boldsymbol{B}), \tag{2}$$

where the function $f(\cdot)$ can be instantiated as summation, concatenation, or gating. Then we adopt a widely used self-attentive architecture, i.e., Transformers (Vaswani et al. 2017). Specifically, it consists of stacks of multi-head self-attention layers (denoted by MHAttn($\cdot$)) and point-wise feed-forward networks (denoted by FFN($\cdot$)). The input and the output can be formalized as follows:

$$\boldsymbol{X}_e = \boldsymbol{E}' + \boldsymbol{P}, \tag{3}$$

$$\boldsymbol{F} = \text{FFN}(\text{MHAttn}(\boldsymbol{X}_e)). \tag{4}$$

The output $\boldsymbol{F} \in \mathbb{R}^{L\times d}$ serves as the contextualized representation after early item-behavior fusion.

Intermediate Item-Behavior Fusion. To better capture behavior semantics, we extend the Transformer with two components: (1) a behavior-aware self-attention (BASA) module, inspired by (Kim et al. 2025), and (2) a behavior-guided mixture-of-experts (BGMoE) module (Jacobs et al. 1991). The input is constructed by summing the item and position embeddings, i.e., $\boldsymbol{X} = \boldsymbol{E} + \boldsymbol{P}$. Meanwhile, the behavior set embeddings $\boldsymbol{B}$ influence the encoding process by implicitly guiding representation learning. In BASA, behavior-informed queries and keys modulate attention scores, assigning higher weights to items associated with similar behavior sets. In BGMoE, expert weights are dynamically computed based on behavior set embeddings, enabling the model to adaptively highlight the representations of diverse behavioral semantics during user modeling.

BASA. For the h-th head, the behavior-aware attention matrix is computed by projecting $\boldsymbol{X}$ and $\boldsymbol{B}$ into query/key spaces, respectively:

$$\boldsymbol{A}_h = \boldsymbol{Q}_h^X (\boldsymbol{K}_h^X)^\top + \boldsymbol{Q}_h^B (\boldsymbol{K}_h^B)^\top. \tag{5}$$

Then, the behavior-aware attention output is:

$$\boldsymbol{R}_h = \left(\text{softmax}\left(\frac{\boldsymbol{A}_h}{\sqrt{d_h}}\right) \odot \Delta\right) \boldsymbol{V}_h^X, \tag{6}$$

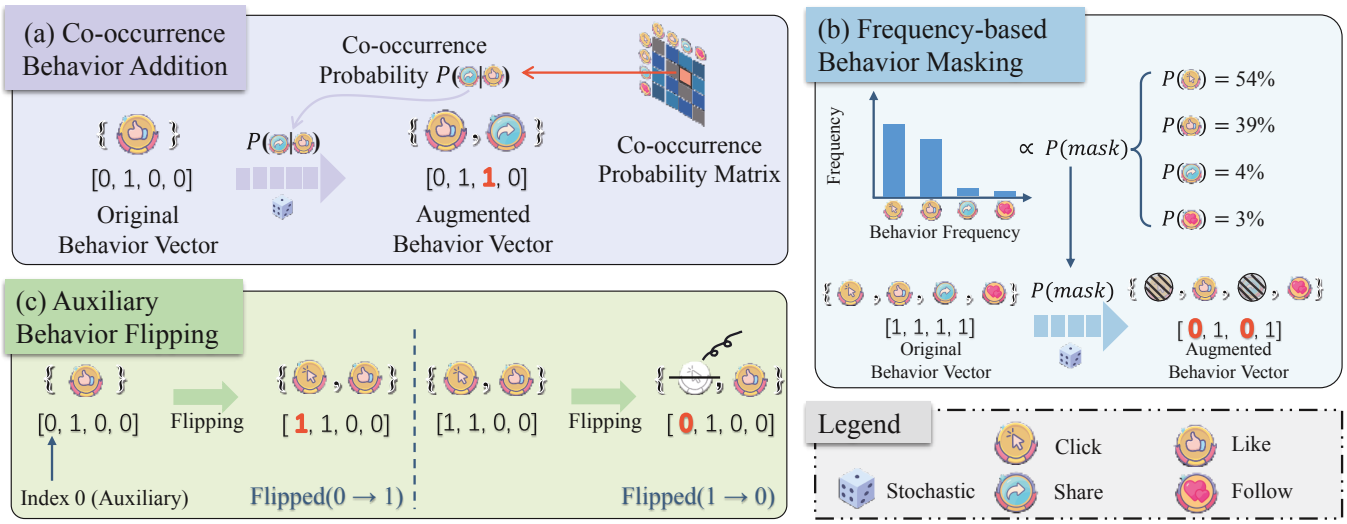

Figure 2: Overview of the proposed behavior-level data augmentation methods: (a) Co-occurrence behavior addition adds an extra behavior based on co-occurrence frequency; (b) Frequency-based behavior masking masks behaviors according to their occurrence frequency; (c) Auxiliary behavior flipping removes or adds an auxiliary behavior.

where $\boldsymbol{V}_h^X$ is the value projection from $\boldsymbol{X}$ and Δ is the lower triangular matrix of causality mask. Outputs from all heads are concatenated and projected to form $\boldsymbol{R}$.

BGMoE. To facilitate the fusion of behavior information, we apply a MoE module where the behavior set embedding guides expert weights. Given the behavior set embeddings $\boldsymbol{B}$ and the attention output $\boldsymbol{R}$, the output is computed as:

$$\boldsymbol{O} = \sum_{i=1}^{n} \phi(\boldsymbol{B})_i e_i(\boldsymbol{R}), \tag{7}$$

where $\phi(\cdot)$ computes routing weights via linear projection followed by softmax normalization, and $e_i(\cdot)$ denotes the i-th expert implemented by an FFN.

User Representation. The fused representation $\check{\boldsymbol{U}}$ is obtained by aggregating the outputs from early and intermediate fusion as follows:

$$\check{\boldsymbol{U}} = \alpha \boldsymbol{O} + (1-\alpha)\boldsymbol{F}, \tag{8}$$

where $\check{\boldsymbol{U}} \in \mathbb{R}^{L\times d}$ and α denotes the representation aggregating hyperparameter. To align user preferences with the next-step behavior set semantics, we apply a cross-attention mechanism between the fused representation $\check{\boldsymbol{U}}$ and the next-step behavior set embeddings $\boldsymbol{T} = [\boldsymbol{\beta}^2; \ldots; \boldsymbol{\beta}^L, \boldsymbol{\beta}^{L+1}]$. Specifically, we treat the $\boldsymbol{T}$ as the query, and the fused representation $\check{\boldsymbol{U}}$ as both the key and value to compute the final user representation $\boldsymbol{U}$.

$$\boldsymbol{U} = \text{FFN}(\text{CrossAttn}(\boldsymbol{T}, \check{\boldsymbol{U}})), \tag{9}$$

$$\text{CrossAttn}(\boldsymbol{T}, \check{\boldsymbol{U}}) = (\text{softmax}\left(\frac{\check{\boldsymbol{Q}}\check{\boldsymbol{K}}^\top}{\sqrt{d}}\right) \odot \Delta)\check{\boldsymbol{V}}, \tag{10}$$

where $\check{\boldsymbol{Q}} = \boldsymbol{T}\boldsymbol{W}_{\check{Q}} \in \mathbb{R}^{L\times d}$, $\check{\boldsymbol{K}} = \check{\boldsymbol{U}}\boldsymbol{W}_{\check{K}} \in \mathbb{R}^{L\times d}$, and $\check{\boldsymbol{V}} = \check{\boldsymbol{U}}\boldsymbol{W}_{\check{V}} \in \mathbb{R}^{L\times d}$. Here, $\boldsymbol{W}_{\check{Q}}, \boldsymbol{W}_{\check{K}}, \boldsymbol{W}_{\check{V}}$ are learnable projection matrices.

3.4 Behavior-Level Data Augmentation

To mitigate data sparsity, we propose three behavior-level data augmentation methods. These operate on user behavior sequences instead of core item sequences to generate diverse interaction views, thereby enhancing model generalization while preserving the semantics of item sequences. Additionally, all three methods are frequency-aware, tending to add low-frequency behaviors and mask high-frequency ones, thus alleviating the effects of behavioral imbalance.

Given a behavior sequence $\mathfrak{B} = [\mathbf{b}^1, \mathbf{b}^2, \ldots, \mathbf{b}^L]$, we first sample a subset of steps for augmentation based on a predefined operation ratio $\rho \in (0, 1)$. Let $\mathcal{I} = \{i_1, i_2, \ldots, i_k\}$ denote the sampled index set, where $k = \lfloor \rho \cdot L \rfloor$. One of the proposed augmentation methods is then applied to the behavior sets at these positions, yielding an augmented behavior sequence $\mathfrak{B}_* = [\mathbf{b}_*^1, \mathbf{b}_*^2, \ldots, \mathbf{b}_*^L]$, where $\mathbf{b}_*^l = \mathbf{b}^l$ if $l \notin \mathcal{I}$, and otherwise $\mathbf{b}_*^l$ is the augmented version of $\mathbf{b}^l$. Figure 2 illustrates the augmentation methods.

Co-occurrence Behavior Addition. To enhance the diversity of behavior combinations and simulate joint behavior patterns (e.g., "like + favorite"), this method supplements the original behavior set with frequently co-occurring yet currently missing behaviors. Specifically, given a co-occurrence probability matrix $\mathbf{M} \in \mathbb{R}^{|\mathcal{B}|\times|\mathcal{B}|}$ and a behavior set $\mathbf{b}$, we compute an aggregated co-occurrence vector as:

$$\mathbf{p} = \mathbf{b} \cdot \mathbf{M}. \tag{11}$$

To exclude the influence of already-present behaviors, we set $\mathbf{p}_k = 0$ for all k where $\mathbf{b}_k = 1$. Then we normalize $\mathbf{p}$ into a probability distribution:

$$\mathbf{p} \leftarrow \frac{\mathbf{p}}{\sum_{k=1}^{|\mathcal{B}|} \mathbf{p}_k}. \tag{12}$$

Finally, a new behavior b^+ is sampled from the distribution $\mathbf{p}$ and added to the set by setting $\mathbf{b}_{b^+} = 1$.

Frequency-based Behavior Masking. To prevent the model from overfitting to high-frequency behaviors and thereby weakening its ability to model long-tail behaviors, we dynamically mask frequently occurring behaviors, thus encouraging the model to focus more on informative long-tail behaviors. Specifically, we first compute a behavior frequency vector $\mathbf{m} \in \mathbb{R}^{|\mathcal{B}|}$. For a given behavior set $\mathbf{b}$, the masking probability for behavior type i is defined as:

$$P(\mathbf{b}_i = 0) = \frac{\mathbf{m}_i^c}{\sum_{k=1}^{|\mathcal{B}|} \mathbf{m}_k^c}, \tag{13}$$

where the exponent c is introduced to smooth the influence of extremely high-frequency behaviors. Each behavior type is then independently masked with its corresponding probability; if selected, we set $\mathbf{b}_i = 0$.

Auxiliary Behavior Flipping. Auxiliary behaviors (e.g., clicks) are common forms of implicit feedback, yet they may introduce noise. To prevent the model from overfitting to such behaviors, we flip the auxiliary behavior b_a in a given behavior set $\mathbf{b}$ as follows:

$$\mathbf{b}_{b_a} = 1 - \mathbf{b}_{b_a}. \tag{14}$$

3.5 Prediction and Model Training

Preferred Item Prediction. In the training phase, predict the next item that user u may interact with under the target behavior set at the (l+1)-th step:

$$\hat{y}_{l+1,v} = \boldsymbol{u}_l \boldsymbol{e}_v^\top, \tag{15}$$

where $\boldsymbol{u}_l$ represents the user representation $\boldsymbol{U}$ at the l-th step, $\hat{y}_{l+1,v}$ is a scalar that signifies the probability score of interacting with item v.

Training and Optimization. We utilize binary cross-entropy (BCE) loss to optimize our model for the next-item prediction task. Additionally, we introduce a behavior richness-based loss weighting to assign higher loss weights to prediction steps where the target behavior set contains multiple behavior types. Let $w_{u,l} = \frac{\|\mathbf{b}_{u,l+1}\|_0}{|\mathcal{B}|}$, which increases the penalty for prediction errors under richer behavior supervision. The formula for the loss is:

$$\mathcal{L}_{\text{next}} = -\frac{1}{|\delta(v)|} \sum_{u \in \mathcal{U}} \sum_{l=1}^{L} \delta(v_u^l) w_{u,l} \Big[\log \sigma(\hat{y}_{l+1,v_u^l}) + \log\big(1 - \sigma(\hat{y}_{l+1,j})\big) \Big], \tag{16}$$

where $\sigma(\cdot)$ denotes the sigmoid function and subscript $j \in \mathcal{V} \setminus \mathcal{S}_u$ denotes a randomly sampled negative item. $\delta(v_u^l)$ is an indicator function: $\delta(v_u^l) = 1$ if v_u^l is a real item and $\delta(v_u^l) = 0$ if it is a padding item. $|\delta(v)|$ represents the total number of valid ground-truth items across all sequences.

To further enhance representation learning, we introduce a sequence-level contrastive loss to enforce consistency between two augmented views of the same user interaction sequence. Given an original sequence $\mathcal{S}_u$, we generate two augmented sequences $\mathcal{S}_u^{aug1}$ and $\mathcal{S}_u^{aug2}$, which are independently encoded into $\boldsymbol{H}_u^1, \boldsymbol{H}_u^2 \in \mathbb{R}^{L \times d}$. We then concatenate the representations across all steps to obtain $\boldsymbol{h}_u^1, \boldsymbol{h}_u^2 \in \mathbb{R}^{L \cdot d}$. The contrastive learning loss is defined as:

$$\mathcal{L}_{\text{SeqCL}} = \mathcal{L}_{\text{CL}}(\boldsymbol{h}_u^1, \boldsymbol{h}_u^2) + \mathcal{L}_{\text{CL}}(\boldsymbol{h}_u^2, \boldsymbol{h}_u^1), \tag{17}$$

$$\mathcal{L}_{\text{CL}}(\boldsymbol{h}_u^1, \boldsymbol{h}_u^2) = -\log \frac{\exp\big(\text{sim}(\boldsymbol{h}_u^1, \boldsymbol{h}_u^2)/\tau\big)}{\sum_{neg} \exp\big(\text{sim}(\boldsymbol{h}_u^1, \boldsymbol{h}_{neg})/\tau\big)}, \tag{18}$$

where $\text{sim}(\cdot)$ denotes the dot product operation, $\boldsymbol{h}_{neg}$ denotes augmented representations from other sequences within the current mini-batch and τ is a temperature parameter. The final objective combines both losses:

$$\mathcal{L} = \mathcal{L}_{\text{next}} + \lambda \mathcal{L}_{\text{SeqCL}}, \tag{19}$$

where λ is a balancing hyperparameter that controls the contribution of the contrastive loss.

4 Experiments

4.1 Experimental Settings

Datasets. We conduct experiments on the recommendation subset of KuaiSAR (Sun et al. 2023) and the Tenrec dataset (Yuan et al. 2022b). i) **KuaiSAR.** The items are short videos. We treat click as the auxiliary behavior, and like, share and follow as target behaviors. ii) **QK-Article.** The items are articles. We treat read as the auxiliary behavior, and like, share, favorite and follow as target behaviors. iii) **QK-Video.** The items are short videos. We treat click as the auxiliary behavior, and like, share and follow as target behaviors. Dataset statistics are summarized in Table 2 and the details of data preprocessing are provided in code link.

Evaluation Metrics. We use two widely adopted ranking-oriented evaluation metrics, i.e., hit ratio (HR@k) and normalized discounted cumulative gain (NDCG@k), where $k \in \{5, 10\}$. We adopt the full-ranking setting in evaluation.

Baselines. We compare BLADE with three categories of baselines: (1) **Single-Behavior Sequential Recommendation**: SASRec (Kang and McAuley 2018) and CL4SRec (Xie et al. 2022); (2) **Multi-Behavior Sequential Recommendation**: DyMuS (Cho et al. 2023), MBHT (Yang et al. 2022) and MB-STR (Yuan et al. 2022a); (3) **Behavior Set-informed Sequential Recommendation**: EIDP (Chen, Pan, and Ming 2024). Baseline descriptions are in code link. Following EIDP, we retain only the most preferred behavior in each set based on global frequency, reducing BSSR to a standard MBSR setup. Necessary adaptations for MBSR models are detailed in code link.

Implementation Details. For a fair comparison, we fix the embedding dimension d to 32 and the sequence length L to 50 for all models. All models are trained on the same truncated sequences. For BLADE, we tune the number of stacked blocks in both early and intermediate fusion from $\{2, 3\}$, the number of attention heads from $\{2, 4, 8\}$, the dropout rate from $\{0.2, 0.3, 0.4, 0.5\}$, and the number of experts in BGMoE from $\{4, 6, 8\}$. More details are provided in code link.

4.2 Overall Performance

We compare the proposed model with baseline models across three datasets. Table 1 summarizes the overall performance. Due to space limitations, the results of BLADE reported in Table 1 are the best among the three augmentations. From the results, we summarize the observations

Dataset	KuaiSAR				QK-Article				QK-Video			
Metrics	NDCG@5	HR@5	NDCG@10	HR@10	NDCG@5	HR@5	NDCG@10	HR@10	NDCG@5	HR@5	NDCG@10	HR@10
SASRec	0.0115	0.0180	0.0159	0.0320	0.0176	0.0289	0.0255	0.0534	0.0078	0.0126	0.0104	0.0204
CL4SRec	0.0096	0.0160	0.0132	0.0276	0.0187	0.0303	0.0257	0.0522	0.0082	0.0126	0.0109	0.0208
DyMuS	0.0020	0.0039	0.0030	0.0073	0.0057	0.0099	0.0075	0.0155	0.0050	0.0077	0.0072	0.0146
DyMuS$^+$	0.0024	0.0045	0.0036	0.0081	0.0082	0.0141	0.0113	0.0238	0.0033	0.0056	0.0058	0.0133
MBHT	0.0110	0.0181	0.0151	0.0312	0.0193	0.0314	0.0250	0.0491	0.0051	0.0081	0.0078	0.0165
MB-STR	0.0077	0.0131	0.0109	0.0228	0.0156	0.0245	0.0221	0.0449	0.0080	0.0130	0.0116	0.0243
EIDP	0.0122	0.0184	0.0169	0.0331	0.0198	0.0317	0.0288	0.0599	0.0095	0.0159	**0.0137**	**0.0291**
BLADE	**0.0135**	**0.0218**	**0.0187**	**0.0380**	**0.0215**	**0.0354**	**0.0301**	**0.0621**	**0.0097**	**0.0161**	0.0127	0.0256

Table 1: Performance comparison on KuaiSAR, QK-Article and QK-Video datasets with NDCG@5/10 and HR@5/10. The best results are marked in bold, and the second best results are underlined.

Dataset	#Users	#Items	#Interactions
KuaiSAR	3,812	10,653	528,242
QK-Article	5,081	13,788	252,069
QK-Video	5,081	20,494	150,396

Table 2: Statistics of datasets used in experiments

as follows: (1) Although MBSR methods introduce additional behavioral information, they do not consistently outperform traditional SR models. In BSSR settings, behavior sets often exhibit higher-order heterogeneous dependencies, which MBSR models—primarily relying on single behavior labels—struggle to capture effectively. Among them, DyMuS performs poorly on all three datasets, likely due to its late fusion strategy that weakly models item-item and behavior-behavior interactions, especially in BSSR settings with richer behavior information. (2) Model performance varies notably across datasets. MBHT excels on KuaiSAR and QK-Article but underperforms on QK-Video, whereas MB-STR shows the opposite trend. This suggests that different models adapt differently to behavioral patterns and scenario-specific characteristics. (3) EIDP, tailored for BSSR, outperforms all MBSR baselines by fully exploiting information within behavior sets, while MBSR methods simplify the behavior set into a single behavior and lose key semantic dependencies. (4) Our BLADE achieves the best performance on most metrics, confirming the effectiveness of dual item-behavior fusion modeling and behavior-level data augmentation in modeling complex user behavioral preferences, alleviating data sparsity, and mitigating behavior distribution imbalance.

4.3 Ablation Study and Component Analysis

We validate the effectiveness of the key components of BLADE on three datasets, KuaiSAR, QK-Article, and QK-Video, through the ablation study as shown in Table 3. (i) The proposed dual item-behavior fusion architecture outperforms variants using only Early Fusion (EF) or Intermediate Fusion (IF), suggesting that the two strategies capture distinct yet complementary aspects of user behavior and their combination enables more comprehensive preference modeling. (ii) Removing the contrastive loss leads to a substantial drop in performance, demonstrating the importance of contrastive learning over augmented user sequences. This result also highlights the effectiveness of our behavior-level data augmentation methods. (iii) The behavior richness-based loss weighting improves performance by assigning higher weights to target items with multiple behaviors. Replacing it with standard BCE results in a performance drop, highlighting the advantage of leveraging richer supervision for next-item prediction.

Dataset	Metric	w/o EF	w/o IF	w/o CL	w/o BRW	BLADE
KuaiSAR	NDCG@5	0.0105	0.0120	0.0107	0.0071	**0.0135**
	HR@5	0.0157	0.0199	0.0171	0.0128	**0.0218**
QK-Article	NDCG@5	0.0215	0.0215	0.0189	0.0208	**0.0215**
	HR@5	0.0352	0.0348	0.0318	0.0350	**0.0354**
QK-Video	NDCG@5	0.0091	0.0081	0.0078	0.0074	**0.0097**
	HR@5	0.0146	0.0138	0.0128	0.0120	**0.0161**

Table 3: Ablation study on KuaiSAR, QK-Article and QK-Video datasets, where "w/o" denotes the removal of the corresponding module in BLADE

4.4 Augmentation Method Comparison

We analyze how different data augmentation operators and their proportion parameters affect model performance as shown in Figure 5. To study the effect of each augmentation strategy, we apply only one type of augmentation at a time during contrastive learning. The figure illustrates the performance trends of the three augmentation methods as the proportion parameter p varies from 0.1 to 0.9. We make the following observations: BLADE with any of the proposed augmentation methods consistently outperforms the version without augmentation across most proportion settings and on both datasets. This demonstrates the effectiveness and robustness of our behavior-level data augmentation methods, as they introduce implicit self-supervised signals embedded in the original data and enrich interaction views by operating user behavior sequences rather than core item sequences, thereby significantly enhancing model generalization while preserving item sequence semantics.

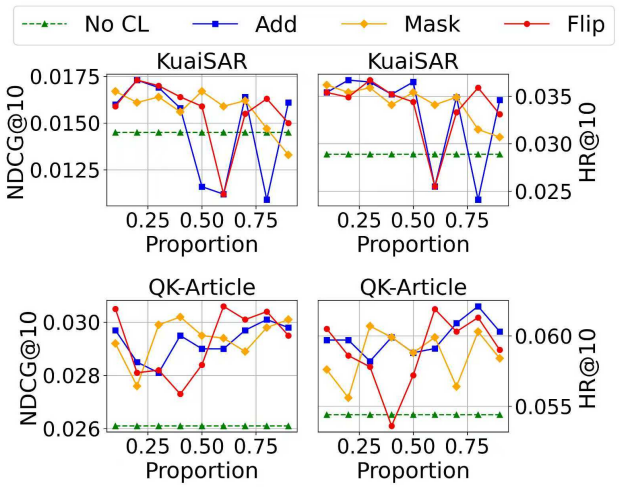

Figure 3: Performance impact of different augmentation methods and operation ratio p.

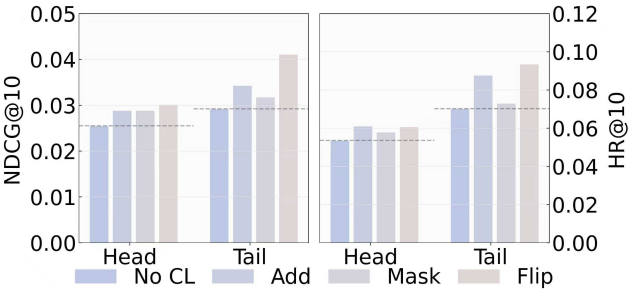

Figure 4: Performance comparison of three data augmentation methods based on the long-tail degree of user interaction behaviors on the QK-Article dataset.

4.5 Impact of Behavior-Level Data Augmentation on Long-Tail Behavior

In Figure 4, we evaluate the model's performance on the QK-Article dataset by dividing the test user sequences into two groups: the tail group consists of samples where the proportion of long-tail behaviors (share and follow) in users' interaction behaviors exceeds 80%, while the remaining samples are categorized into the head group. This partition allows us to evaluate the model's capability in modeling long-tail behaviors. The experimental results show that BLADE consistently outperforms variants without data augmentation and contrastive learning in both groups. Notably, the performance gain is more pronounced in the tail group, indicating that our method is particularly effective at enhancing the model's capacity to model long-tail behaviors. This improvement is mainly due to our frequency-based augmentation methods, which add low-frequency behaviors or mask high-frequency ones, thereby alleviating the imbalance in the original behavior distribution and improving the modeling of long-tail behaviors.

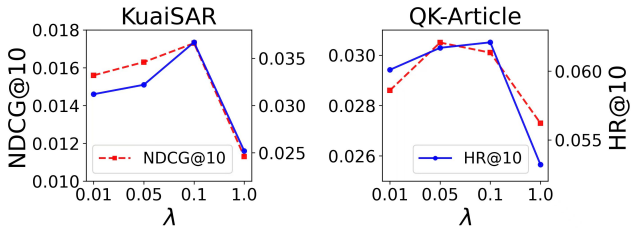

Figure 5: Performance with varying contrastive loss balancing hyperparameter λ.

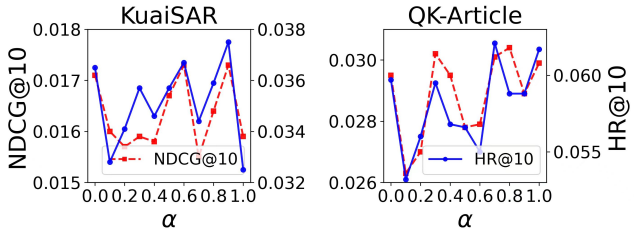

Figure 6: Performance with varying representation aggregation hyperparameter α.

4.6 Hyperparameter Sensitivity Analysis

Loss Balancing Hyperparameter. Figure 5 illustrates the impact of the contrastive loss balancing hyperparameter λ on the KuaiSAR and QK-Article datasets. The results show that the model achieves the best performance when $\lambda \approx 0.1$, while overly small or large values result in suboptimal results. A small λ introduces insufficient contrastive supervision, whereas a large λ overemphasizes the contrastive objective, thereby impairing the model's ability to perform accurate next-item prediction.

Representation Aggregating Hyperparameter. Figure 6 reports the effect of the representation aggregating hyperparameter α on model performance across both KuaiSAR and QK-Article. We observe notable performance fluctuations as α varies, clearly suggesting that the model is highly sensitive to α. These results indicate that early and intermediate fusion are not strictly complementary. An inappropriate setting of α may lead to suboptimal aggregation of early and intermediate fusion representations.

5 Conclusion

In this paper, we propose BLADE, a behavior-level data augmentation framework with dual fusion modeling. To address multi-behavior heterogeneity, BLADE introduces a dual item-behavior fusion architecture that incorporates behavioral information at both the input layer and intermediate layer thereby enhancing semantic representation learning across diverse behaviors. To mitigate data sparsity, BLADE introduces three behavior-level data augmentation methods, which operate on behavior sequences while preserving the semantics of item sequences. Experiments on real-world datasets validate its effectiveness.

Acknowledgements

This research was supported by grants from the National Natural Science Foundation of China (No. 62502486), the grants of Provincial Natural Science Foundation of Anhui Province (No. 2408085QF193), USTC ResearchFunds of the DoubleFirst-Class Initiative (No. YD2150002501), the Fundamental Research Funds for the Central Universities of China (No. WK2150110032).

References

Chen, M.; Pan, W.; and Ming, Z. 2024. Explicit and implicit modeling via dual-path transformer for behavior set-informed sequential recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 329–340.

Chen, W.; He, M.; Ni, Y.; Pan, W.; Chen, L.; and Ming, Z. 2022a. 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*, 268–277.

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

Cheng, M.; Liu, Z.; Liu, Q.; Ge, S.; and Chen, E. 2022. Towards automatic discovering of deep hybrid network architecture for sequential recommendation. In *Proceedings of the ACM Web Conference 2022*, 1923–1932.

Cheng, M.; Luo, Y.; Ouyang, J.; Liu, Q.; Liu, H.; Li, L.; Yu, S.; Zhang, B.; Cao, J.; Ma, J.; et al. 2025. A survey on knowledge-oriented retrieval-augmented generation. *arXiv preprint arXiv:2503.10677*.

Cheng, M.; Yuan, F.; Liu, Q.; Xin, X.; and Chen, E. 2021. Learning transferable user representations with sequential behaviors via contrastive pre-training. In *2021 IEEE International Conference on Data Mining (ICDM)*, 51–60. IEEE.

Cho, J.; Hyun, D.; won Lim, D.; jae Cheon, H.; Park, H.-i.; and Yu, H. 2023. Dynamic multi-behavior sequence modeling for next item recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 4199–4207.

Fang, H.; Zhang, D.; Shu, Y.; and Guo, G. 2020. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Transactions on Information Systems (TOIS)*, 39(1): 1–42.

Gong, S.; Liu, Y.; Dang, Y.; Guo, G.; Zhao, J.; and Wang, X. 2025. Multiple Purchase Chains with Negative Transfer Elimination for Multi-Behavior Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 11717–11725.

He, M.; Pan, W.; and Ming, Z. 2022. BAR: Behavior-aware recommendation for sequential heterogeneous one-class collaborative filtering. *Information Sciences*, 608: 881–899.

He, R.; and McAuley, J. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In *2016 IEEE 16th international conference on data mining (ICDM)*, 191–200. IEEE.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, 173–182.

He, Z.; Liu, W.; Guo, W.; Qin, J.; Zhang, Y.; Hu, Y.; and Tang, R. 2023. A survey on user behavior modeling in recommender systems. *arXiv preprint arXiv:2302.11087*.

Jacobs, R. A.; Jordan, M. I.; Nowlan, S. J.; and Hinton, G. E. 1991. Adaptive mixtures of local experts. *Neural computation*, 3: 79–87.

Kang, W.-C.; and McAuley, J. 2018. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, 197–206. IEEE.

Kim, H.-y.; Choi, M.; Lee, S.; Baek, I.; and Lee, J. 2025. DIFF: Dual Side-Information Filtering and Fusion for Sequential Recommendation. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1624–1633.

Lee, S.; Park, S.; and Lee, J. 2025. Exploiting Fine-Grained Skip Behaviors for Micro-Video Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 12004–12012.

Li, J.; Ren, P.; Chen, Z.; Ren, Z.; Lian, T.; and Ma, J. 2017. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 1419–1428.

Li, Z.; Zhao, H.; Liu, Q.; Huang, Z.; Mei, T.; and Chen, E. 2018a. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1734–1743.

Li, Z.; Zhao, H.; Liu, Q.; Huang, Z.; Mei, T.; and Chen, E. 2018b. Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 1734–1743.

Liu, Q.; Wu, S.; and Wang, L. 2017. Multi-behavioral sequential prediction with recurrent log-bilinear model. *IEEE Transactions on Knowledge and Data Engineering*, 29: 1254–1267.

Luo, J.; He, M.; Lin, X.; Pan, W.; and Ming, Z. 2022. Dual-task learning for multi-behavior sequential recommendation. In *Proceedings of the 31st ACM international conference on information & knowledge management*, 1379–1388.

Luo, Y.; Zhou, Y.; Cheng, M.; Wang, J.; Wang, D.; Pan, T.; and Zhang, J. 2025. Time Series Forecasting as Reasoning: A Slow-Thinking Approach with Reinforced LLMs. *arXiv preprint arXiv:2506.10630*.

Meng, W.; Yang, D.; and Xiao, Y. 2020. 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*, 1091–1100.

Qiu, R.; Huang, Z.; Yin, H.; and Wang, Z. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In *Proceedings of the fifteenth ACM international conference on web search and data mining*, 813–823.

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

Sun, F.; Liu, J.; Wu, J.; Pei, C.; Lin, X.; Ou, W.; and Jiang, P. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*, 1441–1450.

Sun, Z.; Si, Z.; Zang, X.; Leng, D.; Niu, Y.; Song, Y.; Zhang, X.; and Xu, J. 2023. KuaiSar: A unified search and recommendation dataset. In *Proceedings of the 32nd ACM international conference on information and knowledge management*, 5407–5411.

Tang, J.; and Wang, K. 2018. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the eleventh ACM international conference on web search and data mining*, 565–573.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Wang, J.; Cheng, M.; and Liu, Q. 2025. Can slow-thinking llms reason over time? empirical studies in time series forecasting. *arXiv preprint arXiv:2505.24511*.

Wang, S.; Hu, L.; Wang, Y.; Cao, L.; Sheng, Q. Z.; and Orgun, M. 2019. Sequential recommender systems: challenges, progress and prospects. *arXiv preprint arXiv:2001.04830*.

Wang, W.; Ma, J.; Zhang, Y.; Zhang, K.; Jiang, J.; Yang, Y.; Zhou, Y.; and Zhang, Z. 2025. Intent Oriented Contrastive Learning for Sequential Recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 12748–12756.

Wang, W.; Zhang, W.; Liu, S.; Liu, Q.; Zhang, B.; Lin, L.; and Zha, H. 2020. Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction. In *Proceedings of the web conference 2020*, 3056–3062.

Xia, L.; Huang, C.; Xu, Y.; Dai, P.; Zhang, X.; Yang, H.; Pei, J.; and Bo, L. 2021. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 4486–4493.

Xia, L.; Huang, C.; Xu, Y.; and Pei, J. 2022. Multi-behavior sequential recommendation with temporal graph transformer. *IEEE Transactions on Knowledge and Data Engineering*, 35: 6099–6112.

Xiao, J.; Pan, W.; and Ming, Z. 2024. A generic behavior-aware data augmentation framework for sequential recommendation. In *Proceedings of the 47th international ACM SIGIR conference on research and development in information retrieval*, 1578–1588.

Xie, X.; Sun, F.; Liu, Z.; Wu, S.; Gao, J.; Zhang, J.; Ding, B.; and Cui, B. 2022. Contrastive learning for sequential recommendation. In *2022 IEEE 38th international conference on data engineering (ICDE)*, 1259–1273. IEEE.

Yang, Y.; Huang, C.; Xia, L.; Liang, Y.; Yu, Y.; and Li, C. 2022. Multi-behavior hypergraph-enhanced transformer for sequential recommendation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, 2263–2274.

Yuan, E.; Guo, W.; He, Z.; Guo, H.; Liu, C.; and Tang, R. 2022a. Multi-behavior sequential transformer recommender. In *Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval*, 1642–1652.

Yuan, G.; Yuan, F.; Li, Y.; Kong, B.; Li, S.; Chen, L.; Yang, M.; Yu, C.; Hu, B.; Li, Z.; et al. 2022b. Tenrec: A large-scale multipurpose benchmark dataset for recommender systems. *Advances in Neural Information Processing Systems*, 35: 11480–11493.

Zhan, Z.; He, M.; Pan, W.; and Ming, Z. 2022. TransRec++: Translation-based sequential recommendation with heterogeneous feedback. *Frontiers Comput. Sci.*, 16(2): 162615.

Zhang, S.; Chu, H.; Li, J.; Zhou, Y.; Wang, S.; and Sun, Q. 2025. DeMBR: Denoising Model with Memory Pruning and Semantic Guidance for Multi-Behavior Recommendation. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, 521–529.

Zhou, M.; Ding, Z.; Tang, J.; and Yin, D. 2018. Micro behaviors: A new perspective in e-commerce recommender systems. In *Proceedings of the eleventh ACM international conference on web search and data mining*, 727–735.