Asymmetrical Context-aware Modulation for Collaborative Filtering Recommendation

Yi Ouyang
Shanghai Jiao Tong University
Shanghai, China
halaoyy@sjtu.edu.cn

Peng Wu*
School of Electronic Information and Electrical Engineering & Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai Jiao Tong University
Shanghai, China
catking@sjtu.edu.cn

Li Pan
School of Electronic Information and Electrical Engineering & Shanghai Key Laboratory of Integrated Administration Technologies for Information Security, Shanghai Jiao Tong University
Shanghai, China
panli@sjtu.edu.cn

ABSTRACT

Modern learnable collaborative filtering recommendation models generate user and item representations by deep learning methods (e.g. graph neural networks) for modeling user-item interactions. However, most of them may still have unsatisfied performances due to two issues. Firstly, some models assume that the representations of users or items are fixed when modeling interactions with different objects. However, a user may have different interests in different items, and an item may also have different attractions to different users. Thus the representations of users and items should depend on their contexts to some extent. Secondly, existing models learn representations for user and item by symmetrical dual methods which have identical or similar operations. Symmetrical methods may fail to sufficiently and reasonably extract the features of user and item as their interaction data have diverse semantic properties. To address the above issues, a novel model called Asymmetrical context-awaRe modulation for collaBorative filtering REcommendation (ARBRE) is proposed. It adopts simplified GNNs on collaborative graphs to capture homogeneous user preferences and item attributes, then designs two asymmetrical context-aware modulation models to learn dynamic user interests and item attractions, respectively. The learned representations from user domain and item domain are input pair-wisely into 4 Multi-Layer Perceptrons in different combinations to model user-item interactions. Experimental results on three real-world datasets demonstrate the superiority of ARBRE over various state-of-the-arts.

CCS CONCEPTS

• **Information systems → Recommender systems**.

*Corresponding author.

CIKM '22, October 17–21, 2022, Atlanta, GA, USA

ACM ISBN 978-1-4503-9236-5/22/10…$15.00
https://doi.org/10.1145/3511808.3557240

KEYWORDS

recommender systems, graph neural networks, context-aware modulation

ACM Reference Format:
Yi Ouyang, Peng Wu, and Li Pan. 2022. Asymmetrical Context-aware Modulation for Collaborative Filtering Recommendation. In *Proceedings of the 31st ACM International Conference on Information and Knowledge Management (CIKM '22), October 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3511808.3557240

1 INTRODUCTION

Information overload in Internet brings great challenges for recommendation systems to mine personalized information of users for filtering and suggesting items of potential interests to them [6, 11, 19]. Many recently proposed recommendation methods model user preferences and item attributes by exploiting multi-source information other than user-item historical interactions (e.g., rating, clicking), such as social relations [7, 35] and review information [18, 39], and have achieved competitive performances. However, the application of these methods maybe restricted in the cases where none of the additional multi-source information is available. The most widely used technique for recommendation only based on user-item historical interactions is collaborative filtering (CF) [15, 24]. The core idea of CF is to assume that users with similar historical interactions would have similar preferences on items. In order to effectively model users and items, the learnable CF models embed users and items into vectorized representations in latent space where their similarity and relations can be computed [12, 16, 31, 33]. Early shallow CF models like matrix factorization (MF) [16] and BPR [25] directly linearly factorize the interaction matrix as latent vectors of users and items and models their potential interaction with inner product. The complex non-linear relations between users and items are hard to capture by shallow models. With the development of deep learning (DL), many DL-based CF models have been proposed for their effectiveness in assimilating collaborative signals [11, 33] and modeling complex nonlinear interactions [12, 27].

The user-item historical interactions can be naturally expressed as graphs. Recently, graph neural networks (GNNs) have shown effective performance in extracting high-order nonlinear features from complex graph structures. Hence, many GNN-based recommendation models have been proposed [2, 11, 33, 34, 37]. Some works apply GNNs on the bipartite graph where nodes represent

users and items and edges represent their historical interactions [2, 11, 33] to directly model nonlinearly relations between users and items. Some other works apply GNNs on the collaborative graphs, where nodes represent users or items and the edges represent their collaborative similarity computed based on their neighborhood overlap in the bipartite graph, to explicitly model the homogeneous user preferences or item attributes [21, 37]. In order to combine the advantages of both types of graphs, some works integrate the features learned from bipartite graphs and collaborative graphs together for recommendation [36]. Despite the effectiveness of the above methods, following [7, 35], we argue that their performances are still hindered as the learned representation of a user (resp. item) is static for diverse items (resp. users). Intuitively, the representations of user and item should be different when facing different contexts, i.e., predicting different user-item interactions. For instance, the upper part of Figure 1 illustrates the user's interests towards clothes and computers. If the candidate item is a jacket, we should pay more attention on what kind of clothes the user would like by embedding her interest in clothes more into her representation, while her interest in computers is more likely to be noisy information. Also, the lower part of Figure 1 illustrates that a Major League Baseball (MLB) varsity jacket can attract baseball fans and fashionistas. If the target user is a fashionista who has no interest in baseball, the representation should embed its attraction to fashionistas more, while its attraction to baseball fans should be restrained. In other words, the representation of the user (resp. item) should embed more historical interaction information related to the candidate item (resp. target user) than those unrelated interaction information, which is neglected by most of the existing works.

Figure 1: Illustration of context-aware user interest and item attraction. The upper part is a user's clicking history, which can be divided into two categories: clothes and computers; the lower part is a MLB varsity jacket's clicked users, which composed of baseball fans and fashionistas.

The symmetrical structure of user-item interaction data has inspired many works to design dual deep learning models to learn both user and item representations, respectively [5, 7, 9, 18, 35]. The dual models of user and item have symmetrical structures with identical or similar deep learning operations. For example, DANSER [35] develops a dual-graph attention network to model the two-fold social effects for social recommendation. DICER [7] extracts deep context-aware features of users and items with dual side modulation. However, although the user and item have symmetrical interaction data structure, their data have different semantic features for recommendation. For example, let us consider a user who has clicked several laptops, mice and keyboards. Apparently, these clicked items have substitutive relations (e.g. laptop-laptop) and complementary relations (e.g. computer-mouse-keyboard), i.e., there are strong interdependencies among the user's clicked items. However, the historical clicking users of a certain item usually do not have such explicit interdependencies among them. Thus learning representations of users and items from historical interaction data with symmetrical dual models may not be sufficient and reasonable to extract their unique semantic features.

To alleviate the aforementioned limitations, we propose a novel model called ARBRE, which learns user preferences and item attributes with GNNs on collaborative graphs, then designs two asymmetrical context-aware modulation models to learn dynamic user interests and items attractions, respectively. More specifically, the collaborative graphs of users and items are first constructed by evaluating their collaborative similarity on bipartite graph. The homogeneous user preferences and item attributes are learned by the simplified GNNs on the collaborative graphs. The preference and attribute capture intrinsic homogeneous features of users and items within their neighborhood, respectively, which stay unchanged and independent of external contexts. Besides static preference and attribute, the user interests and item attractions should be further dynamically modulated according to their contexts, as discussed above as well as in [7, 35]. However, different from [7, 35], we design two asymmetrical context-aware modulation models for dynamic interest and attraction learning, as interaction data of user and item have different semantic features. In item domain, since the historical clicking users of an item usually have little interdependencies, the item attraction to a target user is calculated by adopting a simple product and pooling operation to modulate its historical users with the target user. In user domain, as discussed above that the user's clicked items usually have some interdependencies. Thus a self-attention block is first adopted on the clicked items to capture their interdependencies, and then the user interests in the candidate item are calculated by an attention-based modulation. The preference and interest of the user and the attribute and attraction of the item are then input pair-wisely into 4 Muti-Layer Perceptrons (MLPs) in different combinations to model their interactions. Experimental studies valid the benefits of ARBRE of capturing accurate features under different contexts. Our contributions can be summarized as:

- To the best of our knowledge, we are the first to highlight that the symmetrical historical interaction data structures of users and items have different semantic features for recommendation, and thus need to adopt different targeted models to learn their representations.

- We propose ARBRE, which learns homogeneous user preferences and item attributes by GNNs on collaborative graphs, and then designs two asymmetrical context-aware modulation models to learn dynamic user interests and item attractions, respectively.
- Experiments on three real-world benchmark datasets are conducted to demonstrate that ARBRE consistently outperforms the several state-of-the-arts.

2 RELATED WORKS

In this section, we briefly review two kinds of existing recommendation methods that are most relevant to our work: 1) Graph neural network based recommendation, 2) Context-aware modulation based recommendation.

Graph Neural Network based Recommendation. In recommendation tasks, user-item interactions can naturally form a bipartite graph where edges represent interactions between users and items. Since graph neural networks (GNNs) have shown great potential in learning graph structure features [2, 11, 20, 35], recent models leverage them to learn features of users and items from interaction graphs. For example, NGCF [20] models the high-order connectivity in user-item bipartite graph. LightGCN [11] further simplifies the GCN design for recommendation by removing feature transformation and nonlinear activation. With the development of the attention mechanism, graph attention network (GAT) [29] which learns the weights of nodes attentively for aggregating neighborhood features is also adoted in recommendation models, such as KGAT [32] and DGSR [38]. To better model the homogeneous collaborative signals among users and items, other works evaluate the behavioral similarities of users and items and construct homogeneous collaborative graphs from bipartite graph for recommendation [21, 37]. Some works integrate the features learned from bipartite graphs and collaborative graphs together to combine their advantages [35, 36]. However, most of the existing GNN-based recommendation methods learn static features of users and items, and ignore the dynamic user interest and item attraction when facing different contexts.

Context-aware Modulation based Recommendation. Incorporating the contextualized information in the recommendation process help to improve accuracy in learning features of users and items [30]. In reality, user interest and item attraction are multifaceted thus dynamic according to different contexts. DIN [40] adaptively learns the representation of user interests from historical behaviors with respect to candidate items in click-through rate prediction. CoSAN [20] proposes a collaborative item representation learning method which is capable to dynamically generate different representations for an item when encountering different users for session recommendation. Considering the symmetry of user-item interaction data, some models dynamically learn both user and item representations concerning different contexts through a symmetrical dual structure. DANSER [35] designes a dual graph attention network in which social and item-to-item influences are modeled with symmetrical modulation model under specific contexts. DICER [7] also adopts a symmetrical model to learn features of user and item with deep contexts. Both of [7, 35] utilized product and pooling based models to modulate the interaction features of both users and items. However, the semantic information in user domain and item domain are different, and learning representations of users and items with symmetrical dual models may not be sufficient and reasonable to extract their unique semantic features. Different from [7, 35], the proposed model devises two asymmetrical modulation models in which user interest and item attraction are exploited and modulated by different methods according to their semantic properties, respectively. Besides asymmetrical modulation structure, our model adopts a simpler graph neural network than that in [7, 35] which removes the feature transformation and nonlinear activation for enhancing the feature aggregation ability. Moreover, in order to fully explore the complex relations between users and items, the learned user preference and interest and item attribute and attraction are input pair-wisely into 4 MLPs in prediction layer where the relation between modulated user interest and item attraction is considered yet neglected by [7].

Table 1: Summary of notations

Symbols	Definitions and Descriptions
$\mathcal{U}$	set of users
$\mathcal{I}$	set of items
$\mathbf{R}$	the user-item interaction matrix
$\mathcal{R}_{\mathrm{I}}(u)$	the interacted item set of user u
$\mathcal{R}_{\mathrm{U}}(i)$	the interacting user set of item i
$\mathbf{U}$	the initial raw feature matrix of users
$\mathbf{I}$	the initial raw feature matrix of items
$\mathcal{G}_{\mathrm{U}}$	user collaborative graph
$\mathcal{G}_{\mathrm{I}}$	item collaborative graph
$\mathcal{N}_{\mathrm{U}}(u)$	the neighbors set of user u in $\mathcal{G}_{\mathrm{U}}$
$\mathcal{N}_{\mathrm{I}}(i)$	the neighbors set of item i in $\mathcal{G}_{\mathrm{I}}$
$\boldsymbol{p}_u^*$	the user preference representation of user u
$\boldsymbol{m}_u^{i*}$	the user interest representation of user u
$\boldsymbol{n}_i^{u*}$	the item attraction representation of item i
$\boldsymbol{q}_i^*$	the item attribute representation of item i
d	the number of embedding dimension
$\otimes$	the element-wise product operation

3 PRELIMINARIES

Let $\mathcal{U}$ and $\mathcal{I}$ be the sets of users and items respectively, u, v index users and i, j index items. The initial raw feature matrices of users and items are denoted as $\mathbf{U} \in \mathbb{R}^{l_{\mathrm{U}} \times M}$ and $\mathbf{I} \in \mathbb{R}^{l_{\mathrm{I}} \times N}$ respectively, where M, N denote the numbers of users and items, l_{U} and l_{I} are their raw feature dimensions, respectively. If no initial raw feature matrix is provided, one-hot feature matrix will be adopted. Let $\mathbf{R} = [r_{ui}] \in \mathbb{R}^{M \times N}$ be the user-item interaction matrix which consists of 0 and 1, where $r_{ui} = 1$ indicates user u has interacted with item i and $r_{ui} = 0$ otherwise. We use $\mathcal{R}_{\mathrm{I}}(u)$ and $\mathcal{R}_{\mathrm{U}}(i)$ to respectively denote the interacted item set of user u and the interacting user set of item i. The mathematical notations used in this paper are summarized in Table 1.

Problem Formulation. The recommendation problem is defined as: given the user-item interaction matrix $\mathbf{R}$, the user raw feature matrix $\mathbf{U}$ and item raw feature matrix $\mathbf{I}$, the goal is to predict unobserved interactions in $\mathbf{R}$, i.e., the probability $\hat{y}_{u,i} \in [0, 1]$

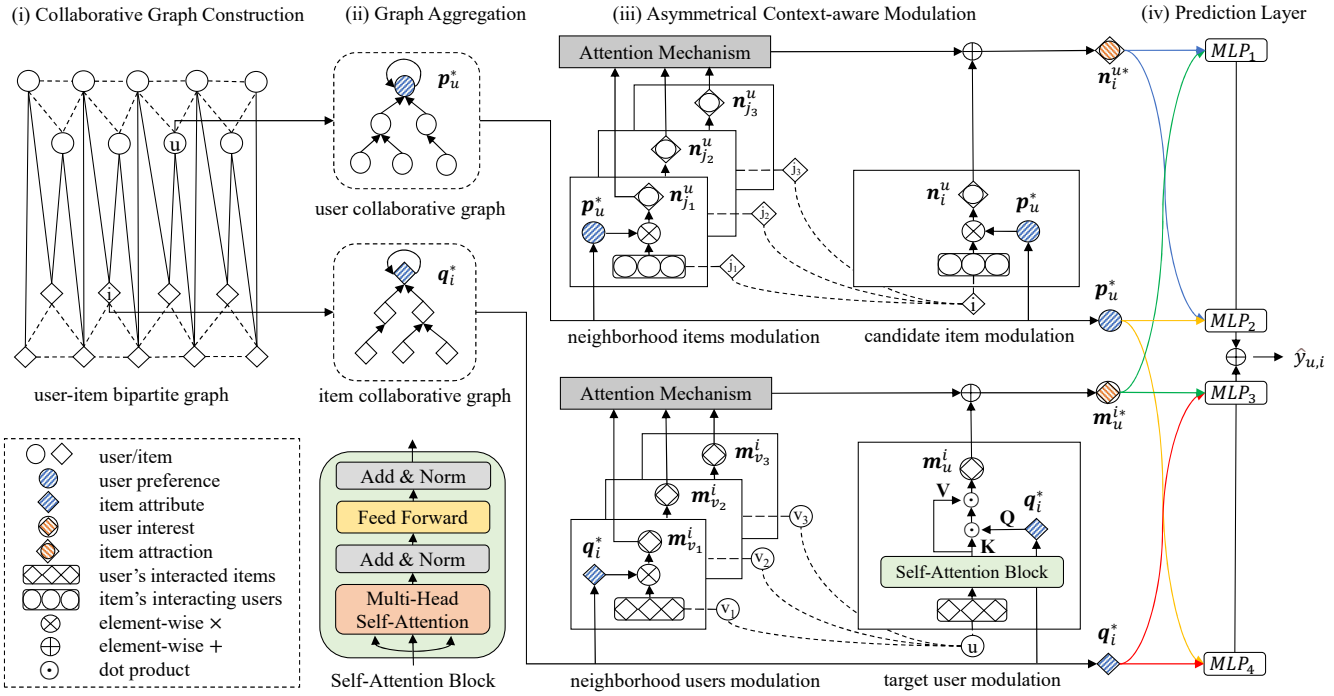

Figure 2: Architecture of the proposed ARBRE model. i) We first construct the collaborative graphs of users and items based on the bipartite graph. ii) In graph aggregation layer, we aggregate the homogeneous collaborative features through the collaborative graphs to generate user preference p_u^* and item attribute q_i^*. iii) In asymmetrical context-aware modulation module, the target user preference p_u^* and candidate item attribute q_i^* are introduced in item domain and user domain respectively to learn context-aware item attraction n_i^{u*} and user interest m_u^{i*} based on the interaction data. iv) Finally, the learned representations in user domain and item domain are input pair-wisely into prediction layer to get final score $\hat{y}_{u,i}$.

of the target user u interacting with an unobserved candidate item i.

4 METHODOLOGIES

The architecture of the proposed model is illustrated in Figure 2. The model consists of the following modules: (i) collaborative graphs construction, which construct two collaborative graphs for user and item from the user-item historical interactions, respectively; (ii) graph aggregation layer, which aggregates the features of users and items on the collaborative graphs to generate homogeneous user preference and item attribute; (iii) an asymmetrical context-aware modulation module, which model dynamic user interest and item attraction from historical interaction data according to their contexts with two asymmetrical modulation structures, respectively; (iv) a prediction layer, which model the user-item interaction by cross combining the user preference and interest and the item attribute and attraction. Each of module is described in details next.

4.1 Collaborative Graphs Construction

Intuitively, the users who share much neighborhood overlap with each other in the user-item bipartite graph may have much common preferences, and are called *collaborative neighborhood users*. Stacking many layers of GNN on the bipartite graph to capture homogeneous user preferences among collaborative neighborhood users may lead to over-smoothing problem [17]. Thus the collaborative graph is constructed from the bipartite graph to explicitly express the collaborative relations among users. The collaborative similarity $sim_{\mathrm{U}}(u, v)$ between any user u and v is firstly calculated based on the Jaccard similarity coefficient as below.

$$sim_{\mathrm{U}}(u,v) = \frac{|\mathcal{R}_{\mathrm{I}}(u)| \cap |\mathcal{R}_{\mathrm{I}}(v)|}{|\mathcal{R}_{\mathrm{I}}(u)| \cup |\mathcal{R}_{\mathrm{I}}(v)|} \tag{1}$$

u and v are considered to be collaborative neighborhood users of each other if $sim_{\mathrm{U}}(u, v) > \eta$, where η is a fixed threshold. Similarly, the collaborative neighborhood items can be obtained with the same operation. Then the user collaborative graph $\mathcal{G}_{\mathrm{U}} = (\mathcal{U}, \mathcal{E}_{\mathrm{U}})$ and item collaborative graph $\mathcal{G}_{\mathrm{I}} = (\mathcal{I}, \mathcal{E}_{\mathrm{I}})$ can be constructed by forming the edge set $\mathcal{E}_{\mathrm{U}}$ and $\mathcal{E}_{\mathrm{I}}$ among the collaborative neighborhood users and items, respectively.

4.2 Graph Aggregation Layer

The model first transforms the raw features of users and items into the low-dimensional latent embedding space through two embedding layers, respectively.

$$\begin{aligned} \mathbf{P} &= \mathbf{W}_{\mathrm{U}}\mathbf{U} \\ \mathbf{Q} &= \mathbf{W}_{\mathrm{I}}\mathbf{I} \end{aligned} \tag{2}$$

where $\mathbf{W}_{\mathrm{U}} \in \mathbb{R}^{d \times l_{\mathrm{U}}}$, $\mathbf{W}_{\mathrm{I}} \in \mathbb{R}^{d \times l_{\mathrm{I}}}$ are the trainable weight matrices, $\mathbf{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \cdots, \boldsymbol{p}_M] \in \mathbb{R}^{d \times M}$ and $\mathbf{Q} = [\boldsymbol{q}_1, \boldsymbol{q}_2, \cdots, \boldsymbol{q}_N] \in \mathbb{R}^{d \times N}$ are the transformed embeddings of users and items respectively. d is the embedding dimension.

Collaborative similar users (resp. items) usually share some common preferences (resp. attributes), which can be extracted as their homogeneous collaborative features by GNN. GNN learns rich high-order semantic features from graph structure by propagating and aggregating neighborhood information. Recent works demonstrate that the common node aggregation designs in GNN, i.e., feature transformation and nonlinear activation, contribute little in CF recommendation tasks [11]. Adapt this idea, we design a simplified GNN module for collaborative graph aggregation to yield homogeneous user preference and item attribute as follows.

In user domain, let $\boldsymbol{p}_u^0 = \boldsymbol{p}_u$ be the first layer input of the GNN module for each user u. The user embedding $\boldsymbol{p}_u^{l+1}$ at $(l+1)$-th layer is updated as:

$$\boldsymbol{p}_u^{l+1} = LN\left(\sum_{v \in \mathcal{N}_{\mathrm{U}}(u)} \boldsymbol{p}_v^l\right) \tag{3}$$

where $\mathcal{N}_{\mathrm{U}}(u)$ is the set of neighbors of u in $\mathcal{G}_{\mathrm{U}}$, and LN is the layer-wise normalization to avoid the scale of embedding increasing with graph aggregation. After L layers of graph aggregation, we average the embeddings from all layers to form the graph enhanced user **preference** $\boldsymbol{p}_u^*$, which captures user's intrinsic homogeneous feature within its neighborhoods.

Similarly, in item domain, let $\boldsymbol{q}_i^0 = \boldsymbol{q}_i$ be the first layer input of the GNN module for each item i. We update the item embedding $\boldsymbol{q}_i^{l+1}$ at $(l+1)$-th layer as:

$$\boldsymbol{q}_i^{l+1} = LN\left(\sum_{j \in \mathcal{N}_{\mathrm{I}}(i)} \boldsymbol{q}_j^l\right) \tag{4}$$

where $\mathcal{N}_{\mathrm{I}}(i)$ is the set of neighbors of i in $\mathcal{G}_{\mathrm{I}}$. After L layers of graph aggregation, the graph enhanced item **attribute** $\boldsymbol{q}_i^*$ is obtained by averaging the embeddings from all layers.

4.3 Asymmetrical Context-aware Modulation

Though modeling user-item interactions simply based on the graph enhanced user preference and item attribute is able to achieve acceptable performance, we argue that the learned preference and attribute are semantically static without considering dynamic recommendation contexts of users and items. Some CF models [7, 35] have attempted to modulate user interest and item attraction based on different contexts. However, these models suffer from a symmetrical structure which is not sufficient for mining the different semantic features in user domain and item domain. To this end, we introduce our asymmetrical context-aware modulation module in this section. We go into details of each side modulation as below.

4.3.1 Item Attraction Modulation. Besides the item attributes that affect the users' decision of interaction with it, an item also has its specific attraction to any target user, and the attraction to a target user can be modeled by the relations between item's historical interacting users and the target user [3, 5, 35]. Since the historical users of an item are generally independent with each other, there is no need to explicitly model their relationships. Following [7, 35], the attraction $\boldsymbol{n}_i^u$ of an item i to a target user u is modeled by the *product & max pooling* modulation which filters out the information of item's interacting users more related to that of the target user.

$$\boldsymbol{n}_i^u = MP_{v \in \mathcal{R}_{\mathrm{U}}(i)}\left(\{\boldsymbol{p}_u^* \otimes \boldsymbol{p}_v^*\}\right) \tag{5}$$

where MP indicates the max pooling operation, and $\otimes$ is the element-wise product. The MP helps to focus on the important features from item's historical users related to the target user and reduce the noisy information.

Intuitively, the collaborative similar items are the items that frequently interact with (e.g. be purchased by) the same users, thus a user who has been attracted by the collaborative neighbors of a candidate item is quite likely to be attracted by the candidate item as well. Hence we enrich the candidate item dynamic attraction by the attractions of its neighbors. The attraction of any neighbor j in $\mathcal{N}_{\mathrm{I}}(i)$ to the target user is learned with the same operation as equation (5). Then the attractions of neighbors are aggregated non-uniformly by a simplified attention mechanism. The attraction attention weight $\alpha_{i,j}$ of any neighbor j to item i is calculated based on their attraction similarity.

$$\alpha_{i,j} = \left(\boldsymbol{n}_i^u\right)^T \cdot \left(\boldsymbol{n}_j^u\right) \tag{6}$$

After normalizing the attention weights by score function softmax, the final candidate item **attraction** $\boldsymbol{n}_i^{u*}$ is got by aggregating its neighbors' attractions.

$$\begin{aligned} \boldsymbol{n}_i^{u*} &= 0.5 \times \left(\boldsymbol{n}_i^u + \textstyle\sum_{j \in \mathcal{N}_{\mathrm{I}}(i)} \alpha'_{i,j} \cdot \boldsymbol{n}_j^u\right) \\ \alpha'_{i,j} &= \frac{\exp(\alpha_{i,j})}{\sum_{j \in \mathcal{N}_{\mathrm{I}}(i)} \exp(\alpha_{i,j})} \end{aligned} \tag{7}$$

4.3.2 User Interest Modulation. Similarly, besides the user preference that affect her decision of interaction with an item, a user also has its specific interest in any candidate item, and the interest in a candidate item can be modeled by the relations between user's historical interacted items and the candidate item [5, 35]. Different from the historical interactions of any item, the semantic feature of historical interactions of any user is more complex. The interacted items of a user may be substitutive items or complementary items of each other, i.e., there are strong interdependencies among the user's interacted items. In order to capture their relations and interdependencies, we employ a self-attention block [28] composed of a multi-head self-attention layer and a feed-forward network to learn their features. Mathematically, let $\mathbf{Q}_u = \left[\boldsymbol{q}_{i_1}^*, \boldsymbol{q}_{i_2}^*, \ldots, \boldsymbol{q}_{i_{|\mathcal{R}_I(u)|}}^*\right] \in \mathbb{R}^{d \times |\mathcal{R}_I(u)|}$ be the attributes of the target user u's interacted items. The multi-head self-attention is defined as:

$$\begin{aligned} MultiheadAttn\left(\mathbf{Q}_u\right) &= \mathbf{W}^{\mathrm{O}} \cdot Concat(\mathbf{head}_1, ..., \mathbf{head}_h) \\ \mathbf{head}_i &= Attention\left(\mathbf{W}_i^{\mathrm{Q}}\mathbf{Q}_u, \mathbf{W}_i^{\mathrm{K}}\mathbf{Q}_u, \mathbf{W}_i^{\mathrm{V}}\mathbf{Q}_u\right) \end{aligned} \tag{8}$$

where h is the number of heads, $\mathbf{W}_i^{\mathrm{Q}}$, $\mathbf{W}_i^{\mathrm{K}}$, $\mathbf{W}_i^{\mathrm{V}} \in \mathbb{R}^{d_h \times d}$ are the projection matrices of each head, $d_h = d/h$ is the dimension of each head, and $\mathbf{W}^{\mathrm{O}} \in \mathbb{R}^{d \times d}$ is a learnable output weight matrix. The scaled dot-product attention function is adopted as follows:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{V} \cdot softmax\left(\frac{\mathbf{K}^T\mathbf{Q}}{\sqrt{d_h}}\right) \tag{9}$$

The non-linearity of the self-attention block is endowed with the feed-forward network FFN. The residual connections [10] and layer normalization LN [1] are also conducted successively to obtain the output of the self-attention block $\mathbf{H}_u$.

$$\begin{aligned} \mathbf{H_u} &= LN\left(\mathbf{Q}'_u + FFN\left(\mathbf{Q}'_u\right)\right) \\ \mathbf{Q}'_u &= LN\left(\mathbf{Q}_u + MultiHeadAttn\left(\mathbf{Q}_u\right)\right) \end{aligned} \tag{10}$$

$\mathbf{H}_u$ capture the interdependencies of the target user's interacted items. The user's interest $\boldsymbol{m}_u^i$ in the candidate item i is modeled by an attention-based modulation, in which the candidate item's attribute $\boldsymbol{q}_i^*$ is linearly transformed and set as the query vector.

$$\boldsymbol{m}_{\boldsymbol{u}}^i = \left(\mathbf{W}^{\mathrm{V}}\mathbf{H}_u\right) \cdot softmax\left(\frac{\left(\mathbf{W}^{\mathrm{K}}\mathbf{H}_u\right)^T\left(\mathbf{W}^{\mathrm{Q}}\boldsymbol{q}_i^*\right)}{\sqrt{d}}\right) \tag{11}$$

where $\mathbf{W}^{\mathrm{Q}}, \mathbf{W}^{\mathrm{K}}, \mathbf{W}^{\mathrm{V}} \in \mathbb{R}^{d\times d}$ are the weight matrices.

Intuitively, the collaborative similar users are users that frequently interact with the same items, thus the collaborative neighbors' interests in the candidate item may also influence the target user's decision on the candidate item. Hence we enrich the target user interest with interests of her neighbors. The interest of any neighbor is modeled by a simpler modulation method without considering the interdependences among her interacted items, as such interdependences have relatively little effect on the target user, while adopting *self-attention block & attention-based modulation* may result in overfitting problem. Specifically, we conduct *product & max pooling* modulation method to model the neighborhood users interest $\boldsymbol{m}_v^i$ in candidate item i.

$$\boldsymbol{m}_v^i = MP_{j\in\mathcal{R}_{\mathrm{I}}(v)}\left(\{\boldsymbol{q}_i^* \otimes \boldsymbol{q}_j^*\}\right) \tag{12}$$

where $v \in \mathcal{N}_{\mathrm{U}}(u)$ is a neighbor of target user u. The final target user **interest** $\boldsymbol{m}_u^{i*}$ is got by aggregating her neighbors' interest with a simplified attention mechanism.

$$\begin{aligned} \boldsymbol{m}_u^{i*} &= 0.5 \times \left(\boldsymbol{m}_u^i + \sum_{v\in\mathcal{N}_{\mathrm{U}}(u)} \alpha'_{u,v} \cdot \boldsymbol{m}_v^i\right) \\ \alpha'_{u,v} &= \frac{\exp\left(\alpha_{u,v}\right)}{\sum_{v\in\mathcal{N}_{\mathrm{U}}(u)}\exp\left(\alpha_{u,v}\right)} \\ \alpha_{u,v} &= \left(\boldsymbol{m}_u^i\right)^T \cdot \left(\boldsymbol{m}_v^i\right) \end{aligned} \tag{13}$$

4.4 Prediction Layer

After the above representation learning modules, we obtain static user preference and dynamic user interest in user domain, and static item attribute and dynamic item attraction in item domain. Since the target user's decision on the interaction with the candidate item depends on both user side factors and item side factors, we send user domain's embeddings and item domain's embedding pair-wisely into MLPs for predicting the interacting scores:

$$\begin{aligned} y_{u,i}^1 &= MLP_1\left(\boldsymbol{m}_u^{i*}, \boldsymbol{n}_i^{u*}\right) \\ y_{u,i}^2 &= MLP_2\left(\boldsymbol{p}_u^*, \boldsymbol{n}_i^{u*}\right) \\ y_{u,i}^3 &= MLP_3\left(\boldsymbol{m}_u^{i*}, \boldsymbol{q}_{i*}\right) \\ y_{u,i}^4 &= MLP_4\left(\boldsymbol{p}_u^*, \boldsymbol{q}_i^*\right) \end{aligned} \tag{14}$$

The final score is calculated by the weighted sum of the four scores above:

$$\hat{y}_{u,i} = \sum_{k=1}^{4} \lambda_k y_{u,i}^k \tag{15}$$

where $\sum_{k=1}^4 \lambda_k = 1, \lambda_k > 0$ denote the importance of the k-th score in predicting the final score. It can be tuned as hyper-parameters manually or optimized automatically as model parameters by using an attention mechanism [4]. In our experiments, we find that setting λ_k uniformly can generally lead to good performance, thus we do not design learnable component to optimize λ_k for model simplicity.

4.5 Model Training

For modeling user-item implicit feedback interaction, the most widely used loss function is known as the cross-entropy, which is defined as below:

$$\mathcal{L} = -\sum_{(u,i)\in\mathbb{D}} y_{u,i}\log\hat{y}_{u,i} + \left(1-y_{u,i}\right)\log\left(1-\hat{y}_{u,i}\right) \tag{16}$$

where $\mathbb{D}$ is the training dataset, $y_{u,i} \in \{0,1\}$ represents whether the user interacted with the item or not. The mini-batch Adaptive Moment Estimation (Adam) [14] is adopted as the optimizer, which can adaptively adjust the learning rate during the training phase. We also adopt the dropout strategy [26] in graph aggregation and MLP to alleviate the overfitting.

5 EXPERIMENTS

To demonstrate the effectiveness of the proposed model, we conduct experiments to answer the following research questions:

RQ1 How does ARBRE perform as compared with state-of-the-art methods for recommendation?

RQ2 Are the key designs in ARBRE, such as the asymmetrical structure, necessary for improving performance?

RQ3 How do hyper-parameters in ARBRE impact recommendation performance?

5.1 Experiment Setup

5.1.1 Dataset. We apply our model to three datasets from dataset collections Amazon[1] and Yelp[2].

- Amazon: This is a series of datasets extracted from real world application AMAZON [23]. Top-level product categories are treated as separated datasets in Amazon. We consider two categories 'Beauty' and 'Video'.
- Yelp: This is a popular dataset for business recommendation. Following [41], local businesses are viewed as items and the transaction records after January 1st, 2019 of the version updated on February 21th, 2020 are used for experiments.

For all datasets, users and items with fewer than 5 interactions are filtered out to ensure data quality. The statistical details of the datasets after prepossessing are summarized in Table 2.

[1] http://jmcauley.ucsd.edu/data/amazon/
[2] http://www.yelp.com/dataset

Table 2: Statistics of the datasets

Dataset	#users	#items	#interactions	Density
Beauty	22,363	12,101	198,502	0.00073
Video	24,303	10,672	231,780	0.00089
Yelp	10,873	17,965	271,363	0.00139

5.1.2 Baseline. To evaluate the performance of the proposed method, we compare ARBRE with several comparative methods, including some state-of-the-art models for recommendation.

The first group of models is shallow CF model:

BPR [25]: This is a latent factor model which designs a pairwise ranking loss for personalized recommendation by assuming users prefer items they interact with compared to unobserved ones.

The second group is GNN-based deep CF models:

GCMC [2]: This method proposes a graph auto-encoder framework containing one convolutional layer to exploit the direct connections between users and items on user-item bipartite graph.

NGCF [33]: This method injects the collaborative signal into graph embedding process and models the high-order connectivity in user-item bipartite graph.

LightGCN [11]: This is a strong baseline, which removes the feature transformation and nonlinear activation designs in GCN to make it more appropriate for recommendation.

UltraGCN [22]: This model proposes an ultra-simplified formulation of GCN by skipping infinite layers of message passing for recommendation.

IMP-GCN [19]: This model performs high-order graph convolution inside subgraphs to avoid propagating negative information into embedding learning.

The third group contains deep-learning based context-aware model:

DICER [7]: This method introduces contextualized information for modulating the representations in dual sides based on the collaborative graph enhanced representations.

5.1.3 Evaluation Metrics. Our model adopts *Recall@K* and *NDCG@K* (Normalized Discounted Cumulative Gain) for evaluating the performance of all methods. This two metrics have been widely used in previous works [7, 11, 33]. *Recall@K* considers whether the ground-truth is ranked among the top K items, while *NDCG@K* assigns greater weights on higher positions.

5.1.4 Implementation Details. We use Pytorch to implement our model[3] and the Xavier initializer [8] to initialize the model parameters. For each dataset, we randomly select 80% as the training set to learn parameters, 10% as the validation set to turn hyper-parameter and 10% as the test set to evaluate the prediction performance. The hyper-parameter settings are as follows: learning rate is 0.001, training batch size is 128, embedding dimension d=64, number of graph neural network layers L=3, number of attention heads h=2, dropout ratio is 0.1, coefficient $\lambda_1 = \lambda_2 = \lambda_3 = \lambda_4 = 1/4$. In the training process, we uniformly sample the items with no observed interaction with a user as her negative samples. The numbers of positive and negative samples for each user are the same. Follow the strategy in [7, 13], for evaluation metrics calculation, we randomly sample 100 unobserved items as negative items and rank them with the ground-truth items. For all baselines, the optimal hyper-parameter settings are determined either by our experiments or suggested by previous works to ensure the best performance. As we address the situations where no social relations can be obtained, the social relations in [7] are replaced by collaborative similar relations in the experiments.

[3]https://github.com/halaoyy/ARBRE_pytorch

5.2 Comparative Results: RQ1

The results of comparing ARBRE with other baseline methods are reported in Table 3. Experimentally, K=5,10,15 are selected as the recommendation lengths for two evaluation metrics. The following observations can be made from the experimental results:

First, deep-learning based models generally outperform the shallow model on all evaluation metrics, which indicates the effectiveness of deep neural networks in modeling complex nonlinear features.

Second, there are some findings in the comparison of GNN-based deep CF models. As we can see in Table 3, NGCF consistently outperforms GCMC. This demonstrates the advantages of NGCF in incorporating collaborative signals and high-order connectivity in representation learning process. Also, LightGCN outperforms NGCF in all cases, which further proves that the feature transformation and nonlinear activation operations of GCN contribute little in recommendation tasks. UltraGCN and IMP-GCN achieve competitive performances, which indicate the effectiveness of approximating the limit of message passing and denoising the high-order embedding learning.

Third, our proposed ARBRE achieves the best performance in all metrics, with an average improvement of 8.36%, 9.64% and 4.62% compared to the second best method for Beauty, Video and Yelp, respectively. Specifically, our method performs better than the GNN-based deep CF models, confirming the effectiveness of introducing contextualized information in learning representations. The substantial improvement of our model over DICER is possibly due to the facts: (1) our model removes the designs of graph convolutional networks that is irrelevant to the recommendation task, which further alleviates the training difficulty; (2) we design two asymmetrical modulation models for user interest and item attraction learning according to their semantic properties, respectively. (3) in prediction layer, the representations from user domain and item domain are input pair-wisely into MLPs for modeling user-item interactions, while DICER ignore the interaction between modulated user interest and item attraction.

5.3 Ablation Study: RQ2

In this subsection, we analyze the impact of key components in our model with some ablation studies.

5.3.1 Effect of Context-aware Modulation. In the last subsection, we have demonstrated the advantages of the proposed ARBRE. The model follows an asymmetrical structure and designs context-aware modulation methods for learning user interest and item attraction, respectively. To validate the effectiveness of the context-aware modulation module, we compare ARBRE with its three variants:

Table 3: Comparisons of different models on three datasets. The best results are in boldface and the second best results are underlined. "Impv." indicates the relative improvement of the best results compared to the second best results.

Datasets	Metric	BPR	GCMC	NGCF	LightGCN	UltraGCN	IMP-GCN	DICER	ARBRE	Impv.
Beauty	Recall@5	0.3554	0.3714	0.4310	0.4733	0.4690	0.4812	0.4718	**0.5223**	8.54%
	Recall@10	0.4436	0.4901	0.5395	0.5814	0.5644	0.5925	0.5903	**0.6465**	9.11%
	Recall@15	0.5006	0.5642	0.6022	0.6406	0.6291	0.6539	0.6573	**0.7202**	9.57%
	NDCG@5	0.3165	0.2981	0.3561	0.3914	0.3939	0.3972	0.3799	**0.4249**	6.97%
	NDCG@10	0.3543	0.3400	0.3947	0.4297	0.4286	0.4367	0.4232	**0.4704**	7.72%
	NDCG@15	0.3738	0.3618	0.4130	0.4472	0.4477	0.4549	0.4431	**0.4923**	8.22%
Video	Recall@5	0.4781	0.5470	0.6187	0.6341	0.6355	0.6332	0.6105	**0.7024**	10.53%
	Recall@10	0.6032	0.6924	0.7473	0.7558	0.7582	0.7541	0.7479	**0.8227**	8.51%
	Recall@15	0.6753	0.7703	0.8123	0.8192	0.8221	0.8145	0.8206	**0.8836**	7.48%
	NDCG@5	0.4172	0.4420	0.5166	0.5338	0.5351	0.5331	0.4970	**0.5957**	11.33%
	NDCG@10	0.4697	0.4935	0.5623	0.5769	0.5802	0.5760	0.5476	**0.6395**	10.22%
	NDCG@15	0.4948	0.5169	0.5819	0.5959	0.5995	0.5943	0.5697	**0.6580**	9.76%
Yelp	Recall@5	0.4210	0.4389	0.4693	0.4884	0.6002	0.5522	0.5264	**0.6266**	4.40%
	Recall@10	0.5731	0.6277	0.6528	0.6645	0.7472	0.7337	0.7012	**0.7910**	5.86%
	Recall@15	0.6466	0.7325	0.7494	0.7585	0.8335	0.8268	0.8113	**0.8784**	5.39%
	NDCG@5	0.4253	0.4044	0.4329	0.4568	0.5287	0.5153	0.4452	**0.5469**	3.44%
	NDCG@10	0.4785	0.4665	0.4935	0.5140	0.5866	0.5748	0.5148	**0.6120**	4.33%
	NDCG@15	0.5066	0.5015	0.5258	0.5453	0.6174	0.6064	0.5540	**0.6439**	4.29%

ARBRE-m, ARBRE-n, ARBRE-m-n. These three variants are defined as follows:

- ARBRE-m: The context-aware modulation in user domain is removed. This variant only uses user preference $\boldsymbol{p}_u^*$, item attribute $\boldsymbol{q}_i^*$ and item attraction $\boldsymbol{n}_i^{u*}$ to predict the score, while ignoring user interest $\boldsymbol{m}_u^{i*}$.
- ARBRE-n: The context-aware modulation in item domain is removed. This variant only uses user preference $\boldsymbol{p}_u^*$, item attribute $\boldsymbol{q}_i^*$ and user interest $\boldsymbol{m}_u^{i*}$ to predict the score, while ignoring item attraction $\boldsymbol{n}_i^{u*}$.
- ARBRE-m-n: The context-aware modulations in both user and item domain are removed. This variant only uses user preference $\boldsymbol{p}_u^*$, item attribute $\boldsymbol{q}_i^*$ to predict the score, while ignoring modulated user interest $\boldsymbol{m}_u^{i*}$ and item attraction $\boldsymbol{n}_i^{u*}$.

The performance of ARBRE and its variants are shown in Table 4. From the results, we have the following findings:

The results prove the effectiveness of the context-aware modulation in user domain. ARBRE-m perform worse than ARBRE in all cases. On average, the relative reduction is 17.41% on Recall metric and 21.03% on NDCG metric. It verifies that capturing context-aware user interest boost the recommendation performance.

The results prove the effectiveness of the context-aware modulation in item domain. Without the context-aware item attraction, the performance decrease significantly. Specifically, the performance of ARBRE-n decrease 6.24% and 8.15% on Recall and NDCG metrics, respectively. This demonstrate the advantage of exploiting context-aware item attraction in improving the performance of recommendation.

Both the context-aware modulation in user domain and item domain contribute to improve the performance of prediction. The variant ARBRE-m-n that removes the entire context-aware modulation module achieve the worst performance in this comparison, i.e., reduce averagely 25.56% and 32.09% on Recall and NDCG metrics compares to ARBRE.

Table 4: Effect of context-aware modulation on Beauty

Model	Recall@5	Recall@10	Recall@15	NDCG@5	NDCG@10	NDCG@15
ARBRE-m	0.4196	0.5373	0.6074	0.3299	0.3730	0.3937
ARBRE-n	0.4769	0.6093	0.6894	3853	0.4331	0.4568
ARBRE-m-n	0.3604	0.4924	0.5627	0.2753	0.3239	0.3449
ARBRE	**0.5223**	**0.6465**	**0.7202**	**0.4249**	**0.4704**	**0.4923**

Table 5: Effect of asymmetrical structure on Beauty

Model	Recall@5	Recall@10	Recall@15	NDCG@5	NDCG@10	NDCG@15
ARBRE-α	0.4657	0.5832	0.6486	0.3761	0.4192	0.4387
ARBRE-β	0.4652	0.5840	0.6612	0.3789	0.4213	0.4440
ARBRE	**0.5223**	**0.6465**	**0.7202**	**0.4249**	**0.4704**	**0.4923**

5.3.2 Effect of Asymmetrical Structure. Comparing to other context-aware models [7, 35], a key difference of our model is that ARBRE follows an asymmetrical structure where different methods are designed for extracting semantic features in user domain and item domain. Therefore, in order to verify the effectiveness of designing different modulation methods in user domain and item domain, we compare the performance of ARBRE with two variant models:

- ARBRE-α: In this variant, the context-aware modulation method for learning target user interest is identical to the approach of learning candidate item attraction in ARBRE, i.e., learning target user interest by conducting element-wise

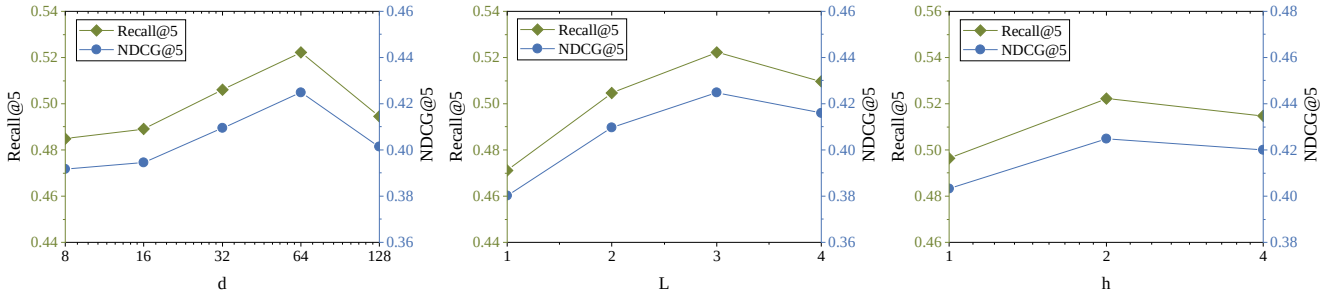

Figure 3: Performance of ARBRE on Beauty w.r.t different hyper-parameters

product and max pooling operations over target user's interacted items.

- ARBRE-β: In this variant, the context-aware modulation method for learning candidate item attraction is the same as the approach of learning target user interest in ARBRE, i.e., learning candidate item attraction with a self-attention block and an attention-based modulation.

The experimental results of ARBRE and its variants are reported in Table 5. The following findings are obtained:

The *self-attention block & attention-based modulation* is more appropriate for learning target user interest than *product & max pooling* operations. On average, the relative reduction of ARBRE-α is 10.19% on Recall metric and 11.08% on NDCG metric comparing to ARBRE. This demonstrates that the self-attention block and attention-based modulation can better exploit the complex relations and interdependencies in user's clicked items.

The *product & max pooling* is more effective in learning candidate item attraction than *self-attention block & attention-based modulation*. On average, ARBRE-β reduce 9.59% on Recall metric and 10.35% on NDCG metric, respectively. It justifies our assumption that it's sufficient to describe item's historical users by extracting their common features.

5.3.3 Effect of Prediction Layer. In ARBRE, the learned representations in user domain and item domain are input pair-wisely into 4 MLPs for modeling user-item interaction. Specifically, ARBRE models the interaction of the user interest $\boldsymbol{m}_u^{i*}$ and item attraction $\boldsymbol{n}_i^{u*}$ which is ignored by [7]. Therefore, a variant named ARBRE-MLP_1 is designed by removing MLP_1 in prediction layer. The experimental results are shown in Table 6. We can find that ARBRE outperform ARBRE-MLP_1 in all metrics. The average reduction on Recall metric is 8.81% and 12.59% on NDCG metric. One likely reason of the improvement is that the modulated representations capture precisely the interest and the attraction shown by the target user and the candidate item when facing the context respectively, between which the interaction is modeled with little noisy information.

5.4 Parameter Sensitivity: RQ3

In this subsection, we investigate the sensitivity of the proposed ARBRE to some hyper-parameters, which include the embedding dimension d, the number of GNN layer L and the number of attention heads h. As shown in Figure 3, from which it can be observed that: (1) The embedding dimension affects the performance of ARBRE. The model lacks expressiveness if the dimension is too small, while the performance degrades if it's too large due to sparsity; (2) Increasing the number of GNN layers can improve the performance by aggregating high-order features, but too many layers may lead to over-smoothing problem; (3) The performance is optimal when the number of attention head is 2, too few or too many heads will degrade the recommendation performance.

Table 6: Effect of the interaction between user interest and item attraction in the prediction layer on Video

Model	Recall@5	Recall@10	Recall@15	NDCG@5	NDCG@10	NDCG@15
ARBRE-MLP_1	0.6180	0.7554	0.8285	0.5111	0.5612	0.5832
ARBRE	**0.7024**	**0.8227**	**0.8836**	**0.5957**	**0.6395**	**0.6580**

6 CONCLUSION

In this paper, we propose ARBRE, which generates static user preference and item attribute by feature propagation through collaborative graphs and designs context-aware modulations for extracting dynamic user interest and item attraction with asymmetrical structures. Our method is equipped with good expressiveness because: (i) The simplified graph neural networks effectively captures homogeneous user preferences and item attributes, respectively; (ii) Two asymmetrical modulation modules are designed based on the different semantic features in user and item domains, which learns the most relevant user interest and item attraction under different contexts; (iii) Our method fully models the user-item interaction in the prediction layer. The comparative experiments and ablation studies on several real-world datasets validate the effectiveness of ARBRE and its accuracy in exploiting related features according to the different contexts.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No. 62002219, 62172278), Shanghai Sailing Program (No. 19YF1424700), Startup Fund for Youngman Research at SJTU (SFYR at SJTU).

REFERENCES

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[2] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).

[3] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *The world wide web conference*. 151–161.

[4] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*. 335–344.

[5] Weiyu Cheng, Yanyan Shen, Yanmin Zhu, and Linpeng Huang. 2018. DELF: A Dual-Embedding based Deep Latent Factor Model for Recommendation.. In *IJCAI*, Vol. 18. 3329–3335.

[6] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.

[7] Bairan Fu, Wenming Zhang, Guangneng Hu, Xinyu Dai, Shujian Huang, and Jiajun Chen. 2021. Dual side deep context-aware modulation for social recommendation. In *Proceedings of the Web Conference 2021*. 2524–2534.

[8] Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 249–256.

[9] Wei Guo, Rong Su, Renhao Tan, Huifeng Guo, Yingxue Zhang, Zhirong Liu, Ruiming Tang, and Xiuqiang He. 2021. Dual Graph enhanced Embedding Neural Network for CTR Prediction. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 496–504.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 770–778.

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

[12] X. He, L. Liao, H. Zhang, L. Nie, and T. S. Chua. 2017. Neural Collaborative Filtering. *International World Wide Web Conferences Steering Committee* (2017).

[13] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 197–206.

[14] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[15] Y. Koren and R. Bell. 2015. Advances in Collaborative Filtering. *Springer US* (2015).

[16] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009), 30–37.

[17] Qimai Li, Zhichao Han, and Xiao-Ming Wu. 2018. Deeper insights into graph convolutional networks for semi-supervised learning. In *Thirty-Second AAAI conference on artificial intelligence*.

[18] Donghua Liu, Jing Li, Bo Du, Jun Chang, and Rong Gao. 2019. Daml: Dual attention mutual learning between ratings and reviews for item recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 344–352.

[19] Fan Liu, Zhiyong Cheng, Lei Zhu, Zan Gao, and Liqiang Nie. 2021. Interest-aware message-passing gcn for recommendation. In *Proceedings of the Web Conference 2021*. 1296–1305.

[20] Anjing Luo, Pengpeng Zhao, Yanchi Liu, Fuzhen Zhuang, Deqing Wang, Jiajie Xu, Junhua Fang, and Victor S Sheng. 2020. Collaborative Self-Attention Network for Session-based Recommendation.. In *IJCAI*. 2591–2597.

[21] Chen Ma, Liheng Ma, Yingxue Zhang, Jianing Sun, Xue Liu, and Mark Coates. 2020. Memory augmented graph neural networks for sequential recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 34. 5045–5052.

[22] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1253–1262.

[23] Julian McAuley, Christopher Targett, Qinfeng Shi, and Anton Van Den Hengel. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th international ACM SIGIR conference on research and development in information retrieval*. 43–52.

[24] Andriy Mnih and Russ R Salakhutdinov. 2007. Probabilistic matrix factorization. *Advances in neural information processing systems* 20 (2007).

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2012. BPR: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618* (2012).

[26] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* 15, 1 (2014), 1929–1958.

[27] Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 world wide web conference*. 729–739.

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[29] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[30] Katrien Verbert, Nikos Manouselis, Xavier Ochoa, Martin Wolpers, Hendrik Drachsler, Ivana Bosnic, and Erik Duval. 2012. Context-aware recommender systems for learning: a survey and future challenges. *IEEE transactions on learning technologies* 5, 4 (2012), 318–335.

[31] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. 2015. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 1235–1244.

[32] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 950–958.

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

[34] Xiao Wang, Ruijia Wang, Chuan Shi, Guojie Song, and Qingyong Li. 2020. Multi-component graph convolutional collaborative filtering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 6267–6274.

[35] Qitian Wu, Hengrui Zhang, Xiaofeng Gao, Peng He, Paul Weng, Han Gao, and Guihai Chen. 2019. Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems. In *The World Wide Web Conference*. 2091–2102.

[36] Min Xie, Hongzhi Yin, Fanjiang Xu, Hao Wang, and Xiaofang Zhou. 2016. Graph-based metric embedding for next poi recommendation. In *International Conference on Web Information Systems Engineering*. Springer, 207–222.

[37] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 974–983.

[38] Mengqi Zhang, Shu Wu, Xueli Yu, Qiang Liu, and Liang Wang. 2022. Dynamic graph neural networks for sequential recommendation. *IEEE Transactions on Knowledge and Data Engineering* (2022).

[39] Lei Zheng, Vahid Noroozi, and Philip S Yu. 2017. Joint deep modeling of users and items using reviews for recommendation. In *Proceedings of the tenth ACM international conference on web search and data mining*. 425–434.

[40] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

[41] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-rec: Self-supervised learning for sequential recommendation with mutual information maximization. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1893–1902.