2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Metaverse)

2022 IEEE Smartworld, Ubiquitous Intelligence & Computing, Scalable Computing & Communications, Digital Twin, Privacy Computing, Metaverse, Autonomous & Trusted Vehicles (SmartWorld/UIC/ScalCom/DigitalTwin/PriComp/Meta) | 979-8-3503-4655-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/SmartWorld-UIC-ATC-ScalCom-DigitalTwin-PriComp-Metaverse56740.2022.00086

A Multi-Head Attention Based Dual Target Graph Collaborative Filtering Network

Qinglong Peng
School of Information Science and Technology
Qingdao University of Science and Technology
Qingdao, China
pengqinglong@mails.qust.edu.cn

Junwei Du
School of Information Science and Technology
Qingdao University of Science and Technology
Qingdao, China
djwqd@163.com

Bin Tang
Collegel of Shipbuilding Engineering
Harbin Engineering University
Harbin, China
tangbin@hrbeu.edu.cn

Yan Lu
School of Information Science and Technology
Qingdao University of Science and Technology
Qingdao, China
luyan@mails.qust.edu.cn

Jinhuan Liu
School of Information Science and Technology
Qingdao University of Science and Technology
Qingdao, China
liujinhuan.sdu@gmail.com

Feng Jiang
School of Information Science and Technology
Qingdao University of Science and Technology
Qingdao, China
jiangfeng@qust.edu.cn

Shuang Cui
School of Information Science and Technology
Qingdao University of Science and Technology
Qingdao, China
cuishuang@mails.qust.edu.cn

Xu Yu
School of Information Science and Technology
Qingdao University of Science and Technology
Qingdao, China
yuxu0532@163.com

***Abstract*—Recently, cross-domain collaborative filtering (CDCF) has been widely used to solve the data sparsity problem in recommendation systems. Therein, the dual-target cross-domain recommendation becomes a research hotspot, which aims to improve the recommendation performance of both target and source domains. Most existing approaches tend to use fixed weights or self-attention in a single representation space for the bi-directional inter-domain transfer of the user representation. However, a single representation space leads to limited representation capability, which makes the transfer of the user representation coarse-grained and inaccurate. In this paper, Multi-head Attention Based Dual Target Graph Collaborative Filtering Network (MA-DTGCF) is proposed. The core of the model is the bi-directional transfer graph convolution layer, consisting of a graph convolution layer and a bi-directional transfer layer based on a multi-head attention mechanism. The latter can achieve fine-grained and adaptive transfer of user features in multiple representation subspaces. It is worth noting that by stacking multiple bi-directional transfer graph convolutional layers, we can get high-order user and item features and achieve adaptive transfer of each order user features. Experimental results on three real datasets show that the proposed MA-DTGCF model significantly outperforms the state-of-the-art models in terms of HR and NDCG.**

***Index Terms*—Cross-domain collaborative filtering, Multi-head attention, Graph neural network**

I. Introduction

Recently, with the development of the internet, various kinds of information are filling people's lives. In order to solve the problem of information overload, personalized recommendation systems [1]–[3] are widely used in real life and gradually become an integral part of the Internet, such as the recommendation system in Taobao. The core of the personalized recommendation system is to recommend a set of products that users are likely to interact with, such as clicking, buying, etc., by using the user's historical data. Collaborative filtering [1], [4], [5] is the most widely used model in the personalized recommendation, which learns the model by mining the relationship between users and items in historical interaction data. Its core is to learn the implicit features of users and items with the help of their relationships. Then, it makes recommendation prediction based on the implicit features. However, in real life, users

This work is jointly sponsored by National Natural Science Foundation of China (Nos. 62172249, 61973180), Natural Science Foundation of Shandong Province (Nos. ZR2019MF014, ZR2021MF092). *(Corresponding author: Xu Yu)*

979-8-3503-4655-8/22/$31.00 ©2022 IEEE

DOI 10.1109/SmartWorld-UIC-ATC-ScalCom-DigitalTwin-PriComp-Metaverse56740.2022.00086

often do not like ratings, and the number of users and items is huge. Whatever collaborative filtering algorithm will face a very serious data sparsity problem.

Cross-domain collaborative filtering(CDCF) is the most commonly used method to solve the data sparsity problem, which transfers the information from the source domain with relatively dense data to the target domain with sparse data to improve the recommendation performance of the target domain. It consists of single-target and dual-target CDCF. For single-target CDCF [6]–[8], the optimization objective is to improve the recommendation performance of the target domain. But, some researchers have found that the information in the target domain can also be used to improve the recommendation performance of the auxiliary domain, i.e., the dual-target CDCF [2], [9], [10]. It achieves the simultaneous improvement of recommendation performance in both domains through bi-directional transfer of information. Most existing dual-target CDCF approaches [2], [10] tend to use fixed weights or self-attention in a single representation space for the bi-directional inter-domain transfer of the user representation. However, The representation capability of single feature space is limited and coarse-grained [11], which can only represent users and perform feature transfer at a single perspective and lead to inaccurate calculation of user feature transfer weights or attention mechanisms. Also, They use same transfer strategy to transfer each order user features in user-item graph. The amount of transferable information in each order user features is different, So, the same transfer strategy to each order user features is inappropriate. Finally, it may hurt performance of target domain.

To address the limitations of the existing dual-target cross-domain recommendation models, this paper proposes a **M**ulti-head **A**ttention Based **D**aul **T**arget **G**raph **C**ollaborative **F**iltering Network(MA-DTGCF). First, we construct user-item heterogeneous graphs for each of the two domains. Then, the bi-directional transfer graph convolution layer is used to aggregate the information on two heterogeneous graphs respectively and transfer the features of the common users from two domains in multiple representation subspaces. To get the high-order features of users and items and achieve adaptive transfer of each order user features, the bi-directional transfer graph convolution layer will be stacked multiple times. Finally, the click-through rate prediction of the two domains is performed. The main contributions of this paper are as follows:

1) We design a bi-direction transfer graph convolution layer, which can aggregate information in the user-item graph of each domain respectively and achieve fine-grained, adaptive transfer of user features in multiple representation subspaces.
2) By stacking multiple bi-directional transfer graph convolutional layers, we can get high-order user and item features and achieve adaptive transfer of each order user features.
3) Adequate experiments are conducted on three real datasets, and the experiment results show that MA-DTGCF significantly outperforms the state-of-the-art model in terms of HR and NDCG.

The remainder is organized in the following. Related work will be reviewed in Section II. The details of the model will be given in Section III. Section IV includes a detailed experimental procedure and a comparison analysis. Finally, we present the conclusion in Section V.

II. Related Work

A. Cross-domain Collaborative Filtering

Existing cross-domain collaborative filtering models can be classified into single-target and dual-target cross-domain collaborative filtering. For single-target cross-domain collaborative filtering models, Berkovsky et al. proposed Neighbor-based CDCF (N-CDCF) [6], and Singh and Gordon proposed CMF [7], both of which achieved improved recommendation performance through shared users/items. Hu et al. proposed CoNet [8], which establishes cross-connections between the networks of two domains. Yu et al. proposed PPCDHWRec [12] model to transfer user latent features and achieve privacy protection for user raw ratings. For the dual-target cross-domain collaborative filtering, Zhu et al. proposed the DTCDR [13] model, a preliminary study of the user feature fusion approach in the dual-target scenario. By applying the graph collaborative filtering model with good performance in a single domain to the dual-target cross-domain recommendation scenario, Zhao et al. proposed PPGN [9]. For the negative transfer problem in PPGN, Liu et al. proposed the bi-directional transfer graph collaborative filtering model BITGCF [10]. Unlike the fixed weights fusion of features in BITGCF, Zhu et al. proposed GA-DTCDR [2], a dual-target cross-domain recommendation model based on self-attention mechanism and graph representation learning. Both BITGCF and GA-DTGCF tend to use fixed weights or self-attention in a single representation space for the bi-directional inter-domain transfer of the user representation. However, a single representation space leads to limited representation capability, which makes the transfer of the user representation inaccurate.

B. Graph Collaborative Filtering

By mining the information on the user-item heterogeneous graph and modeling the higher-order interaction between users and items, the relationship between users and items can be fully explored. Earlier models like ItemRank [14] used a label propagation mechanism to propagate user preference scores directly on the graph by encouraging connected nodes to have similar labels. Then, with the proposal of graph neural networks [15], many researchers used it in recommender systems due to its ability to mine higher-order interactions in graphs. Berg, Kipf, and Welling proposed GC-MC [16], a collaborative filtering model based on graph convolutional neural networks. For large-scale scenarios, Ying et al. proposed PinSage [17]. Recently, Wang et al. proposed a graph collaborative filtering model NGCF [18] with user-item feature interaction terms. He et al. demonstrated that the weight matrices and nonlinear part in the traditional graph collaborative filtering model are worthless for collaborative filtering, and proposed LightGCN [1] that achieves the simultaneous improvement of recommendation performance and computational efficiency.

III. The Proposed Algorithm

In this section, the **M**ulti-head **A**ttention Based **D**ual **T**arget **G**raph **C**ollaborative **F**iltering Network (MA-DTGCF) will be introduced. First, we define the dual-target cross-domain recommendation problem. Then, we will present the main structure of the model. Finally, the components and details of the model will be presented in the remaining subsections.

A. Problem Definition

The scenario in this paper is a dual-target cross-domain recommendation scenario with fully overlapping users and no overlapping items. We consider two domains A and B. Let U denotes the set of users, and the length of U is m. Let the set of items in the two domains be I_A and I_B, respectively, and the lengths of them are n_A and n_B. Let $\mathbf{Y}_A = [y_{u^A i^A}]_{m \times n_A} \in \{0{,}1\}^{m \times n_A}$ and $\mathbf{Y}_B = [y_{u^B i^B}]_{m \times n_B} \in \{0{,}1\}^{m \times n_B}$ denote the implicit feedback matrices, where “1” means that the interaction between a user and a item is observed, and “0” otherwise. In this paper, we convert the traditional rating prediction into a click-through rate prediction problem and use the implicit feedback information for Top-K recommendation. The optimization objective is to improve the recommendation performance of both domains simultaneously, i.e., dual-target cross-domain recommendation.

B. Overview of MA-DTGCF

To address the inaccurate transfer of the user representation and transfer each order user features by different strategies, this paper proposes a Multi-head Attentio Based Dual Target Graph Collaborative Filtering Network (MA-DTGCF). The structure of the model is shown in Fig. 1. Firstly, we use the embedding layer to generate the vector representation of users and items. Then, to learn more accurate user and item representations and transfer user features appropriately, we construct user-item heterogeneous graphs $\mathcal{G}_A = (\mathcal{V}_A, \mathcal{E}_A)$ and $\mathcal{G}_B = (\mathcal{V}_B, \mathcal{E}_B)$ for domain A and domain B, respectively, where $\mathcal{V}_A$ and $\mathcal{V}_B$ are the sets of nodes, $\mathcal{E}_A$ and $\mathcal{E}_B$ are the sets of edges. The sets of nodes include all users and items in domain A and B, respectively. If a user has a positive interaction with a item, there will be an edge between them. And, The bi-directional transfer graph convolution layer, which consists of a graph convolution layer and a bi-directional transfer layer, is used to aggregate the information on two heterogeneous graphs respectively and transfers the features of the common users from two domains in multiple representation subspaces. To get the high-order features of users and items and achieve adaptive transfer of each order user features, the bi-directional transfer graph convolution layer will be stacked multiple times. Finally, we use a output layer based on multi-layer neural networks to predict the the probability that the given user-item pair is a positive interaction in the two domains. The details of our model will be presented in subsequent subsections.

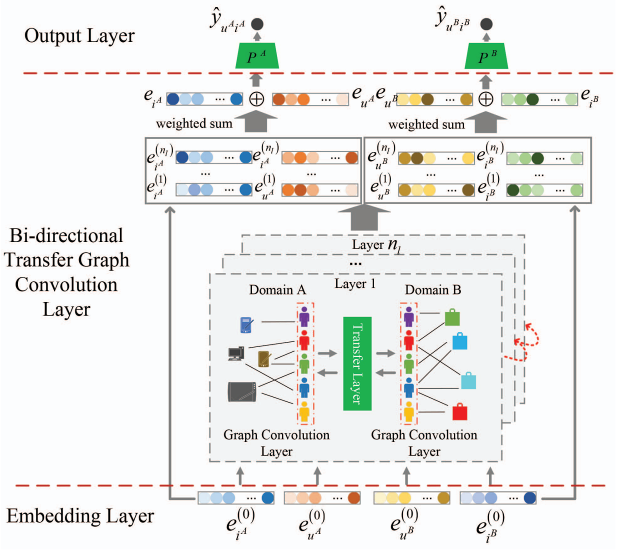

Fig. 1. The structure of MA-DTGCF

C. Embedding Layer

The computation of the model requires a vectorized representation of users and items. We design the embedding layer to map the user ID and item ID into embedding vectors $e_{u^{dom}} \in \mathbb{R}^d$ and $e_{i^{dom}} \in \mathbb{R}^d$, where $dom = \{A, B\}$ is the domain identity, and d is the dimension of the embedding vector. The formulations of the embedding layer are as follows,

$$\begin{aligned} e^{(0)}_{u^{dom}} &= \left(\mathbf{W}^{dom}_U\right)^{\mathrm{T}} \mathrm{ID}^{dom}_u \\ e^{(0)}_{i^{dom}} &= \left(\mathbf{W}^{dom}_I\right)^{\mathrm{T}} \mathrm{ID}^{dom}_i \end{aligned} \tag{1}$$

where $\mathbf{W}^{dom}_U$ and $\mathbf{W}^{dom}_I$ are weight matrices, and ID^{dom}_u and ID^{dom}_i denote the one-hot encoding of user u and item i of the domain dom, respectively. In the end, we can get the user and item embedding matrices $\mathbf{E}^{dom}_U \in \mathbb{R}^{m \times d}$ and $\mathbf{E}^{dom}_I \in \mathbb{R}^{n_{dom} \times d}$ in domain dom.

D. Graph Convolution Layer

The characteristics of users and items are often reflected in their related users and items. So, we use a graph convolution layer to mine the relationships between nodes within the domain. The following is an example of the matrix form calculation process of graph convolution layer in domain A. Let the initial node embedding matrix in graph $\mathcal{G}_A$ be $\mathbf{E}_A^{(0)} = \left(\mathbf{E}_U^A \oplus \mathbf{E}_I^A\right) \in \mathbb{R}^{(m+n_A)\times d}$, and the adjacency matrix of domain A is represented as follows,

$$\mathbf{M}^A = \begin{pmatrix} \mathbf{0} & \left(\mathbf{Y}^A\right) \\ \left(\mathbf{Y}^A\right)^T & \mathbf{0} \end{pmatrix} \tag{2}$$

where $\mathbf{0}$ is a matrix with all zero elements, and $\mathbf{M}^A \in \mathbb{R}^{(m+n_A)\times(m+n_A)}$ is a symmetric matrix. The degree matrix of domain A is $\mathbf{T}^A \in \mathbb{R}^{(m+n_A)\times(m+n_A)}$, which is a diagonal matrix, where $\mathbf{T}_{jj}^A = \sum_i \mathbf{M}_{ij}^A$. The node embedding representation of layer l be calculated as follows.

$$\mathbf{E}_A^{(l)} = \left(\left(\mathbf{T}^A\right)^{-\frac{1}{2}}\mathbf{M}^A\left(\mathbf{T}^A\right)^{-\frac{1}{2}}\right)\mathbf{E}_A^{(l-1)} \tag{3}$$

After the $l-th$ layer of convolution, the user embedding matrix is $\left(\mathbf{E}_U^A\right)^{(l)} = \mathbf{E}_A^{(l)}[0:m]$, i.e., the first m rows of $\mathbf{E}_A^{(l)}$. Similarly, The embedding matrix of the items is $\left(\mathbf{E}_I^A\right)^{(l)} = \mathbf{E}_A^{(l)}[m:m+n_A]$.

Then, we will input the users embedding to the $l-th$ bi-directional transfer layer. The transfer process of user features is as follows,

$$[(\tilde{\mathbf{E}}_U^A)^{(l)}, (\tilde{\mathbf{E}}_U^B)^{(l)}] = \text{Transfer}^{(l)}((\mathbf{E}_U^A)^{(l)}, (\mathbf{E}_U^B)^{(l)}) \tag{4}$$

where Transfer$^{(l)}(\cdot)$ is the multi-head attention based bi-directional transfer layer of the l-th bi-directional transfer graph convolution layer, which will be described in detail in Section E. After transfer, the user embedding matrix and item embedding matrix in the same domain are reassembled into the node embedding matrix $\mathbf{E}_A^{(l)} = [(\tilde{\mathbf{E}}_U^A)^{(l)} \oplus (\tilde{\mathbf{E}}_I^A)^{(l)}] \in \mathbb{R}^{(m+n_A)\times d}$, which will be used in the next bi-directional transfer graph convolution layer. After n_l layers of convolution, the final user and item embeddings are calculated as follows,

$$\begin{aligned} &\mathbf{E}_A = \beta_0\mathbf{E}_A^{(0)} + \beta_1\mathbf{E}_A^{(1)} + \beta_2\mathbf{E}_A^{(2)} + \ldots + \beta_{n_l}\mathbf{E}_A^{(n_l)} \\ &\mathbf{E}_U^A = \mathbf{E}_A[0:m] \\ &\mathbf{E}_I^A = \mathbf{E}_A[m:m+n_A] \end{aligned} \tag{5}$$

where $\beta_0 = \beta_1 = ... = \beta_{n_l} = \frac{1}{n_l+1}$ is the weight of each layer, $\mathbf{E}_A^{(l)}$ is the output of $l-th$ bi-directional transfer graph convolution layer.

E. Bi-directional Transfer Layer

The limited representation capability of single feature [11] leads to inaccurate calculation of user feature transfer weights or attention mechanisms, which may hurt the recommendation performance. Moreover, as the order of the interaction increases, the user features are more accurate, the strategy of transfer should be different. Therefore, this paper designs a bi-directional transfer module based on the multi-head attention mechanism, which uses multiple representation subspaces to represent the user in multiple perspectives and performs attention calculation in each of the multiple subspaces to achieve fine-grained and more accurate transfer of user features.

1) Transfer in Single Representation Space: The multi-head attention based transfer layer consists of multiple self-attention based transfer units, and each unit transfer the feature in its feature space. In the following, a single self-attention based transfer unit is introduced. Let the features of user u in layer l in domain A and domain B are $e_{u^A}^{(l)} \in \mathbb{R}^d$ and $e_{u^B}^{(l)} \in \mathbb{R}^d$, respectively, and then we concatenate them as the feature matrix $\mathbf{E}_u^{(l)} \in \mathbb{R}^{2\times d}$ of user u. The calculation process of a single transfer unit is as follows,

$$\begin{aligned} \tilde{\mathbf{E}}_u^{(l)} &= \textit{Attention}\left(\mathbf{Q}_u^{(l)}, \mathbf{K}_u^{(l)}, \mathbf{V}_u^{(l)}\right) \\ &= \operatorname{softmax}\left(\frac{\mathbf{Q}_u^{(l)}\left(\mathbf{K}_u^{(l)}\right)^T}{\sqrt{d_{att}}}\right)\mathbf{V}_u^{(l)} \\ \mathbf{Q}_u^{(l)} &= \mathbf{E}_u^{(l)}\left(\mathbf{W}^Q\right)^{(l)} \\ \mathbf{K}_u^{(l)} &= \mathbf{E}_u^{(l)}\left(\mathbf{W}^K\right)^{(l)} \\ \mathbf{V}_u^{(l)} &= \mathbf{E}_u^{(l)}\left(\mathbf{W}^V\right)^{(l)} \end{aligned} \tag{6}$$

where $\tilde{\mathbf{E}}_u^{(l)}$ is the feature matrix of user u after transfer. $\mathbf{Q}_u^{(l)}, \mathbf{K}_u^{(l)}, \mathbf{V}_u^{(l)} \in \mathbb{R}^{2\times d_{att}}$ are the query, key and value matrices of user u at the $l-th$ layer, respectively. $\left(\mathbf{W}^Q\right)^{(l)}, \left(\mathbf{W}^K\right)^{(l)}, \left(\mathbf{W}^V\right)^{(l)} \in \mathbb{R}^{d\times d_{att}}$ are the weight matrices of the single transfer unit at layer l. $d_{att} = d$ is the feature dimension of user u after mapping, and $\sqrt{d_{att}}$ is the normalization factor to avoid gradient vanishing.

2) Transfer in Multiple Representation Subspaces: Feature transfer in a single feature space is limited and coarse-grained [11]. However, using multiple self-attention baed transfer units to transfer user features in different feature spaces can be better adapted to this complex scenario. For the feature matrix $\mathbf{E}_u^{(l)} \in \mathbb{R}^{2\times d}$ of user u, the multi-head attention transfer equation is as follows,

$$\begin{aligned} \tilde{\mathbf{E}}_u^{(l)} &= \operatorname{MultiHead}(\mathbf{E}_u^{(l)}) \\ &= \operatorname{Concat}\left(\operatorname{head}_1^{(l)}, \ldots, \operatorname{head}_{n_h}^{(l)}\right)\left(\mathbf{W}^O\right)^{(l)} \\ \operatorname{head}_i^{(l)} &= \operatorname{Attention}\begin{pmatrix} \mathbf{Q}_u^{(l)}\left(\mathbf{W}^Q\right)_i^{(l)}, \\ \mathbf{K}_u^{(l)}\left(\mathbf{W}^K\right)_i^{(l)}, \\ \mathbf{V}_u^{(l)}\left(\mathbf{W}^V\right)_i^{(l)} \end{pmatrix} \end{aligned} \tag{7}$$

where $\left(\mathbf{W}^O\right)^{(l)} \in \mathbb{R}^{n_h d_{att}\times d}$ is the weight matrix, and we set $d_{att} = d/n_h$. n_h is the number of feature spaces and is even. This setting ensures that the computational loss of multi-head attention based transfer is comparable to that of a single tarnsfer unit. After completing the transfer, the updated user features can be obtained as follows,

$$\begin{aligned} \tilde{e}_{u^A}^{(l)} &= \tilde{\mathbf{E}}_u^{(l)}[0] \\ \tilde{e}_{u^B}^{(l)} &= \tilde{\mathbf{E}}_u^{(l)}[1] \end{aligned} \tag{8}$$

where $\tilde{\mathbf{E}}_u^{(l)}[0]$ and $\tilde{\mathbf{E}}_u^{(l)}[1]$ represent the 0th and 1st rows of the updated user feature matrix. By using Eqs. (5-8) to transfer the features of each user in the user embedding representation matrices $\left(\mathbf{E}_U^A\right)^{(l)}$ and $\left(\mathbf{E}_U^B\right)^{(l)}$ at layer l, i.e., the $\text{Transfer}^{(l)}(\cdot)$ in Eq.(4), we can get the updated user feature matrices $(\tilde{\mathbf{E}}_U^A)^{(l)}$ and $(\tilde{\mathbf{E}}_U^B)^{(l)}$, which will be used in the information aggregation of the next bi-directional transfer graph convolution layer.

F. Output Layer

After multiple bi-directional transfer graph convolution layers, we obtain the final embedding of users and items. Given that the traditional inner product prediction of probability is linear and its fitting ability is limited, this paper will use neural networks for probability prediction. Then we will introduce the probability prediction module in domain A. The final embeddings of user u^A and item i^A are represented as e_{u^A} and e_{i^A}. We concatenate them into a vector $e_{u^A i^A} \in \mathbb{R}^{2d}$ as the input of the neural network, and a pyramid shape neural network [19] is used to predict the probability. The computation process of probability predictor P^A is as follows,

$$
\begin{aligned}
f^{(0)} &= e_{u^A i^A} \\
f^{(1)} &= relu\left(f^{(0)}\mathbf{W}_{p^A}^{(1)} + b_{p^A}^{(1)}\right) \\
f^{(2)} &= relu\left(f^{(1)}\mathbf{W}_{p^A}^{(2)} + b_{p^A}^{(2)}\right) \\
&\cdots \\
f^{(n_p)} &= sigmoid\left(f^{(n_p-1)}\mathbf{W}_{p^A}^{(n_p)} + b_{p^A}^{(n_p)}\right) \\
\hat{y}_{u^A i^A} &= f^{(n_p)}
\end{aligned}
\tag{9}
$$

where $\mathbf{W}_{p^A}$ and b_{p^A} are the weights and biases, and *relu*, *sigmoid* are activation functions. The probability predictor P^B of the domain B can be obtained, similarly.

G. Model Training

The performance of a deep learning model usually depends on the quality of the loss function. A good loss function can avoid the model from falling into a local optimum and accelerate convergence. In this paper, we use cross-entropy loss as the loss function for probability prediction,

$$
\begin{aligned}
&L_{join} = L_y^A\left(\hat{y}_{u^A i^A}, y_{u^A i^A}\right) + L_y^B\left(\hat{y}_{u^B i^B}, y_{u^B i^B}\right) + \lambda\|\theta\|^2 \\
&L_y\left(\hat{y}_{ui}, y_{ui}\right) = -\sum_{(u,i)\in D} y_{ui}\log\hat{y}_{ui} + (1-y_{ui})\log\left(1-\hat{y}_{ui}\right)
\end{aligned}
\tag{10}
$$

where $D \in Y^+ \cup Y^-$ is the set of training samples, Y^+ and Y^- are the set of positive and negative samples in a domain, λ is the coefficient of the regularization, and $\theta = \left\{\mathbf{E}_A^{(0)}, \mathbf{E}_B^{(0)}, \mathbf{W}_{Transfer}, \mathbf{W}_{P^A}, \mathbf{W}_{P^B}, b_{P^A}, b_{P^B}\right\}$ is the set of parameters, where $\mathbf{W}_{Transfer}$ is the weights of bi-directional transfer layer. Adam [20] is selected in this paper to optimize the parameters of the model, which is a stochastic gradient descent based optimizer that allows the model to converge quickly at a large learning rate. The MA-DTGCF algorithm is shown in Algorithm 1. Lines 4-14 is the process of bi-directional transfer graph convolution layer, and output layer is denoted in lines 15-16.

Algorithm 1 MA-DTGCF

```
Input: The user-item heterogeneous graphs $\mathcal{G}_A = (\mathcal{V}_A, \mathcal{E}_A)$ and $\mathcal{G}_B = (\mathcal{V}_B, \mathcal{E}_B)$, initial node embedding matrices $\mathbf{E}_A^{(0)}$ and $\mathbf{E}_B^{(0)}$, adjacency matrices $\mathbf{M}^A$ and $\mathbf{M}^B$, degree matrices $\mathbf{T}^A$ and $\mathbf{T}^B$, the positive sample set $\mathbf{Y}_A^+$ and negative sample set $\mathbf{Y}_A^-$, the positive sample set $\mathbf{Y}_B^+$ and negative sample set $\mathbf{Y}_B^-$, the number of graph convolution layers $n_l$.
Output: $\theta = \left\{\mathbf{E}_A^{(0)}, \mathbf{E}_B^{(0)}, \mathbf{W}_{Transfer}, \mathbf{W}_{PA}, \mathbf{W}_{PB}, b_{PA}, b_{PB}\right\}$
 1: repeat
 2:     for each $\left(u^A, i^A, y_{u^A i^A}\right) \in \mathbf{Y}_A^+ \cup \mathbf{Y}_A^-$ do
 3:         Randomly select a sample $\left(u^B, j^B, y_{u^B j^B}\right) \in \mathbf{Y}_B^+ \cup \mathbf{Y}_B^-$
 4:         for $l = 1 : n_l$ do
 5:             $\mathbf{E}_A^{(l)} = \left(\left(\mathbf{T}^A\right)^{-\frac{1}{2}}\mathbf{M}^A\left(\mathbf{T}^A\right)^{-\frac{1}{2}}\right)\mathbf{E}_A^{(l-1)}$
 6:             $\mathbf{E}_B^{(l)} = \left(\left(\mathbf{T}^B\right)^{-\frac{1}{2}}\mathbf{M}^B\left(\mathbf{T}^B\right)^{-\frac{1}{2}}\right)\mathbf{E}_B^{(l-1)}$
 7:             $\left(\mathbf{E}_U^A\right)^{(l)} = \mathbf{E}_A^{(l)}[0:m]$, $\left(\mathbf{E}_U^B\right)^{(l)} = \mathbf{E}_B^{(l)}[0:m]$
 8:             Get $\left(\tilde{\mathbf{E}}_U^A\right)^{(l)}$ and $\left(\tilde{\mathbf{E}}_U^B\right)^{(l)}$ by Eq. (4).
 9:             $\mathbf{E}_A^{(l)}[0:m] = \left(\tilde{\mathbf{E}}_U^A\right)^{(l)}$, $\mathbf{E}_B^{(l)}[0:m] = \left(\tilde{\mathbf{E}}_U^B\right)^{(l)}$
10:         end for
11:         $\mathbf{E}_A = \beta_0\mathbf{E}_A^{(0)} + \beta_1\mathbf{E}_A^{(1)} + \beta_2\mathbf{E}_A^{(2)} + \ldots + \beta_{n_l}\mathbf{E}_A^{(n_l)}$
12:         $\mathbf{E}_B = \beta_0\mathbf{E}_B^{(0)} + \beta_1\mathbf{E}_B^{(1)} + \beta_2\mathbf{E}_B^{(2)} + \ldots + \beta_{n_l}\mathbf{E}_B^{(n_l)}$
13:         $\mathbf{E}_U^A = \mathbf{E}_A[0:m], \mathbf{E}_I^A = \mathbf{E}_A[m:m+n_A]$
14:         $\mathbf{E}_U^B = \mathbf{E}_B[0:m], \mathbf{E}_I^B = \mathbf{E}_B[m:m+n_B]$
15:         $\hat{y}_{u^A i^A} = P^A\left(e_{u^A}, e_{i^A}\right)$
16:         $\hat{y}_{u^B j^B} = P^B\left(e_{u^B}, e_{j^B}\right)$
17:         Calculate the loss $L_{join}$ by Eq. (10)
18:         Update parameter $\theta$
19:     end for
20: until Convergence
21: Return $\theta$
```

IV. Experiment

First, we introduce the basic setup of the experiment in Section A, and then Section B will show the results of comparing the model in this paper with the state-of-the-art methods. Section C analyzes the effectiveness of the transfer layer. Finally, Section D analyzes the impact of key parameters in the model on the recommendation performance.

A. Experimental Setup

1) Experiment Dataset: The experimental data was obtained from the real dataset Amazon (2018) [21], which includes items from 24 categories. The raw data of this dataset contains 233.1 million ratings and textual information such as user reviews and item descriptions. We extracted three category combinations for the experiments, which are Books & Movies and TV(MT), Cell Phones and Accessories(CPA) & Electronics, and Clothing, Shoes and Jewelry(CSJ) & Sports_and_Outdoors(SO). We will use the rating data corresponding to these three data sets for the experiments in this paper. First, we convert the explicit rating data into implicit feedback data, i.e., if a user has a rating for a item, its corresponding label is 1. Then, we removed users with less than 5 ratings and items with less than 10 ratings in each category. Finally, the data related to the common users of each combination were extracted as the final experimental data. Detailed statistics about the data are shown in Table I.

TABLE I
STATISTICAL INFORMATION OF THE DATASET

Dataset	#Users	#Items	#Interactions	Density
Books	11524	34485	145155	0.037%
MT	11524	33493	311540	0.081%
Electronics	38127	43460	442366	0.027%
CPA	38127	20467	310234	0.040%
SO	14020	37893	166480	0.031%
CSJ	14020	13170	109309	0.059%

2) Evaluation: We use the leave-one-out method, which is widely used in recent work, to valid and test our model. The training, validation, and test data are partitioned in the same way as in Ref. [5]. Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) [22] will be used to evaluate the rank performance of the model. For a user, HR is used to measure the proportion of positive samples appearing in the recommendation list among all positive samples, which is either 0 or 1 since a user has only one positive sample, and NDCG is used to measure the quality of the ranking. We will calculate the HR and NDCG of all users separately during validation or testing, and take the average value as the final result.

3) Compared Methods: We compare MA-DTGCF with seven models, including two single-domain recommendation models, two single-target cross-domain recommendation models, two dual-target cross-domain recommendation models. The details of these models are as follows.

- **NeuMF** [5] is a single-domain recommendation model that uses generalized matrix decomposition and a multi-layer perceptron to learn the complex interactions between the user and item features.
- **LightGCN** [1] is a single-domain graph collaborative filtering model that removes the activation function and the transformation matrices from the original GCN [15] model.
- **CMF** [7] is a single-target cross-domain recommendation model that transfers information by learning shared user features through a collaborative factorization of the rating matrices in both domains.
- **CoNet** [8] is a single-target cross-domain recommendation model that transfers information by cross-connecting the layers of neural networks and uses joint learning to optimize the model.
- **GA-DTCDR** [13] is a dual-target cross-domain model that uses Node2Vec [23] to learn the embeddings of users and items, and uses element-wise attention [24] to achieve bi-directional transfer of user features.
- **BITGCF** [10] is a dual-target cross-domain model, and the intra-domain and inter-domain information aggregation models are designed to achieve bi-directional information transfer.

4) Parameters Setting: The random initialization of the hidden features in all compared methods obeys Gaussian $\mathcal{N}(0, 0.01)$, and the learning rate is set to 0.01 except for CMF which is set to 0.1. All of the key parameters of the compared methods are tuned by cross-validation. The number of layers of NCF in NeuMF is set to 3, and the dimensions of hidden features in each layer are 64→32→16. For LightGCN, we use the code provided by the authors for our experiments but modified its loss function to cross-entropy, and we set the number of graph convolution layers as 3, the message dropout as 0, the hidden feature dimension as 64, the regularization coefficient λ as 0.00001. CoNet is set according to the optimal parameters in the original paper, and the dimensions of each layer are set to 64→32→16→8. GA-DTCDR uses the review information of users and items in the Amazon dataset as text features, and the hidden feature dimension is set to 64. The number of convolutional layers in BITGCF is set to 3, and the coefficients $\alpha_A = 0.4$ and $\alpha_B = 0.7$. For MA-DTGCF, we tune the number of convolution layers in the range of [1, 2, 3, 4, 5], the number of attention heads in the range of [1, 2, 4, 6, 8], the regularization coefficient in the range of [0.000001, 0.00001, 0.0001, 0.001, 0.01] and the dimensionality of features in the range of [16, 32, 64, 128]. The number of layers of the probability predictor is set to 3, and if the dimensionality of the features is 64, the dimensions of each layer are 128→64→1. The optimal parameters were obtained by cross-validation, and the optimal parameters of three combinations are same, $d = 64, n_l = 3, n_h = 4, \lambda = 0.00001$. The convergence condition of the model is that the loss value decreases by less than 0.001 for 5 epochs.

B. Performance Comparison

For the single-domain recommendation model in the comparison methods, we will train the model separately within a single domain and show the test results for each domain separately. For the single-target cross-domain recommendation models, we train the model using data from both domains, but only show the test results for the target domain (the domain with relatively sparse data). For the dual-target cross-domain recommendation models, we train the model using data from both domains, test it in both domains, and present the results of both domains. From Table II, we have the following observations:

1) Among the single-domain methods, LightGCN performs significantly better than NeuMF and even outperforms the two single-target cross-domain recommendation algorithms CMF and CoNet, which shows the importance of mining the higher-order interactions between users and items to improve the recommendation performance.
2) In the single-target cross-domain recommendation model, the neural network-based CoNet performs better than CMF because the neural network model has a better fitting ability compared to matrix factorization.

TABLE II
THE EXPERIMENTAL RESULTS (HR@10 & NDCG@10)

Dataset	Metrices	Single Domain		Single Target		Dual Target		Ours
		NeuMF	LightGCN	CMF	CoNet	GA-DTCDR	BITGCF	MA-DTGCF
Books	HR	0.3016	0.4383	0.3258	0.4213	0.4416	0.4456	0.4747
	NDCG	0.2332	0.3204	0.2556	0.2924	0.3288	0.3314	0.3601
MT	HR	0.5049	0.5899	—	—	0.5936	0.5966	0.6325
	NDCG	0.2886	0.3725	—	—	0.3756	0.3776	0.4033
Electronics	HR	0.3681	0.4402	0.3869	0.4165	0.4465	0.4437	0.4781
	NDCG	0.2168	0.2694	0.2457	0.2543	0.2701	0.2646	0.2963
CPA	HR	0.4073	0.4546	—	—	0.4615	0.4597	0.4904
	NDCG	0.2396	0.2963	—	—	0.2983	0.2977	0.3329
SO	HR	0.3823	0.4406	0.3852	0.4275	0.4448	0.4453	0.4793
	NDCG	0.2265	0.2754	0.2589	0.2664	0.2801	0.2762	0.3065
CSJ	HR	0.3157	0.3706	—	—	0.3755	0.3788	0.4158
	NDCG	0.1893	0.2511	—	—	0.2537	0.2557	0.2881

3) The dual-target cross-domain models are all based on graph models, and their recommendation performance is better than all single-domain and single-target cross-domain recommendation models in both the auxiliary and target domains. The combination of graph representation learning and bi-directional information transfer is very important for the improvement of recommendation performance.
4) The recommendation performance of MA-DTGCF is significantly better than all the comparison models, which indicates that adaptive transfer of user features in multiple representation subspaces and adaptive transfer of each order user features play an important role in improving the recommendation performance.

C. Effectiveness of Transfer Layer

To prove the effectiveness of the proposed bi-directional transfer layer, we replace the transfer layer in MA-DTGCF with the transfer layer proposed in GA-DTCDR and BITGCF to get the models GA-DTGCF and BI-DTGCF. The parameters of the transfer layer are consistent with the GA-DTCDR and BITGCF, and the rest of the parameters are consistent with MA-DTGCF. Figure 2 shows the results on the test datasets under different transfer layers. From the above figure, it can be seen that the proposed multi-head attention based bi-directional transfer layer has obvious advantages in both HR and NDCG. This fully demonstrates that using multiple representation subspaces to represent the user in multiple perspectives, performing attention calculation in each of the multiple subspaces and learning different transfer strategies in different bi-directional transfer graph convolution layers can achieve fine-grained and more accurate transfer of user features.

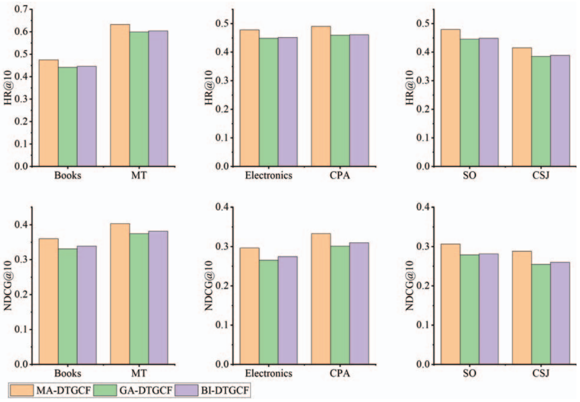

Fig. 2. Results of different transfer layers

D. Impact of Key Parameters

The key parameters in the model are the number of attention heads in the bi-directional transfer layer and the number of layers of the bi-directional transfer graph convolution layer. Fig. 3 shows the impact of the two key parameters on the recommended performance in test datasets, and the basic parameter settings for the two sets of experiments are the optimal parameter settings. As shown in Fig. 3(a) and 3(b), the recommendation performance of the model gradually improves as the attention head increases. This indicates that more accurate feature transfer can be achieved as the number of feature spaces increases, but the model performance decreases after the number of attention heads exceeds 4, which indicates that more feature space for feature transfer is not better. It can be seen from Fig. 3(c) and 3(d) that the recommendation performance increases with the increase of the number of layers of the bi-directional transfer graph convolution layer, however, the performance decreases after the number of layers exceeds 3. It can be seen that the higher the interaction order is not better.

V. CONCLUSION

In this paper, the **M**ulti-head **A**ttention Based **D**ual **T**arget **G**raph **C**ollaborative **F**iltering Network (MA-DTGCF) is proposed. We design a multi-head attention based bi-directional transfer graph convolution layer. Transferring

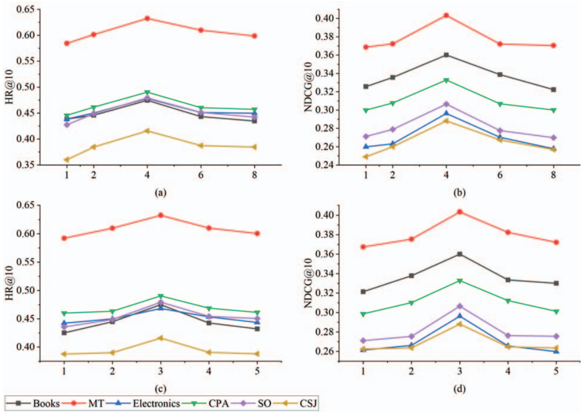

Fig. 3. Impact of the key parameters

user features in multiple representation subspaces and learning different transfer strategies in different bi-directional transfer graph convolution layers can achieve fine-grained and more accurate transfer of user features. Ultimately, MA-DTGCF shows significant improvement compared to the state-of-the-art methods on several real data sets, which fully proves the effectiveness of MA-DTGCF. But, in this work, we ignore the enhancement of the item features in the dual-target cross-domain recommendation scenario, which will be our research priority in the future.

Acknowledgment

This work is jointly sponsored by National Natural Science Foundation of China (Nos. 62172249, 61973180), Natural Science Foundation of Shandong Province (Nos. ZR2019MF014, ZR2021MF092).

References

[1] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "Lightgcn: Simplifying and powering graph convolution network for recommendation," in *the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020, pp. 639–648.

[2] F. Zhu, Y. Wang, C. Chen, G. Liu, and X. Zheng, "A graphical and attentional framework for dual-target cross-domain recommendation," in *the 29th International Joint Conference on Artificial Intelligence and the 17th Pacific Rim International Conference on Artificial Intelligence*, 2020, pp. 3001–3008.

[3] X. Yu, Q. Peng, L. Xu, F. Jiang, J. Du, and D. Gong, "A selective ensemble learning based two-sided cross-domain collaborative filtering algorithm," *Information Processing and Management*, vol. 58, no. 6, p. 102691, 2021.

[4] M. Fu, H. Qu, Z. Yi, L. Lu, and Y. Liu, "A novel deep learning-based collaborative filtering model for recommendation system," *IEEE Transactions on Cybernetics*, vol. 49, no. 3, pp. 1084–1096, 2018.

[5] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *the 26th International Conference on World Wide Web*, 2017, pp. 173–182.

[6] S. Berkovsky, T. Kuflik, and F. Ricci, "Cross-domain mediation in collaborative filtering," in *the International Conference on User Modeling*, 2007, pp. 355–359.

[7] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008, pp. 650–658.

[8] G. Hu, Y. Zhang, and Q. Yang, "Conet: Collaborative cross networks for cross-domain recommendation," in *the 27th ACM International Conference on Information and knowledge Management*, 2018, pp. 667–676.

[9] C. Zhao, C. Li, and C. Fu, "Cross-domain recommendation via preference propagation graphnet," in *the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 2165–2168.

[10] M. Liu, J. Li, G. Li, and P. Pan, "Cross domain recommendation via bi-directional transfer graph collaborative filtering networks," in *the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 885–894.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 6000–6010.

[12] X. Yu, D. Zhan, L. Liu, H. Lv, L. Xu, and J. Du, "A privacy-preserving cross-domain healthcare wearables recommendation algorithm based on domain-dependent and domain-independent feature fusion," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 5, pp. 1928–1936, 2022.

[13] F. Zhu, C. Chen, Y. Wang, G. Liu, and X. Zheng, "Dtcdr: A framework for dual-target cross-domain recommendation," in *the 28th ACM International Conference on Information and Knowledge Management*, 2019, pp. 1533–1542.

[14] M. Gori and A. Pucci, "Itemrank: A random-walk based scoring algorithm for recommender engines," in *the 20th International Joint Conference on Artifical Intelligence*, 2007, pp. 2766–2771.

[15] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *the 5th International Conference on Learning Representations*, 2016. [Online]. Available: https://arxiv.org/abs/1609.02907

[16] R. V. D. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," in *the KDD Workshop on Deep Learning Day*, 2018. [Online]. Available: http://arxiv.org/abs/1706.02263

[17] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2018, pp. 974–983.

[18] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*, 2019, pp. 165–174.

[19] F. Yuan, L. Yao, and B. Benatallah, "Darec: Deep domain adaptation for cross-domain recommendation via transferring rating patterns," in *the 28th International Joint Conference on Artificial Intelligence*, 2019, pp. 4227–4233.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *the 3rd International Conference on Learning Representations*, 2015.

[21] J. Ni, J. Li, and J. J. McAuley, "Justifying recommendations using distantly-labeled reviews and fine-grained aspects," in *the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019, pp. 188–197.

[22] X. He, T. Chen, M.-Y. Kan, and X. Chen, "Trirank: Review-aware explainable recommendation by modeling aspects," in *the 24th ACM International on Conference on Information and Knowledge Management*, 2015, pp. 1661–1670.

[23] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016, pp. 855–864.

[24] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *the 3rd International Conference on Learning Representations*, 2014. [Online]. Available: http://arxiv.org/abs/1409.0473