# Cross-domain Recommendation with Behavioral Importance Perception

Hong Chen
h-chen20@mails.tsinghua.edu.cn
Tsinghua University

Xin Wang*
xin_wang@tsinghua.edu.cn
Tsinghua University

Ruobing Xie
xrbsnowing@163.com
WeChat, Tencent

Yuwei Zhou
zhou-yw21@mails.tsinghua.edu.cn
Tsinghua University

Wenwu Zhu*
wwzhu@tsinghua.edu.cn
Tsinghua University

## ABSTRACT

Cross-domain recommendation (CDR) aims to leverage the source domain information to provide better recommendation for the target domain, which is widely adopted in recommender systems to alleviate the data sparsity and cold-start problems. However, existing CDR methods mostly focus on designing effective model architectures to transfer the source domain knowledge, ignoring the behavior-level effect during the loss optimization process, where behaviors regarding different aspects in the source domain may have different importance for the CDR model optimization. The ignorance of the behavior-level effect will cause the carefully designed model architectures ending up with sub-optimal parameters, which limits the recommendation performance. To tackle the problem, we propose a generic behavioral importance-aware optimization framework for cross-domain recommendation (BIAO). Specifically, we propose a behavioral perceptron which predicts the importance of each source behavior according to the corresponding item's global impact and local user-specific impact. The joint optimization process of the CDR model and the behavioral perceptron is formulated as a bi-level optimization problem. In the lower optimization, only the CDR model is updated with weighted source behavior loss and the target domain loss, while in the upper optimization, the behavioral perceptron is updated with implicit gradient from a developing dataset obtained through the proposed reorder-and-reuse strategy. Extensive experiments show that our proposed optimization framework consistently improves the performance of different cross-domain recommendation models in 7 cross-domain scenarios, demonstrating that our method can serve as a generic and powerful tool for cross-domain recommendation[1].

## CCS CONCEPTS

• **Information systems** → **Recommender systems**.

## KEYWORDS

cross-domain recommendation, behavior-aware optimization

**ACM Reference Format:**
Hong Chen, Xin Wang, Ruobing Xie, Yuwei Zhou, and Wenwu Zhu. 2023. Cross-domain Recommendation with Behavioral Importance Perception. In *Proceedings of the ACM Web Conference 2023 (WWW '23), April 30–May 04, 2023, Austin, TX, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3543507.3583494

## 1 INTRODUCTION

Recommender systems aim to provide personalized recommendation to users according to their historical behaviors, and have played an important role in various applications [6, 28]. However, the existing recommender systems often suffer from the data sparsity and the cold-start problems [8, 9, 28, 29, 31], where the data sparsity problem is caused by the insufficient user-item interaction records and the cold-start problem is generally caused by the new users. To address these problems, cross-domain recommendation, which improves the recommendation accuracy in the target domain by utilizing the user's behaviors in the source domain, has achieved great success recently [8, 12, 20, 27].

Existing cross-domain recommendation(CDR) methods mainly focus on designing effective model architectures to transfer knowledge from the source domain to the target domain. In particular, EMCDR-based methods [18, 33] aim to share the knowledge in the user's embedding, where in each domain the user's embedding is optimized with the source and target recommendation loss respectively, and then different mapping functions like global nonlinear function [18] and user-specific function [33] are learned to align the user's embedding in two domains. However, the transfer mechanism of EMCDR simply focuses on user embedding mapping, limiting their performance in recommendation. More advanced architectures [8, 12, 13, 17, 20] are recently proposed for more precise cross-domain recommendation. CoNet [8] adopts the cross-stitch structure in multi-task learning to transfer the source features to the target domain. MiNet [20] and DASL [12] design different kinds of attentive mechanisms to select the target-domain-related interest from the source behaviors. These methods [8, 12, 13, 17, 20] share not only user embedding but also deep model parameters through utilizing the joint loss of both the source and target domain to optimize the whole model, and thus achieve promising target recommendation performance.

Despite the effectiveness of the existing methods, they neglect the fact that behaviors regarding different aspects in the source

[*] Corresponding authors.
[1] Our code is available at https://github.com/forchchch/BIAO_WWW23

*WWW '23, April 30–May 04, 2023, Austin, TX, USA*
© 2023 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9416-1/23/04.
https://doi.org/10.1145/3543507.3583494

domain may have different importance for the CDR model optimization. They lack the behavior-level consideration for the joint loss to be optimized, which leads to sub-optimal model parameters for the carefully designed CDR models. Specifically, the joint loss adopted by existing works [8, 12, 13, 17, 20, 26, 30] is generally obtained as follows, (1) average the losses of all the behaviors in the target/source domain to obtain the target/source loss, (2) use a hyper-parameter weighted to the source loss to balance information from the two domains, (3) and add the two loss terms together to obtain the joint loss. However, an appropriate behavior-level consideration requires that not only should the losses between two domains be balanced, but the loss of each behavior in the source domain should also be balanced. For example, if we want to utilize the user's behavior in the book domain to help recommendation in the target toy domain, interacting with the books related to "children" or "toy stories" can provide beneficial information to the target domain, but interacting with "love story" books provides little helpful information. Additionally, as indicated by Man et al. [18], using the losses of users that have only few interactions in the source domain tend to harm the target domain recommendation accuracy. Therefore, existing works averaging the source behavior losses without behavior-level consideration fail to filter out the harmful information, thus only obtaining sub-optimal parameters for the carefully designed model architectures.

To address the problem, we propose to consider behavior-level importance through assigning an individual importance weight to each source behavior loss, which faces two challenges.

- When assigning a proper behavior-level importance, we will have to decide millions of hyper-parameters for each of the millions of source behaviors if each weight is regarded as a hyper-parameter, which brings explosive computational cost.
- Even though we can reduce the number of required hyper-parameters to an affordable level, it still remains a problem how we optimize the reduced hyper-parameters.

To tackle the challenges, we propose a generic behavioral importance-aware optimization framework (BIAO) for cross-domain recommendation, which involves a novel behavioral perceptron for the first challenge and a tailored bi-level optimization algorithm for the second challenge. The proposed behavioral perceptron learns the importance of each source behavior (i.e., a user-item pair) according to the item's global impact on the target domain as well as its local impact on a specific user, where the former is modeled by a global MLP and the latter is modeled with self-attention followed by a local MLP, which reduces the required hyper-parameters from million level to thousand level. The tailored bi-level optimization algorithm jointly optimizes the recommendation model parameters and the behavioral perceptron parameters. In the lower optimization, we utilize the target loss together with the importance weighted source loss to optimize the recommendation model parameters, while in the upper optimization, we update the perceptron parameters with implicit gradient from a developing dataset obtained through the designed reorder-and-reuse strategy. This strategy makes full use of all the target behaviors both in the upper and lower optimization, alleviating the potential bias and information loss in previous bi-level optimization works [2, 15]. The proposed behavioral perceptron is learned in a bi-level data-driven manner, thus being generic enough to automatically fit different recommendation models and datasets. Extensive experiments and analysis show that our proposed BIAO framework consistently improves the performance of different cross-domain recommendation models in seven cross-domain scenarios. Our contributions are summarized as follows,

- To the best of our knowledge, we are the first to consider behavior-level effect through assigning an individual importance weight to each source domain behavior loss for cross-domain recommendation.
- We propose a generic behavioral importance-aware optimization (BIAO) framework for cross-domain recommendation, which includes a novel behavioral perceptron and a tailored effective bi-level optimization algorithm.
- We conduct extensive experiments with different models on different datasets. Empirical results show that our proposed BIAO framework brings consistent performance improvement, demonstrating its ability to serve as a generic and powerful tool for cross-domain recommendation.

## 2 RELATED WORK

In this section, we review related work for cross-domain recommendation and bi-level optimization.

**Cross-Domain Recommendation**. Cross-domain recommendation aims to utilize the source domain information to provide more precise recommendation to users in the target domain. A line of typical methods are based on EMCDR [1, 10, 18, 32, 33]. The original EMCDR [18] utilizes the latent factor model to learn the user's embedding in the source domain and target domain respectively. Then a non-linear mapping from the source user embedding to the target domain is learned to transfer the knowledge. [32] proposes task-oriented loss to utilize the mapped embedding to predict the target behavior instead of target user embedding. [33] further proposes a personalized mapping function for each user and [10] gives a more reasonable metric learning for EMCDR. However, the EMCDR-based methods simply consider the user embedding mapping, limiting its performance in recommendation. More advanced cross-domain recommendation models were proposed recently [8, 12, 17, 20, 26]. CoNet [8] adopts the cross-stitch structure in multi-task learning to share the features between the source and target domain. $\pi$-Net [17], MiNet [20] and DASL [12] considers the sequential cross-domain recommendation, where attention mechanisms are adopted to select the useful source information. Despite the effectiveness of the proposed models, they simply utilize the linear combination of the source loss and the target loss to optimize the model, ignoring different impacts of behaviors in the source domain. To further exploit the potential of these models, we propose to conduct the behavioral importance-aware optimization for cross-domain recommendation.

**Bi-level Optimization**. The bi-level optimization problem arises in many scenarios of deep learning, like meta learning [5, 21], neural architecture search [14], and auxiliary learning [2, 3, 19]. To conduct the upper optimization, some works adopt the unrolled differentiation to calculate the upper-level gradient [4, 23]. However, several steps of unrolling will be memory-consuming [15] and the efficient one-step unrolling will suffer from short horizons [25]. The

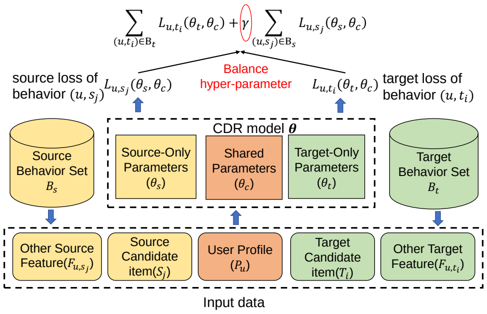

$\sum_{(u,t_i)\in B_t} L_{u,t_i}(\theta_t,\theta_c) + \gamma \sum_{(u,s_j)\in B_s} L_{u,s_j}(\theta_s,\theta_c)$
source loss of behavior $(u, s_j)$
$L_{u,s_j}(\theta_s,\theta_c)$
Balance hyper-parameter
$L_{u,t_i}(\theta_t,\theta_c)$
target loss of behavior $(u, t_i)$
Source Behavior Set $B_s$
CDR model $\boldsymbol{\theta}$
Source-Only Parameters $(\theta_s)$
Shared Parameters $(\theta_c)$
Target-Only Parameters $(\theta_t)$
Target Behavior Set $B_t$
Other Source Feature($F_{u,s_j}$)
Source Candidate item($S_j$)
User Profile ($P_u$)
Target Candidate item($T_i$)
Other Target Feature($F_{u,t_i}$)
Input data

**Figure 1: The schematic diagram of typical CDR models.**

implicit gradient is another widely adopted strategy for bi-level optimization. However, to fully obtain the implicit gradient, it needs to obtain the inverse of the Hessian matrix, which is computationally exhausted for deep models. Several works try to approximate the inverse of the Hessian, like using identity matrix [16], conjugate gradient [22] and truncated Neumann series [15]. All these methods generally split the training dataset into two disjoint subsets for the lower and upper optimization, which easily causes bias in the upper optimization and information loss in the lower optimization. In this paper, we adopt [15] to optimize the proposed novel behavioral perceptron, but propose a reorder-and-reuse strategy to alleviate the potential bias and information loss problems.

## 3 THE PROPOSED METHOD

In this section, we present preliminaries, the proposed behavioral perceptron and the effective optimization strategy.

### 3.1 Preliminaries

Assuming that there exists a set of users $U$ that have interactions with items in both the source domain $S$ and the target domain $T$. The widely CDR model schematic diagram is shown in Figure 1. The CDR dataset includes a source behavior set $B_s$ and a target behavior set $B_t$, which contains several user-item behavior pairs for training. The CDR model generally contains the user's profile $P_u$ (like user ID or age), the candidate item in the source domain $s_j$, the candidate item in the target domain $t_i$, and some other source/target features $F_{u,s_j}/F_{u,t_i}$ as input, where $(u, s_j)$ is a behavior from the source behavior set $B_s$ and $(u, t_i)$ is from the target behavior set $B_t$. The other features in the source/target domain $F_{u,s_j}/F_{u,t_i}$ could be some historical sequential features for sequential recommendation. The CDR model parameters $\theta$ generally includes target-only/source-only parameters $\theta_t/\theta_s$, like the final predictor of each domain in MiNet [20] or CoNet [8], and the shared parameters by both domains $\theta_c$, like the user profile embedding. Given the input data and the CDR model, the loss of the target behavior $L_{u,t_i}$ and the loss of the source behavior $L_{u,s_j}$ will be calculated. Finally, the loss of all the source behaviors and all the target behaviors are added together with a balanced factor $\gamma$, and the loss sum is used to optimize the CDR model parameters $\theta$:

$$\sum_{(u,t_i)\in B_t} L_{u,t_i}(\theta_t;\theta_c) + \gamma \sum_{(u,s_j)\in B_s} L_{u,s_j}(\theta_s;\theta_c). \tag{1}$$

The current methods only rely on a global hyper-parameter $\gamma$ to balance the source and target information. However, the differences of different behaviors $(u, s_j) \in B_s$ to the target domain recommendation are ignored, leading to sub-optimal $\theta$.
**Note:** Although the source and target loss are optimized together, CDR model only cares about the performance in the target domain.

### 3.2 The Behavioral Perceptron

To conduct the behavioral importance-aware optimization, our proposed optimization objective is as follows,

$$\min_{\theta} \sum_{(u,t_i)\in B_t} L_{u,t_i}(\theta_t;\theta_c) + \sum_{(u,s_j)\in B_s} \gamma_{uj} L_{u,s_j}(\theta_s;\theta_c), \tag{2}$$

where each source behavior loss is given an individual weight so that the beneficial behavior can be preserved while the harmful ones are discarded. Currently, the most important problem is how to decide the weights $\gamma_{uj}$ for the source-domain behavior. We propose a behavioral perceptron to perceive the importance of each behavior, whose structure is shown in Figure 2.

Specifically, for a behavior $(u, s_j)$ in the source domain, the behavioral perceptron judges its importance from the global importance of $s_j$ to the target domain and the local importance of $s_j$ to user $u$ as follows.

**Global item importance.** Since different kinds of items in the source domain have different impacts on the target domain, e.g., "cartoon" books are more beneficial than "love story" books when the target domain is toy, we adopt a multi-layer-perceptron(MLP) to map the features of $s_j$ to its importance weight. We concatenate all fields of its features like its ID, category, etc. to an embedding $E_{s_j}$, and then the global item weight is obtained as follows,

$$\gamma_{1,uj} = MLP(E_{s_j}; \phi_G), \tag{3}$$

where $\phi_G$ is the learnable parameters of the global MLP. Since this weight captures the importance of item $s_j$ to the whole target domain, we call it the global importance.

**User-specific item importance.** Besides the item information, we also judge the importance of $(u, s_j)$ from the user's historical interactions. If $s_j$ is more related to user's recent interactions, the $(u, s_j)$ behavior should be highly weighted. Specifically, the recent $N$ interactions of user $u$ in the source domain are $[x_S^1, x_S^2, \cdots, x_S^N]$, and in the target domain are $[x_T^1, x_T^2, \cdots, x_T^N]$. We first map each of the historical items to its embedding, and obtain $[e_S^1, e_S^2, \cdots, e_S^N]$ and $[e_T^1, e_T^2, \cdots, e_T^N]$. To make the perceptron find the interest that is related to $s_j$, we concatenate the embedding $E_{s_j}$ to each of item embedding in the two historical sequences, and we obtain $[e_{S,j}^1, e_{S,j}^2, \cdots, e_{S,j}^N]$ and $[e_{T,j}^1, e_{T,j}^2, \cdots, e_{T,j}^N]$, where $e_{S,j}^i = [e_S^i; E_{s_j}]$ and $e_{T,j}^i = [e_T^i; E_{s_j}]$ for $i \in \{1, 2, \cdots, N\}$. After obtaining the item-aware historical embeddings, we use two Multi-head Attention(MHA) [24] modules to extract the user's recent interest about $s_j$ in the source and target domain respectively.

$$[m_{S,j}^1, m_{S,j}^2, \cdots, m_{S,j}^N] = MHA_S([e_{S,j}^1, e_{S,j}^2, \cdots, e_{S,j}^N]; \phi_s), \tag{4}$$

$$[m_{T,j}^1, m_{T,j}^2, \cdots, m_{T,j}^N] = MHA_T([e_{T,j}^1, e_{T,j}^2, \cdots, e_{T,j}^N]; \phi_t), \tag{5}$$

where $\phi_s$ and $\phi_t$ are the parameters of MHA. After the MHA process, we use mean pooling to obtain the recent source/taregt interest

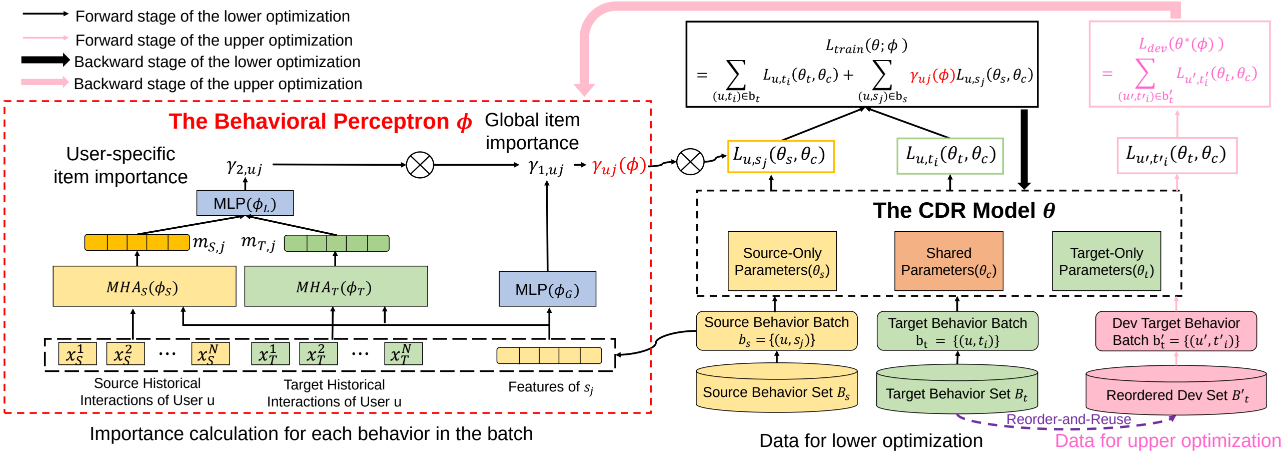

Forward stage of the lower optimization
Forward stage of the upper optimization
Backward stage of the lower optimization
Backward stage of the upper optimization
The Behavioral Perceptron $\phi$
Global item importance
User-specific item importance
$\gamma_{2,uj}$
$\gamma_{1,uj}$
$\gamma_{uj}(\phi)$
MLP($\phi_L$)
$m_{S,j}$
$m_{T,j}$
$MHA_S(\phi_S)$
$MHA_T(\phi_T)$
MLP($\phi_G$)
$x_S^1$ $x_S^2$ ⋯ $x_S^N$
$x_T^1$ $x_T^2$ ⋯ $x_T^N$
Source Historical Interactions of User u
Target Historical Interactions of User u
Features of $s_j$
Importance calculation for each behavior in the batch
$L_{train}(\theta;\phi) = \sum_{(u,t_i)\in b_t} L_{u,t_i}(\theta_t,\theta_c) + \sum_{(u,s_j)\in b_s} \gamma_{uj}(\phi) L_{u,s_j}(\theta_s,\theta_c)$
$L_{dev}(\theta^*(\phi)) = \sum_{(u\prime,t\prime_i)\in b_t'} L_{u',t_i'}(\theta_t,\theta_c)$
$L_{u,s_j}(\theta_s,\theta_c)$
$L_{u,t_i}(\theta_t,\theta_c)$
$L_{u\prime,t\prime_i}(\theta_t,\theta_c)$
The CDR Model $\theta$
Source-Only Parameters($\theta_s$)
Shared Parameters($\theta_c$)
Target-Only Parameters($\theta_t$)
Source Behavior Batch $b_s = \{(u,s_j)\}$
Target Behavior Batch $b_t = \{(u,t_i)\}$
Dev Target Behavior Batch $b_t' = \{(u',t'_i)\}$
Source Behavior Set $B_s$
Target Behavior Set $B_t$
Reordered Dev Set $B'_t$
Reorder-and-Reuse
Data for lower optimization
Data for upper optimization

**Figure 2: The proposed BIAO framework for cross-domain recommendation. In the lower optimization, the parameters of the behavioral perceptron $\phi$ are fixed and we only use the weighted training loss $L_{train}(\theta;\phi)$ to optimize the CDR model parameters $\theta$. In the upper optimization, we only use the loss on the reordered target set $L_{dev}(\theta^*(\phi))$ to update $\phi$ with the implicit gradient.**

about $s_j$, which are $m_{S,j}$ and $m_{T,j}$. Then we concatenate them together and use an MLP to obtain the final importance as follows,

$$m_{S,j} = \frac{1}{N}\sum_{i=1}^{N} m_{S,j}^i, m_{T,j} = \frac{1}{N}\sum_{i=1}^{N} m_{T,j}^i, \tag{6}$$

$$\gamma_{2,uj} = MLP([m_{S,j}; m_{T,j}]; \phi_L), \tag{7}$$

where $\phi_L$ is the paramters of the local MLP. This importance considers the information from the specific user $u$, and we call it the local user-specific importance.

Finally, we multiply the two importance weights and obtain the final importance of $(u, s_j)$:

$$\gamma_{uj}(\phi) = \gamma \cdot \sigma(\gamma_{1,uj} \cdot \gamma_{2,uj}), \tag{8}$$

where $\sigma$ is the normalization function, $\gamma$ is the learnable global scalar to balance information of the two domains, and $\gamma_{uj}$ is a function of $\phi$, and $\phi = \{\phi_G, \phi_s, \phi_t, \phi_L, \gamma\}$ contains all the behavioral perceptron parameters. Compared to directly assigning each source behavior a weight which requires millions of parameters, the behavior perceptron only involves several fully connected layers. Assuming that the embedding dimension of the item is $d$(typical value 32 or 64), the behavioral perceptron only requires $O(d^2)$ parameters, which is thousand level. However, how to optimize $\phi$ still remains a problem. Directly using Eq. (2) to optimize $\phi$ will easily cause $\gamma_{uj}$ to be zero, a trivial solution that cannot utilize the source loss information. Next, we present our bi-level optimization framework that jointly optimizes $\theta$ and $\phi$.

## 3.3 Overall Bi-level Optimization Formulation

The proposed behavioral importance-aware optimization framework is shown in Figure 2. Note that our ultimate goal is to obtain the optimal behavioral perceptron parameters $\phi$ which can select the most beneficial source behaviors to optimize the CDR model $\theta$, so that the CDR model performs best in the target domain. This goal can be formulated as a bi-level optimization problem as follows,

$$\phi^* = \arg\min_{\phi} L_{dev}(\theta^*(\phi)), \tag{9}$$
$$s.t. \quad \theta^*(\phi) = \arg\min_{\theta} L_{train}(\theta;\phi),$$

where $L_{train}(\theta;\phi)$ is the loss in Eq. (2) with $\gamma_{uj}$ obtained through Eq. (8). The lower optimization aims to find the optimal $\theta^*(\phi)$ that minimizes $L_{train}(\theta;\phi)$, i.e., optimize the CDR model parameters when the behavioral perceptron is fixed. Note that if $\phi$ changes, $\theta^*$ will also be changed, so $\theta^*$ is an implicit function of $\phi$, and we denote it as $\theta^*(\phi)$. $L_{dev}(\theta^*(\phi))$ is the loss of the CDR model on a new developing dataset $B_t'$ in the target domain. Assuming that if we can obtain an additional dataset in the target domain, our final optimization goal is that the CDR model $\theta^*(\phi)$ can achieve the best performance on the new developing target dataset, whose function is just like the validation dataset. Later we will explain how existing works and how we obtain the additional developing target dataset. Now, we still focus on how to conduct the lower and the upper optimization in Eq. (9).

***Lower Optimization.*** The lower optimization is quite straightforward. With the parameters of the behavioral perceptron $\phi$ fixed, we can use any optimizer like SGD or Adam [11] to optimize $\theta$, so that $L_{train}(\theta;\phi)$ is minimized.

***Upper Optimization.*** The upper optimization is a little more complex, since $L_{dev}(\theta^*(\phi))$ is the loss on the target domain and it directly relies on $\theta$ instead of $\phi$. We cannot use the autograd tools like SGD to calculate $\nabla_\phi L_{dev}(\theta^*(\phi))$. Therefore, considering that $\theta^*(\phi)$ is an implicit function of $\phi$, we utilize the chain rule to obtain the implicit gradient as follows,

$$\nabla_\phi L_{dev}(\theta^*(\phi)) = \nabla_\theta L_{dev}(\theta^*(\phi)) \nabla_\phi \theta^*(\phi), \tag{10}$$

where $\nabla_\theta L_{dev}(\theta^*(\phi))$ is easily obtained using the autograd tools and our target now is to obtain $\nabla_\phi \theta^*(\phi)$. Note that $\theta^*(\phi)$ is the minimal point of $L_{train}(\theta;\phi)$, so we have:

$$\nabla_\theta L_{train}(\theta^*(\phi), \phi) = 0. \tag{11}$$

Further calculating the gradient with respect to $\phi$ in both sides of Eq. (11), we can obtain the following results:

$$\nabla^2_{\theta} L_{train}(\theta^*(\phi),\phi)\nabla_{\phi}\theta^*(\phi) + \nabla_{\phi}\nabla_{\theta} L_{train}(\theta^*(\phi),\phi) = 0. \quad (12)$$

Therefore, the gradient $\nabla_{\phi}\theta^*(\phi)$ can be obtained as follows (we omit the parameters in $L_{train}$ for the sake of brevity.),

$$\nabla_{\phi}\theta^*(\phi) = -(\nabla^2_{\theta} L_{train})^{-1}\nabla_{\phi}\nabla_{\theta} L_{train}. \quad (13)$$

However, $(\nabla^2_{\theta} L_{train})^{-1}$, the Hessian inverse of the CDR model(a neural network), is usually intractable. We adopt the $K$-truncated Neumann series to approximate this inverse, where $(\nabla^2_{\theta} L_{train})^{-1} = \sum_{n=0}^{\infty}(I - \nabla^2_{\theta} L_{train})^n \approx \sum_{n=0}^{K}(I - \nabla^2_{\theta} L_{train})^n$. With these derivations, we can obtain the upper implicit gradient $\nabla_{\phi} L_{dev}$ as follows,

$$\nabla_{\phi} L_{dev} = -\nabla_{\theta} L_{dev} \cdot \sum_{n=0}^{K}(I - \nabla^2_{\theta} L_{train})^n \cdot \nabla_{\phi}\nabla_{\theta} L_{train}, \quad (14)$$

which can be efficiently calculated by the vector-Jacobi product [15].

Now, we can jointly optimize the CDR model parameters $\theta$ and the behavioral perceptron parameters $\phi$. Specifically, in the lower optimization, with $\phi$ fixed, we update the CDR parameters $\theta$ with the popular optimizer like SGD or Adam until convergence, and obtain the optimal $\theta^*(\phi)$. When the optimal $\theta^*(\phi)$ is reached, we switch to the upper optimization and use the gradient in Eq. (14) to update the perceptron parameters $\phi$. The lower and upper optimization are conducted in an alternating way until convergence.

## 3.4 The Practical Optimization Algorithm

Note that the optimization strategy discussed so far is still conducted on the whole dataset, which is impractical in real recommendation scenarios. Additionally, conducting the lower optimization until $\theta$ converges in each loop is inefficient and how to obtain the additional developing target dataset $B'_t$ still remains unresolved. To tackle these problems, we present a practically efficient and effective solution of our optimization algorithm in Algorithm 1. Note that there are three key points that make the algorithm more practical and efficient compared to previous theoretical analyses.

**Batch Optimization.** Note that in both the lower and upper optimization, we fetch batches from the whole dataset to calculate the loss and the gradient of parameters. Since we always cannot calculate the gradient of the whole dataset due to memory limit, this kind of batch optimization has been widely adopted in current deep learning and is also effective for the bi-level optimization.

***Interval* rounds of lower optimization instead of convergence.** Theoretically, with fixed $\phi$, we need to train the $\theta$ to its optimal point $\theta^*(\phi)$ and then we can conduct the upper optimization to update $\phi$. However, it is quite time-consuming because each time we update $\phi$, we need to experience a new complete lower training process. To make the algorithm more effective, we only conduct a fixed *interval* rounds of lower optimization for approximation, which is found effective in previous works [2, 15, 19].

**Algorithm 1** The BIAO Algorithm.

```
Input: source behavior set $B_s$, target behavior set $B_t$, interval between
two upper optimizations $interval$, lower and upper learning rate $\eta_1, \eta_2$,
lower optimizer $opt_1$, Neumann series truncated number $K$;
Initialize: CDR model parameter $\theta$, behavioral perceptron parameter $\phi$;
// the reorder and reuse trick
$B'_t = random\_shuffle(B_t)$
// the alternating lower and upper optimization loop
while not converged do
    // lower optimization
    for $round = 1$ to $interval$ do
        fetch train data batch $b_s = fetch(B_s), b_t = fetch(B_t)$;
        for each behavior $(u, s_j) \in b_s$ and $(s, t_i) \in b_t$ do
            $L_{u,s_j} = CDR(u, s_j; \theta), L_{u,t_i} = CDR(u, t_i; \theta)$;
            obtain $\gamma_{uj}(\phi)$ using Eq. (3) - (8);
        end for
        $L_{train}(\theta;\phi) = \sum_{(u,t_i)\in b_t} L_{u,t_i}(\theta) + \sum_{(u,s_j)\in b_s} \gamma_{uj}(\phi) L_{u,s_j}(\theta)$ ;
        $\theta \leftarrow opt_1(L_{train}(\theta;\phi),\eta_1)$
    end for
    // upper optimization
    fetch dev data batch: $b'_t = fetch(B'_t)$
    $L_{dev}(\theta) = \sum_{(u',t'_i)\in b'_t} L_{u',t'_i}(\theta)$;
    fetch a new train data batch: $bn_s = fetch(B_s), bn_t = fetch(B_t)$
    for each behavior $(u, s_j) \in bn_s$ and $(s, t_i) \in bn_t$ do
        $L_{u,s_j} = CDR(u, s_j; \theta), L_{u,t_i} = CDR(u, t_i; \theta)$;
        obtain $\gamma_{uj}(\phi)$ using Eq. (3) - (8);
    end for
    $L_{train}(\theta;\phi) = \sum_{(u,t_i)\in bn_t} L_{u,t_i}(\theta) + \sum_{(u,s_j)\in bn_s} \gamma_{uj}(\phi) L_{u,s_j}(\theta)$ ;
    //efficient vector-Jacobi calculation for Eq. (14)
    $v = p = \nabla_{\theta} L_{dev}(\theta)$;
    for $n = 1$ to $K$ do
        $v = v - v \cdot \nabla^2_{\theta} L_{train}(\theta;\phi)$
        $p = p + v$
    end for
    $\nabla_{\phi} L_{dev}(\theta) = -p\nabla_{\phi}\nabla_{\theta} L_{train}(\theta;\phi)$
    $\phi \leftarrow \phi - \eta_2 \nabla_{\phi} L_{dev}(\theta)$;
end while
Return $\theta^*(\phi^*)$
```

**The reorder-and-reuse strategy.** As we mentioned before, we need an additional developing target dataset to calculate $L_{dev}(\theta^*(\phi))$. Previous works [2, 15, 19] usually split a small dataset $B'_t$ from the target dataset $B_t$. These works use the rest set $B_t - B'_t$ for lower optimization and $B'_t$ for upper optimization. However, this kind of split easily causes bias in the upper optimization and information loss in the lower optimization. Specifically, in the upper optimization, we expect $\theta^*(\phi)$ has best performance on the developing dataset. If this developing dataset is only a small dataset split from the target dataset, it may only contain information of part of the users or items, thus easily biased. In the lower optimization in Figure 1, we note that the target-only parameter $\theta_t$ is only optimized with the target loss. If we only use $B_t - B'_t$ in the lower optimization, $\theta_t$ will easily become sub-optimal because of the lost information in $B'_t$. To tackle this problem, we propose the reorder-and-reuse strategy thanks to batch optimization. We reorder the target dataset $B_t$ and obtain the dataset $B'_t$ and use this reordered dataset to calculating $L_{dev}(\theta^*(\phi))$. This reordered dataset on the whole is the same as $B_t$ which cannot be regarded as a validation set when using the whole dataset for optimization, but luckily, we adopt batch optimization, so the batch $b_t$ from $B_t$ used in the lower optimization is different from the batch $b'_t$ from the reordered $B'_t$ in the upper optimization, and $b'_t$ can be regarded as a validation batch used to tune $\phi$. With this strategy, we can reuse $B_t$ to effectively conduct the bi-level optimization without requiring additional data in the target domain. The superiority of the reorder-and-reuse strategy compared to previous methods is validated in the experiments. The schematic diagram of the reorder-and-reuse strategy and previous methods is in Appendix C.

## 4 EXPERIMENTAL RESULTS

In this section, we empirically assess the efficacy of our proposed method on various datasets with different base models. Additionally, we provide ablations to show how our proposed method works.

### 4.1 Dataset

We conduct our experiments on the Amazon datasets [7], which contain users with their behaviors in different domains like book, movie and clothing. Specifically, to validate the generalization ability of our proposed method, we totally choose 7 domains from the Amazon datasets, which are Books, Movies, CDs, Cloth, Electronics(Elec), Home&Kitchen(Kitchen) and Toys. Based on these selected domains, we create 7 source-target cross-domain scenarios, i.e., Books-Movies, Books-CDs, Books-Elec, Books-Toys, CDs-Cloth, CDs-Kitchen, Elec-Cloth. These scenarios include both the intuitively highly-correlated domains like Books-Movies and intuitively less correlated domains like Elec-Cloth. The behaviors of the source and target domains are filtered by the common users between domains. Detailed statistics of the filtered datasets are presented in the appendix. The target domain behavior numbers of different scenarios are also quite different, e.g., the target behavior number of Books-Movies is 792,319 and CDs-Cloth only 36,319. The dataset split for the target domain is the same as that of [8, 20], where the test set is composed of the last behavior of each user, the validation set is composed of the second to last behavior, and the rest behaviors belong to the training set. The feature adopted for each user is its ID information, and the features for each item contain both its ID and category. Note that we keep the users with less than 5 ratings and items without metadata, different from [20], and thus our setting is closer to real scenarios.

### 4.2 Competitors and Evaluation Metrics

We choose CoNet [8] and MiNet [20], two typical models for cross-domain recommendation, as our base models, to which we apply our proposed method. Additionally, we also investigate other variants of the two base models to better present how the source domain information influences the model performance. Specifically, details of different models are as follows,

- **CoNet** [8] is a cross-domain recommendation model which adopts the cross-stitch structure in multi-task learning to share information between the source and target domain.
- **MiNet** [20] jointly considers the user's long-term interest across domains, short term interest from the target domain and the source domain with attentive mechanisms.
- **Base-0** indicates the variant that keeps the source branch information from the base model, but sets the coefficient for the source domain loss to 0.0, where the base model can be CoNet or MiNet. This variant can be used to validate whether the loss on the source domain can help the cross-domain recommendation, and can be regarded as the version where we only use the source domain features to help the target domain recommendation.
- **Base-S** indicates the variant that removes the source branch information from the base model. This variant not only removes the source loss, but also removes the features from the source domain. This variant is a single-domain version of the base model.
- **Base+ours** is the variant that utilizes our proposed BIAO method to reweight the loss of each source behavior.

We adopt **AUC** and **RelaImpr**, the same metrics as that of [20], to evaluate the models. Higher values indicate better performances.

### 4.3 Implementation

We implement all the methods with PyTorch. We optimize all the base models with Adam [11] optimizer, whose learning rate is searched from {1e-3, 5e-3, 1e-2} to fit different datasets, where the batch size is the same as that of the original paper. As for the hyperparameters in the upper optimization, the truncated number $K$ is fixed to 3 as adopted by previous works [2, 15], the *interval* for conducting upper optimization is searched from {20, 100, 500}, the length of the historical sequence used in the behavioral perceptron is 20 for MiNet and 50 for CoNet, the head number of the multi-head attention is 4, and the adopted upper optimizer is SGD with learning rate 1e-2 for all the scenarios. Note that the embedding table used in the perceptron is the same as that of the CDR model, but we stop its gradient in the perceptron so that the embeddings can be regarded as input instead of learnable parameters of the perceptron.

**Time Complexity Analysis.** During optimization, regarding the optimization time of MiNet/CoNet as unit "1", and considering that the additional computation brought by our method mainly comes from the backward process for the upper gradient, the time complexity of our method is as follows: In one lower-upper loop, the model conducts *interval* times of lower optimization and 1 upper optimization. The original MiNet/CoNet only needs *interval* lower backwards. Our method has the additional upper optimization which requires ($K$+2) backwards for the Jacobi calculation where $K$ is the truncated number, so it needs total $interval$+$K$+2 backwards, which results in $O((interval+K+2)/interval)$ complexity compared to the unit. Since $K$ in our experiments is fixed to 3, our method needs about O(1+5/*interval*) complexity. During inference, we do not change the model structure but only the parameters, the inference complexity is the same as that of the original model.

### 4.4 Recommendation Performance

The overall recommendation performance of different models is presented in Table 1. We have the following observations:

- **Our proposed BIAO method brings consistent improvements.** Whether utilizing MiNet or CoNet as the base model, our proposed method brings further improvement to the base model on all the datasets. Especially under the MiNet on Books-Toys, Elec-Cloth, and the CoNet on CDs-Cloth and Elec-Cloth settings, our BIAO method brings more than 10% RelaImpr improvement without changing model structures.
- **Our method has greater potential in tackling less irrelevant source-target transfer and cold-start scenarios.** It's worth noting that our proposed method achieves about 0.3% absolute AUC improvement in the Books-Movies scenario, but

Table 1: Overall recommendation performance. We run each method with 5 random seeds and report the mean performance. The * indicates 0.05 level, paired t-test of our method vs. the best baselines.

| Dataset | Books-Movies | | Books-CDs | | Books-Elec | | Books-Toys | | CDs-Cloth | | CDs-Kitchen | | Elec-Cloth | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | AUC | RelaI | AUC | RelaI | AUC | RelaI | AUC | RelaI | AUC | RelaI | AUC | RelaI | AUC | RelaI |
| MiNet-S | 0.7557 | 0.00% | 0.7019 | 0.00% | 0.6585 | 0.00% | 0.6706 | 0.00% | 0.5913 | 0.00% | 0.6161 | 0.00% | 0.6001 | 0.00% |
| MiNet-0 | 0.7508 | -1.92% | 0.6995 | -1.19% | 0.6523 | -3.91% | 0.6658 | -2.81% | 0.5950 | 4.05% | 0.6094 | -5.77% | 0.5885 | -11.59% |
| MiNet | 0.7616 | 2.31% | 0.7080 | 3.02% | 0.6540 | -2.84% | 0.6711 | 0.29% | 0.5876 | -4.05% | 0.6079 | -7.06% | 0.5966 | -3.50% |
| **MiNet+ours** | **0.7639*** | **3.21%** | **0.7145*** | **6.24%** | **0.6678*** | **5.87%** | **0.6883*** | **10.38%** | **0.6020*** | **11.72%** | **0.6212*** | **4.39%** | **0.6099*** | **9.79%** |
| CoNet-S | 0.7633 | 0.00% | 0.7165 | 0.00% | 0.6758 | 0.00% | 0.6918 | 0.00% | 0.5999 | 0.00% | 0.6465 | 0.00% | 0.6152 | 0.00% |
| CoNet-0 | 0.7643 | 0.38% | 0.7196 | 1.43% | 0.6808 | 2.84% | 0.6927 | 0.47% | 0.6134 | 13.51% | 0.6471 | 0.41% | 0.6206 | 4.69% |
| CoNet | 0.7693 | 2.28% | 0.7239 | 3.42% | 0.6764 | 0.34% | 0.6989 | 3.70% | 0.6110 | 11.11% | 0.6471 | 0.41% | 0.6238 | 7.47% |
| **CoNet+ours** | **0.7721*** | **3.34%** | **0.7274*** | **5.03%** | **0.6864*** | **6.03%** | **0.7048*** | **6.78%** | **0.6176*** | **17.72%** | **0.6558*** | **6.35%** | **0.6308*** | **13.54%** |

Table 2: Effectiveness of behavioral perceptron modules. We run each method with 5 seeds and report the mean performance.

| Dataset | Books-Movies | | Books-CDs | | Books-Elec | | Books-Toys | | CDs-Cloth | | CDs-Kitchen | | Elec-Cloth | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Model** | AUC | RelaI | AUC | RelaI | AUC | RelaI | AUC | RelaI | AUC | RelaI | AUC | RelaI | AUC | RelaI |
| **MiNet+ours** | **0.7639** | **3.21%** | **0.7145** | **6.24%** | **0.6678** | **5.87%** | **0.6883** | **10.38%** | **0.6020** | **11.72%** | 0.6212 | 4.39% | **0.6099** | **9.79%** |
| w/o global | 0.7632 | 2.93% | 0.7110 | 4.51% | 0.6645 | 3.79% | 0.6843 | 8.03% | 0.6016 | 11.28% | 0.6194 | 2.84% | 0.6060 | 5.89% |
| w/o user | 0.7618 | 2.39% | 0.7050 | 1.54% | 0.6657 | 4.54% | 0.6792 | 5.04% | 0.5910 | -0.33% | **0.6216** | **4.74%** | 0.6048 | 4.70% |
| **CoNet+ours** | **0.7721** | **3.34%** | **0.7274** | **5.03%** | **0.6864** | **6.03%** | **0.7048** | **6.78%** | **0.6176** | **17.72%** | **0.6558** | **6.35%** | **0.6308** | **13.54%** |
| w/o global | 0.7715 | 3.11% | 0.7265 | 4.62% | 0.6827 | 3.92% | 0.7032 | 5.94% | 0.6172 | 17.32% | 0.6524 | 4.03% | 0.6266 | 9.90% |
| w/o user | 0.7692 | 2.24% | 0.7254 | 4.11% | 0.6829 | 4.04% | 0.6935 | 0.89% | **0.6176** | **17.72%** | 0.6521 | 3.82% | 0.6263 | 9.64% |

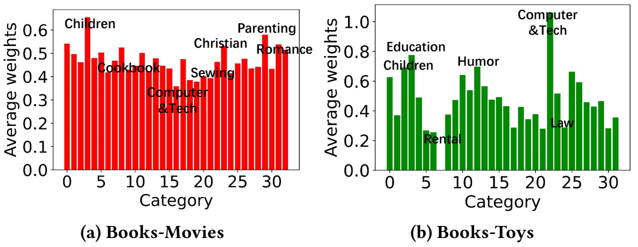
achieves more significant improvement in other scenarios like Books-Toys and CDs-Cloth. In the Books-Movies setting, the source domain and target domain information is quite similar, thus the optimal weights for different behaviors do not have large differences(which is also validated in 4.5), making the improvement brought by our method comparatively small. However, in the Books-Toys setting, where only specific categories of books have intuitive influence on recommendation in toys, our method brings significant improvement. Additionally, the RelaImpr of our method in the CDs-Cloth and Elec-Cloth is quite significant compared to other settings, where target behaviors in these two settings are quite inadequate, demonstrating the potential of our method to tackle the cold-start problem.

- **Our method has the ability of exploiting the beneficial information and discarding the harmful information in the source domain.** Note that under the Books-Elec, CDs-Kitchen and Elec-Cloth settings, both MiNet-0 and MiNet perform worse than the single-domain MiNet-S. This means both the feature and the loss from the source domain on average are harmful to the target domain. However, our method still surprisingly brings improvement compared to the MiNet-S baseline, indicating the strong ability of our method to discover the beneficial information from the on average harmful source domain behaviors.

## 4.5 The Learned Behavioral Weight

We record the learned weight of each source behavior loss on the Books-Movies and Books-Toys dataset with MiNet, and analyze their statistical characteristics. Figure 3 shows the average weight of items within each category. We have the following observations:

- **The behavioral perceptron selects highly-related behaviors from the source domain.** For example, on the Books-Toys dataset, the books with topics about "Children, Education, Humor, Computer" are highly weighted, indicating that the behavior of buying these books has greater influence on the user's behavior in the target toy domain. These highly weighted categories indeed have higher correlation with toys from human intuitions. Additionally, the books about "Rental" and "Law" are regarded as less important to the user's interest in toys.
- **The weight distribution varies with the datasets.** On the Books-Movies dataset, the weights of different categories are closer to each other, while on the Books-Toys dataset, the weights of different categories are quite different. It's not hard to understand this phenomenon. Compared to Books-Toys, Books and Movies are more similar. Almost for each category in books, we can find a similar category in Movies. Therefore, different categories in Books tend to have equal contributions to the target Movies domain. However, the Books and Toys are two less similar domains. Only the categories about toys in the Books domain will have obvious influence, like "Children".

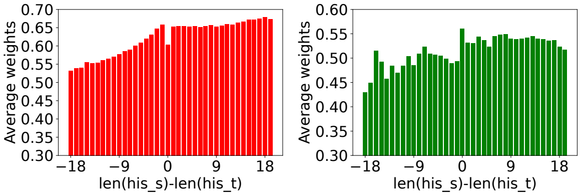


(a) Books-Movies

(b) Books-Toys

Figure 3: The average weight for items of each category in the source domain.

Besides the item categories, we also want to know whether the length of user historical behaviors in the source/target domain has influence on the learned weights. The x-axis of Figure 4 means the difference between the user's source behavior length and the target length(e.g., if a user has 5 interactions in the source domain and 10 interactions in the target domain, the difference is 5-10=-5). The y-axis means the learned average weight under each difference value. We can see that larger difference tends to lead to larger weights. This phenomenon is also intuitive. If a user has more interactions in the source domain and fewer interactions in the target domain, the source behavior of this user has higher probability to be valuable, which is consistent with [18]. Case analysis of the learned user-specific importance is given in Appendix B.



**(a) Books-Movies** **(b) Books-Toys**

**Figure 4: The weight changes with the user source-target behavior number.**

## 4.6 Behavioral Perceptron Effectiveness

We conduct an ablation study to validate the effectiveness of the designed *Global item importance* and the *User-specific item importance*. We report the performance of the model that removes any of the two components in Table 2. The w/o global refers to the variant that removes *Global item importance* and w/o user is the variant that removes the *User-specific item importance*. The results show that in almost all the settings, both of the two components are effective for the cross-domain recommendation.

## 4.7 Reorder-and-Reuse Strategy Effectiveness

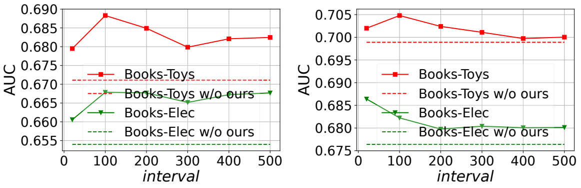
Previous works [2, 19] that utilize bi-level optimization split a small dataset from the training set, and use this small dataset for upper optimization. However, our proposed method reuses the whole training set and relies on batch optimization to make the data in the lower and upper optimization different during training. We compare our proposed method with the previous methods which utilize different split ratios. Specifically, we split 0.01, 0.05 and 0.1 of the whole target set for upper optimization, and the rest 0.99, 0.95 and 0.9 for lower optimization, respectively. Results in Table 3 show that the previous way to split a small dataset from the whole training set is not as effective as our proposed reorder-and-reuse method, where splitting too few samples makes the upper optimization biased and splitting too many samples does harm to the lower optimization. In most cases, splitting 0.1 data from the training set results in the worst performance, indicating that the benefits from weighting the source behavior cannot compensate for the degradation caused by worse lower optimization of the target-only parameters $\theta_t$. The proposed simple but effective reorder-and-reuse strategy can also be applied to other bi-level optimization problems.

**Table 3: Effectiveness of the Reorder-and-Reuse strategy. The top and bottom parts of the table give the results using MiNet and CoNet as base models, respectively.**

| Dataset | Books-CDs | | CDs-Cloth | | Elec-Cloth | |
|---|---|---|---|---|---|---|
| Split | AUC | RelaI | AUC | RelaI | AUC | RelaI |
| split 0.01 | 0.7037 | 0.89% | 0.6016 | 11.28% | 0.5996 | -0.53% |
| split 0.05 | 0.7019 | 0.00% | 0.5982 | 7.56% | 0.5993 | -0.80% |
| split 0.1 | 0.6960 | -2.92% | 0.5945 | 3.50% | 0.5970 | -3.10% |
| **ours** | **0.7145** | **6.24%** | **0.6020** | **11.72%** | **0.6099** | **9.79%** |
| split 0.01 | 0.7227 | 2.88% | **0.6176** | **17.72%** | 0.6244 | 8.02% |
| split 0.05 | 0.7204 | 1.79% | 0.6160 | 16.15% | 0.6237 | 7.41% |
| split 0.1 | 0.7176 | 0.52% | 0.6114 | 11.54% | 0.6254 | 8.88% |
| **ours** | **0.7274** | **5.03%** | **0.6176** | **17.72%** | **0.6308** | **13.54%** |

## 4.8 Hyper-parameter Sensitivity

Almost all the hyper-parameters involved in the optimization framework are fixed, except for the *interval* between two upper optimizations searched from {20, 100, 500}. Figure 5 shows the impact of *interval* on Books-Toys and Books-Elec with both MiNet and CoNet, *interval* set to {20, 100, 200, 300, 400, 500}. A larger *interval* makes $\phi$ in the upper optimization update more slowly, but the approximation errors of the implicit gradient will be smaller. Although the best *interval* varies with the base model and the dataset, our optimization brings consistent improvement with different *interval* compared to the original MiNet or CoNet(the dotted line named dataset w/o ours in the figure). Therefore, our method brings little HPO (Hyper-Parameter Optimization) burden for performance improvement and can be easily adopted.



**(a) MiNet** **(b) CoNet**

**Figure 5: Impact of the *interval* hyper-parameter, which determines how frequently we conduct the upper optimization.**

## 5 CONCLUSION

In this paper, we propose a behavioral importance-aware optimization framework for cross-domain recommendation, which automatically selects the most beneficial behaviors from the source domain to improve the target recommendation performance. The proposed framework involves the behavioral perceptron and the bi-level optimization based strategy, whose effectiveness has been validated through extensive experiments. Our proposed method can be combined with various cross-domain recommendation methods that jointly optimize the source and target loss, serving as a powerful tool for cross-domain recommendation. Future work like exploring more effective behavioral perceptron designs is interesting.

## ACKNOWLEDGMENTS

This work was supported in part by the National Key Research and Development Program of China No. 2020AAA0106300 and National Natural Science Foundation of China (No. 62250008, 62222209, 62102222).

## REFERENCES

[1] Jiangxia Cao, Jiawei Sheng, Xin Cong, Tingwen Liu, and Bin Wang. 2022. Cross-Domain Recommendation to Cold-Start Users via Variational Information Bottleneck. In *38th IEEE International Conference on Data Engineering, ICDE 2022, Kuala Lumpur, Malaysia, May 9-12, 2022*. IEEE, 2209–2223.

[2] Hong Chen, Xin Wang, Chaoyu Guan, Yue Liu, and Wenwu Zhu. 2022. Auxiliary Learning with Joint Task and Data Scheduling. In *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA (Proceedings of Machine Learning Research, Vol. 162)*. PMLR, 3634–3647.

[3] Hong Chen, Xin Wang, Yue Liu, Yuwei Zhou, Chaoyu Guan, and Wenwu Zhu. 2022. Module-Aware Optimization for Auxiliary Learning. In *Advances in Neural Information Processing Systems*.

[4] Justin Domke. 2012. Generic methods for optimization-based modeling. In *Artificial Intelligence and Statistics*. PMLR, 318–326.

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*. PMLR, 1126–1135.

[6] Xiaobo Hao, Yudan Liu, Ruobing Xie, Kaikai Ge, Linyao Tang, Xu Zhang, and Leyu Lin. 2021. Adversarial Feature Translation for Multi-domain Recommendation. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*. ACM, 2964–2973.

[7] Ruining He and Julian J. McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proceedings of the 25th International Conference on World Wide Web, WWW 2016, Montreal, Canada, April 11 - 15, 2016*, Jacqueline Bourdeau, Jim Hendler, Roger Nkambou, Ian Horrocks, and Ben Y. Zhao (Eds.). ACM, 507–517.

[8] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*. ACM, 667–676.

[9] Xiaowen Huang, Jitao Sang, Jian Yu, and Changsheng Xu. 2022. Learning to Learn a Cold-start Sequential Recommender. *ACM Trans. Inf. Syst.* 40, 2 (2022), 30:1–30:25.

[10] SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-Supervised Learning for Cross-Domain Recommendation to Cold-Start Users. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*. ACM, 1563–1572.

[11] D. Kingma and J. Ba. 2014. Adam: A Method for Stochastic Optimization. *Computer Science* (2014).

[12] Pan Li, Zhichao Jiang, Maofei Que, Yao Hu, and Alexander Tuzhilin. 2021. Dual Attentive Sequential Learning for Cross-Domain Click-Through Rate Prediction. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*. ACM, 3172–3180.

[13] Pan Li and Alexander Tuzhilin. 2020. DDTCDR: Deep Dual Transfer Cross Domain Recommendation. In *WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020*. ACM, 331–339.

[14] Hanxiao Liu, Karen Simonyan, and Yiming Yang. 2019. DARTS: Differentiable Architecture Search. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

[15] Jonathan Lorraine, Paul Vicol, and David Duvenaud. 2020. Optimizing Millions of Hyperparameters by Implicit Differentiation. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020 (Proceedings of Machine Learning Research)*. PMLR, 1540–1552.

[16] Jelena Luketina, Mathias Berglund, Klaus Greff, and Tapani Raiko. 2016. Scalable gradient-based tuning of continuous regularization hyperparameters. In *International conference on machine learning*. PMLR, 2952–2960.

[17] Muyang Ma, Pengjie Ren, Yujie Lin, Zhumin Chen, Jun Ma, and Maarten de Rijke. 2019. $\pi$-Net: A Parallel Information-sharing Network for Shared-account Cross-domain Sequential Recommendations. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2019, Paris, France, July 21-25, 2019*. ACM, 685–694.

[18] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. ijcai.org, 2464–2470.

[19] Aviv Navon, Idan Achituve, Haggai Maron, Gal Chechik, and Ethan Fetaya. 2021. Auxiliary Learning by Implicit Differentiation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*.

[20] Wentao Ouyang, Xiuwu Zhang, Lei Zhao, Jinmei Luo, Yu Zhang, Heng Zou, Zhaojie Liu, and Yanlong Du. 2020. MiNet: Mixed Interest Network for Cross-Domain Click-Through Rate Prediction. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*. ACM, 2669–2676.

[21] Aravind Rajeswaran, Chelsea Finn, Sham M. Kakade, and Sergey Levine. 2019. Meta-Learning with Implicit Gradients. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*. 113–124.

[22] Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. 2019. Meta-learning with implicit gradients. *Advances in neural information processing systems* 32 (2019).

[23] Amirreza Shaban, Ching-An Cheng, Nathan Hatch, and Byron Boots. 2019. Truncated back-propagation for bilevel optimization. In *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 1723–1732.

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 5998–6008.

[25] Yuhuai Wu, Mengye Ren, Renjie Liao, and Roger Grosse. 2018. Understanding short-horizon bias in stochastic meta-optimization. (2018).

[26] Ruobing Xie, Qi Liu, Liangdong Wang, Shukai Liu, Bo Zhang, and Leyu Lin. 2022. Contrastive Cross-domain Recommendation in Matching. In *KDD '22: The 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, August 14 - 18, 2022*. ACM, 4226–4236.

[27] Feng Yuan, Lina Yao, and Boualem Benatallah. 2019. DARec: Deep Domain Adaptation for Cross-Domain Recommendation via Transferring Rating Patterns. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*. ijcai.org, 4227–4233.

[28] Tianzi Zang, Yanmin Zhu, Haobing Liu, Ruohan Zhang, and Jiadi Yu. 2021. A Survey on Cross-domain Recommendation: Taxonomies, Methods, and Future Directions. *CoRR* abs/2108.03357 (2021).

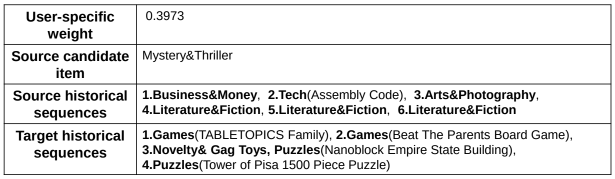
[29] Cheng Zhao, Chenliang Li, Rong Xiao, Hongbo Deng, and Aixin Sun. 2020. CATN: Cross-Domain Recommendation for Cold-Start Users via Aspect Transfer Network. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 229–238.

[30] Cheng Zhao, Chenliang Li, Rong Xiao, Hongbo Deng, and Aixin Sun. 2020. CATN: Cross-Domain Recommendation for Cold-Start Users via Aspect Transfer Network. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 229–238.

[31] Feng Zhu, Yan Wang, Chaochao Chen, Jun Zhou, Longfei Li, and Guanfeng Liu. 2021. Cross-Domain Recommendation: Challenges, Progress, and Prospects. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021*. 4721–4728.

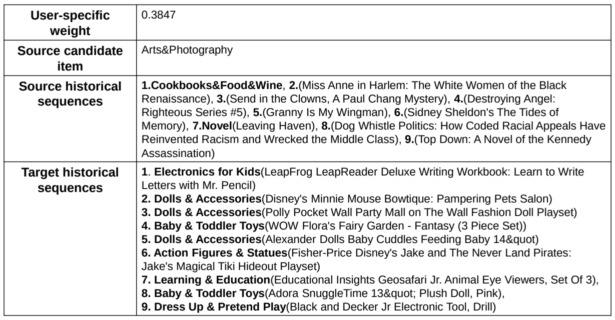
[32] Yongchun Zhu, Kaikai Ge, Fuzhen Zhuang, Ruobing Xie, Dongbo Xi, Xu Zhang, Leyu Lin, and Qing He. 2021. Transfer-Meta Framework for Cross-domain Recommendation to Cold-Start Users. In *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*. ACM, 1813–1817.

[33] Yongchun Zhu, Zhenwei Tang, Yudan Liu, Fuzhen Zhuang, Ruobing Xie, Xu Zhang, Leyu Lin, and Qing He. 2022. Personalized Transfer of User Preferences for Cross-domain Recommendation. In *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*. ACM, 1507–1515.



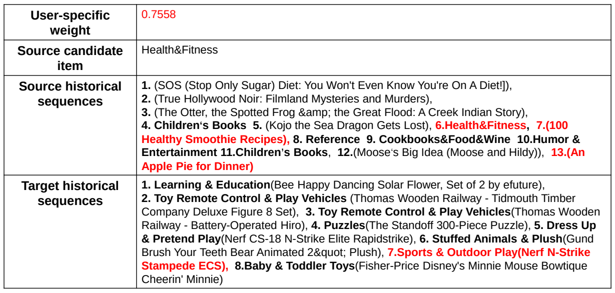
| | |
|---|---|
| **User-specific weight** | 0.3973 |
| **Source candidate item** | Mystery&Thriller |
| **Source historical sequences** | **1.Business&Money**, **2.Tech**(Assembly Code), **3.Arts&Photography**, **4.Literature&Fiction**, **5.Literature&Fiction**, **6.Literature&Fiction** |
| **Target historical sequences** | **1.Games**(TABLETOPICS Family), **2.Games**(Beat The Parents Board Game), **3.Novelty& Gag Toys, Puzzles**(Nanoblock Empire State Building), **4.Puzzles**(Tower of Pisa 1500 Piece Puzzle) |

(a) case1



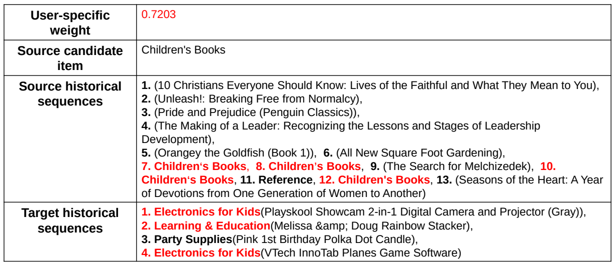
| | |
|---|---|
| **User-specific weight** | 0.3847 |
| **Source candidate item** | Arts&Photography |
| **Source historical sequences** | **1.Cookbooks&Food&Wine**, **2.**(Miss Anne in Harlem: The White Women of the Black Renaissance), **3.**(Send in the Clowns, A Paul Chang Mystery), **4.**(Destroying Angel: Righteous Series #5), **5.**(Granny Is My Wingman), **6.**(Sidney Sheldon's The Tides of Memory), **7.Novel**(Leaving Haven), **8.**(Dog Whistle Politics: How Coded Racial Appeals Have Reinvented Racism and Wrecked the Middle Class), **9.**(Top Down: A Novel of the Kennedy Assassination) |
| **Target historical sequences** | 1. **Electronics for Kids**(LeapFrog LeapReader Deluxe Writing Workbook: Learn to Write Letters with Mr. Pencil)<br>**2. Dolls & Accessories**(Disney's Minnie Mouse Bowtique: Pampering Pets Salon)<br>**3. Dolls & Accessories**(Polly Pocket Wall Party Mall on The Wall Fashion Doll Playset)<br>**4. Baby & Toddler Toys**(WOW Flora's Fairy Garden - Fantasy (3 Piece Set))<br>**5. Dolls & Accessories**(Alexander Dolls Baby Cuddles Feeding Baby 14&quot)<br>**6. Action Figures & Statues**(Fisher-Price Disney's Jake and The Never Land Pirates: Jake's Magical Tiki Hideout Playset)<br>**7. Learning & Education**(Educational Insights Geosafari Jr. Animal Eye Viewers, Set Of 3),<br>**8. Baby & Toddler Toys**(Adora SnuggleTime 13&quot; Plush Doll, Pink),<br>**9. Dress Up & Pretend Play**(Black and Decker Jr Electronic Tool, Drill) |

(b) case2



| | |
|---|---|
| **User-specific weight** | 0.7558 |
| **Source candidate item** | Health&Fitness |
| **Source historical sequences** | **1.** (SOS (Stop Only Sugar) Diet: You Won't Even Know You're On A Diet!]),<br>**2.** (True Hollywood Noir: Filmland Mysteries and Murders),<br>**3.** (The Otter, the Spotted Frog &amp; the Great Flood: A Creek Indian Story),<br>**4. Children‘s Books 5.** (Kojo the Sea Dragon Gets Lost), **6.Health&Fitness**, **7.(100 Healthy Smoothie Recipes),** **8. Reference 9. Cookbooks&Food&Wine 10.Humor & Entertainment 11.Children’s Books**, **12.**(Moose‘s Big Idea (Moose and Hildy)), **13.(An Apple Pie for Dinner)** |
| **Target historical sequences** | **1. Learning & Education**(Bee Happy Dancing Solar Flower, Set of 2 by efuture),<br>**2. Toy Remote Control & Play Vehicles** (Thomas Wooden Railway - Tidmouth Timber Company Deluxe Figure 8 Set), **3. Toy Remote Control & Play Vehicles**(Thomas Wooden Railway - Battery-Operated Hiro), **4. Puzzles**(The Standoff 300-Piece Puzzle), **5. Dress Up & Pretend Play**(Nerf CS-18 N-Strike Elite Rapidstrike), **6. Stuffed Animals & Plush**(Gund Brush Your Teeth Bear Animated 2&quot; Plush), **7.Sports & Outdoor Play(Nerf N-Strike Stampede ECS),** **8.Baby & Toddler Toys**(Fisher-Price Disney's Minnie Mouse Bowtique Cheerin' Minnie) |

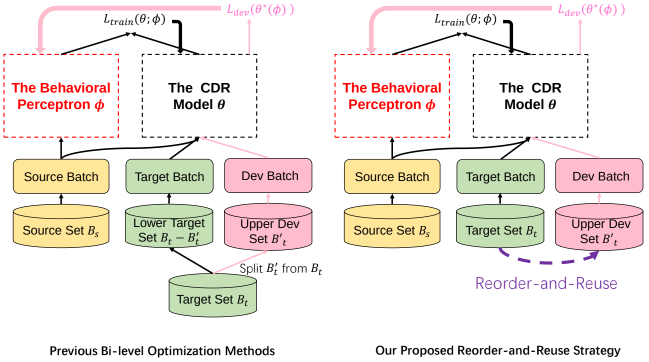
(c) case3



| | |
|---|---|
| **User-specific weight** | 0.7203 |
| **Source candidate item** | Children's Books |
| **Source historical sequences** | **1.** (10 Christians Everyone Should Know: Lives of the Faithful and What They Mean to You),<br>**2.** (Unleash!: Breaking Free from Normalcy),<br>**3.** (Pride and Prejudice (Penguin Classics)),<br>**4.** (The Making of a Leader: Recognizing the Lessons and Stages of Leadership Development),<br>**5.** (Orangey the Goldfish (Book 1)), **6.** (All New Square Foot Gardening),<br>**7. Children‘s Books**, **8. Children’s Books**, **9.** (The Search for Melchizedek), **10. Children‘s Books**, **11. Reference**, **12. Children's Books**, **13.** (Seasons of the Heart: A Year of Devotions from One Generation of Women to Another) |
| **Target historical sequences** | **1. Electronics for Kids**(Playskool Showcam 2-in-1 Digital Camera and Projector (Gray)),<br>**2. Learning & Education**(Melissa &amp; Doug Rainbow Stacker),<br>**3. Party Supplies**(Pink 1st Birthday Polka Dot Candle),<br>**4. Electronics for Kids**(VTech InnoTab Planes Game Software) |

(d) case4

**Figure 6: Case study for user-specific importance weights in Books-Toys. Each sub-figure gives 4 aspects of a source behavior, i.e., the final learned user-specific weight, the item in the source behavior, the user’s historical sequences in the source domain, and the user’s historical sequences in the target domain. Each item is represented in the form of category(title) if existed. Case3 and case4 assign high weights to the source behavior, where in case4 the source candidate item is related to the target domain and the users’ recent behaviors, while in case3 the ’Health&Fitness’ source item is less related to the target domain in global sense but quite fits the user’s specific interests in both domains.**



Previous Bi-level Optimization Methods

Our Proposed Reorder-and-Reuse Strategy

**Figure 7: Comparison between the reorder-and-reuse strategy and previous methods.**

## A DATASET STATISTICS

We provide the data statistics of our experiments in Table 4. We also conduct several experiments on a more recent backbone DASL [12]. And the results are shown in Table 5. The results further show that our proposed optimization method can be applied to various current CDR models.

## B USER-SPECIFIC IMPORTANCE CASE STUDY

In section 4.5, we analyze the statistical characteristics of the learned weights. The learned average weights under each category show that the global item importance is important. In this section, we further provide some case studies in the Books-Toys scenario to analyze the user-specific importance weights in Figure 6. In case1 and case2, the two behaviors are assigned to low user-specific weights because the source item has very low correlations with the source historical behaviors and target historical behaviors. For example, in case2, the candidate source item is about photography, but the source sequences are almost about novel and the target sequences are about kid’s toys, making this source behavior less worth learning. In case3 and case4, the source candidate item has high correlations with both source and target historical sequences (highlighted in red), thus are assigned higher weights to learn. Particularly, in case3, the behavior “Health&Fitness” book in the source domain intuitively will have little importance to the target toy domain. However, due to the user’s specific interest in “health” and “sports” shown in his historical behaviors, it is assigned a high weight. This phenomenon also indicates the significance of considering the local user-specific importance.

## C COMPARISON BETWEEN PREVIOUS METHODS AND THE PROPOSED REORDER-AND-REUSE STRATEGY

Figure 7 presents how previous bi-level optimization methods obtain the developing target dataset and how our proposed reorder-and-reuse strategy obtains the developing dataset. In previous methods, they split the original target dataset $B_t$ into two disjoint dataset for the lower and upper optimization, respectively. This kind of split easily causes bias and information loss as analyzed in the paper,

**Table 4: Data Statistics**

| | Books-Movies | Books-CDs | Books-Elec | Books-Toys | CDs-Cloth | CDs-Kitchen | Elec-Cloth |
|---|---|---|---|---|---|---|---|
| #source behavior | 944,533 | 418,653 | 735,683 | 317,646 | 71,491 | 160,358 | 99,594 |
| #target behavior | 792,319 | 380,675 | 364,267 | 84,564 | 36,319 | 96,670 | 66,470 |
| #users | 37,388 | 16,738 | 28,506 | 7,576 | 4,283 | 8,144 | 8,235 |
| #source item | 269,301 | 191,942 | 229,189 | 134,350 | 26,178 | 36,018 | 25,816 |
| #target item | 49,273 | 61,201 | 52,134 | 11,567 | 15,445 | 22,550 | 18,703 |

**Table 5: Additional experiments with the DASL backbone.**

| Model | DASL-S | DASL-0 | DASL | DASL+ours |
|---|---|---|---|---|
| **Books-CDs** | 0.7106 | 0.7128 | 0.7100 | **0.7165** |
| **Books-Toys** | 0.6771 | 0.6786 | 0.6734 | **0.6842** |

especially in the cross-domain recommendation setting where the behaviors in the target domain are quite sparse. However, the proposed reorder-and-reuse strategy enables that both the lower and upper optimization make full use of all the target data, alleviating the potential bias and information loss.