Self-supervised Graph Neural Networks for Multi-behavior Recommendation

Shuyun Gu , **Xiao Wang** , **Chuan Shi*** and **Ding Xiao**
Beijing University of Posts and Telecommunications
gsy793048702@163.com, xiaowang@bupt.edu.cn, shichuan@bupt.edu.cn, dxiao@bupt.edu.cn

Abstract

Traditional recommendation usually focuses on utilizing only one target user behavior (e.g., *purchase*) but ignoring other auxiliary behaviors (e.g., *click*, *add to cart*). Early efforts of multi-behavior recommendation often emphasize the differences between multiple behaviors, i.e., they aim to extract useful information by distinguishing different behaviors. However, the commonality between them, which reflects user's common preference for items associated with different behaviors, is largely ignored. Meanwhile, the multi-behavior recommendation still severely suffers from limited supervision signal issue. In this paper, we propose a novel self-supervised graph collaborative filtering model for multi-behavior recommendation named S-MBRec. Specifically, for each behavior, we execute the GCNs to learn the user and item embeddings. Then we design a supervised task, distinguishing the importance of different behaviors, to capture the differences between embeddings. Meanwhile, we propose a star-style contrastive learning task to capture the embedding commonality between target and auxiliary behaviors, so as to alleviate the sparsity of supervision signal, reduce the redundancy among auxiliary behavior, and extract the most critical information. Finally, we jointly optimize the above two tasks. Extensive experiments, in comparison with state-of-the-arts, well demonstrate the effectiveness of S-MBRec, where the maximum improvement can reach to 20%.

1 Introduction

Personalized recommender system has become a widely deployed technology in today's web platforms and applications to alleviate the issue of information overload nowadays. Most recommendation models are designed based on single behavior (called single-behavior recommendation model), i.e., one type of association between users and items. For example, as shown in Figure 1(a), only *purchase* behavior is used in building recommendation model. However, in the real scene, user behavior is usually more than one type. For example, as shown in Figure 1(b), in addition to *purchase* behavior, *click* and *add to cart* can also reflect users' preferences to a certain extent. We usually consider the *purchase* behavior as target behavior and other types of behaviors as auxiliary behaviors. Recently, more and more works realize that only utilizing purchase behavior is distant from satisfactory, and the auxiliary behavior holds great potential to help predict target behavior [Gao *et al.*, 2019].

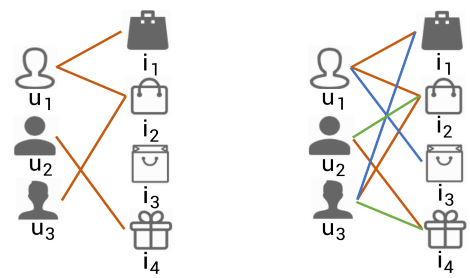


Figure 1: Examples of single-behavior and multi-behavior in e-commerce scene. (a) is *single-behavior* and (b) is *multi-behavior*. The **red** line indicates *purchase* behavior, the **blue** line indicates *click* behavior, and the **green** line indicates *add to cart* behavior.

In order to take full advantage of these various types of behavior, some multi-behavior recommendation models have emerged in recent years [Gao *et al.*, 2019; Jin *et al.*, 2020]. A straightforward approach is to directly model all types of behaviors and apply the single-behavior recommendation model without considering the differences between behaviors [He *et al.*, 2020]. In order to distinguish the semantics of different behaviors, some works assign different learnable weights to different edges to model the importance of the behaviors [Xia *et al.*, 2021a]. In addition, some recent studies provide an embedding representation for each behavior, which can cooperate with the node embedding to participate in graph convolution operation [Chen *et al.*, 2021].

Despite their success, these models still face the following disadvantages. Firstly, they mainly focus on effectively fusing multiple behaviors and capturing the differences of these behaviors. However, they mostly ignore to exploit the com-

*CorrespondingAuthor.

monality of these behaviors, which are also very important for recommendation performances. For example, as shown in Figure 1(b), there must be differences between items associated with user u_3 through target behavior *purchase* and those associated with auxiliary behavior *add to cart*, which leads to user's different behaviors towards them. But at the same time, these items are connected to the user u_3 through any behavior, so they also have certain commonalities (e.g., identical style or price interval) for u_3. These commonalities can reflect the overall preferences of users in different behaviors which often play decisive roles in whether users conduct target behavior (e.g., *purchase*). So we need to mine the commonalities between the target behavior and other behaviors, and integrate the commonalities into target behavior to enhance the quality of embeddings under the target behavior, which can achieve higher-precision recommendation effect. Therefore, how to capture the commonality between target and auxiliary behaviors is an important yet not well explored problem.

Secondly, most of recommendation models are based on a supervised paradigm [Wang *et al.*, 2019], where the observed target behaviors between users and items are usually regarded as supervised signals. However, sparse supervised signals cannot guarantee the quality of graph learning. Even with multiple behaviors, the above problem still exists. There are some efforts in single-behavior recommendation to solve the problem [Wu *et al.*, 2021], which split the single-behavior graph into two views to carry out contrastive learning. However, these methods cannot be directly applied to the multi-behavior recommendation, because they ignore the impact of auxiliary behavior on target behavior and abandon their synergy. Therefore, it is particularly important to develop a new scheme to solve this problem in the field of multi-behavior recommendation.

In this paper, we propose a novel model named S-MBRec, a multi-behavior recommendation model that considers the discrepancies and commonalities of multiple behaviors from the perspective of two types of tasks, and can effectively alleviate the problem of sparse supervised signals as well. Specifically, for each behavior, we execute the GCNs [Kipf and Welling, 2016] to learn the user and item embeddings. In order to distinguish the importance of different behaviors, supervised task is considered and we use automatic learning weights to aggregate the embeddings under multiple behaviors. At the same time, considering the commonalities between multiple behaviors and effectively alleviating the problems of data sparsity, we propose a star-style contrastive learning task, which only performs contrastive learning between the target and each auxiliary behavior. Finally, we jointly optimize these two tasks.

We summarize the contributions of this work as below:

- Based on the multi-behavior recommendation scenario, different from previous works distinguishing the differences of each type of behavior, we make the first effort to study how to retain the commonality of them, and solve the problem of data sparsity simultaneously.
- We propose a novel multi-behavior recommendation model named S-MBRec, consisting of supervised and self-supervised learning tasks. In particular, we design a star-style contrastive learning strategy, which constructs a contrastive view pair for target and each auxiliary behavior subgraph respectively.
- The effectiveness of our S-MBRec model is verified on three real-world datasets, which proves that our model advances the recommendation performance compared with other baselines.

2 Related Work

In recent years, using Graph Neural Networks (GNNs) to solve the recommendation problem has become an extremely important field [Gao *et al.*, 2021]. It is capable of capturing high-order similarity among users and items as well as structural connectivity. In this way, high-quality embeddings for users and items can be obtained, which is critical to the recommendation performance. For example, NGCF [Wang *et al.*, 2019] proposes a spatial GNN in recommendation and obtains superior performance compared with conventional CF methods. LightGCN [He *et al.*, 2020] learns the embeddings of users and items and calculates the weighted sum of the embedding of all layers as the final embedding.

Meanwhile, multi-behavior recommendation is a new branch of recommender system research. Compared with the single-behavior recommendation model, the newly added behaviors in the multi-behavior recommendation model can be regarded as auxiliary behaviors. By adding auxiliary behaviors, the performance of users' accurate recommendation of target behaviors can be improved [Chen *et al.*, 2020a; Wang *et al.*, 2021]. Most of the multi-behavior recommendation methods with GNNs are based on heterogeneous graphs [Jin *et al.*, 2020]. MB-GMN [Xia *et al.*, 2021b] empowers the user-item interaction learning with the capability of uncovering type-dependent behavior representations, which automatically distills the behavior heterogeneity and interaction diversity for recommendations. Summarizing existing multi-behavior recommendation methods, they do not consider capturing the commonality of multiple behaviors, and the problem of data sparsity still exists.

3 Our Proposed Method

3.1 Problem Definition

We define graph $G = (V, E)$, in which nodes V consists of user nodes $u \in U$ and item nodes $i \in I$. The edges in E is user-item interaction edges in G. Assume that there are K ($K \geq 2$) behaviors between users and items, and the edges under the $k^{th}(1 \leq k \leq K)$ behavior is represented as E_k. At the same time, E_k together with all users and items nodes can be extracted to generate a subgraph $G_k = (V, E_k)$, which is formalized as an interaction matrix $\boldsymbol{R_k} \in \boldsymbol{R}^{|\boldsymbol{U}|*|\boldsymbol{I}|}$. We assume that the first behavior is target behavior, and other $K-1$ behaviors are auxiliary behaviors. Our goal is to predict the possibility of interaction between user and item under target behavior with the help of all types of behavior.

3.2 Overall Framework

Figure 2 shows the overall framework of our S-MBRec model. As we can see, we first split the subgraphs of the

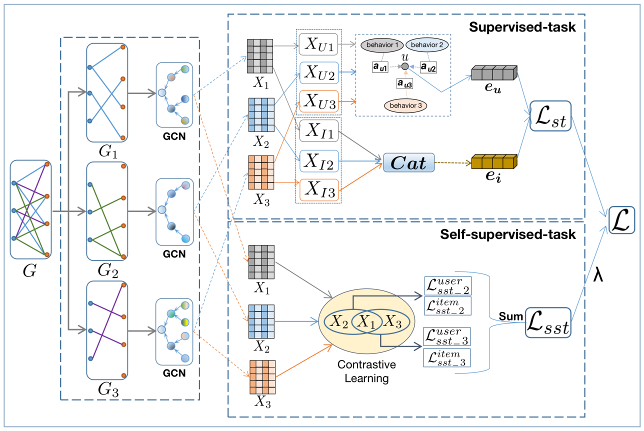


Figure 2: The model architecture of S-MBRec. (We take an example that $K = 3$, i.e., there are three kinds of behavior, in which the first is target behavior and the other two are auxiliary behaviors.)

multi-behavior graph, and perform GCNs operations on each subgraph so as to get the node embedding under each behavior subgraph. Then we set up two tasks: adaptive supervised task and star-style self-supervised task. In supervised task, in order to distinguish the importances of different behaviors and capture the differences between embeddings, we introduce automatic learning weight coefficients to fuse the embeddings under each behavior with a supervised loss function to control this task. In self-supervised task, in order to capture the commonality between target and auxiliary behaviors and alleviate data sparsity, we perform contrastive learning on the target behavior subgraph and each auxiliary behavior subgraph to form a star-style contrastive structure, and then use multiple contrastive learning loss functions to control this task. Finally, we jointly optimize the two tasks.

3.3 Node Representation Learning

Firstly, we need to learn the embedding representation of users and items under each subgraph (behavior). For the k^{th} behavior graph G_k, we can get the adjacency matrix $\boldsymbol{A_k}$ corresponding to the matrix $\boldsymbol{R_k}$ as below,

$$\boldsymbol{A_k} = \begin{pmatrix} 0 & \boldsymbol{R_k} \\ \boldsymbol{R_k^T} & 0 \end{pmatrix}, \tag{1}$$

Then we can obtain the multi-layer message propagation formula of GCNs, as below,

$$\boldsymbol{X_k^{(l+1)}} = \sigma(\widehat{\boldsymbol{A_k}}\boldsymbol{X_k^{(l)}}\boldsymbol{W_k}), \tag{2}$$

where $\widehat{\boldsymbol{A_k}} = \boldsymbol{D_k^{-\frac{1}{2}}}(\boldsymbol{A_k} + \boldsymbol{I_k})\boldsymbol{D_k^{-\frac{1}{2}}}$ is a normalized adjacency matrix with self-connections, in which $\boldsymbol{D_k}$ is a $|V|*|V|$ degree matrix of the k^{th} behavior, and $|V| = |U| + |I|$. $\boldsymbol{I_k}$ is a $|V| * |V|$ identity matrix. $\boldsymbol{X_k^{(l)}} \in \boldsymbol{R}^{|V|*d}$ is the embedding matrix of nodes under the k^{th} behavior in the l^{th} layer of convolution, in which d indicates embedding dimension. $\boldsymbol{W_k}$ and σ are the parameter of model training and a non-linear activation function, respectively. In order to ensure that the short-range neighbor nodes contribute more to the generated embeddings, we use a function f to merge the results of all layers as below,

$$\boldsymbol{X_k} = f(\boldsymbol{X_k^{(l)}}), \tag{3}$$

where $l = [0, ..., L]$. $\boldsymbol{X_k}$ consists of user embedding matrix $\boldsymbol{X_{Uk}} \in \boldsymbol{R}^{|\boldsymbol{U}|*\boldsymbol{d}}$ and item embedding matrix $\boldsymbol{X_{Ik}} \in \boldsymbol{R}^{|\boldsymbol{I}|*\boldsymbol{d}}$. The common designs of f are last layer only [Ying *et al.*, 2018], concatenation [Wang *et al.*, 2019], and weighted sum [He *et al.*, 2020], and we choose the concatenation operation in this paper.

3.4 Adaptive Supervised Task

In this section, we integrate the representation of nodes under different behaviors with an automatic learning weight coefficient which considers the number and impact intensity of different types of behaviors, so as to automaticly distinguish the strength of multiple behaviors. Finally, we use a supervised learning loss function to control the optimization of this module, so as to improve the embedding similarity of associated users and items under the target behavior.

Firstly, we design a coefficient of semantic fusion a_{uk} for the k^{th} behavior of user u, in which not only we need to consider the proportion of the k^{th} behavior of user u in all behaviors, but also the strength of different behaviors (for all users) also needs to be identified, as below,

$$a_{uk} = \frac{exp(w_k * n_{uk})}{\sum_{m=1}^{K} exp(w_m * n_{um})}, \tag{4}$$

where w_k is considered as a strength weight of behavior k which is the same for all users and has the ability of automatic learning in the model. n_{uk} is the number of associations of user u under behavior k.

We have obtained embedding matrix $\boldsymbol{X_{Uk}}$ and $\boldsymbol{X_{Ik}}$ under behavior k, in which the embedding of user u and item i is represented by $\boldsymbol{x_{uk}}$ and $\boldsymbol{x_{ik}}$ respectively. Then we will merge representations under all behaviors. For user u, with the coefficient a_{uk} of k^{th} behavior, we can integrate all behaviors to generate the final representation of user u as below,

$$\boldsymbol{e_u} = \sigma\left\{\boldsymbol{W}(\sum_{m=1}^{K} a_{uk} * \boldsymbol{x_{uk}}) + \boldsymbol{b}\right\}, \tag{5}$$

where $\boldsymbol{W}$ and $\boldsymbol{b}$ are the weight and bias of neural network. However, the fusion between multiple behaviors of item is different from that of user, because the features of items are static. Therefore, we can combine the representations of item i under different behaviors through concatenation operation, as below,

$$\boldsymbol{e_i} = g\left\{Cat(\boldsymbol{x_{ik}})\right\}, \tag{6}$$

where $k = [1, ..., K]$, g is a Multi-Layer Perceptron (MLP), and Cat denotes the concatenation operation between K vectors.

To optimize the current module, we use the pairwise Bayesian Personalized Ranking (BPR) loss [Rendle *et al.*, 2012], which makes the similarity between associated nodes higher than that of non-association nodes. BPR loss function is as below,

$$\mathcal{L}_{st} = \sum_{(u,i,j)\in O} -log\left\{\sigma(\boldsymbol{e_u^T e_i} - \boldsymbol{e_u^T e_j})\right\}, \tag{7}$$

where $O = \{(u,i,j)|(u,i) \in O_+, (u,j) \in O_-\}$ is training data, and O_+ is the observed interactions. $O_- = (U \times V) - O_+$, which represents all unobserved interactions.

3.5 Star-style Self-supervised Task

In this section, we introduce self-supervised learning task. Assume there are K behaviors, we usually need to perform contrastive learning for any two behaviors, i.e., there are $K(K-1)$ contrastive learning pairs and the complexity will be $O(K^2)$. However, considering that the main goal is to capture the relationships between the target and auxiliary behaviors, we propose a star-style contrastive structure, i.e., we only need to perform contrastive learning between the target and each auxiliary behavior subgraph. By this way, we can capture the commonalities of these multiple behaviors and use them to enhance the representing ability of embeddings under target behavior.

So far, we have obtained the embeddings of users and items under each subgraph by GCNs in Eq.(3). A very important step in contrastive learning is to select reasonable positive and negative examples. Most practices are that the positive pairs emphasize the consistency between different views of the same node, while the negative pairs enforce the divergence among different nodes. However, users (or items) with similar associated information under the target behavior should also be regarded as positive example. So we introduce point-wise mutual information (PMI) [Yao *et al.*, 2019] to calculate the similarity between two users (or items) under target behavior. PMI of users is calculated as below

$$PMI(u,u') = log\frac{p(u,u')}{p(u)p(u')}, \tag{8}$$

$$p(u) = \frac{|I(u)|}{|I|}, \tag{9}$$

$$p(u,u') = \frac{|I(u) \cap I(u')|}{|I|}, \tag{10}$$

where $I(u)$ is the item set associated with user u, and $I(u) \cap I(u')$ denotes the item set associated with both users u and u'. In this way, the similarity of any two users under target behavior can be calculated. On the basis of retaining the traditional positive pairs selection scheme, we stipulate that users whose similarity is higher than a threshold t can also be used as a positive pair. The PMI calculation method of the items is consistent with that of the users. For the selection of negative examples, we adopt the strategy of random selection.

After finding the positive and negative examples, we adopt the contrastive loss, InfoNCE [Gutmann and Hyvärinen, 2010], to maximize the agreement of positive pairs and minimize that of negative pairs. When the first target behavior and auxiliary behavior k' ($k' = [2,...,K]$) carry out contrastive learning, the loss function is as below,

Dataset	User	Item	Interaction	Behavior Type
Beibei	21716	7977	3.36×10^6	{View, Cart, Purchase}
Taobao	48749	39493	2.0×10^6	{Click, Cart, Purchase}
Yelp	19800	22734	1.4×10^6	{Tip, Dislike, Neutral, Like}

Table 1: Statistics of experimented datasets

$$\mathcal{L}_{sst\_k'}^{user} = \sum_{u\in U} -log\frac{\sum\limits_{u^+\in U} exp\left\{(\boldsymbol{x_{uK}})^T \boldsymbol{x_{u^+k'}}/\tau\right)\}}{\sum\limits_{u^-\in U} exp\left\{(\boldsymbol{x_{uK}})^T \boldsymbol{x_{u^-k'}}/\tau\right\}}, \tag{11}$$

where $(\boldsymbol{x_{uK}}, \boldsymbol{x_{u^+k'}})$ is the the positive pair and $(\boldsymbol{x_{uK}}, \boldsymbol{x_{u^-k'}})$ is the negative pair. τ is a hyper-parameter, known as the $temperature$ coefficient in softmax. Analogously, we can obtain the contrastive loss $\mathcal{L}_{sst\_k'}^{item}$. By combining all the loss functions under the users and items, the loss function of the self-supervised task can be obtained as below,

$$\mathcal{L}_{sst} = \sum_{k'=2}^{K} (\mathcal{L}_{sst\_k'}^{user} + \mathcal{L}_{sst\_k'}^{item}). \tag{12}$$

3.6 Joint optimization

In order to combine the above two tasks, we jointly optimize the recommendation model, which is as below,

$$\mathcal{L} = \mathcal{L}_{st} + \lambda\mathcal{L}_{sst} + \mu\left\|\Theta\right\|_2^2, \tag{13}$$

where Θ denotes all trainable parameters in two tasks; λ and μ are hyperparameters to control the proportion of self-supervised task and L_2 regularization, respectively.

4 Experimental Results and Analysis

In this section, we will test the effectiveness of our proposed model on real-world datasets and compare it with other existing advanced models. Finally, we conduct a series of parametric and ablation study to analyze of our model.

4.1 Experimental settings

DataSet. In order to evaluate the superior performance of S-MBRec, we verify the effect of our model on three real-world datasets: **Beibei** [1] [Xia *et al.*, 2021b], **Taobao** [2] [Xia *et al.*, 2021b] and **Yelp** [3] [Xia *et al.*, 2021a]. We describe the data details as below,

- **Beibei.** There are three behaviors in this dataset, including *view*, *add to cart* and *purchase*, and *purchase* is the target behavior.
- **Taobao.** There are three behaviors in this dataset, including *click*, *add to cart* and *purchase*, and *purchase* is the target behavior.
- **Yelp.** There are four behaviors in this dataset, including *tip*, *dislike*, *neutral* and *like*, and *like* is the target behavior.

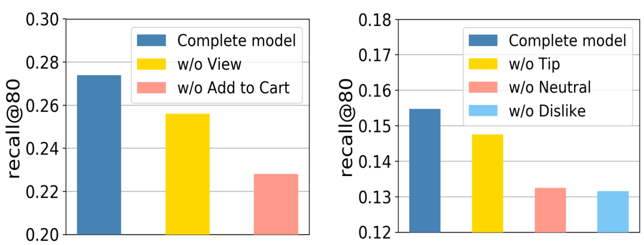
[1] https://www.beibei.com/

[2] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649

[3] https://www.yelp.com/dataset/download

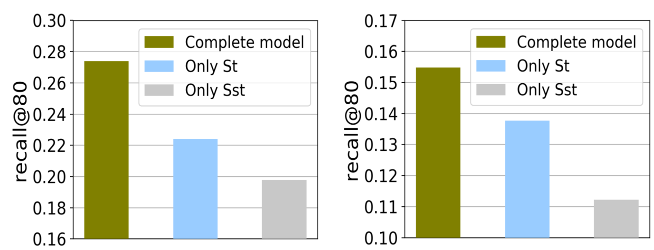
		Single-Behavior Models				Multi-Behavior Models				Our Model
Dataset	**Metric**	**NCF**	**NGCF**	**ENMF**	**LightGCN**	**NMTR**	**EHCF**	**RGCN**	**MB-GMN**	**S-MBRec**
Beibei	**Recall@10**	0.0251	0.0389	0.0377	0.0452	0.0462	0.0459	0.0480	*0.0497*	**0.0529**
	Recall@40	0.0554	0.0754	0.0633	0.1211	0.1366	0.1271	0.1263	*0.1498*	**0.1647**
	Recall@80	0.0641	0.0933	0.0812	0.1939	0.1992	0.1923	0.1912	*0.2017*	**0.2740**
	NDCG@10	0.0117	0.0121	0.0109	0.0127	0.0129	0.0134	0.0123	*0.0139*	**0.0148**
	NDCG@40	0.0164	0.0154	0.0171	0.0187	0.0193	0.0214	0.0226	*0.0397*	**0.0429**
	NDCG@80	0.0228	0.0206	0.0312	0.0334	0.0423	0.0439	0.0443	*0.0465*	**0.0615**
Taobao	**Recall@10**	0.0141	0.0219	0.0198	0.3177	0.0369	0.0295	0.0372	*0.0438*	**0.0608**
	Recall@40	0.0204	0.0297	0.0224	0.0405	0.0487	0.0599	0.0706	*0.0873*	**0.1027**
	Recall@80	0.0311	0.0763	0.0459	0.0795	0.0983	0.1030	0.1527	*0.1559*	**0.1647**
	NDCG@10	0.0094	0.0105	0.0129	0.0216	0.0237	0.0284	0.0214	*0.0326*	**0.0391**
	NDCG@40	0.0141	0.0162	0.0226	0.0287	0.0305	0.0374	0.0304	*0.0398*	**0.0464**
	NDCG@80	0.0196	0.0206	0.0248	0.0265	0.0336	0.0390	0.0448	*0.0476*	**0.0583**
Yelp	**Recall@10**	0.0114	0.0175	0.0163	0.0148	0.0197	0.0186	0.0205	*0.0243*	**0.0259**
	Recall@40	0.0375	0.0398	0.0407	0.0676	0.0724	0.0705	0.0843	*0.0879*	**0.1135**
	Recall@80	0.0498	0.0604	0.0535	0.0823	0.0634	0.0980	0.1090	*0.1398*	**0.1548**
	NDCG@10	0.0044	0.0095	0.0102	0.0178	0.0190	0.0164	0.0214	*0.0273*	**0.0287**
	NDCG@40	0.0141	0.0162	0.0126	0.0187	0.0305	0.0294	0.0204	*0.0248*	**0.0337**
	NDCG@80	0.0164	0.0216	0.0227	0.0235	0.0354	0.0342	0.0398	*0.0416*	**0.0438**

Table 2: Overall model performance on Beibei, Taobao and Yelp datasets, with the metrics of Recall@K and NDCG@K (K=10, 40, 80).



(a) Beibei (b) Yelp

Figure 3: The result comparison of removing different auxiliary behaviors. (Take Beibei and Yelp datasets as examples, and the evaluation index is $Recall@80$.)



(a) Beibei (b) Yelp

Figure 4: The result comparison of removing different tasks. (Take Beibei and Yelp datasets as examples, and the evaluation index is $Recall@80$. St represents supervised task. Sst represents self-supervised task.)

In the above three datasets, there are at least five associations of target behavior, in which we randomly choose two associations, one is test data and the other is verification data. The rest are used for training. The statistics of the three datasets are recorded in Table 1.

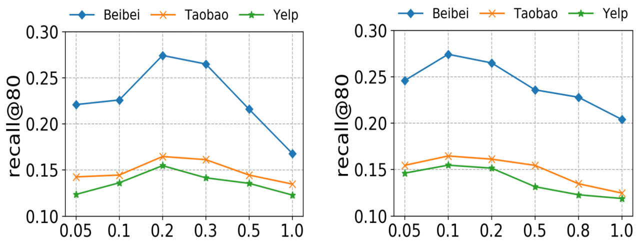
Baseline. We compare S-MBRec with several state-of-the-art methods.The baseline can be divided into two categories: single-behavior models and multi-behavior models. The single-behavior models include: NCF [He *et al.*, 2017], NGCF [Wang *et al.*, 2019], ENMF [Chen *et al.*, 2020a], LightGCN [He *et al.*, 2020]. The multi-behavior models include: NMTR [Gao *et al.*, 2019], EHCF [Chen *et al.*, 2020b], RGCN [Schlichtkrull *et al.*, 2018], MB-GMN [Xia *et al.*, 2021b].

Evaluation Metrics. In order to fully evaluate the effectiveness of our model, we adopt two representative evaluation metrics in the field of recommendation: Recall@K and NDCG@K [Krichene and Rendle, 2020].

Parameters Settings. Our S-MBRec model is implemented in Pytorch. The model is optimized by the Adam optimizer with learning rate of $1e^{-4}$. The training batch-size is selected from $\{1024, 2048, 4096, 6114\}$. The embedding dim is searched from $\{64, 128, 256, 512\}$. The task weight parameter λ is searched from $\{0.05, 0.1, 0.2, 0.5, 1.0\}$, and L_2 regularization coefficient is selected in ranges of $\{0.05, 0.1, 0.2, 0.5, 1.0\}$. The temperature coefficient τ is searched in $\{0.1, 0.2, 0.5, 1.0\}$.

4.2 Overall Performance

We conduct a large number of experiments and recorded the experimental results, which are shown in Table 2. In order



(a) λ (b) τ

Figure 5: Impact of λ and τ to our model under three datasets.

to fully evaluate the results, we take N=10, 40 and 80 respectively in experiment. From Table 2, we summarize the following observations:

Firstly, Table 2 shows that our model S-MBRec consistently outperforms all the baselines. The average improvement of our model to the second best result is approximately 14.1% on Beibei dataset, 20% on Taobao dataset and 15.4% on Yelp dataset. The biggest difference between S-MBRec and baseline model is that we not only need to distinguish the importances of different behaviors, but also improve the similarity of node representation under target and auxiliary behavior subgraphs, So we consider that capturing the commonalities between target and auxiliary behaviors can improve the effect of recommendation. At the same time, all baseline models are trained based on supervised tasks, which are limited by the problem of sparse supervision signals. So it can be concluded that our model can well alleviate this problem with the introduction of star-style self-supervised task among multiple behaviors.

Secondly, muti-behavior models generally outperform single-behavior models, which fully reflects that adding auxiliary behavior enriches semantics and has positive effects on predicting target behavior. Compared to the best single-behavior method (LightGCN), our model has average improvements of 53.9% on Beibei dataset, 102.2% on Taobao dataset and 76.5% on Yelp dataset, which also well verifies the above conclusion.

4.3 Ablation Study

In order to more fully verify the perfection of our model function, we conduct the following ablation experiments.

Firstly, we explore the impact of auxiliary behavior. In order to explore the importance of each auxiliary behavior in our model, we remove each auxiliary behavior and then test the experimental results. As shown in Figure 3, taking Beibei and Yelp datasets as examples, we remove each auxiliary behavior respectively, and the experimental results are significantly lower than our complete model. By analyzing the experimental result, we find that different auxiliary behaviors have different effects on the prediction results of target behavior. For example, in Beibei dataset, the prediction accuracy of removing *add to cart* behavior is much lower than that of removing *view* behavior, which shows that *add to cart* behavior has a greater impact on user's target behavior.

Secondly, we explore the importance of the two tasks in our model. We remove them respectively, and then compare the experimental results. As shown in Figure 4, taking Beibei and Yelp datasets as examples, we find that the experimental results decrease more significantly after removing the supervised task. So we can conclude that two tasks both play important roles in our model, in which the effect of supervised task is more obvious. At the same time, the newly added self-supervised task can also play a key role of assistance in improving the overall effect.

4.4 Hyper-parameter Study

As our model jointly optimizes the supervised and self-supervised tasks with hyperparameter λ in Eq.(13), we first explore the effect of λ on the model performance. Moreover, we analyze the influence of $temperature$ coefficient τ in Eq.(11). The changing trend of the two parameters is shown in Figure 5.

Firstly, we tune λ in $\{0.05, 0.1, 0.2, 0.3, 0.5, 1.0\}$, and then check the corresponding results. As can be seen in Figure 5(a), when $\lambda \leq 0.1$, the experimental results are obviously not satisfactory, which is because the proportion of self-supervised task is relatively small, and its role can be ignored. With λ increasing from 0.1 to 0.2, it can be seen that the experimental results are greatly improved, which shows that the proportion of the two tasks is gradually reasonable. When $\lambda \geq 0.2$, the experimental results show a decreasing trend, which shows that the proportion of self-supervised task is too large, reducing the impact of the traditional supervised task.In conclusion, the two tasks play different roles in our model, in which the proportion of supervised task is larger.

Next, we analyze how the experimental results change with different parameter τ in eq.(11). τ controls the smoothness of embedding similarity in eq.(11). We perform experiments on three datasets and tune it in $\{0.05, 0.1, 0.2, 0.5, 0.8, 1.0\}$. As can be seen in Figure 5(b), As τ gets closer to 0.1, the experimental results gradually improve. When τ is larger than 0.1, the experimental results gradually decrease with the increase of it. Obviously, τ = 0.2 is the best choice for our model. In fact, when the value of τ is small, the similarity will be sharped, while the similarity will be smooth with a large τ. It can be seen that setting τ = 0.2 can ensure a suitable smoothness of similarity.

5 Conclusion

In this work, we propose a novel model named S-MBRec, a multi-behavior recommendation model that considers the discrepancies and commonalities of multiple behaviors from the perspective of two types of tasks, and can cooperate with various types of behavior data to effectively alleviate the problem of sparse supervised signals. We conduct comprehensive experiments, which show that the proposed method improves recommender performance on three datasets.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (No. U20B2045, 62192784, 62172052, 61772082, 62002029, U1936104). It is also supported in part by The Fundamental Research Funds for the Central Universities 2021RC28.

References

[Chen *et al.*, 2020a] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. Efficient neural matrix factorization without sampling for recommendation. *ACM Transactions on Information Systems (TOIS)*, 38(2):1–28, 2020.

[Chen *et al.*, 2020b] Chong Chen, Min Zhang, Yongfeng Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. Efficient heterogeneous collaborative filtering without negative sampling for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 19–26, 2020.

[Chen *et al.*, 2021] Chong Chen, Weizhi Ma, Min Zhang, Zhaowei Wang, Xiuqiang He, Chenyang Wang, Yiqun Liu, and Shaoping Ma. Graph heterogeneous multi-relational recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3958–3966, 2021.

[Gao *et al.*, 2019] Chen Gao, Xiangnan He, Dahua Gan, Xiangning Chen, Fuli Feng, Yong Li, Tat-Seng Chua, and Depeng Jin. Neural multi-task recommendation from multi-behavior data. In *2019 IEEE 35th international conference on data engineering (ICDE)*, pages 1554–1557. IEEE, 2019.

[Gao *et al.*, 2021] Chen Gao, Yu Zheng, Nian Li, Yinfeng Li, Yingrong Qin, Jinghua Piao, Yuhan Quan, Jianxin Chang, Depeng Jin, Xiangnan He, et al. Graph neural networks for recommender systems: Challenges, methods, and directions. *arXiv preprint arXiv:2109.12843*, 2021.

[Gutmann and Hyvärinen, 2010] Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 297–304. JMLR Workshop and Conference Proceedings, 2010.

[He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[He *et al.*, 2020] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

[Jin *et al.*, 2020] Bowen Jin, Chen Gao, Xiangnan He, Depeng Jin, and Yong Li. Multi-behavior recommendation with graph convolutional networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 659–668, 2020.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[Krichene and Rendle, 2020] Walid Krichene and Steffen Rendle. On sampled metrics for item recommendation. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1748–1757, 2020.

[Rendle *et al.*, 2012] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[Schlichtkrull *et al.*, 2018] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer, 2018.

[Wang *et al.*, 2019] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, pages 165–174, 2019.

[Wang *et al.*, 2021] Xiao Wang, Nian Liu, Hui Han, and Chuan Shi. Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1726–1736, 2021.

[Wu *et al.*, 2021] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. Self-supervised graph learning for recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 726–735, 2021.

[Xia *et al.*, 2021a] Lianghao Xia, Chao Huang, Yong Xu, Peng Dai, Xiyue Zhang, Hongsheng Yang, Jian Pei, and Liefeng Bo. Knowledge-enhanced hierarchical graph transformer network for multi-behavior recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4486–4493, 2021.

[Xia *et al.*, 2021b] Lianghao Xia, Yong Xu, Chao Huang, Peng Dai, and Liefeng Bo. Graph meta network for multi-behavior recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 757–766, 2021.

[Yao *et al.*, 2019] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 7370–7377, 2019.

[Ying *et al.*, 2018] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 974–983, 2018.