B-STAR: Behavior-aware Sequential Transformer with Adaptive Representations for Multi-behavior Recommendation

Kaiwen Zhou, Hongjuan Liu*
Software College, Northeastern University, Shenyang, Liaoning, China
2471459@stu.neu.edu.cn, liuhj@swc.neu.edu.cn

***Abstract*—Multi-behavior sequential recommendation (MBSR) leverages heterogeneous user interactions (e.g., views, purchases) to capture dynamic preferences. However, existing Transformer-based MBSR models still face three limitations: (1) they model different behaviors via simple additive embeddings or separate sequences, failing to adaptively capture the varying influence between behavior pairs; (2) they mainly rely on local sequential patterns and overlook high-order collaborative signals in the global user–item graph; and (3) they rarely exploit explicit negative feedback, missing useful information for delimiting user preference boundaries. We propose Behavior-aware Sequential Transformer with Adaptive Representations (B-STAR), which unifies global graph reasoning with local sequence modeling. A global graph encoder first injects high-order collaborative signals into item representations. A behavior-aware attention mechanism with a learnable interaction matrix then re-weights attention scores according to the affinity between behavior types. Furthermore, a hard negative push strategy, combined with a multi-task learning objective encourages the model to separate preferred and disliked items and yields more robust representations. Experiments on three real-world datasets (Yelp, Taobao, and Tianchi) show that B-STAR consistently outperforms recent state-of-the-art baselines. Visualizations of learned attention weights indicate that B-STAR captures meaningful behavioral patterns (e.g., the e-commerce purchase funnel).**

***Index Terms*—Sequential Recommendation, Multi-behavior Recommendation, Graph Neural Networks, Transformer, Negative Feedback**

I. Introduction

Recommender systems are essential to online platforms for alleviating information overload. Sequential recommendation (SR) models the temporal order of user–item interactions to predict the next item, yet most SR methods assume a single interaction type (e.g., purchase). In practical e-commerce and content platforms, user engagement is inherently heterogeneous, involving behaviors such as View, Favorite, Add-to-Cart, and Buy. This motivates multi-behavior sequential recommendation (MBSR), which leverages auxiliary behaviors to enrich supervision and mitigate sparsity of target behaviors.

Despite recent progress, we identify three limitations that remain insufficiently addressed (as shown in Fig. 1). *(i) Rigid behavior dependency modeling.* Existing methods often represent behaviors via separate sequences or additive embeddings [1], implicitly assuming uniform influence across behavior transitions, while in reality high-intent actions (e.g., Add-to-Cart) are more predictive of Buy than casual View. *(ii) Local-only sequential bias.* Transformer-based sequential encoders mainly capture within-sequence patterns and overlook high-order collaborative signals in the global user–item graph, which are crucial under sparse histories. *(iii) Under-utilization of explicit negative feedback.* In datasets such as Yelp, explicit dislikes/low ratings provide strong boundary signals, but are often discarded or treated as unobserved, blurring the separation between preferred and disliked items.

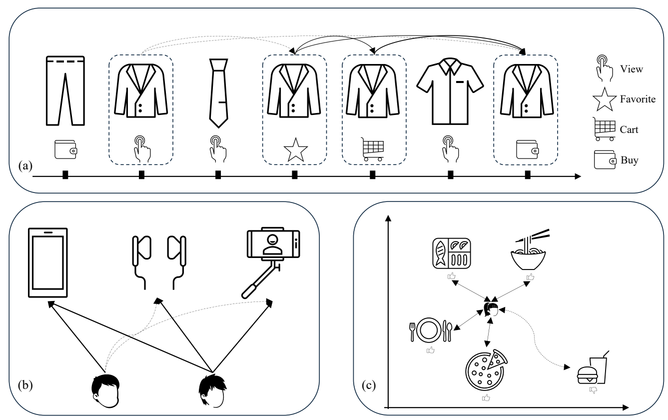

Fig. 1. Motivation of B-STAR. (a) Behavior dependencies: different behavior transitions carry varying intent strengths. (b) Local vs. Global: sequential models miss high-order collaborative signals. (c) Negative feedback: explicit dislikes provide crucial boundary information but are often ignored.

To address these issues, we propose Behavior-aware Sequential Transformer with Adaptive Representations **(B-STAR)**, which unifies global graph reasoning with local multi-behavior sequence modeling. B-STAR first injects high-order collaborative signals into item representations via a lightweight LightGCN-style global graph encoder. It then introduces a *behavior-aware attention* mechanism with a learnable behavior interaction matrix to adaptively bias self-attention according to cross-behavior affinities. Finally, we incorporate (when available) explicit negative feedback through a hard negative push loss, together with an auxiliary next-behavior prediction

*Corresponding author.

objective to improve behavior-sensitive representation learning.

The key contributions of this work are as follows:

- We propose B-STAR, a unified framework that integrates global user–item graph signals into multi-behavior sequential recommendation.
- We design a behavior-aware attention mechanism with a learnable behavior interaction matrix to model adaptive cross-behavior dependencies.
- We leverage explicit negative feedback via a hard negative push loss (when available) and introduce an auxiliary next-behavior prediction task for robust representation learning.

Code is publicly available at https://github.com/kaiwenzhou1/B-STAR-multi-behavior-recommendation.

II. Related Work

A. Sequential and Multi-Behavior Recommendation

Sequential recommendation predicts a user's next interaction from her historical sequence. Early methods mainly relied on Markov chains, factorization models, and later RNNs to capture temporal dependencies. More recently, Transformer-based models such as SASRec [2] and BERT4Rec [3] have shown strong ability in modeling long-range dependencies. However, most of these methods assume a single interaction type and treat all feedback as homogeneous and implicitly positive.

To exploit richer supervision signals, multi-behavior recommendation (MBR) incorporates auxiliary behaviors such as view and cart. Early studies like MB-GCN [4] modeled dependencies among behaviors through multi-task learning or graph propagation. In multi-behavior sequential recommendation (MBSR), models must capture both temporal dynamics and behavior heterogeneity. Representative methods include RNN-based BINN [5], multi-Transformer architectures such as DIPN [6] and DMT [7], hypergraph-based models like MB-GMN [8] and MBHT [9], and more recent approaches including PBAT [1], MB-STR [10], MISSL [11], MBSRec [12], GEAR [13]. These methods improve multi-behavior modeling through pattern-aware, self-supervised, generative, or preference-learning mechanisms. Despite the progress, most existing MBSR methods still encode behavior types with simple additive embeddings, extra tokens, or separate behavior streams, without explicitly modeling *adaptive* attention biases between behavior pairs within self-attention. As a result, they are limited in distinguishing different influence strengths among behavior transitions. Moreover, explicit negative feedback is rarely utilized and is often ignored or treated as missing data, leaving user disinterest boundaries insufficiently modeled.

B. Graph Learning for Recommendation

Graph neural networks (GNNs) have shown immense power in capturing high-order collaborative signals. LightGCN [14] explicitly model the connectivity between users and items, addressing the data sparsity issue effectively by propagating embeddings on the user-item bipartite graph. GCSAN [15] attempted to combine GNNs with sequential models by constructing session graphs. Recent multi-behavior works have also tried to fuse temporal signals into graphs. However, efficiently integrating global graph structure into multi-behavior sequential modeling without excessive complexity remains challenging.

III. Preliminaries

A. Notation

Let $\mathcal{U}$ and $\mathcal{V}$ denote the sets of users and items, respectively. Each user–item interaction is associated with a behavior type. We define the behavior set as $\mathcal{B} = \{b_1, b_2, \ldots, b_K\}$, where each $b_k \in \mathcal{B}$ denotes a specific interaction type (e.g., Page View, Favorite, Cart, Buy). For datasets with explicit feedback (e.g., Yelp), $\mathcal{B}$ may also include negative behaviors such as Dislike or Negative Rating.

For each user $u \in \mathcal{U}$, her chronological multi-behavior interaction sequence is denoted as

$$S_u = \{(v_1, b_1), (v_2, b_2), \ldots, (v_T, b_T)\}, \tag{1}$$

where $v_t \in \mathcal{V}$ and $b_t \in \mathcal{B}$ indicate that user u interacted with item v_t through behavior type b_t at position t, and T is the sequence length.

B. Problem Definition

We build a global user–item interaction graph $\mathcal{G} = (\mathcal{U} \cup \mathcal{V}, \mathcal{E})$ from all observed interactions, where an edge $(u, v) \in \mathcal{E}$ means that user u has interacted with item v under at least one behavior type. Given a user's historical sequence S_u and the global graph $\mathcal{G}$, our goal is to predict the next item v_{T+1} that user u is most likely to interact with under a **target behavior** b_{tar} (e.g., Buy or Like).

Formally, we learn a scoring function $f(\cdot)$ that estimates, for each candidate item $v \in \mathcal{V}$, the probability that v will be the next item interacted with by user u via the target behavior b_{tar}:

$$P(v_{T+1} = v \mid S_u, \mathcal{G}, b_{\text{tar}}) = f(v, S_u, b_{\text{tar}}; \mathcal{G}). \tag{2}$$

IV. Methodology

Fig. 2 shows the architecture of B-STAR, which contains three components: (i) an **Input & Global Graph Encoder** for capturing high-order collaborative signals from the global user–item graph; (ii) a **Behavior-Aware Transformer** for modeling dynamic preferences with adaptive behavior dependencies; and (iii) a **Multi-Task Prediction Module** with a Hard Negative Push strategy to refine user representations.

A. Input & Global Graph Encoder

Sequential models based only on ID embeddings often suffer from sparsity and limited receptive fields. To address this, we construct a global user–item interaction graph $\mathcal{G} = (\mathcal{U} \cup \mathcal{V}, \mathcal{E})$ and propagate information on it to obtain graph-enhanced item representations.

We use a simplified graph convolutional network following LightGCN [14]. Let $\mathbf{e}_u^{(0)} \in \mathbb{R}^d$ and $\mathbf{e}_v^{(0)} \in \mathbb{R}^d$ denote the

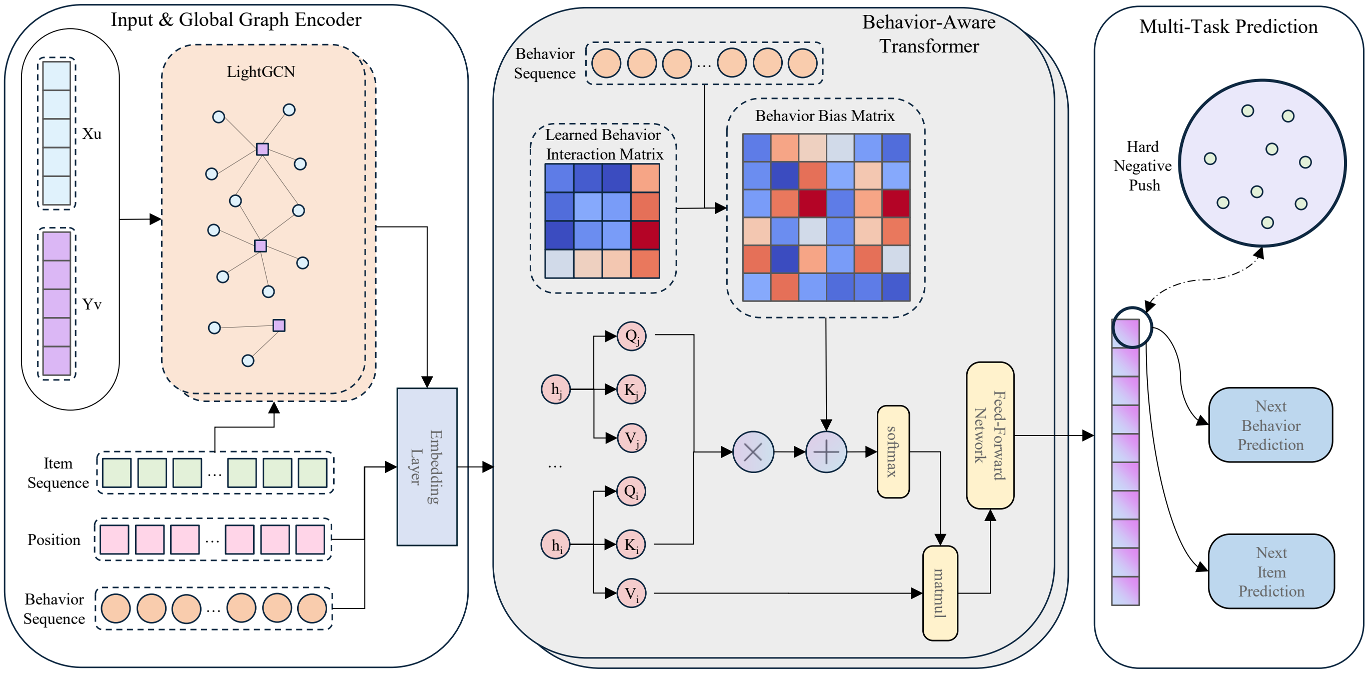

Fig. 2. The overall architecture of B-STAR, comprising the Global Graph Encoder, Behavior-Aware Transformer, and Multi-Task Prediction Module.

initial embeddings of user u and item v. Node embeddings are updated by symmetrically normalized neighbor aggregation:

$$\mathbf{e}_u^{(l+1)} = \sum_{i\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|\,|\mathcal{N}_i|}} \mathbf{e}_i^{(l)}, \tag{3}$$

$$\mathbf{e}_v^{(l+1)} = \sum_{u\in\mathcal{N}_v} \frac{1}{\sqrt{|\mathcal{N}_v|\,|\mathcal{N}_u|}} \mathbf{e}_u^{(l)}, \tag{4}$$

where $\mathcal{N}_u$ and $\mathcal{N}_v$ are the neighbor sets of user u and item v. Following LightGCN, we remove non-linear activations and self-loops to reduce complexity and alleviate over-smoothing. After L layers, the final graph-based item representation is

$$\mathbf{e}_v^{\text{global}} = \frac{1}{L+1}\sum_{l=0}^{L} \mathbf{e}_v^{(l)}. \tag{5}$$

These graph-enhanced item embeddings $\mathbf{e}_v^{\text{global}}$ initialize the sequential encoder, injecting high-order collaborative information beyond an individual sequence.

B. Behavior-Aware Transformer

1) Multi-Feature Embedding Layer: For a user u with interaction sequence $S_u = \{(v_1, b_1), \ldots, (v_T, b_T)\}$, we build the input at each position t by combining item, position, and behavior information. Let $\mathbf{P} \in \mathbb{R}^{T\times d}$ be the positional embedding matrix and $\mathbf{B} \in \mathbb{R}^{|\mathcal{B}|\times d}$ the behavior embedding matrix. The input vector at step t is

$$\mathbf{h}_t^{(0)} = \mathbf{e}_{v_t}^{\text{global}} + \mathbf{P}_t + \mathbf{B}_{b_t}, \tag{6}$$

where $\mathbf{P}_t$ is the positional embedding and $\mathbf{B}_{b_t}$ is the embedding of behavior type b_t. We apply dropout to $\mathbf{h}_t^{(0)}$ and stack all positions into $\mathbf{H}^{(0)} = [\mathbf{h}_1^{(0)}, \ldots, \mathbf{h}_T^{(0)}] \in \mathbb{R}^{T\times d}$ as Transformer input.

2) Behavior-Aware Attention Mechanism: Standard self-attention computes weights mainly from item content similarity, implicitly treating interactions as homogeneous. In MBSR, however, different behavior transitions convey different intent strengths (e.g., *Cart* $\to$ *Buy* is usually more informative than *View* $\to$ *Buy*).

To model this heterogeneity explicitly, we introduce a **learnable Behavior Interaction Matrix** $\mathbf{\Phi} \in \mathbb{R}^{|\mathcal{B}|\times|\mathcal{B}|}$. Each entry $\mathbf{\Phi}_{ij}$ is a scalar that measures the influence from behavior type b_i at a query position to behavior type b_j at a key position. For a given sequence, we derive a behavior bias matrix $\mathbf{M}_{\text{bias}} \in \mathbb{R}^{T\times T}$:

$$\mathbf{M}_{\text{bias}}[i,j] = \mathbf{\Phi}(b_i, b_j), \tag{7}$$

where b_i and b_j denote the behavior types at positions i and j.

Given the input $\mathbf{H}^{(0)}$, we compute $\mathbf{Q} = \mathbf{H}\mathbf{W}^Q$, $\mathbf{K} = \mathbf{H}\mathbf{W}^K$, $\mathbf{V} = \mathbf{H}\mathbf{W}^V$, where $\mathbf{W}^Q, \mathbf{W}^K, \mathbf{W}^V \in \mathbb{R}^{d\times d}$ are learnable projection matrices. The **behavior-aware scaled dot-product attention** is defined as

$$\text{Attention}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{softmax}\Big(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d}} + \mathbf{M}_{\text{bias}} + \mathbf{M}_{\text{mask}}\Big)\mathbf{V}, \tag{8}$$

where $\mathbf{M}_{\text{mask}}$ is the causal mask preventing attention to future positions. By adding $\mathbf{M}_{\text{bias}}$, attention scores are adaptively adjusted according to behavior pairs, allowing the model to emphasize informative transitions and suppress weak or noisy ones.

We use standard Transformer blocks with a position-wise FFN, residual connections, and Layer Normalization. After stacking N blocks, we obtain contextualized representations $\mathbf{H}^{(N)} = [\mathbf{h}_1^{(N)}, \ldots, \mathbf{h}_T^{(N)}]$. The last hidden state $\mathbf{h}_T^{(N)}$ is used as the representation of the user's current intent.

C. Multi-Task Prediction with Hard Negative Push

1) Main Task: Next-Item Prediction: The main objective of B-STAR is to predict the next item v_{T+1} under the target behavior b_{tar}. Given $\mathbf{h}_T^{(N)}$, we compute the relevance score between user u and a candidate item v by

$$s(v \mid S_u) = \mathbf{h}_T^{(N)} \cdot \mathbf{e}_v^{\text{global}}. \tag{9}$$

We optimize a sampled binary cross-entropy loss over one positive and multiple negative items. Let v_{pos}^{+} be the ground-truth next item and $\mathcal{N}_{\text{neg}}$ the sampled negatives. The loss is

$$\mathcal{L}_{\text{item}} = -\log \sigma\big(\mathbf{h}_T^{(N)} \cdot \mathbf{e}_{v_{\text{pos}}^{+}}\big) - \textstyle\sum_{v^- \in \mathcal{N}_{\text{neg}}} \log\big(1 - \sigma(\mathbf{h}_T^{(N)} \cdot \mathbf{e}_{v^-})\big), \tag{10}$$

where $\sigma(\cdot)$ is the sigmoid function.

2) Auxiliary Task: Next-Behavior Prediction: To better exploit heterogeneous behaviors, we add an auxiliary task for predicting the *type* of the next behavior. Given $\mathbf{h}_T^{(N)}$, a small MLP head produces

$$\hat{\mathbf{y}}_{\text{beh}} = \text{MLP}_{\text{beh}}(\mathbf{h}_T^{(N)}), \tag{11}$$

and we optimize

$$\mathcal{L}_{\text{beh}} = \text{CrossEntropy}\big(\hat{\mathbf{y}}_{\text{beh}}, b_{T+1}\big). \tag{12}$$

This task regularizes the model to capture not only *which* item will be interacted with, but also *how* the interaction will occur.

3) Hard Negative Push: Explicit negative feedback (e.g., low ratings or dislikes) provides strong signals of user disinterest and differs fundamentally from unobserved items. However, prior methods often ignore such feedback.

In B-STAR, for each user u, we maintain a hard negative pool $\mathcal{N}_u^{\text{hard}}$ containing items that received explicit negative feedback. We then define a **Hard Negative Push** loss:

$$\mathcal{L}_{\text{neg}} = \frac{1}{|\mathcal{N}_u^{\text{hard}}|} \sum_{v_{\text{neg}} \in \mathcal{N}_u^{\text{hard}}} \sigma\big(\mathbf{h}_T^{(N)} \cdot \mathbf{e}_{v_{\text{neg}}}\big), \tag{13}$$

which is computed only for users with at least one hard negative. Minimizing $\mathcal{L}_{\text{neg}}$ pushes the user representation away from explicitly disliked items, thereby sharpening the boundary between preference and disinterest. This module is used when explicit negative signals are available, such as in Yelp.

4) Final Objective: The final training objective of B-STAR is

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{item}} + \lambda_{\text{beh}}\mathcal{L}_{\text{beh}} + \lambda_{\text{neg}}\mathcal{L}_{\text{neg}}, \tag{14}$$

where λ_{beh} and λ_{neg} control the contributions of behavior prediction and hard negative push. In experiments, λ_{neg} is set to a positive value only for datasets with explicit negative feedback, and to 0 otherwise.

TABLE I
DATASET STATISTICS

Dataset	#User	#Item	#Interactions	Behavior types
Yelp	19,800	22,734	1,400,000	{Like, Neutral, Dislike, Tip}
Taobao	147,894	99,037	7,658,926	{View, Favorite, Cart, Buy}
Tianchi	25,000	500,900	4,619,389	{View, Favorite, Cart, Buy}

V. EXPERIMENTS

In this section, we wish to explore the following questions:

- **RQ1:** How does B-STAR perform compared with other baseline models?
- **RQ2:** What is the contribution of each key component of B-STAR?
- **RQ3:** How do different hyperparameters affect the performance?
- **RQ4:** Can B-STAR learn meaningful and interpretable dependencies between heterogeneous behaviors?

A. Experimental Setting

Datasets. Taobao, collected from the Taobao e-commerce platform, includes user behaviors such as Page View, Favorite, Add-to-Cart, and Purchase, where Purchase is treated as the target behavior. **Tianchi**, originating from the Tianchi Big Data Competition, is sourced from Tmall and shares similar behavior types with Taobao. **Yelp** is a business review dataset. Following prior work [11], we construct behavior types based on explicit rating scores: Dislike (rating ≤ 2), Neutral ($2 <$ rating < 4), Like (rating ≥ 4), and Tip, with Like serving as the target behavior. The dataset statistics are summarized in Table I.

Baselines. To comprehensively demonstrate the effectiveness of B-STAR, we compare it with various baselines form different lines of research topic: SASRec [2], BERT4Rec [3], MB-GCN [4], MB-GMN [8], BINN [5], DIPN [6], MGNN-SPred [16], DMT [7], ASLI [17], MB-STR [10], MBHT [9], and PBAT [1], MBSRec [12], MBGen [18], MISSL [11], and GEAR [13].

Implementation Details. We implement B-STAR in PyTorch. For a fair comparison, we follow the experiment setting of GEAR [13]. The batch size is set to 128, and the maximum sequence length is set to 50 for all datasets. The embedding dimension is set to $d = 16$ for Yelp and $d = 32$ for Taobao and Tianchi. We use $N = 2$ Transformer layers for Yelp and $N = 4$ for Taobao and Tianchi. All model parameters are initialized with a Gaussian distribution $\mathcal{N}(0, 0.02)$ and optimized using Adam with a learning rate of 0.001. For the loss weights, we set $\lambda_{\text{beh}} = 0.4$, $\lambda_{\text{neg}} = 0.1$ for Yelp, and $\lambda_{\text{beh}} = 0.8$, $\lambda_{\text{neg}} = 0$ for Taobao and Tianchi. We adopt the leave-one-out evaluation protocol and report Hit Ratio (HR@K) and Normalized Discounted Cumulative Gain (NDCG@K) at $K = 5, 10$.

B. Overall Performance Comparison (RQ1)

The overall performance comparison is reported in Table II. We summarize the key observations as follows: **(1) Superiority of Multi-Behavior Modeling.** B-STAR consistently

TABLE II

EXPERIMENTAL RESULTS ON THREE DATASETS. THE BEST RESULTS ARE BOLDFACED AND THE SECOND-BEST RESULTS ARE UNDERLINED.

Model	Yelp				Taobao				Tianchi			
	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10	HR@5	NDCG@5	HR@10	NDCG@10
SASRec [2]	0.728	0.451	0.793	0.501	0.253	0.182	0.377	0.226	0.568	0.425	0.623	0.470
BERT4Rec [3]	0.745	0.473	0.824	0.528	0.278	0.193	0.396	0.241	0.579	0.439	0.641	0.488
MB-GCN [4]	0.734	0.468	0.807	0.513	0.247	0.172	0.362	0.215	0.565	0.422	0.619	0.468
MB-GMN [8]	0.736	0.525	0.862	0.580	0.394	0.221	0.498	0.304	0.665	0.481	0.712	0.497
BINN [5]	0.375	0.344	0.489	0.413	0.358	0.224	0.463	0.306	0.592	0.445	0.648	0.494
DIPN [6]	0.557	0.452	0.683	0.517	0.256	0.157	0.332	0.184	0.552	0.412	0.608	0.458
MGNN-SPred [16]	0.372	0.335	0.485	0.402	0.353	0.218	0.459	0.304	0.588	0.441	0.644	0.490
DMT [7]	0.549	0.471	0.662	0.525	0.547	0.355	0.654	0.408	0.642	0.468	0.698	0.512
ASLI [17]	0.417	0.392	0.528	0.457	0.382	0.255	0.503	0.325	0.615	0.458	0.672	0.502
MB-STR [10]	0.750	0.574	0.874	0.615	0.691	0.585	0.772	0.611	0.680	0.493	0.727	0.533
MBHT [9]	0.727	0.550	0.861	0.594	0.687	0.590	0.764	0.615	0.665	0.482	0.704	0.529
PBAT [1]	0.741	0.575	0.867	0.619	0.712	0.630	0.794	0.656	0.695	0.528	0.745	0.565
MBSRec [12]	0.764	0.583	0.892	0.616	0.761	0.689	0.835	0.711	0.702	0.548	0.755	0.584
MBGen [18]	0.398	0.376	0.492	0.431	0.486	0.380	0.533	0.395	0.638	0.472	0.692	0.515
MISSL [11]	0.766	0.589	0.884	0.625	0.758	0.673	0.821	0.685	0.869	0.802	0.912	0.810
GEAR [13]	0.769	0.591	0.887	0.628	0.776	0.701	0.834	0.720	0.871	0.793	0.914	0.807
B-STAR	**0.770**	**0.593**	**0.894**	**0.633**	**0.781**	**0.714**	**0.837**	**0.732**	**0.897**	**0.863**	**0.923**	**0.871**
p-val.	1e-2	7e-3	9e-3	5e-3	6e-3	3e-3	2e-3	7e-3	4e-3	2e-3	3e-3	6e-3

TABLE III

ABLATION STUDY ON KEY COMPONENTS OF B-STAR. NOTE THAT THE CUT-OFF OF THE RANKED LIST IS 10.

	Yelp		Taobao		Tianchi	
	HR	NDCG	HR	NDCG	HR	NDCG
w/o GGE	0.885	0.618	0.819	0.712	0.917	0.861
w/o BB	0.887	0.621	0.824	0.716	0.919	0.864
w/o BP	0.868	0.592	0.813	0.704	0.914	0.858
w/o HNP	0.884	0.620	-	-	-	-
B-STAR	**0.894**	**0.633**	**0.837**	**0.732**	**0.923**	**0.871**

outperforms single-behavior baselines across all datasets. This confirms that auxiliary behaviors provide valuable context for inferring user preferences, especially when the target behavior is sparse. **(2) Effectiveness of Adaptive Representations.** B-STAR achieves significant gains over strong Transformer-based MBR baselines. Unlike these methods that often rely on rigid fusion or summation strategies, B-STAR explicitly parameterizes the interaction biases between behavior pairs. This allows the model to adaptively up-weight high-intent transitions while suppressing noise, yielding more expressive user representations. **(3) Advances over State-of-the-Arts.** Even compared with recent advanced models like MISSL and GEAR, B-STAR maintains a clear advantage. The consistent improvements demonstrate the necessity of injecting high-order collaborative signals from the global graph and refining decision boundaries via hard negatives.

C. Ablation Study (RQ2)

To explore the contribution of each module within B-STAR, we conducted an ablation study on the following variables: (i) w/o GGE: removing the Global Graph Encoder to use randomly initialized item embeddings; (ii) w/o BB: removing the behavior bias matrix $\mathbf{M}_{\text{bias}}$ to make the attention mechanism behavior-agnostic; (iii) w/o BP: disabling the auxiliary Behavior Prediction task; and (iv) w/o HNP: omitting the Hard Negative Push loss on the Yelp dataset.

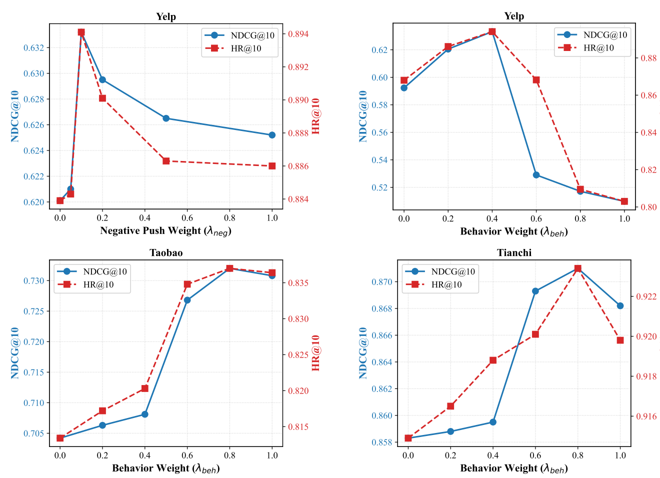

Fig. 3. Performance comparison w.r.t. hyperparameters λ_{neg} and λ_{beh}.

Results in Table III support the following conclusions: (1) *GGE provides consistent benefits.* The performance degradation in w/o GGE highlights the limitations of modeling local interaction sequences in isolation. The global user–item graph injects necessary high-order collaborative signals that are robust against data sparsity. (2) *Behavior-aware attention captures fine-grained intent.* The decline in w/o BB metrics indicates that modeling cross-type behavior affinities is superior to treating all transitions homogeneously. This mechanism effectively encodes diverse interaction semantics. (3) *Auxiliary objectives act as strong regularizers.* The removal of behavior prediction (w/o BP) leads to the sharpest performance drop, suggesting that supervising how a user interacts is critical for shaping robust

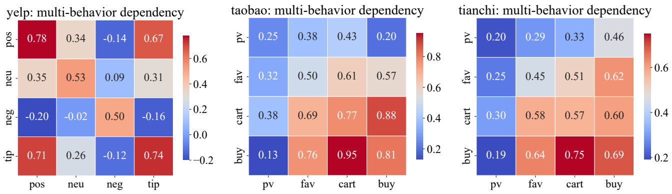

Fig. 4. Visualization of the cross-type behavior dependencies.

sequence representations. On Yelp, the additional drop in w/o HNP confirms that pushing away negative samples further discriminates user preferences.

D. Hyperparameter Sensitivity (RQ3)

We investigate the sensitivity of B-STAR to two pivotal hyperparameters: the explicit negative push weight and the auxiliary behavior prediction weight, as visualized in Fig. 3.

Impact of Negative Push Weight (λ_{neg}). On Yelp, performance exhibits a notable trade-off: introducing a moderate negative push improves accuracy by better separating explicitly disliked items from the preference space, while setting λ_{neg} too large eventually hurts performance.

Impact of Behavior Prediction Weight (λ_{beh}). The optimal strength of the behavior prediction task is dataset-dependent. On Taobao and Tianchi, larger values of λ_{beh} consistently improve performance, indicating that in e-commerce scenarios with clear behavioral funnels (e.g., *View* $\rightarrow$ *Cart* $\rightarrow$ *Buy*), predicting the next behavior strongly complements next-item prediction. On Yelp, a moderate weight works best: too small under-utilizes the auxiliary signal, while too large tends to overfit noisy and irregular behavior patterns in review data.

E. Model Interpretability (RQ4)

To verify whether B-STAR captures meaningful behavioral logic, we visualize the learned Behavior Interaction Matrix Φ in Fig. 4. Based on these results, several key observations can be made: **(1) Sentiment Logic & Negative Suppression.** On Yelp, B-STAR exhibits a sophisticated understanding of sentiment. Positive transitions (e.g., *Tip* $\rightarrow$ *Pos*) receive high weights, while transitions involving explicit dislikes (e.g., *Neg* $\rightarrow$ *Pos*) are assigned negative weights. This semantic awareness capability, which is missing in standard attention mechanisms, drives the superior performance on datasets with explicit feedback. **(2) E-commerce Funnel Pattern.** On Taobao and Tianchi, the learned weights align perfectly with the real-world shopping funnel. High-intent transitions (e.g., *Favorite/Cart* $\rightarrow$ *Buy*) receive significantly higher attention weights compared to casual browsing, and *Buy* $\rightarrow$ *Buy* shows strong self-reinforcement. This confirms that B-STAR autonomously learns to prioritize strong causal signals over weak dependencies.

VI. Conclusion

In this paper, we proposed B-STAR for multi-behavior sequential recommendation, which integrates global collaborative signals and local behavior dependencies. With a global graph encoder, behavior-aware attention, and hard negative push, B-STAR adaptively models interaction strengths and sharpens preference boundaries. For future work, we aim to explore personalized behavior interaction modeling and extend negative feedback mechanisms to implicit scenarios via soft negative mining.

Acknowledgment

This work was supported by the Fundamental Research Funds for the Central Universities under Grant No. N25LPY032.

References

[1] J. Su, C. Chen, Z. Lin, X. Li, W. Liu, and X. Zheng, “Personalized behavior-aware transformer for multi-behavior sequential recommendation,” in *Proceedings of the 31st ACM International Conference on Multimedia (MM ’23)*, 2023, pp. 6321–6331.
[2] W.-C. Kang and J. McAuley, “Self-attentive sequential recommendation,” in *Proc. of ICDM*, 2018, pp. 197–206.
[3] F. Sun, J. Liu, J. Wu, C. Pei, X. Lin, W. Ou, and P. Jiang, “Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer,” in *Proc. of CIKM*, 2019, pp. 1441–1450.
[4] B. Jin, C. Gao, X. He, D. Jin, and Y. Li, “Multi-behavior recommendation with graph convolutional networks,” in *Proc. of SIGIR*, 2020, pp. 659–668.
[5] Z. Li, H. Zhao, Q. Liu, Z. Huang, T. Mei, and E. Chen, “Learning from history and present: Next-item recommendation via discriminatively exploiting user behaviors,” in *Proc. of KDD*, 2018, pp. 1734–1743.
[6] L. Guo, L. Hua, R. Jia, B. Zhao, X. Wang, and B. Cui, “Buying or browsing?: Predicting real-time purchasing intent using attention-based deep network with multiple behavior,” in *Proc. of KDD*, 2019, pp. 1984–1992.
[7] Y. Gu, Z. Ding, S. Wang, and D. Yin, “Deep multifaceted transformers for multi-objective ranking in large-scale e-commerce recommender systems,” in *Proc. of CIKM*, 2020, pp. 2493–2500.
[8] L. Xia, Y. Xu, C. Huang, P. Dai, and L. Bo, “Graph meta network for multi-behavior recommendation,” in *Proc. of SIGIR*, 2021, pp. 757–766.
[9] Y. Yang, C. Huang, L. Xia, and C. Li, “Multi-behavior hypergraph-enhanced transformer for sequential recommendation,” in *Proc. of KDD*, 2022, pp. 2263–2274.
[10] E. Yuan, W. Guo, Z. He, H. Guo, C. Liu, and R. Tang, “Multi-behavior sequential transformer recommender,” in *Proc. of SIGIR*, 2022, pp. 1642–1652.
[11] B. Wu, Y. Cheng, H. Yuan, and Q. Ma, “When multi-behavior meets multi-interest: Multi-behavior sequential recommendation with multi-interest self-supervised learning,” in *Proc. of ICDE*, 2024, pp. 845–858.
[12] S. Elsayed, A. Rashed, and L. Schmidt-Thieme, “Multi-behavioral sequential recommendation,” in *Proceedings of the 18th ACM Conference on Recommender Systems (RecSys ’24)*, 2024, pp. 902–906.
[13] J. Jiang, K. Zhang, J. Kang, Y. Luo, and M. Gao, “Gear: Generalized alternating regressor for multi-behavior sequential recommendation,” in *Proc. of SIGIR*, 2025, pp. 2870–2874.
[14] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, “Lightgcn: Simplifying and powering graph convolution network for recommendation,” in *Proc. of SIGIR*, 2020, pp. 639–648.
[15] C. Xu, P. Zhang, P. Zhao, H. Piao, D. Wei, and Z. Yang, “Gcsan: Graph-contextualized self-attention network for session-based recommendation,” in *Proc. of IJCAI*, 2019, pp. 3940–3946.
[16] W. Wang, W. Zhang, S. Liu, Q. Liu, B. Zhang, L. Lin, and H. Zha, “Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction,” in *Proc. of WWW*, 2020, pp. 3056–3062.
[17] M. M. Tanjim, C. Su, E. Benjamin, D. Hu, L. Hong, and J. McAuley, “Attentive sequential models of latent intent for next item recommendation,” in *Proc. of WWW*, 2020, pp. 2528–2534.
[18] Z. Liu, Y. Hou, and J. McAuley, “Multi-behavior generative recommendation,” in *Proc. of CIKM*, 2024, pp. 1575–1585.