Bi-directional Contrastive Distillation for Multi-behavior Recommendation

Yabo Chu[1[0000-0002-1694-9179]], Enneng Yang[1[0000-0001-5419-5286]], Qiang Liu[2[0000-0002-9233-3827]], Yuting Liu[1[0000-0002-0774-0007]], Linying Jiang[1[0000-0001-7492-0473]], and Guibing Guo[1(✉)[0000-0002-1709-5056]]

[1] Software College, Northeastern University, Shenyang, Liaoning, People's Republic of China 2071268@stu.neu.edu.cn, ennengyang@stumail.neu.edu.cn, 20185246@stu.neu.edu.cn, {jiangly,guogb}@swc.neu.edu.cn

[2] Center for Research on Intelligent Perception and Computing (CRIPAC), National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing, People's Republic of China qiang.liu@nlpr.ia.ac.cn

Abstract. Multi-behavior recommendation leverages auxiliary behaviors (e.g., view, add-to-cart) to improve the prediction for target behaviors (e.g., buy). Most existing works are built upon the assumption that all the auxiliary behaviors are positively correlated with target behaviors. However, we empirically find that such an assumption may not hold in real-world datasets. In fact, some auxiliary feedback is too noisy to be helpful, and it is necessary to restrict its influence for better performance. To this end, in this paper we propose a **B**i-directional **C**ontrastive **D**istillation (BCD) model for multi-behavior recommendation, aiming to distill valuable knowledge (about user preference) from the interplay of multiple user behaviors. Specifically, we design a forward distillation to distill the knowledge from auxiliary behaviors to help model target behaviors, and then a backward distillation to distill the knowledge from target behaviors to enhance the modelling of auxiliary behaviors. Through this circular learning, we can better extract the common knowledge from multiple user behaviors, where noisy auxiliary behaviors will not be involved. The experimental results on two real-world datasets show that our approach outperforms other counterparts in accuracy.

Keywords: recommender system · contrastive distillation · multi-behavior recommender.

1 Introduction

Modern applications heavily rely on recommender systems as an essential tool to overcome the issue of information overload and improve user experience and satisfaction. Conventional recommenders aim to learn users preference from their target behaviors on items (e.g., 'buy' in e-commerce, 'watch' in movies). Recently, it has become a hot research topic to involve auxiliary behaviors of users (e.g., 'view' and 'add-to-cart' in e-commerce) for performance enhancement. The basic assumption is that auxiliary behaviors are positively correlated with target ones, and can directly reveal user interest to some extent. Hence, most existing research takes into account all the auxiliary behaviors [1, 17, 20], and implicitly works on the same conversion paths among user behaviors [3,8]. Take e-commerce as an example, the general conversion paths are 'view → add-to-cart → buy' and 'view → buy'. That is, users generally browse products on the website or apps, and then add the products of interest into the shopping cart (and then purchase) or directly purchase them without add-to-cart. For simplicity, hereafter we use 'cart' to represent the 'add-to-cart' behavior.

In this paper, we revisit the above assumption on two real-world datasets (**Taobao**[3], **Beibei**[4]), and find that the assumption may not hold in real applications. Specifically, we conduct data analysis by applying a funnel model on the two conversion paths. The results show that conversion paths on Beibei are valid while those on Taobao are less helpful: 'cart' is too noisy to be involved in a conversion path. By removing 'cart' from 'view' data, we obtain a refined conversion path that reaches higher conversion rate than the original one. As a conclusion, not all auxiliary behaviors are positively correlated with target behaviors and some noisy behaviors should be removed from existing conversion paths for better conversion rates.

Therefore, we propose a novel **B**i-directional **C**ontrastive **D**istillation (BCD) model for multi-behavior recommendation, aiming to distill valuable knowledge (about user preference) from the refined conversion paths. Specifically, we design a forward distillation to learn the knowledge from auxiliary feedback to help model target behaviors, and a backward distillation to learn the knowledge from target behaviors to enhance auxiliary ones. In this way, we can highlight the common knowledge (about user preference) from both kinds of user behaviors, and thus improve recommendation performance. To sum up, the main contributions of this paper are summarized as follows:

- We conduct a thorough data analysis on two real datasets and find that the previous assumption may not hold, since some auxiliary behaviors are too noisy to be involved in the conversion paths.
- We propose a novel bi-directional contrastive distillation model to distill and transfer the knowledge from one kind of user behaviors to help model the other kind of user behaviors, whereby better representations of users and items can be learned.
- We conduct extensive experiments on two real-world datasets, and the experimental results demonstrate the effectiveness of our proposed approach in comparison with five competing methods.

2 Data Analysis

In this section, we will revisit the underlying assumption of existing multi-behavior recommenders based on two real-world datasets, that is, all auxiliary

[3] https://github.com/chenchongthu/GHCF
[4] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649

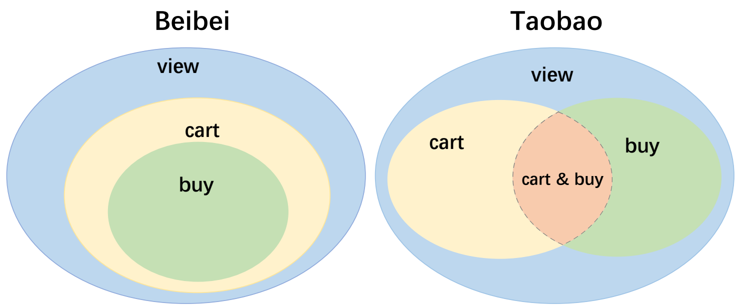

Fig. 1: The distribution of different kinds of user behaviors on the Beibei and Taobao datasets.

user behaviors are positively correlated with target behaviors. The two datasets are Taobao and Beibei. Beibei is a vertical e-commerce platform in China specializing in maternal and infant products, while Taobao is one of China's largest integrated e-commerce platforms. Both datasets consist of three different types of user behaviors: view, cart (i.e., add-to-cart) and buy. The distributions of user behaviors are given in Figure 1.

For the ease of discussion, we first introduce a number of notations to describe the datasets. Let $U = \{u_1, u_2, \ldots, u_m\}$ and $V = \{v_1, v_2, \ldots, v_n\}$ be the set of users and items, where m and n are the number of users and items, respectively. Let $B = \{b_1, b_2, \ldots, b_k\}$ be the set of behavior types, where k is the number of behavior types and b_k is the k-th type of user behaviors. Each user may have multiple interactions with a same item, resulting in multiple types of user behaviors. Let V_{u,b_k} be the set of items that user u has interacted with by behavior type b_k. In our experiments, we have three different sets of items for user u, namely $V_{u,view}, V_{u,cart}, V_{u,buy}$.

From Figure 1, we can observe that the relations of user behaviors follow $V_{u,buy} \subset V_{u,cart} \subset V_{u,view}$ in Beibei, indicating that all the purchased items were added to the shopping cart and browsed by the user in the first place. In this respect, all the auxiliary behaviors are positively correlated with target behaviors. In other words, the assumption in question holds in Beibei. However, this assumption does not hold in Taobao. Specifically, although the area of 'cart' and 'buy' has some overlaps, the ratio is less than 40%. That is, more than 60% Taobao users have only two kinds of behaviors (rather than all of them) on items, either 'view/cart' or 'view/buy'. Users in Taobao have only purchased a small portion of products added to their carts. Hence, we may conclude that 'cart' is likely to be a noisy behavior since 'cart' in most cases does not imply 'buy'. As a general e-commerce platform, Taobao covers such a large variety of product categories that users may not have a strong shopping intent when they go window-shopping online (maybe for exploring interesting products). In the contrast, users likely bring a strong intent to meet their needs when visiting a specialized platform like Beibei.

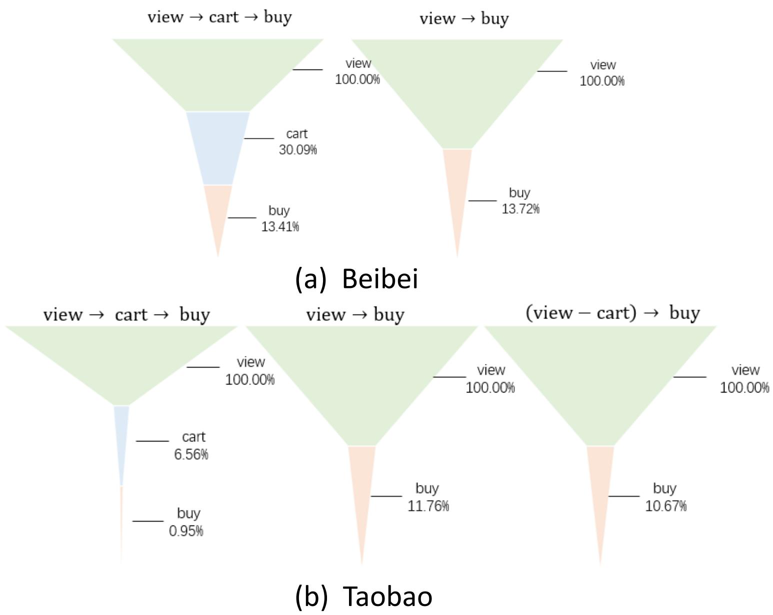

Fig. 2: Funnel model diagram of different conversion paths on the Beibei dataset and Taobao dataset.

To further validate our inference, we apply a funnel model [11] to study the conversion rate of two general conversion paths (i.e., 'view→cart→buy' and 'view→buy'). The purchase funnel is a consumer-focused marketing model that depicts the hypothetical customer journey toward buying a product or service [11]. Since 'cart' is possibly noisy in Taobao, we devise a refined conversion path by considering the products viewed but not added to cart (and finally purchased), denoted by '(view-cart)→buy'. The conversion rate can be calculated as follows:

$$r_{b_1 \to b_2} = \frac{1}{|U|} \sum_{u} \frac{|V_{u,b_1} \cap V_{u,b_2}|}{|V_{u,b_1}|}$$

$$r_{b_1 \to b_2 \to b_3} = r_{b_1 \to b_2} \times r_{b_2 \to b_3}$$

where $r_{b_1 \to b_2}$ denotes the conversion rate from behavior b_1 to b_2, and $r_{b_1 \to b_2 \to b_3}$ represents the conversion rate from behavior b_1 to behavior b_3 through behavior b_2. $|\cdot|$ is the cardinality of a given set. The experimental results on Beibei and Taobao are illustrated in Figure 2.

Specifically, in Figure 2(a) the conversion rates from both paths ('view→cart→buy' and 'view→buy') are very close, indicating that users in Beibei have similar purchase pattern, i.e., either immediately buy products after browsing or firstly add products to cart and then purchase. In fact, there is 40% conversion from 'cart' to 'buy'. In Figure 2(b), the conversion rate from 'view→cart→buy' is extremely small in comparison with other conversion paths. It shows that users in Taobao only purchase a small portion of products added to cart, and the conversion rate from 'cart' to 'buy' is around 14.5%. By removing 'cart' from 'view', as illustrated in Figure 3, the conversion rate of path '(view-cart)→buy' is close to that of path 'view→buy', which is much greater than that of path 'view→cart→buy', implying that 'cart' is quite a noisy behavior. This conclusion will be further validated by our experiments in Section 4.6.

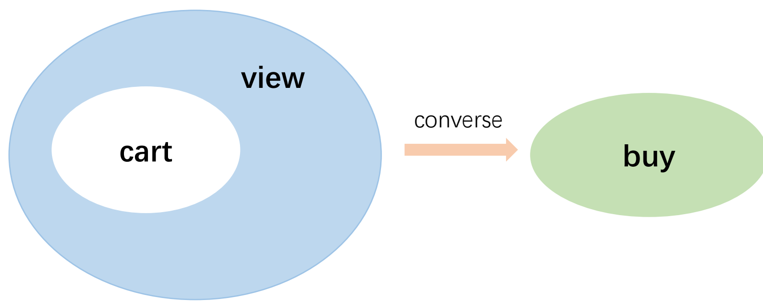

Fig. 3: Illustration of our refined conversion path. The white part of the view indicates that we have removed the items that the user added to the cart from the collection of items she viewed.

To sum up, we find that not all auxiliary behaviors are valuable in providing knowledge about user preference. Their usefulness is domain dependent, that is, a same auxiliary behavior (e.g., 'cart') is helpful in one dataset but maybe noisy in another one. The potential noise involved in auxiliary behaviors motivates us to design a bi-directional contrastive learning for common knowledge distillation from multiple user behaviors.

3 Our Proposed Model

The overall architecture of our proposed BCD model is illustrated in Figure 4. It contains three main components: graph convolutional network (GCN), Bi-directional Contrastive Distillation (BCD) and prediction modules. Specifically, the GCN module is to learn representations of users, items and behaviors from the graph structure of user-item interactions by each kind of behaviors. The BCD module further refines those representations by applying contrastive learning, i.e., to increase the similarity among the same kind of behaviors and highlight the difference among different kinds of behaviors. Then, a forward distillation process is used to distill valuable knowledge (about user preference) from auxiliary to target behaviors. After that, a backward distillation is designed to distill knowledge from target to auxiliary behaviors, to strength the common knowledge among user behaviors and to enhance the task of auxiliary behavior prediction.

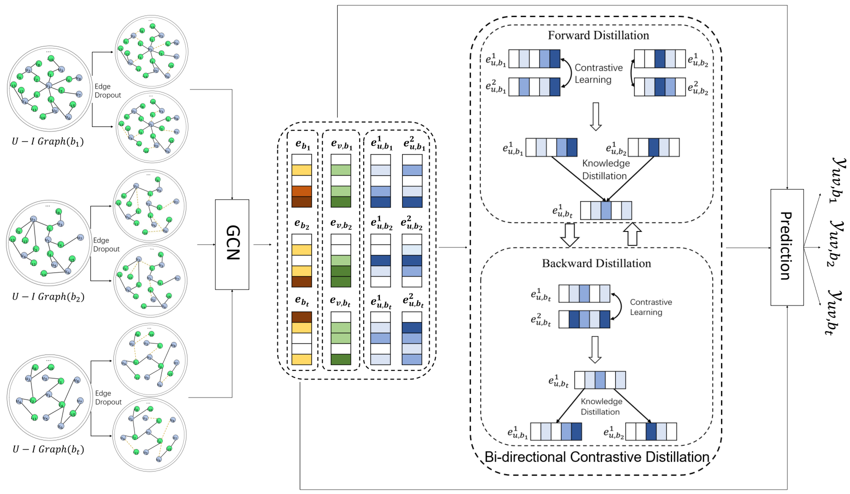

Fig. 4: An overview of our model, which contains three main modules, namely Graph Convolutional Network (GCN), Bi-directional Contrastive Distillation (BCD) and multi-behavior prediction.

Lastly, the prediction module is to predict user's possible behaviors on a given item. We will elaborate each module in next subsections.

3.1 Multi-behavior GCN

Since each user may have multiple behaviors on a same item, we construct an indirected graph to accommodate user-item interactions for each kind of behaviors. The graph nodes are users and items, and the edges are interactions among users and items. We aim to learn user and item representations from the graph structure through graph convolutional networks (GCN) [9, 22], which models nodes propagation based on message-passing architecture. Specifically, user u's embedding $\mathbf{e}_{u,b_k}^{(l)}$ in layer l (under behavior b_k) can be learned by aggregating the representations of his/her neighbors (i.e., interacted items in layer $(l-1)$), which is formally defined as follows:

$$\mathbf{e}_{u,b_k}^{(l)} = \sigma\left(\sum_{v\in\mathcal{N}_u} \frac{1}{\sqrt{|\mathcal{N}_u|\,|\mathcal{N}_v|}}\mathbf{W}^{(l)}\phi\left(\mathbf{e}_v^{(l-1)} \odot \mathbf{e}_{b_k}^{(l-1)}\right)\right)$$

where $\mathcal{N}_u$ and $\mathcal{N}_v$ are the set of immediate neighbors of user u and item v, respectively; $\mathbf{W}^{(l)}$ is a weight matrix in the l-th propagation step, ϕ is a composition operator to incorporate behavior embeddings. $\sigma(\cdot)$ is the LeakyReLU [18] activation function. $\odot$ denotes the element-wise product of two vectors. The normalization term $\frac{1}{\sqrt{|\mathcal{N}_u||\mathcal{N}_v|}}$ is used to avoid the scale of embeddings increasing with graph convolution operations. The representation of item v can be learned in the similar way. Besides, we update behavior embedding $\mathbf{e}_{b_k}^{(l)}$ in layer l by applying a linear transformation to its representation in previous layer, defined by:

$$\mathbf{e}_{b_k}^{(l)} = \mathbf{W}_{b_k}^{(l)} \mathbf{e}_{b_k}^{(l-1)}$$

where $\mathbf{W}_{b_k}^{(l)}$ is a layer-specific weight matrix representing the linear transformation. In this way, behaviors are also transformed into the same embedding space as users and items, whereby mathematical operations can be imposed on them. To start with, we adopt an ID embedding layer to initialize the embeddings of users, items and behaviors in the first hop, denoted as $\mathbf{e}_{u,b_k}^{(0)}, \mathbf{e}_{v,b_k}^{(0)}, \mathbf{e}_{b_k}^{(0)}$.

3.2 Bi-directional Contrastive Distillation

The previous module separately learns multiple representations for each user and item by applying GCN technology on a constructed user-item interaction graph (for each kind of behaviors). We now proceed to refine those representations by taking into account two kinds of relations among user behaviors, namely intra-behavior and inter-behavior relations. Specifically, for the intra-behavior relation, we devise a contrastive learning strategy [21, 30, 32] to strengthen the similarity between two different forms of the same user, and to increase the difference between two different users in the meanwhile. Hence, we design *edge dropout* on the user-item interaction graph by randomly dropping out a certain ratio of edges. By varying different dropout ratios, we obtain two variants of original interaction graph, from which two forms of user presentations (i.e., e_{u,b_k}^1 and e_{u,b_k}^2) can be learnt by the GCN module. For clarity, we use symbols b_a, b_t, b_k to denote auxiliary behavior, target behavior and any kind of user behaviors, respectively. For auxiliary behavior b_a, the objective function of contrastive learning can be formulated as follows.

$$\mathcal{L}_{b_a \to b_a} = -\sum_{u \in U} \mathbb{E}\left[\log \frac{\exp\left(\mathrm{sim}\left(\mathbf{e}_{u,b_a}^1, \mathbf{e}_{u,b_a}^2\right)/\tau\right)}{\sum\limits_{p \in U} \exp\left(\mathrm{sim}\left(\mathbf{e}_{u,b_a}^1, \mathbf{e}_{p,b_a}^1\right)/\tau\right)}\right]$$

where $\mathrm{sim}(\cdot,\cdot)$ is a similarity function to measure the closeness of two input vectors, and inner product is often used for easy computation. τ is a temperature parameter. The numerator term is to maximize the similarity between two embeddings of user u, while the denorminator term is to maximize the difference between user u and any other user p. To avoid mode collapse [14], hereafter we only adopt the first variant of user embeddings during the model learning other than the numerator term.

For the inter-behavior relation, we aim to distill valuable knowledge about user preference from auxiliary behaviors b_a to guide the learning of target behavior b_t. The basic idea is similar with that of intra-behavior relation, i.e., to maximize the similarity between auxiliary and target behaviors of a same user, and meanwhile to maximize the difference of auxiliary and target behaviors between two different users. Formally, the objective loss function of contrastive learning from auxiliary behavior b_a to target behavior b_t can be formulated as follows:

$$\mathcal{L}_{\mathrm{b_a \rightarrow b_t}} = -\sum_{u \in U} \mathbb{E}\left[\log \frac{\exp\left(\operatorname{sim}\left(\mathbf{e}_{u,b_a}^1, \mathbf{e}_{u,b_t}^1\right)/\tau\right)}{\sum\limits_{p \in U} \exp\left(\operatorname{sim}\left(\mathbf{e}_{u,b_a}^1, \mathbf{e}_{p,b_t}^1\right)/\tau\right)}\right]$$

Through the above two-stage operations which we denote as *forward distillation*, the knowledge from auxiliary behaviors can be well learned and transferred to model target behaviors. Inspired by the bi-directional sequence learning in natural language processing [6, 19], we design a *backward distillation* to distill the knowledge from target behavior and help model auxiliary behaviors in return (see Figure 4). Specifically, we apply contrastive learning on target behaviors to refine its representation and then perform knowledge distillation where (modelling of) target behavior is used as a teacher and auxiliary behaviors as a student. This curriculum learning strategy is to better extract the common knowledge between the two kinds of user behaviors and enhance not only the predictive task of target behavior, but also the prediction of auxiliary behaviors.

Hence, the overall objective of our BCD module is to minimize the following loss function:

$$\mathcal{L}_{\text{BCD}} = \underbrace{\sum_{b_a} (\mathcal{L}_{b_a \rightarrow b_a} + \mathcal{L}_{b_a \rightarrow b_t})}_{\text{forward distillation}} + \underbrace{\sum_{b_a} (\mathcal{L}_{b_t \rightarrow b_t} + \mathcal{L}_{b_t \rightarrow b_a})}_{\text{backward distillation}}$$

3.3 Prediction and Learning

We first adopt weighted average across all the behavior-specific embeddings to get the final representations of both users and items, defined by:

$$\mathbf{e}_u = \sum_{b_k} \lambda_{b_k} \mathbf{e}_{u,b_k}, \quad \mathbf{e}_v = \sum_{b_k} \lambda_{b_k} \mathbf{e}_{v,b_k}$$

where λ_{b_k} indicates the importance of behavior b_k relative to target behavior b_t. The setting of $(\lambda_{view}, \lambda_{cart}, \lambda_{buy})$ follows the suggestions given by [1] on two dataset. Then the likelihood that user u will perform the k-th behavior on item v can be estimated by:

$$\hat{y}_{uv,b_k} = \mathbf{e}_u^{\top} \cdot \operatorname{diag}\left(\mathbf{e}_{b_k}\right) \cdot \mathbf{e}_v = \sum_{i}^{d} e_{u,i} e_{b_k,i} e_{v,i}$$

where $\operatorname{diag}(\cdot)$ is a function that converts an input vector into a diagonal matrix, and d is the embedding size.

The main purpose of behavior prediction is to minimize the error of prediction and ground truth, defined by:

$$\mathcal{L}_{b_k} = \sum_{u \in U} \sum_{v \in V} c_{uv,b_k} \left(y_{uv,b_k} - \hat{y}_{uv,b_k}\right)^2$$

where c_{uv,b_k} denotes the weight of entry y_{uv,b_k}. To learn model parameters more effectively and stably, we apply the efficient non-sampling learning technique [1, 7, 16] to optimize our model. Specifically, we simplify c_{uv,b_k} to c_{v,b_k} and reformulate the above loss function as follows:

$$\mathcal{L}_{b_k} = \sum_{u \in U} \sum_{v \in V_{u,b_k}} \left(\left(c^{+}_{v,b_k} - c^{-}_{v,b_k} \right) \hat{y}^2_{uv,b_k} - 2c^{+}_{v,b_k} \hat{y}_{uv,b_k} \right)$$
$$+ \sum_{i=1}^{d} \sum_{j=1}^{d} \left(\left(e_{b_k,i} e_{b_k,j} \right) \left(\sum_{u \in U} e_{u,i} e_{u,j} \right) \left(\sum_{v \in V_u} c^{-}_{v,b_k} e_{v,i} e_{v,j} \right) \right)$$

where V_u represents the items set that user u has interacted with; c^{+}_{v,b_k} and c^{-}_{v,b_k} are the weights if item v has been interacted with behavior b_k and other behaviors, respectively.

The final loss function consists of three components: loss value of behavior predictions, loss value of bi-directional contrastive learning and regularisation terms, which is:

$$\mathcal{L}(\Theta) = \sum_{b_k} \lambda_{b_k} \mathcal{L}_{b_k} + \mathcal{L}_{\text{BCD}} + \mu ||\Theta||^2_2$$

where Θ is the set of model parameters; μ is a regularisation parameter, and $\| \cdot \|_2$ denotes the Frobenius norm.

4 Experiments

4.1 Datasets

As discussed before, Beibei and Taobao are used for our experiments. Their statistics is presented in Table 1. Both datasets are publicly available and the training, validation and test sets are given as well. Specifically, the last purchase records of users are used as test set, the second last records are used as validation set, and the remaining records are used for training. The same two datasets are also used in previous works [1].

Table 1: Statistics of our experimental datasets.

Dataset	#User	#Item	#View	#Add-to-cart	#Purchase
Beibei	21,716	7,977	2,412,586	642,622	304,576
Taobao	48,749	39,493	1,548,126	193,747	259,747

4.2 Comparison Methods

To demonstrate the effectiveness of our BCD model, we compare it with several state-of-the-art methods. The baselines are classified into two categories based on whether they utilize single-behavior or heterogeneous data. The compared single-behavior methods include:

- **ENMF** [2]: This is a state-of-the-art nonsampling recommendation method for Top-N recommendation.
- **LightGCN** [10] : This is a state-of-the-art graph neural network model that simplifies the design of GNNs and thus makes them more suitable for single-behavior recommendation.

The second category that leverages heterogeneous data are as follows:

- **NMTR** [8] : This is a state-of-the-art method which combines the recent advances of NCF modeling and the efficacy of multi-task learning.
- **EHCF** [3]: This is a multi-behavior recommendation algorithm which correlates the prediction of each behavior in a transfer way and adopts non-sampling learning for multi-behavior recommendation.
- **CML** [27]: This is a state-of-the-art multi-behavior recommendation algorithm which proposes a multi-behavior contrastive learning framework to distill transferable knowledge across different types of behaviors via the constructed contrastive loss.
- **GHCF** [1]: This is a state-of-the-art graph neural network model that simulates high-order heterogeneous connectivities beneath each behavior in the user-item integration graph.

4.3 Parameter Settings

We empirically search for optimal parameter settings on the validation set. To speed up searching, model-specific parameters are initialized by the suggested values in the original papers. Other parameters are set as follows. Batch size is 256, the dimension of embeddings is 64, and learning rate is set to 0.0001. For sampling-based methods (LightGCN, NMTR), we set the negative sampling ratio to 1:1. For non-sampling methods (EHCF, GHCF, BCD), we set the negative weight to 0.01 on Beibei and 0.1 on Taobao. The number of graph layers is set to 4 on Beibei and 2 on Taobao to avoid over-fitting, the dropout ratio is set to 0.8 on both datasets. For Beibei, the temperature parameters of forward distillation are set to (2, 1.5) and those for backward distillation is set to (0.5, 3). For Taobao, we set the temperature parameters of forward distillation to (1.5, 4) and those for backward distillation to (1, 1.5).

4.4 Evaluation Metrics

We adopt three popular evaluation metrics, namely HR (Hit Ratio) [12], NDCG (Normalized Discounted Cumulative Gain) [26], and MRR (Mean Reciprocal

Table 2: Performance of all the comparison methods on the Taobao and Beibei datasets. Row 'p-value' indicates the significance score of BCD relative to the second best approach (i.e., GHCF) on each evaluation metric.

Taobao	HR@K				NDCG@K				MRR@K			
	K=3	K=5	K=10	K=15	K=3	K=5	K=10	K=15	K=3	K=5	K=10	K=15
ENMF	.0116	.0152	.0227	.0283	.0091	.0106	.0129	.0144	.0082	.0090	.0100	.0104
LightGCN	.0195	.0258	.0371	.0443	.0155	.0181	.0217	.0236	.0141	.0155	.0170	.0176
NMTR	.0263	.0317	.0417	.0549	.0189	.0232	.0268	.0284	.0184	.0197	.0219	.0231
EHCF	.0286	.0358	.0482	.0572	.0234	.0264	.0304	.0327	.0216	.0233	.0249	.0256
CML	.0346	.0469	.0693	.0885	.0284	.0294	.0413	.0432	.0236	.0268	.0299	.0328
GHCF	.0380	.0521	.0777	.0977	.0291	.0349	.0432	.0485	.0261	.0293	.0327	.0343
BCD	**.0385**	**.0533**	**.0799**	**.0983**	**.0297**	**.0360**	**.0442**	**.0492**	**.0266**	**.0299**	**.0333**	**.0348**
p-value	0.125	2.4e-4	7.6e-5	0.006	0.018	7.0e-4	2.5e-4	0.002	0.078	0.002	6.8e-4	0.002
Beibei	HR@K				NDCG@K				MRR@K			
	K=3	K=5	K=10	K=15	K=3	K=5	K=10	K=15	K=3	K=5	K=10	K=15
ENMF	.0129	.0196	.0356	.0493	.0099	.0126	.0177	.0213	.0089	.0104	.0124	.0135
LightGCN	.0227	.0295	.0325	.0390	.0177	.0206	.0216	.0232	.0159	.0176	.0180	.0185
NMTR	.0221	.0356	.0679	.1005	.0157	.0212	.0316	.0401	.0135	.0166	.0207	.0233
EHCF	.0688	.0966	.1483	.1862	.0531	.0645	.0811	.0911	.0477	.0540	.0608	.0638
CML	.0783	.1069	.1785	.2163	.0574	.0706	.0911	.1018	.0496	.0595	.0673	.0704
GHCF	.0814	.1214	.1900	.2343	.0600	.0765	.0983	.1103	.0527	.0618	.0708	.0744
BCD	**.0836**	**.1233**	**.1906**	**.2380**	**.0614**	**.0777**	**.0996**	**.1120**	**.0538**	**.0628**	**.0718**	**.0755**
p-value	0.033	0.007	2.1e-5	0.001	3e-4	0.015	0.012	0.013	0.078	0.040	0.048	0.043

Rank) [23]. Specifically, HR measures to what extent a recommendation list contains items that users actually like. NDCG gives more weights to the relevant items if being ranked top in the recommendation list. MRR scores high if the first relevant item appears early in the recommendation list.

4.5 Performance Comparison

Table 1 summarizes the performance of all comparison methods on the two datasets. The results show that multi-behavior recommendation methods are superior to single-behavior ones in terms of all evaluation metrics, implying the usefulness of multiple auxiliary behaviors. Among multi-behavior recommenders, our approach BCD consistently outperforms the other methods. Since other methods may also adopt the non-sampling technique for model learning other than multiple user behaviors, we believe it is our bi-directional contrastive distillation module that leverages both intra- and inter-behaviors relations of user behaviors and thus improves the recommendation performance. Specifically, the average improvements relative to the second best approach (i.e., GHCF) are around 1.61% and 1.89% on Beibei and Taobao, respectively. The larger improvements on Taobao can be explained by the fact that we refine the original conversion path by removing the noisy auxiliary behavior 'cart'.

Table 3: Effects of two important components, where 'BCD-ib' and 'BCD-bd' indicate the variants of BCD without the consideration of intra-behavior relations and backward distillation, respectively.

Methods	Beibei				Taobao			
	HR@5	@10	NDCG@5	@10	HR@5	@10	NDCG@5	@10
BCD-ib	0.1196	0.1880	0.0755	0.0976	0.0530	0.0789	0.0356	0.0439
BCD-bd	0.1191	0.1882	0.0753	0.0976	0.0531	0.0783	0.0356	0.0438
BCD	**0.1233**	**0.1906**	**0.0777**	**0.0996**	**0.0533**	**0.0799**	**0.0360**	**0.0442**

We conduct statistical significance test (paired t-tests, confidence 0.95) between our approach and GHCF on both datasets, and the results are presented in the last row of Table 2. The results (all p-values much smaller than 0.05) demonstrate that our approach is statistically significant in comparison with the second best comparison method.

4.6 Ablation Study

Intra-behavior Learning & Backward Distillation In this section, we will study two important components of our approach, namely intra-behavior contrastive learning and backward distillation. We denote 'BCD-ib' and 'BCD-bd' as the variants of our BCD model without the component of intra-behavior contrastive learning and that of backwared distillation, respectively. The results on two datasets are presented in Table 3. It can be observed that BCD outperforms both BCD-ib and BCD-bd variants. We may conclude that (1) it is beneficial to take into account intra-behavior relations for recommendation performance. We can obtain better representations of users/items by applying contrastive learning on the same behaviors. (2) backward distillation is indeed useful to distill the knowledge from target behaviors to enhance the modelling of auxiliary behaviors.

Auxiliary Behaviors for Inter-behavior Learning The inter-behavior distillation will learn better user preference from the knowledge distillation of auxiliary behaviors on target behaviors. As discussed in Section 2, not all auxiliary behaviors are helpful. We further validate this finding by conducting a series of experiments based on different auxiliary behaviors. Specifically, we select different behaviors for the inter-behavior distillation to investigate their usefulness. The results are given in Table 4. On Beibei, we can find that both inter-behaviors ('view→buy' and 'cart→buy') have similar results and are slightly smaller than our BCD method, which considers both kinds of inter-behavior relations. On Taobao, the effect of inter-behavior relation ('cart→buy') is worse than the other ones, indicating the noisy information brought by 'cart' behaviors. By removing 'cart' information from 'view' (i.e., the relation 'view-cart→buy'), we can improve the recommendation performance.

Table 4: Effect of Inter-behavior Learning, where 'v-c→buy' is short for 'view-cart→buy'.

Inter-behavior	Beibei				Taobao			
	HR@5	@10	NDCG@5	@10	HR@5	@10	NDCG@5	@10
view→buy	0.1206	0.1900	0.0758	0.0982	0.0528	0.0792	0.0358	0.0441
cart→buy	0.1206	0.1897	0.0759	0.0985	0.0520	0.0784	0.0354	0.0436
v-c→buy	-	-	-	-	0.0525	0.0791	0.0356	0.0439
BCD	**0.1233**	**0.1906**	**0.0777**	**0.0996**	**0.0533**	**0.0799**	**0.0360**	**0.0442**

4.7 Parameter Analysis

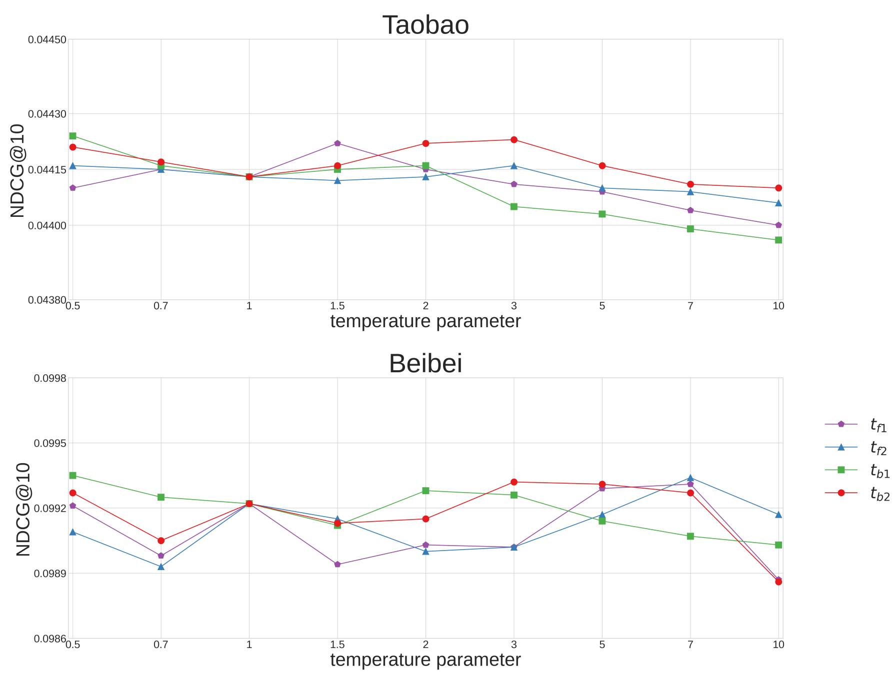

Fig. 5: Effect of temperature parameters on two datasets. (t_{f1}, t_{f2}) are the temperature parameters of forward distillation and (t_{b1}, t_{b2}) are for backward distillation.

Analyses on Temperature Parameter The temperature parameter has been a key parameter for most existing knowledge distillation-based recommendation algorithms. In this section, we intend to analyze the effect of temperature parameters on our recommendation performance. Specifically, we denote (t_{f1}, t_{f2}) as the temperature parameters of forward distillation and (t_{b1}, t_{b2}) of backward

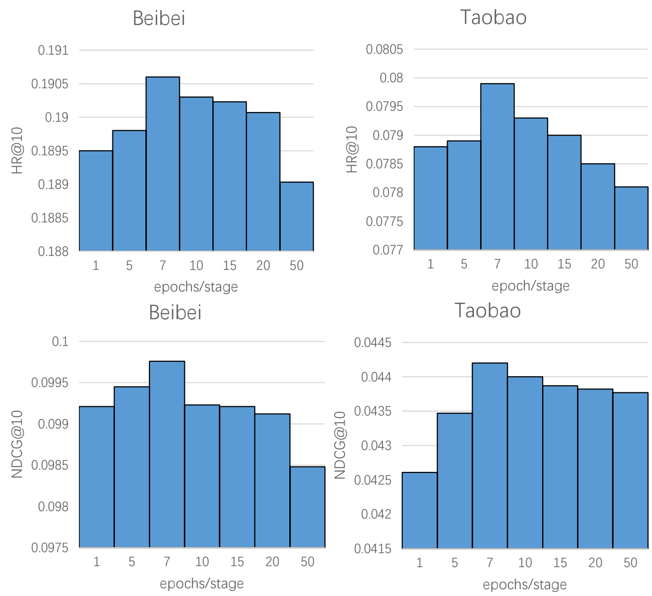

Fig. 6: Effect of different number of training epochs for either forward or backward distillation before moving to the next step. Other metrics follow similar trends and are omitted for space saving.

distillation. We adjust the value of a single temperature parameter in the range $\{0.5, 0.7, 1, 1.5, 2, 3, 5, 7, 10\}$, and when we adjust this temperature parameter, we will fix all other temperatures to 1. The experimental results on two datasets as given in Figure 5. It is observed that the metric values for any temperature parameter fluctuate within a small range, implying that our model is insensitive to temperature parameters. We attribute it to the fact that our model is able to distinguish the preference representations of two different users during training for smaller temperature parameters and thus learn more personalized knowledge. Meanwhile, for a larger temperature, our model can learn some common knowledge from other users and behaviors to assist with recommendations. As a result, our approach is not limited to a strict setting of temperature parameters.

Parameter Analysis of Stage Epochs An important parameter for our model training is the number of training epochs for either forward or backward distillation. The results are illustrated in Figure 6. We can find that the best performance can be reached when each stage of distillation is trained 7 epochs before moving to the next step. Other parameter settings will decrease the performance to a certain extent.

5 Related Work

5.1 Multi-behavior Recommendation

We will briefly review a number of representative multi-behavior recommenders that are most relevant to our work. These works are mainly built upon conversion paths among user behaviors. For example, ChainRec [24] explores the monotonic behavior chains based on conversion paths to model the effect of auxiliary behaviors on target behaviors. NMTR [8] construct a cascade prediction structure from conversion paths, which is then used to predict subsequent behaviors according to the prediction of previous behaviors. EHCF [3] propose to share parameters based on cascading relationships of conversion paths for evolutionary knowledge learning. MRIG [25] opt to construct a graph structure (rather than a cascading chain) from auxiliary behaviors to enhance the prediction of target behaviors. MBGCN [13] learns discriminative behavior representations using graph convolutional network. MB-GMN [29] uses graph meta network for learning the heterogeneity and diversity among different behaviors. GHCF [1] further adopt the operation of graph convolution to capture higher-order representations of users under each behavior. CML [28] proposes a multi-behavior contrastive learning framework to distill transferable knowledge across different types of behaviors via the constructed contrastive loss.

However, all the above works implicitly assume that all the auxiliary behaviors are positively correlated with target ones, which may not hold in real datasets as discussed in Section 2. Our work argues that it is better not to involve noisy auxiliary behaviors for performance improvement.

5.2 Contrastive Distillation in Recommendations

Contrastive learning, which aims to learn high-quality representations via self-supervised pretext tasks, recently achieves remarkable success in the field of computer vision [4, 5]. Till now, only few works have been proposed to leverage contrastive distillation for recommendation. DE-RDD [15] designs a contrastive distillation loss function to make better use of the knowledge from a teacher model to guide the learning of a student recommendation model. MICRO [31] adopts contrastive distillation to maximize the agreement between item representations under each modality and the fused multi-modal representation, whereby a more precision item representation can be obtained. Hence, our work focuses on different problem settings (i.e., multi-behavior recommendation) from the above two relevant works (one for model compression and the other for multi-modal recommendation).

6 Conclusions and Future work

In this paper, we conducted thorough data analysis on two real datasets, and found that not all auxiliary behaviors were positively correlated with target behaviors. We proposed a Bi-directional Contrastive Distillation (BCD) model to distill the common knowledge from multiple user behaviors via forward and backward distillation. We conducted a series of experiments and verified that our approach beat other methods in terms of ranking accuracy. For future work, we plan to explore the manners of data augmentation to enhance recommendation performance under each auxiliary behavior. Besides, we will try to reduce the computational cost due to the additional graph convolution operations on users' interaction graph and further speed up our algorithm.

Acknowledgement

This work is partially supported by the National Natural Science Foundation of China under Grant (No. 62032013, 61972078), and Hebei Natural Science Foundation No. G2021203010.

References

1. Chen, C., Ma, W., Zhang, M., Wang, Z., He, X., Wang, C., Liu, Y., Ma, S.: Graph heterogeneous multi-relational recommendation. In: Proc. of AAAI (2021)
2. Chen, C., Zhang, M., Wang, C., Ma, W., Li, M., Liu, Y., Ma, S.: An efficient adaptive transfer neural network for social-aware recommendation. In: Proc. of SIGIR (2019)
3. Chen, C., Zhang, M., Zhang, Y., Ma, W., Liu, Y., Ma, S.: Efficient heterogeneous collaborative filtering without negative sampling for recommendation. In: Proc. of AAAI (2020)
4. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Proc. of ICML (2020)
5. Chen, X., Fan, H., Girshick, R., He, K.: Improved baselines with momentum contrastive learning. arXiv preprint arXiv:2003.04297 (2020)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Devooght, R., Kourtellis, N., Mantrach, A.: Dynamic matrix factorization with priors on unknown values. In: Proc. of KDD (2015)
8. Gao, C., He, X., Gan, D., Chen, X., Feng, F., Li, Y., Chua, T.S., Jin, D.: Neural multi-task recommendation from multi-behavior data. In: 2019 IEEE 35th international conference on data engineering (ICDE) (2019)
9. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. Proc. of NeurIPS (2017)
10. He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M.: Lightgcn: Simplifying and powering graph convolution network for recommendation. In: Proc. of SIGIR (2020)
11. Jansen, B.J., Schuster, S.: Bidding on the buying funnel for sponsored search and keyword advertising. Journal of Electronic Commerce Research (2011)
12. Järvelin, K., Kekäläinen, J.: Cumulated gain-based evaluation of ir techniques. ACM Transactions on Information Systems (TOIS) (2002)
13. Jin, B., Gao, C., He, X., Jin, D., Li, Y.: Multi-behavior recommendation with graph convolutional networks. In: Proc. of SIGIR (2020)

14. Jing, L., Vincent, P., LeCun, Y., Tian, Y.: Understanding dimensional collapse in contrastive self-supervised learning. arXiv preprint arXiv:2110.09348 (2021)
15. Kang, S., Hwang, J., Kweon, W., Yu, H.: De-rrd: A knowledge distillation framework for recommender system. In: Proc. of CIKM (2020)
16. Liang, D., Charlin, L., McInerney, J., Blei, D.M.: Modeling user exposure in recommendation. In: Proc. of WWW (2016)
17. Liu, Q., Wu, S., Wang, L.: Multi-behavioral sequential prediction with recurrent log-bilinear model. IEEE Trans. Knowl. Data Eng. (2017)
18. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. of ICML (2013)
19. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE transactions on Signal Processing (1997)
20. Tang, L., Long, B., Chen, B.C., Agarwal, D.: An empirical study on recommendation with multiple types of feedback. In: Proc. of KDD (2016)
21. Tian, Y., Krishnan, D., Isola, P.: Contrastive multiview coding. In: Proc. of ECCV (2020)
22. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. In: Proc. of ICLR (2018)
23. Voorhees, E.M., et al.: The trec-8 question answering track report. In: Trec (1999)
24. Wan, M., McAuley, J.: Item recommendation on monotonic behavior chains. In: Proceedings of the 12th ACM conference on recommender systems (2018)
25. Wang, W., Zhang, W., Liu, S., Liu, Q., Zhang, B., Lin, L., Zha, H.: Beyond clicks: Modeling multi-relational item graph for session-based target behavior prediction. In: Proc. of WWW (2020)
26. Wang, Y., Wang, L., Li, Y., He, D., Liu, T.Y.: A theoretical analysis of ndcg type ranking measures. In: Conference on learning theory (2013)
27. Wei, W., Huang, C., Xia, L., Xu, Y., Zhao, J., Yin, D.: Contrastive meta learning with behavior multiplicity for recommendation. In: Proc. of WSDM (2022)
28. Wei, Y., Wang, X., Li, Q., Nie, L., Li, Y., Li, X., Chua, T.S.: Contrastive learning for cold-start recommendation. In: Proc. of ACM MM (2021)
29. Xia, L., Xu, Y., Huang, C., Dai, P., Bo, L.: Graph meta network for multi-behavior recommendation. In: Proc. of SIGIR (2021)
30. Xu, G., Liu, Z., Li, X., Loy, C.C.: Knowledge distillation meets self-supervision. In: Proc. of ECCV (2020)
31. Zhang, J., Zhu, Y., Liu, Q., Zhang, M., Wu, S., Wang, L.: Latent structures mining with contrastive modality fusion for multimedia recommendation. arXiv preprint arXiv:2111.00678 (2021)
32. Zhang, J., Xu, X., Shen, F., Lu, H., Liu, X., Shen, H.T.: Enhancing audio-visual association with self-supervised curriculum learning. In: Proc. of AAAI (2021)