MORO: A Multi-behavior Graph Contrast Network for Recommendation

Weipeng Jiang[1], Lei Duan[1,2](✉), Xuefeng Ding[1], and Xiaocong Chen[1]

[1] School of Computer Science, Sichuan University, Chengdu, China
{jiangweipeng,chenxiaocong}@stu.scu.edu.cn,
{leiduan,dingxf}@scu.edu.cn

[2] Med-X Center for Informatics, Sichuan University, Chengdu, China

Abstract. Multi-behavior recommendation models exploit diverse behaviors of users (*e.g.*, page view, add-to-cart, and purchase) and successfully alleviate the data sparsity and cold-start problems faced by classical recommendation methods. In real-world scenarios, the interactive behaviors between users and items are often complex and highly dependent. Existing multi-behavior recommendation models do not fully utilize multi-behavior information in the following two aspects: (1) The diversity of user behavior resulting from the individualization of users' intents. (2) The loss of user multi-behavior information due to inappropriate information fusion. To fill this gap, we hereby propose a multi-behavior graph contrast network (MORO). Firstly, MORO constructs multiple behavior representations of users from different behavior graphs and aggregate these representations based on behavior intents of each user. Secondly, MORO develops a contrast enhancement module to capture information of high-order heterogeneous paths and reduce information loss. Extensive experiments on three real-world datasets show that MORO outperforms state-of-the-art baselines. Furthermore, the preference analysis implies that MORO can accurately model user multi-behavior preferences.

Keywords: Multi-behavior recommendation · Graph neural network · Multi-task learning · Representation learning

1 Introduction

Recommender systems are widely used in online retail platforms and review sites as techniques to alleviate information overload. How to exploit user behavior data to learn effective user/item representations is the key problem of effective recommendations [5,7,20]. Practically, users will perform different behaviors to items under different intents. As shown in Fig. 1, there are three behaviors (*i.e.*, "page view", "add-to-cart", and "purchase") between users and items, which indicate three kinds of user intents. However, the classical methods [2,5,12,13,20]

This work was supported by the National Natural Science Foundation of China (61972268), the National Key Research and Development Program of China (2018YFB0704301-1), Med-X Center for Informatics Funding Project (YGJC001).

B. Li et al. (Eds.): APWeb-WAIM 2022, LNCS 13423, pp. 117–131, 2023.
https://doi.org/10.1007/978-3-031-25201-3_9

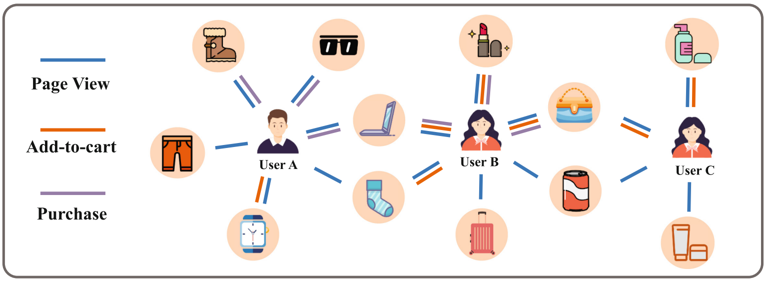


Fig. 1. A toy example of users' multi-behavior data. Best viewed in color. (Color figure online)

consider only single behavior (*e.g.*, purchase) instead of the diversity of multiple behaviors, making it difficult to provide a good list of recommendations to users who have no purchase behavior (*e.g.*, user C in Fig. 1).

To utilize the diversity of user behaviors, several efforts on multi-behavior recommender systems have been made. These works can be roughly classified into two categories. *Deep collaborate filtering-based methods* [3,4,17] use neural network techniques to enhance model representation learning. For instance, MATN [17] employs a transformer module to uniformly capture the dependencies among user behaviors. However, these methods ignore high-order information between users and items [1,16]. Then, *graph neural network-based methods* [1,6,19,21–23] are proposed recently, which model user multi-behavior in two different ways: (1) constructing a unified graph of multi-behavior data and learning user representations on the unified graph [1,6]; (2) constructing subgraph for each user behavior type, learning the representations on different subgraphs, and finally aggregating them [16,19]. However, these methods are still insufficient in their use of multi-behavioral data in two main ways.

- **The lack of modeling diverse user behavior intents.** As shown in Fig. 1, there are various user multi-behavior patterns. Specifically, user A likes to "purchase" items directly after "page view", while user B will "add-to-cart" and then "purchase" items. Uniform modeling the user behaviors as MATN [17] and GNMR [16] will lose this customized feature which is important for modeling user preference. Therefore, modeling users' personalized multi-behavior patterns is one of the goals in this paper.
- **The loss of fusion multi-behavior information.** Due to the influence of other behaviors, modeling on a unified graph cannot fully explore users' preferences of a specific behavior [6]. Moreover, modeling on different behavior subgraphs is difficult to capture the information of high-order heterogeneous paths between users and items (*e.g.*, user→page view→item→purchase →user) [19]. Thus, another goal of this paper is to reduce the information loss in multi-behavior information fusion.

To address these limitations, we propose *multi-behavior graph contrast networks* (MORO) to model complex user behaviors effectively. First, MORO uses graph convolution networks to construct user/item representations under different behavior graphs. Then, we mine the user's personalized multi-behavior patterns with a behavior perceptron module, which is our proposed novel idea enabling MORO to leverage personalized information about user behavior intents. In addition, we propose a contrast enhancement module to reduce the information loss of the multi-behavior information aggregation.

The main contributions of this work are summarized as follows:

- We emphasize the importance of modeling user personalized multi-behavior patterns and the loss of fusion multi-behavior information.
- We propose a novel recommendation model, MORO, which exploits users' personalized behavior patterns and tries to reduce the information loss in the aggregation process of different behavior representations.
- We conduct extensive experiments on three real-world datasets whose results demonstrate that our model outperforms baselines. Further studies on user preferences validate the interpretability of our model.

2 Related Work

In this section, we review works on multi-behavior recommendation. We roughly divide these works into two categories: *deep collaborate filtering-based methods* and *graph neural network-based methods*.

Deep collaborate filtering-based methods leveraged neural networks to learn effective representations of users and items from the interaction data. For example, NCF [5] employed a multilayer perceptron to replace the inner product to calculate the user's acceptance probability of the item. DMF [20] projected representations into the same semantic space, which reduced the noise of inner product computation. Moreover, AutoRec [12] and CDAE [15] introduced autoencoders into recommendation systems by minimizing the representation reconstruction loss. Recently, NMTR [3] proposed a multi-task framework for performing the cascade prediction of different behaviors. DIPN [4] leveraged attention mechanism to model user representations from user multi-behaviors and predict the purchase intent of users. MATN [17] introduced transformer to the multi-behavior recommendation. However, these methods are difficult to capture higher-order neighbor information which makes improvement limited.

Graph neural network-based methods employed graph neural network (GNN) [14,18,21] to aggregate the high-order neighbor information by message propagation mechanism. Wang et al. [13] proposed NGCF to aggregate neighbor information from the user-item bipartite graph. GNMR [16] leveraged GNN to learn representations from multi-behavior interactions between users and items. MBGCN [6] learned user preferences through multi-behavior interaction and learned item representations by item-relevance aware propagation. Moreover, MGNN [23] constructed a multiplex graph and proposed a multiplex graph neural network for multi-behavior recommendation. Recently, GHCF [1] employed efficient multitask learning without sampling in multi-behavior recommendation. MB-GMN [19] combined meta-learning with GNN to learn the meta-knowledge of user

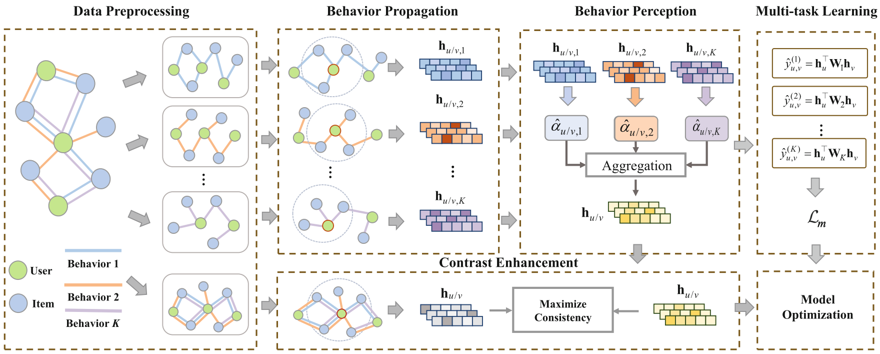


Fig. 2. Illustration of the proposed MORO. Best viewed in color. (Color figure online)

behaviors and improved accuracy of recommendation results. However, due to the differences in building methods, these methods will have some information loss in multi-behavior information fusion.

Different from these methods, we propose MORO to exploit user personalized behavior patterns from multi-behavior data. Moreover, we design a contrastive enhancement module to reduce the information loss on representation aggregation from different behaviors.

3 Preliminary

In multi-behavior recommendation, we need to define a behavior (*e.g.*, purchase) as *target behavior* which we aim to predict. Other relevant behaviors (*e.g.*, page view, add-to-cart, and add-to-favorite) will be defined as *source behaviors*. We begin with introducing key notations and considering a multi-behavior recommendation scenario with users and items:

Definition 1. ***Behavior Graph.*** *A behavior graph* $\mathcal{G}_k = (\mathcal{U}_k, \mathcal{V}_k, \mathcal{E}_k)$ *represents the behavior* k *from* $|\mathcal{U}_k|$ *users over* $|\mathcal{V}_k|$ *items. Specifically, each* $e_{u,v,k} \in \mathcal{E}_k$ *denotes an observed behavior* k *between user* $u \in \mathcal{U}_k$ *and item* $v \in \mathcal{V}_k$*.*

Definition 2. ***Multi-behavior Graph.*** *A multi-behavior graph* $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ *represents all kinds of behaviors* $\mathcal{E} = \bigcup_{k=1}^{K} \mathcal{E}_k$ *from users* $\mathcal{U} = \bigcup_{k=1}^{K} \mathcal{U}_k$ *over items* $\mathcal{V} = \bigcup_{k=1}^{K} \mathcal{V}_k$*. Particularly,* K *denotes the number of behavior categories. In the multi-behavior recommendation scenario,* K *is fixed.*

Task Formulation. Given a multi-behavior graph $\mathcal{G} = (\mathcal{U}, \mathcal{V}, \mathcal{E})$ and a target behavior k, the task of multi-behavior recommendation is to learn a predictive model $\mathcal{F}$ which outputs the probability $\hat{y}_{uv}^{(k)}$ that user $u \in \mathcal{U}$ performs target behavior k to item $v \in \mathcal{V}$.

4 Methodology

We now present our proposed MORO, which exploits multi-behavior data to learn users' preferences. Figure 2 illustrates the framework of MORO, which consists of four key components: (1) behavior propagation module, which models the representations of users and items in different behavior graphs (Sect. 4.1); (2) behavior perception module, which exploits user personalized multi-behavior preferences and aggregates representations from different behavior graphs (Sect. 4.2); (3) contrast enhancement module, which maximizes consistency between multi-behavior aggregated representations and global behavior representations to reduce information loss (Sect. 4.3); (4) multi-task learning module, which makes full use of the information of multi-behavior data (Sect. 4.4).

4.1 Behavior Propagation

In the E-commerce scenario, different behaviors between users and items contain different user intents. For example, "purchase" performs more obvious user preference than "page view". However, the "purchase" behavior is often sparse in practical application scenarios. To take full advantage of multi-behavior, MORO constructs the representations of users and items from different behavior graphs.

As shown in Fig. 2, MORO splits the multi-behavior graph $\mathcal{G}$ into several behavior graphs $\mathcal{G}_k \subset \mathcal{G}$ according to the behavior types. For each behavior graph $\mathcal{G}_k$, MORO performs a behavior propagation to engage the neighborhood information and obtain user/item representations:

$$\mathbf{h}_{u,k}^{l} = \sum_{v \in \mathcal{N}_k(u)} \beta_{u,v}^{(k)} \mathbf{h}_{v,k}^{l-1} \odot \mathbf{h}_k \tag{1}$$

$$\mathbf{h}_{v,k}^{l} = \sum_{u \in \mathcal{N}_k(v)} \beta_{v,u}^{(k)} \mathbf{h}_{u,k}^{l-1} \odot \mathbf{h}_k \tag{2}$$

where $\mathbf{h}_{u,k}^{l} \in \mathbb{R}^d$ and $\mathbf{h}_{v,k}^{l} \in \mathbb{R}^d$ denote the representations of user u and item v collecting the information from l-hops in behavior graph $\mathcal{G}_k$, respectively; $\mathbf{h}_k$ denotes the ID embedding of behavior k; and $\odot$ is the Hadamard product. Particularly, $\mathbf{h}_{u,k}^{0}$ and $\mathbf{h}_{v,k}^{0}$ denote the ID embeddings of user u and item v, respectively.

Inspired by UltraGCN [8], we calculate propagation weight $\beta_{v,u}^{(k)}$ of user u to item v in the behavior graph $\mathcal{G}_k$ in the following way:

$$\beta_{u,v}^{(k)} = \frac{1}{f(u,\mathcal{G}_k)} \sqrt{\frac{f(u,\mathcal{G}_k)+1}{f(v,\mathcal{G}_k)+1}} \tag{3}$$

where $f(u,\mathcal{G}_k)$ denotes the degrees of user u in behavior graph $\mathcal{G}_k$; $f(v,\mathcal{G}_k)$ denotes the degrees of item v in $\mathcal{G}_k$.

After L-layer propagations, we obtain the user/item representations at different layers. The user/item representations about behavior k are obtained:

$$\mathbf{h}_{u,k} = \frac{1}{L+1} \sum_{l=0}^{L} \mathbf{h}_{u,k}^{l}, \quad \mathbf{h}_{v,k} = \frac{1}{L+1} \sum_{l=0}^{L} \mathbf{h}_{v,k}^{l} \tag{4}$$

4.2 Behavior Perception

In the multi-behavior recommendation, user behaviors tend to show personalized. Taking the E-commerce scenario as an example, some users "purchase" the products after "page view", while others will "purchase" the products they prefer after they have "page view" and "add-to-cart". Therefore, it is necessary to model users' multi-behavior preferences individually.

In order to fuse the different behavior information, we define behavior perception weight for a particular behavior k for user u denoted as $\hat{\alpha}_{u,k}$:

$$\alpha_{u,k} = \mathbf{W}_1 \sigma_1(\mathbf{W}_2 \mathbf{h}_{u,k} + \mathbf{b}_2) + b_1 \tag{5}$$

$$\hat{\alpha}_{u,k} = \frac{exp(\alpha_{u,k})}{\sum_{k'=1}^{K} exp(\alpha_{u,k'})} \tag{6}$$

where $\mathbf{W}_1 \in \mathbb{R}^{1 \times d'}$ and $\mathbf{W}_2 \in \mathbb{R}^{d' \times d}$ denote two project matrices; $b_1 \in \mathbb{R}$ and $\mathbf{b_2} \in \mathbb{R}^{d'}$ are bias terms; and the activate function $\sigma_1(\cdot)$ is LeakyReLU. d' and d denote the embedding dimensions ($d' < d$). Symmetrically, the behavior-aware weight of a specific item v can also be calculated by Eqs. (5–6).

MORO aggregates the representations from multiple behavior graphs, and obtains the multi-behavior aggregated representations about user u and item v:

$$\mathbf{h}_u = \sum_{k=1}^{K} \hat{\alpha}_{u,k} \mathbf{h}_{u,k}, \quad \mathbf{h}_v = \sum_{k=1}^{K} \hat{\alpha}_{v,k} \mathbf{h}_{v,k} \tag{7}$$

Therefore, MORO is able to aggregate the information of different behavior graph and generate multi-behavior aggregated representations of users and items. The perception weights also explain the multi-behavior preferences of users.

4.3 Contrast Enhancement

Since the multi-behavior aggregated representations (*e.g.*, $\mathbf{h}_u$ and $\mathbf{h}_v$) are aggregated from different behavior graphs, there will be information loss about high-order heterogeneous path. As shown in Fig. 2, we design a contrast enhancement module to reduce the information loss of high-order heterogeneous path (*e.g.*, user→purchase→item→page view →user). Here, we use R-GCN [11] to encode global behavior representations from multi-behavior graph:

$$\hat{\mathbf{h}}_u^l = \sigma(\sum_{k=1}^{K} \sum_{v \in \mathcal{N}_k(u)} \frac{1}{|\mathcal{N}_k(u)|} \mathbf{W}_k^{l-1} \hat{\mathbf{h}}_v^{l-1} + \mathbf{W}_0^{l-1} \hat{\mathbf{h}}_u^{l-1}) \tag{8}$$

$$\hat{\mathbf{h}}_v^l = \sigma(\sum_{k=1}^{K} \sum_{u \in \mathcal{N}_k(v)} \frac{1}{|\mathcal{N}_k(v)|} \mathbf{W}_k^{l-1} \hat{\mathbf{h}}_u^{l-1} + \mathbf{W}_0^{l-1} \hat{\mathbf{h}}_v^{l-1}) \tag{9}$$

where $\mathbf{W}_k^{l-1} \in \mathbb{R}^{d \times d}$ denotes the transition matrix for $(l-1)$-hop neighbors in the semantic space of behavior k; $\mathbf{W}_0^{l-1} \in \mathbb{R}^{d \times d}$ is the transition matrix for self-loop in all behaviors; and the activate function $\sigma(\cdot)$ is LeakyReLU. Particularly, $\hat{\mathbf{h}}_u^0$ and $\hat{\mathbf{h}}_v^0$ denote the ID embeddings of user u and item v, respectively.

Thus, the multi-behavior aggregated representation $\mathbf{h}_u$ and the global behavior representation $\hat{\mathbf{h}}_u$ to form two representations for user u from different perspectives. With these representations, we can efficiently model the relation between $\mathbf{h}_u$ and $\hat{\mathbf{h}}_u$. Based on InfoNCE [9], we design the graph-contrastive learning objective to maximize the consistency between $\mathbf{h}_u$ and $\hat{\mathbf{h}}_u$ as follows:

$$\mathcal{L}_{user} = \sum_{i \in \mathcal{U}} -log \frac{exp(s(\mathbf{h}_i, \hat{\mathbf{h}}_i)/\tau)}{\sum_{j \in \mathcal{U}} exp(s(\mathbf{h}_i, \hat{\mathbf{h}}_j)/\tau)} \tag{10}$$

where τ is the temperature hyperparameter of softmax; and $s(\cdot)$ denotes the cosine similarity function. Similarly, the graph-contrastive loss of the item side $\mathcal{L}_{item}$ can be obtained. And the complete graph-contrastive loss is the weighted sum of the above two losses:

$$\mathcal{L}_c = \mathcal{L}_{user} + \gamma \mathcal{L}_{item} \tag{11}$$

where γ is a hyperparameter to balance the weight of the two terms in graph-contrastive loss.

4.4 Multi-task Learning

In the multi-behavior recommendation scenario, the target behavior presents more obvious sparsity and cold-start problems than the source behaviors. As shown in Fig. 1, user C only performs "page view" and "add-to-cart" behaviors, and no purchase records. Therefore, using only "purchase" behavior data as target behavior to provide supervision signals is insufficient. We design a multi-task learning module to make full use of multi-behavior data to learn accurate user/item representations. Specifically, the probability of user u and item v having behavior k is calculated as follows:

$$\hat{y}_{u,v}^{(k)} = \mathbf{h}_u^\top \mathbf{W}_k \mathbf{h}_v \tag{12}$$

where $\mathbf{W}_k \in \mathbb{R}^{d \times d}$ denotes the project matrix from global semantic space to the semantic space of target behavior k. We optimize BRP loss [10] for each target behavior and sum them up to obtain the final loss:

$$\mathcal{L}_m = \sum_{u \in \mathcal{U}} \sum_{k=1}^{K} \sum_{(s,s') \in \mathcal{S}_{u,k}} -log(\sigma_2(\hat{y}_{u,s}^{(k)} - \hat{y}_{u,s'}^{(k)})) \tag{13}$$

where $\mathcal{S}_{u,k}$ denotes the set of positive and negative item sample pair of user u under behavior k. For each pair $(s, s') \in \mathcal{S}_{u,k}$, s denotes the positive item sample, s' denotes the negative item sample. The activate function $\sigma_2(\cdot)$ is softmax.

By combining $\mathcal{L}_c$ and $\mathcal{L}_m$, the following objective function for training model can be obtained:

$$\mathcal{L} = \mathcal{L}_m + \lambda_1 \mathcal{L}_c + \lambda_2 ||\boldsymbol{\Theta}||^2 \tag{14}$$

where λ_1 and λ_2 are the hyper-parameters to control the weights of contrastive object and the regularization term, respectively, and $\boldsymbol{\Theta}$ is the model parameters.

Table 1. Statistics of datasets.

Dataset	#User	#Item	#Interaction	Interaction behavior type
Taobao	147,894	99,037	7,658,926	{Page View, Favorite, Cart, Purchase}
Beibei	21,716	7,977	3,338,068	{Page View, Cart, Purchase}
Yelp	19,800	22,734	1,400,036	{Tip, Dislike, Neutral, Like}

5 Experiments

In this section, we evaluate the proposed MORO on three real-world datasets. The experiments are designed to answer the following research questions:

- **RQ1:** How does MORO perform compared with other baselines?
- **RQ2:** What is the impact of module designs on the improvement of MORO?
- **RQ3:** Can MORO successfully capture users' personalized behavior patterns?

5.1 Experimental Setup

We first introduce the datasets, evaluation metrics, baseline methods, and parameter settings involved in the experiments.

Data Description. To evaluate the effectiveness of MORO, we utilize three real-world datasets: Taobao[1], Beibei[2], and Yelp[3], which are publicly accessible and vary in terms of domain, size, and sparsity. The statistical information of them is shown in Table 1.

- *Taobao Dataset.* It is a benchmark dataset for the performance evaluation of multi-behavior recommendations. There are four types of user behaviors contained in this dataset, *i.e.*, page view (pv), add-to-cart (cart), tag-as-favorite (fav), and purchase (buy). We use "purchase (buy)" as the target behavior to evaluate the effectiveness of MORO.
- *Beibei Dataset.* This benchmark dataset is collected from infant product online retailing site Beibei. It involves three types of user behaviors, including page view (pv), add-to-cart (cart), and purchase (buy). We use "purchase (buy)" as the target behavior to evaluate the effectiveness of MORO.
- *Yelp Dataset.* This dataset is collected from the public data repository from Yelp platform. We differentiate user's behaviors over items in terms of the rating scores r, *i.e.*, negative behavior ($r \leqslant 2$), neutral behavior ($2 < r < 4$), and positive behavior ($r \geqslant 4$). Besides the users' rating behaviors, this data also contains the tip behavior if a user gives a tip on his/her visited venues. We use "positive behavior" as the target behavior.

[1] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649.
[2] https://www.beibei.com/.
[3] https://www.yelp.com/dataset/download.

Evaluation Metrics. We adopt two widely-used evaluation metrics: *Hit Ratio* (Hit@N) and *Normalized Discounted Cumulative Gain* (NDCG@N) [16,17]. The higher Hit@N and NDCG@N, the better the model performance. Following the same experimental settings in [16,19], the leave-one-out evaluation is leveraged for training and test set partition. For efficient and fair model evaluation, we pair each positive item instance with 99 randomly sampled no-interactive items for each user, which shares the same settings in [16,19].

Baselines. In order to comprehensively verify the performance of MORO, we consider following baselines:

- **BiasFM** [7] is a classical matrix factorization model that considers the biased information from users and items.
- **DMF** [20] introduces a neural network to matrix factorization and leverages explicit interactions and implicit feedback to refine the representations.
- **NCF** [5] is a collaborative filtering-based method that replaces inner product computation with multilayer perceptrons.
- **AutoRec** [12] stacks multilayer auto-encoder to transfer user-item interaction into a low-dimensional space and fetch user/item representations.
- **NGCF** [13] uses message passing architecture to aggregate information over the user-item interaction and exploits high-order relationships.
- **NMTR** [3] proposes a multi-task framework for performing the cascade prediction of different types of behaviors.
- **DIPN** [4] is a classical multi-behavior recommendation method that leverages an attention mechanism to predict users' purchase intent.
- **MATN** [17] employs a transformer module to capture the relationships among user behaviors and refine the representations learning.
- **GNMR** [16] leverages GNN to learn representations from multi-behavior interactions between users and items.
- **R-GCN** [11] is a graph neural network-based method that leverages relation type information for knowledge graph completion.
- **GHCF** [1] is a stat-of-the-art method that proposes efficient multi-task learning without sampling for parameter optimization.
- **MBGCN** [6] is one of the state-of-the-art methods which uses a graph convolution network to perform behavior-aware embedding propagation.
- **MB-GMN** [19] is another state-of-the-art method that combines GNN with meta-learning to learn the meta-knowledge of user behaviors.

Parameter Settings. We implement the proposed MORO using Pytorch and release our implementation[4] (including the codes, datasets, parameter settings, and training logs) to facilitate reproducibility. MORO is optimized using Adam Optimizer during the training phase. We set the dimension of MORO d as 64 and the number of propagation layer L as 2. The batch size and the learning rate in MORO is set as 256 and 10^{-2}. In addition, we turn the hyper-parameters λ_1 and λ_2 in $[10^{-8}, 10^{-4}]$, τ in $[0.1, 1]$ with grid search.

[4] https://github.com/1310374310/MORO.

Table 2. Overall performance comparison.

Dataset	Taobao		Beibei		Yelp	
Model	Hit@10	NDCG@10	Hit@10	NDCG@10	Hit@10	NDCG@10
BiasMF [7]	0.262	0.153	0.588	0.331	0.775	0.481
DMF [20]	0.305	0.189	0.597	0.336	0.756	0.485
NCF [5]	0.319	0.191	0.595	0.332	0.714	0.429
AutoRec [12]	0.313	0.190	0.607	0.341	0.765	0.472
NGCF [13]	0.302	0.185	0.611	0.375	0.789	0.501
NMTR [3]	0.332	0.179	0.613	0.349	0.790	0.478
DIPN [4]	0.317	0.178	0.631	0.394	0.811	0.540
MATN [17]	0.463	0.271	0.626	0.385	0.822	0.523
GNMR [16]	0.424	0.249	0.604	0.367	0.848	0.559
R-GCN [11]	0.338	0.191	0.605	0.344	0.826	0.520
MBGCN [6]	0.369	0.222	0.642	0.376	0.779	0.465
GHCF [1]	0.377	0.218	0.693	0.411	0.791	0.485
MB-GMN [19]	0.491	0.301	0.691	0.410	0.852	0.567
MORO	**0.619**	**0.403**	**0.754**	**0.455**	**0.877**	**0.583**

5.2 Overall Performance (RQ1)

The performance comparison results are presented in Table 2. From the results, We have the following observations.

Compared to state-of-the-art baseline methods, MORO achieves the best performance on all datasets. Specifically, MORO achieves relatively 26%, 9%, and 3% improvements in terms of Hit@10, and 33%, 10%, and 3% improvements in terms of NDCG@10 on Taobao, Beibei, and Yelp datasets, respectively. It reflects the effectiveness of MORO on multi-behavior recommendation tasks. The improvements can be attributed to three reasons: (1) the advantages of behavior propagation and perception modules which effectively exploit user multi-behavior preferences; (2) the contrastive enhancement, which maximizes the consistencies of representations and reduces the information loss of multi-behavior aggregated representations; (3) multi-task learning, which fully utilizes the signals of multi-behavior data to refine representations.

Further analysis reveals that the methods in which injection multi-behavior information could boost the performance (*e.g.*, MATN, GNMR, GHCF, MBGCN, and MB-GMN). It illustrates the importance of considering multiple behaviors. Moreover, better performance is achieved due to the GNN-based methods to encode the neighbor information in the graph. Furthermore, the GNN-based methods considering the heterogeneity of edges (*e.g.*, MBGCN) perform better than the GNN-based algorithms on homogeneous graphs (*e.g.*, NGCF).

In addition, we also work with different N to validate the effectiveness of top-N recommendations. The experimental results on the Beibei dataset are given in Table 3. We can observe that MORO achieves the best performance under different values of N. It indicates the consistent superiority of MORO as compared to other baselines in assigning higher scores to the user's interested item in the top-N list. We attribute this to the fact that MORO exploits the

Table 3. Comparison results on Beibei dataset with varying N value in terms of Hit@N and NDCG@N.

Model	$N = 1$		$N = 3$		$N = 5$		$N = 7$	
	Hit	NDCG	Hit	NDCG	Hit	NDCG	Hit	NDCG
BiaMF [7]	0.118	0.118	0.310	0.228	0.453	0.287	0.537	0.316
NCF [5]	0.123	0.123	0.317	0.232	0.447	0.283	0.530	0.315
AutoRec [12]	0.128	0.128	0.321	0.236	0.456	0.291	0.540	0.322
MATN [17]	0.184	0.184	0.361	0.286	0.467	0.330	0.543	0.356
GNMR [16]	0.168	0.168	0.336	0.265	0.436	0.307	0.504	0.328
R-GCN [11]	0.134	0.134	0.323	0.242	0.453	0.295	0.535	0.323
MBGCN [6]	0.167	0.167	0.374	0.284	0.498	0.337	0.541	0.322
GHCF [1]	0.179	0.179	0.390	0.300	0.525	0.356	0.611	0.385
MB-GMN [19]	0.183	0.183	0.411	0.306	0.527	0.359	0.608	0.389
MORO	**0.201**	**0.201**	**0.451**	**0.344**	**0.591**	**0.402**	**0.676**	**0.431**

information of multi-behavior and achieves less information loss in representation aggregation by contrast enhancement.

5.3 Study of MORO (RQ2)

Impact of Module. To evaluate the rationality of designed modules in MORO, we consider four model variants as follows:

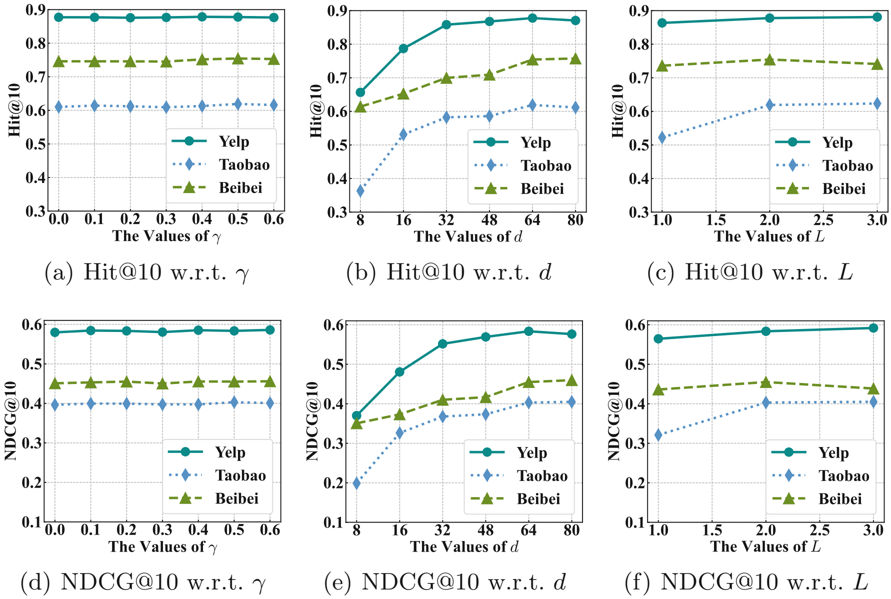
- **MORO-GCN**: To verify the effectiveness of the behavior propagation module, we replace the behavior propagation module with GCN, *i.e.*, the β of Eq. (1) is replaced by $1/\sqrt{|\mathcal{N}_u||\mathcal{N}_v|}$.
- **MORO-Rel**: We replace the behavior perception module with MLP, and aggregate the information from concatenated representations, *i.e.*, $\mathbf{h}_u = MLP(\mathbf{h}_{u,1}||\cdots||\mathbf{h}_{u,K})$, where $||$ denotes the concatenation operation.
- **MORO-Con**: We remove the contrast enhancement loss $\mathcal{L}_c$ from $\mathcal{L}$ to evaluate the effect of the contrast enhancement module.
- **MORO-Task**: We only sample positive and negative examples under the target behavior (*e.g.*, "purchase") and use them to train the model.

The ablation study results are shown in Table 4. From the evaluation results, we have the following observations.

- MORO outperforms all variants on all datasets. It shows the validation of each module in MORO. Specifically, in the e-commerce scenario (*i.e.*, Taobao and Beibei), the improvement of recommendation accuracy by each module is more pronounced. It demonstrates the application value of our proposed model in complex scenarios such as e-commerce.
- The performance gap between MORO and MORO-Rel indicates the advantage of the behavior perception module, which exploits user multi-behavior preferences and aggregates multiple graph representations. It also shows that capturing personalized user multi-behavior preferences is more important than modeling them uniformly.

Table 4. Ablation study on key components of MORO.

Dataset	Taobao		Beibei		Yelp	
Model	Hit@10	NDCG@10	Hit@10	NDCG@10	Hit@10	NDCG@10
MORO-GCN	0.584	0.370	0.694	0.427	0.869	0.580
MORO-Rel	0.577	0.364	0.702	0.418	0.871	0.574
MORO-Con	0.605	0.392	0.746	0.446	0.872	0.579
MORO-Task	0.578	0.368	0.603	0.351	0.843	0.550
MORO	**0.619**	**0.403**	**0.754**	**0.455**	**0.878**	**0.584**



(a) Hit@10 w.r.t. γ (b) Hit@10 w.r.t. d (c) Hit@10 w.r.t. L

(d) NDCG@10 w.r.t. γ (e) NDCG@10 w.r.t. d (f) NDCG@10 w.r.t. L

Fig. 3. Hyperparameter study of MORO

- Moreover, the contrast enhancement module improves the performance of MORO. We attribute this to the ability of the contrast enhancement module to reduce the information loss in multi-behavior aggregated representations, making the representations more effective.
- The evaluation results illustrate the limitations of single-task learning in multi-behavior recommendation tasks (MORO-task). We believe that it is caused by data sparsity for a single target behavior. That multi-task learning can help alleviate this problem.

Hyperparameter Study. To analyze the effect of different parameter settings, we perform experiments to evaluate the performance of MORO with different hyperparameter configurations (*i.e.*, coefficient γ, embedding size d, and number of propagation layers L). The results are shown in Fig. 3.

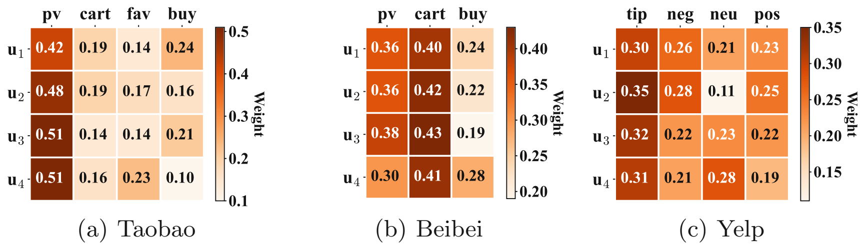


Fig. 4. Weights of user's behaviors

In Eq. (11), the coefficient γ can balance the two terms of $\mathcal{L}_c$ for optimization. To analyze the influence of coefficient γ, we turn γ in $[0, 0.6]$. The model's performance improves slightly as gamma increases, and the model overall achieves the best performance when $\gamma = 0.5$. With the further increase of γ, the performance remains stable, which shows that MORO is robust to γ.

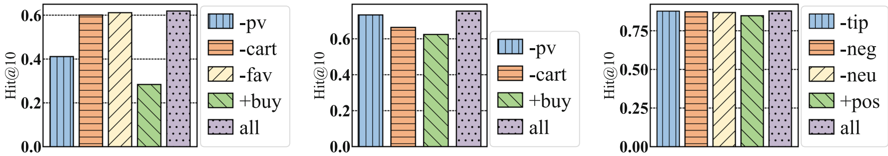
We turn the embedding size d from 8 to 80. With the increasing embedding size d from 8 to 64, the performance improves due to a stronger representation space. After d reaches 64, better performance is not always obtained as d continues to increase. The reason is that a larger representation dimension reduces the ability of MORO to learn representations.

Finally, we analyze the number of propagation layers L. MORO achieves the best performance on all three data sets when stacking two propagation modules (*e.g.*, $L = 2$). Increasing the number of propagation layers L brings the noise to the representations, which affects the performance of MORO (*e.g.*, $L = 3$).

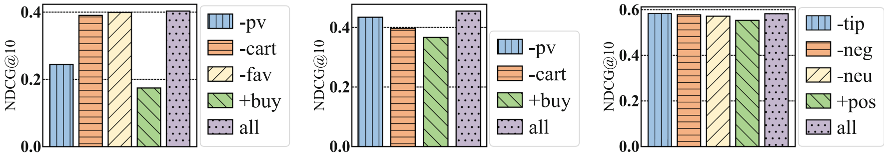
5.4 Study of User Preferences (RQ3)

To analyze user preferences, we visualize user attention weights calculated by Eqs. (5–6). As shown in Fig. 1, we can observe that for each user, MORO successfully captures his/her personalized multi-behavior patterns. The darker the color of the grid, the greater the contribution of that behavior to modeling user preferences. Moreover, for different datasets, the contributions of user behaviors show different distributions. Specifically, the "page view (pv)" behavior of users in the Taobao dataset contributes more to the preference modeling of users. Users' "add-to-cart (cart)" behavior in the Beibei dataset contributes more than other behaviors.

To verify the accuracy of user multi-behavior preferences in Fig. 4, we conduct data ablation experiments, removing specific source behavior data (*e.g.*, -pv) or only using the target behavior data (*e.g.*, +buy) for MORO. Figure 5 shows the results of data ablation experiment. We can find that after removing the "page view" (-pv) behavior data, the model's performance on the Taboao dataset has dropped significantly. For the Beibei dataset, removing the "add-to-cart" (-cart) behavior data impacts the performance of the recommendation. It shows that MORO can accurately model users' multi-behavior preferences and improve



(a) Hit@10 on Taobao (b) Hit@10 on Beibei (c) Hit@10 on Yelp



(d) NDCG@10 on Taobao (e) NDCG@10 on Beibei (f) NDCG@10 on Yelp

Fig. 5. Results of data ablation. Here, "-pv", "-cart", "-fav", "-tip", "-neg", and "-neu" represent MORO without incorporation of "page view", "add-to-cart", "add-to-favorite", "tip","negative", and "neutral" behavior data, respectively. "+buy" and "+pos" denote the variants which only relies on the target behavior data.

model performance with all behaviors. Further analysis shows that the more behavior data eliminated, the worse the performance of MORO. For example, in the Beibei and Taobao datasets, if only "purchase" behavior data is kept (+buy), the model performance will decline because of the sparsity of the "purchase" behavior data. It also indicates that the source behaviors are vital in multi-behavior recommendation tasks.

6 Conclusion and Future Work

In this paper, we propose MORO for multi-behavior recommendation tasks. Considering the different information of user behaviors, we use the behavior propagation module and the behavior perception module to aggregate the representations of different behavior graphs. Then we employ the contrast enhancement module to enhance the multi-behavior aggregated representations. Extensive experiments on three real-world datasets demonstrate the superiority of MORO over other methods. In the future, we plan to capture the dynamic multi-behavior preferences of users from their multi-behavior time series data.

References

1. Chen, C., et al.: Graph heterogeneous multi-relational recommendation. In: AAAI, pp. 3958–3966 (2021)
2. Du, C., Li, C., Zheng, Y., Zhu, J., Zhang, B.: Collaborative filtering with user-item co-autoregressive models. In: AAAI, pp. 2175–2182 (2018)
3. Gao, C., et al.: Neural multi-task recommendation from multi-behavior data. In: ICDE, pp. 1554–1557 (2019)

4. Guo, L., Hua, L., Jia, R., Zhao, B., Wang, X., Cui, B.: Buying or browsing?: predicting real-time purchasing intent using attention-based deep network with multiple behavior. In: KDD, pp. 1984–1992 (2019)
5. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.: Neural collaborative filtering. In: WWW, pp. 173–182 (2017)
6. Jin, B., Gao, C., He, X., Jin, D., Li, Y.: Multi-behavior recommendation with graph convolutional networks. In: SIGIR, pp. 659–668 (2020)
7. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer **42**(8), 30–37 (2009)
8. Mao, K., Zhu, J., Xiao, X., Lu, B., Wang, Z., He, X.: Ultragcn: ultra simplification of graph convolutional networks for recommendation. In: CIKM, pp. 1253–1262 (2021)
9. van den Oord, A., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. CoRR abs/1807.03748 (2018)
10. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: UAI, pp. 452–461 (2009)
11. Schlichtkrull, M.S., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: ESWC, vol. 10843, pp. 593–607 (2018)
12. Sedhain, S., Menon, A.K., Sanner, S., Xie, L.: AutoRec: autoencoders meet collaborative filtering. In: WWW, pp. 111–112 (2015)
13. Wang, X., He, X., Wang, M., Feng, F., Chua, T.: Neural graph collaborative filtering. In: SIGIR, pp. 165–174 (2019)
14. Wu, S., Zhang, Y., Gao, C., Bian, K., Cui, B.: GARG: anonymous recommendation of point-of-interest in mobile networks by graph convolution network. Data Sci. Eng. **5**(4), 433–447 (2020)
15. Wu, Y., DuBois, C., Zheng, A.X., Ester, M.: Collaborative denoising auto-encoders for top-n recommender systems. In: WSDM, pp. 153–162 (2016)
16. Xia, L., Huang, C., Xu, Y., Dai, P.: Multi-behavior enhanced recommendation with cross-interaction collaborative relation modeling. In: ICDE, pp. 659–668 (2021)
17. Xia, L., Huang, C., Xu, Y., Dai, P., Zhang, B., Bo, L.: Multiplex behavioral relation learning for recommendation via memory augmented transformer network. In: SIGIR, pp. 2397–2406 (2020)
18. Xia, L., et al.: Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In: AAAI, pp. 4486–4493 (2021)
19. Xia, L., Xu, Y., Huang, C., Dai, P., Bo, L.: Graph meta network for multi-behavior recommendation. In: SIGIR, pp. 757–766 (2021)
20. Xue, H., Dai, X., Zhang, J., Huang, S., Chen, J.: Deep matrix factorization models for recommender systems. In: IJCAI, pp. 3203–3209 (2017)
21. Yu, S., et al.: Leveraging tripartite interaction information from live stream e-commerce for improving product recommendation. In: KDD, pp. 3886–3894 (2021)
22. Zang, Y., Liu, Y.: GISDCN: a graph-based interpolation sequential recommender with deformable convolutional network. In: DASFAA, vol. 13246, pp. 289–297 (2022)
23. Zhang, W., Mao, J., Cao, Y., Xu, C.: Multiplex graph neural networks for multi-behavior recommendation. In: CIKM, pp. 2313–2316 (2020)