Correlative Preference Transfer with Hierarchical Hypergraph Network for Multi-Domain Recommendation

Zixuan Xu‡, Penghui Wei‡, Shaoguo Liu*, Liang Wang and Bo Zheng
Alibaba Group
Beijing, China
{xuzixuan.xzx,wph242967,shaoguo.lsg,liangbo.wl,bozheng}@alibaba-inc.com

ABSTRACT

Advanced recommender systems usually involve multiple domains (such as scenarios or categories) for various marketing strategies, and users interact with them to satisfy diverse demands. The goal of multi-domain recommendation (MDR) is to improve the recommendation performance of all domains simultaneously. Conventional graph neural network based methods usually deal with each domain separately, or train a shared model to serve all domains. The former fails to leverage users' cross-domain behaviors, making the behavior sparseness issue a great obstacle. The latter learns shared user representation with respect to all domains, which neglects users' domain-specific preferences. In this paper we propose $\mathsf{H^3Trans}$, a **h**ierarchical **h**ypergrap**h** network based correlative preference **trans**fer framework for MDR, which represents multi-domain user-item interactions into a unified graph to help preference transfer. $\mathsf{H^3Trans}$ incorporates two hyperedge-based modules, namely dynamic item transfer (Hyper-I) and adaptive user aggregation (Hyper-U). Hyper-I extracts correlative information from multi-domain user-item feedbacks for eliminating domain discrepancy of item representations. Hyper-U aggregates users' scattered preferences in multiple domains and further exploits the high-order (not only pair-wise) connections to improve user representations. Experiments on both public and production datasets verify the superiority of $\mathsf{H^3Trans}$ for MDR.

CCS CONCEPTS

• **Information systems** → **Personalization**; **Recommender systems**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Multi-domain Recommendation, Preference Transfer, Hypergraph Learning, Behavior Sparseness

†Co-first authorship. *Correspondence to: S. Liu.

WWW '23, April 30–May 4, 2023, Austin, Texas, USA

ACM ISBN 978-1-nnnn-nnnn-n/nn/nn…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

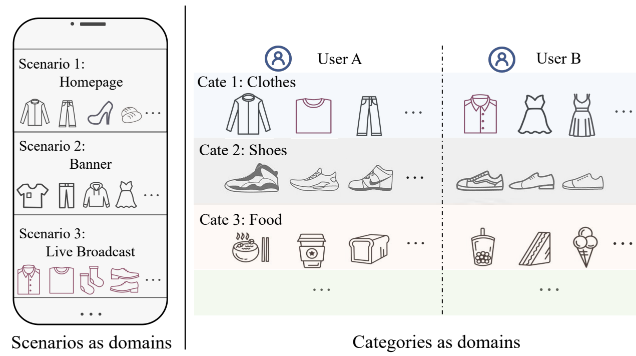

Figure 1: Illustration of multi-domain recommendation. The definition of *domain* can be *recommendation scenario* or *item categories*.

1 INTRODUCTION

Personalized recommender systems aim to make effective and satisfying choices for users. They usually involve multiple recommendation **scenarios** or domains, and each scenario contains a set of items that is related to the scenario's topic and marketing strategy. Users interact with these scenarios to satisfy diverse demands. For example, the E-commerce platform Taobao[1] provides diversified shopping spots including product search, homepage feed, banner, live broadcast and so on, as shown in the left part of Fig. 1. Baidu[2] serves as a comprehensive website where users can read news, watch videos and more. Broadly speaking, different item **categories** can also be regarded as multiple domains. As in the right part of Fig. 1, users usually interact with various categories such as clothes, food and more for their different demands.

Multi-domain recommendation (MDR) has attracted increasing research attention, the goal of which is to improve the recommendation performance of all domains simultaneously. There are both commonality and diversity among domains. For the commonality, multiple domains usually have common users and overlapped items, and a user may have similar behavior patterns across domains (for example, preferring ordinary or fashionable goods). The users' domain-invariant preference and items' static information can be shared across domains. For the diversity, the domains have different topics with specific items, thus attract different audiences and cause discrepant data distributions.

[1]https://www.taobao.com/
[2]https://www.baidu.com/

Graph neural networks (GNNs) have proven to be powerful for recommendations because user-item interactions are naturally suitable for modeling as a graph. Conventional GNN-based methods for MDR can be divided into two types. The first type deals with domains separately. That is, for each domain we construct user-item interaction graph and train model independently, which learns separate representations for different domains to characterize users' domain-specific preferences. However, the sparseness of interaction behaviors in emerging domains [3, 5] is a crucial obstacle. The second type alternatively constructs a unified interaction graph using multi-domain data and train a shared model to serve all domains [31]. Considering the intrinsic difference among domains' data distributions, the shared model neglects domain-specific characteristics which results in limited performance.

Researchers have proposed some advanced methods [1, 13, 19, 20, 32] that exert the prominent feature extracting ability of GNN and incorporate knowledge transfer to alleviate the sparseness. For example, pretrain-finetune diagram which transfers a pre-trained graph encoder to initialize the node embedding on the target domain is a widely used way [20]. Considering the pretrain-finetune paradigm only improves the recommendation accuracy on a single target domain, some works exploit to improve the recommendation accuracy on both domains simultaneously [13, 32]. Despite their effectiveness, these methods focus on knowledge transfer between only two domains. When employed in more than two domains, they only capture pair-wise relations between domains and dismiss the high-order connections.

For effective MDR, the key is to learn from the interactions in all domains and acquire transferable knowledge to obtain better user representations that characterize their domain-specific preferences. In this paper we propose H^{3}Trans, a **h**ierarchical **h**ypergrap**h** network based correlative preference **trans**fer framework to improve MDR. As a general topological structure, a hyperedge can connect an arbitrary number of nodes, and thus hypergraph provides a means for modeling high-order connections in multiple domains. We integrate users' multi-domain behaviors into a unified graph and incorporate hyperedges to help preference transfer. Specifically, each user is viewed as multiple nodes w.r.t. to different domains, where the representation of each user node characterizes the domain-specific preference. For item nodes, because items' properties are relatively static than users, we view each item as a single node shared by all domains.

The core of the hypergraph structure constructed by H^{3}Trans is two novel types of hyperedges for improving user and item representation learning. We first design a dynamic item transfer module named Hyper-I. For a given domain, we dynamically seek out related items from user-item interactions of other domains, and construct a hyperedge (named hyperedge-i) to connect them as cross-domain item relations. Hyperedge-i helps build relations between the items of different domains and capture users' correlative preference from the cross-domain behaviors without interference information. Moreover, we propose a structure-aware aggregator with attention mechanism to model the message passing procedure through hyperedge-i, which adjusts item representation much more correlative to the target domain and thus improves the recommendation performance in multiple domains.

We further introduce an adaptive user aggregation module named Hyper-U. Each user is viewed as a separate node per domain, that is, for a given user we can acquire separate user representations in multiple domains. We utilize a hyperedge (named hyperedge-u) to connect these separate user nodes of a given user, which aggregates the scattered user preferences among multiple domains. To effectively model the high-order connections among domains, we propose to employ attention mechanism into the message propagation within such hyperedges. Hyperedge-u contributes to transferring correlative preferences from source domains and capturing the commonality among multiple domains. Note that each domain can be viewed as the target domain (and the others as the sources), thus our proposed H^{3}Trans can improve the quality of user representation for all domains simultaneously.

The contributions are as follows:

- We propose H^{3}Trans, a hierarchical hypergraph network based correlative preference transfer framework for MDR. To our knowledge, this is the first work that investigates hypergraph-based preference transfer in MDR.
- To improve item representations for cross-domain transfer, Hyper-I performs dynamic item transfer which helps extract correlative preference from the cross-domain behaviors without interference information.
- To model the high-order connections among users' multi-domain behaviors, Hyper-U aggregates users' scattered preferences in multiple domains and exploits the high-order connections with an attention based propagation layer.
- Extensive experiments on large-scale production datasets and public datasets are conducted to analyze our proposed H^{3}Trans, and the results demonstrate the superiority.

2 PRELIMINARY

2.1 Definition of Hypergraph

Compared to an ordinary graph, a hypergraph is a more general topological structure where a **hyperedge** can connect an arbitrary number of nodes. Formally, a hypergraph is composed of a node set and a hyperedge set. The connectivity of a hypergraph can be represented by an incidence matrix H, where $h_{ve} = 1$ if the hyperedge e contains the node v, otherwise $h_{ve} = 0$. Besides, we use E_v to denote a set of hyperedges that connect to node v, and use V_e to denote a set of nodes connected to hyperedge e. Also, we can define the neighbors $\mathcal{N}_v$ of node v as a set of nodes that share at least one hyperedge with node v.

2.2 Problem Definition

Given domains $\{\mathcal{D}_m\}_{m=1}^{T}$, where T denotes the number of domains. For domain $\mathcal{D}_m$, we utilize $\boldsymbol{U}^m$ and $\boldsymbol{I}^m$ to denote its user ID set and item ID set respectively. Let $\mathcal{R}^m \in \mathbb{R}^{|\boldsymbol{U}^m|\times|\boldsymbol{I}^m|}$ denotes the user-item interaction matrix of domain $\mathcal{D}_m$. If its entry $r_{ui}^m = 1$, it means that the user u interacted with the item i under domain m. In this work, we consider click behavior as the interaction type.

Given a specific domain $\mathcal{D}_m$, the problem of single-domain recommendation is to estimate the scores of unobserved entries in one interaction matrix $\mathcal{R}^m$, and we compute the score between a user and an item as:

$$\hat{r}_{u,i}^m = f(z_u, z_i \mid \mathcal{D}_m) \tag{1}$$

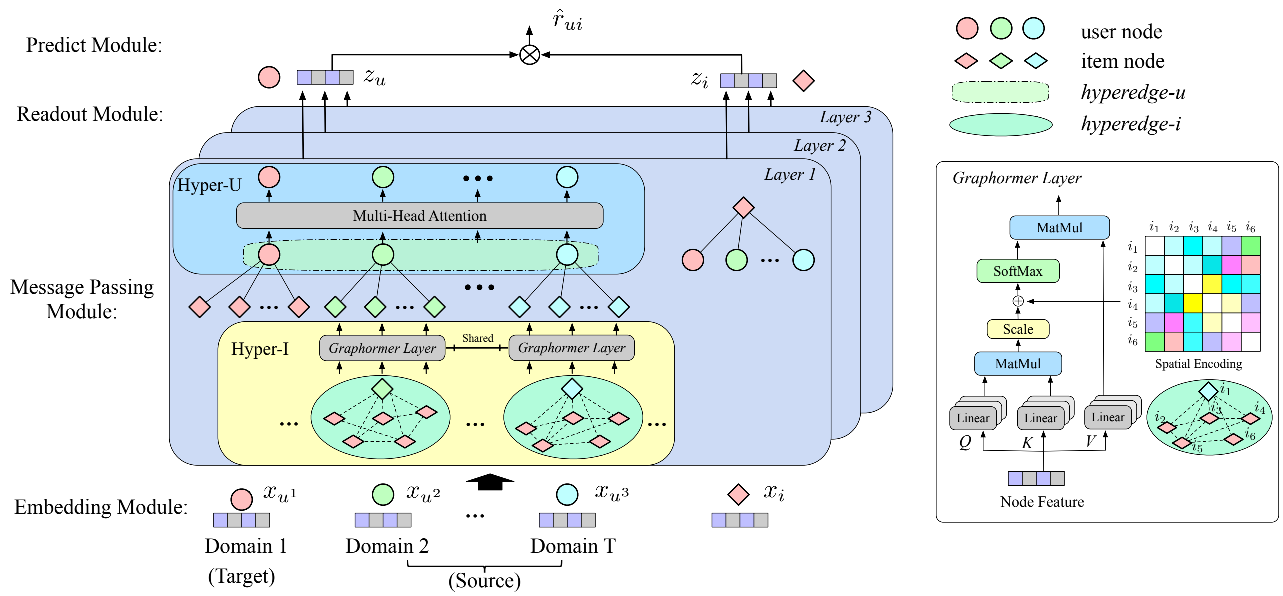

Figure 2: Overall architecture of H^{3}Trans. It contains two hyperedge-based modules: adaptive user aggregation (Hyper-U) and dynamic item transfer module (Hyper-I). These two modules compose a hierarchical hypergraph neural network. Different colors refer to different domains. Here we regard the first domain $\mathcal{D}_1$ as target domain and the others are sources.

Here z_u and z_i denote the learned representations of user $u \in \boldsymbol{U}^m$ and item $i \in \boldsymbol{I}^m$ for domain $\mathcal{D}_m$, and $f(\cdot)$ is the similarity function.

The problem of multi-domain recommendation is to estimate the unobserved scores for all interaction matrices $\{\mathcal{R}^m\}_{m=1}^{T}$. Specifically, the **user set** $\boldsymbol{U}$ is shared among all T domains, i.e., $\boldsymbol{U} = \boldsymbol{U}^1 = \boldsymbol{U}^2 = \cdots = \boldsymbol{U}^T$, because each user may actively interact with all domains. For the **item set** $\boldsymbol{I}$, each domain has its own set and we denote the total item candidate pool as $\boldsymbol{I} = \boldsymbol{I}^1 \cup \boldsymbol{I}^2 \cup \cdots \cup \boldsymbol{I}^T$.

3 METHODOLOGY

Fig. 2 shows the overall architecture of H^{3}Trans. We introduce the construction of multi-domain graph, and basic graph neural network in subsections 3.1 and 3.2. Two core modules, namely dynamic item transfer and adaptive user aggregation, compose a hierarchical hypergraph neural network, detailed in subsection 3.3.1 and 3.3.2. Finally, the training procedure and optimization strategy will be introduced in subsection 3.4.

3.1 Unified Multi-domain Graph

To improve recommendation performance in all domains, instead of constructing individual graph for each domain, we integrate users' multi-domain behaviors into a unified graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In details, the node set $\mathcal{V}$ consists of user nodes and item nodes, i.e., $\mathcal{V} = \mathcal{U} \cup \mathcal{I}$. For **user nodes**, considering the domain discrepancy and the diversity of users' multi-domain behaviors, it is necessary to acquire separate representations for different domains. Thus we regard each user as separate nodes positioned in different domains (these nodes share the same attributes). Specifically, for a given user $u \in \boldsymbol{U}$, it corresponds to T nodes $(u^1, u^2, \cdots, u^T)$, thus the relation between user node set size $|\mathcal{U}|$ and user ID set size $|\boldsymbol{U}|$ meets the condition of $|\mathcal{U}| = |\boldsymbol{U}| \cdot T$. Each user node representation characterizes user's preference under a specific domain. For **item nodes**, items' properties are relatively static than users. Thus we treat each item $i \in \boldsymbol{I}$ as a single node across various domains. In other words, each item i only corresponds to one node in the graph. The item i's node is also denoted as i.

The **basic edge set** collects the user-item history interactions from all domain, i.e. , $\mathcal{R} = (\mathcal{R}^1, \mathcal{R}^2, \cdots, \mathcal{R}^T)$, where $\mathcal{R}^m$ denotes the user-item interaction matrix of domain $\mathcal{D}_m$. This work considers click behavior as the interaction type. For an entry $r_{ui}^m = 1$, it means that the user u has interacted with the item i under domain $\mathcal{D}_m$, and we build an interaction edge between the corresponding user node u^m and item node i, denoted as $e(u^m, i)$. To clarify which domain the edges belong to, we utilize distinct edge types for different domains. For domain $\mathcal{D}_m$, the edge subset is denoted as $\mathcal{E}^m$, and the whole edge set is the union of all domains, i.e., $\mathcal{E} = \mathcal{E}^1 \cup \mathcal{E}^2 \cup \cdots \cup \mathcal{E}^T$.

With access to user-item interactions in any domain, it's convenient to additionally leverage hyperedges to build cross-domain relations and capture the correlative knowledge during transfer.

3.2 Basic Graph Neural Network

Based on the unified multi-domain graph, we employ basic GNN that applies neighborhood aggregation scheme to obtain expressive feature representation for nodes. The basic GNN includes four modules: (1) an embedding module that transforms nodes' sparse attribute features into low-dimensional embedding vectors; (2) a message-passing module with several layers that refine node representations by aggregating information from neighbors; (3) a readout module that generates nodes' final representation; (4) a prediction module that generates the prediction score.

3.2.1 ***Embedding Module.*** This module maps each node into a d-dimensional embedding vector x_{u^m} (or x_i). For each user node $u^m \in \mathcal{U}$ (or item node $i \in \mathcal{I}$), we acquire its embedding x_{u^m} (or x_i) from a learnable embedding look-up table $X \in \mathbb{R}^{(|\mathcal{U}|+|\mathcal{I}|)\times d}$. Noted that each user corresponds to T nodes, and these nodes share the same initial embedding vector.

3.2.2 ***Message Passing Module.*** The message-passing module consists of several layers that follow the neighborhood aggregation scheme. It can be taken as a two-stage process to refine node representations by aggregating information from neighbors. The two stages are neighbor aggregation and node update:

Neighbor aggregation:

$$\begin{aligned} h_{\mathcal{N}_{u^m}}^{(l)} &= \mathrm{AGG_U}\left(\left\{h_i^{(l-1)} \mid i \in \mathcal{N}_{u^m}\right\}\right) \\ h_{\mathcal{N}_i}^{(l)} &= \mathrm{AGG_I}\left(\left\{h_{u^m}^{(l-1)} \mid u^m \in \mathcal{N}_i\right\}\right) \end{aligned} \tag{2}$$

Node update:

$$\begin{aligned} h_{u^m}^{(l)} &= \mathrm{UP_U}\left(h_{u^m}^{(l-1)}, h_{\mathcal{N}_{u^m}}^{(l)}\right) \\ h_i^{(l)} &= \mathrm{UP_I}\left(h_i^{(l-1)}, h_{\mathcal{N}_i}^{(l)}\right) \end{aligned} \tag{3}$$

where l denotes the l-th message passing layer. $h_{u^m}^{(l)}$ and $h_i^{(l)}$ refer to the hidden representation of user node u^m and item node i respectively. $\mathrm{AGG_U}$ and $\mathrm{AGG_I}$ are the aggregation functions for user and item nodes. The same is to the node update function $\mathrm{UP_U}$ and $\mathrm{UP_I}$. There are a lot of designs for aggregate and update function. Here we use mean pooling for the aggregator and linear transforming for node update. Noted that the initial representation is acquired from embedding module, i.e., $h_{u^m}^{(0)} = x_{u^m}, h_i^{(0)} = x_i$.

3.2.3 ***Readout Module.*** After obtaining L layers representations, we utilize a readout layer to generate the final representation:

$$z_v = \mathrm{Readout}\left(h_v^{(l)} \mid l \in [1, \ldots, L]\right), \tag{4}$$

where the subscript v can denote user node u^m or item node i. Common designs for the readout function include last-layer only, concatenation, and weighted sum. Here we adopt last-layer only.

3.2.4 ***Prediction Module.*** The prediction module produces the prediction score that how likely a user u would interact with item i under domain $\mathcal{D}_m$. It is formulated as:

$$\hat{r}_{u,i}^m = f(z_{u^m}, z_i) \tag{5}$$

where f is the score function and we usually adopt similarity function such as inner product and cosine function.

3.3 Hierarchical Hypergraph Network

Based on the unified multi-domain graph, we further utilize hyperedge to explore the high-order connections among users' multi-domain behaviors. In this section, we will introduce two core hyperedge-based modules: dynamic item transfer module (Hyper-I) & adaptive user aggregation module (Hyper-U). These two modules have a hierarchical connection structure and compose a hierarchical hypergraph neural network.

3.3.1 ***Hyper-I: Dynamic Item Transfer Module.*** In MDR, each domain contains a set of items that is related to the domain's topic and marketing strategy. Due to the intrinsic difference, directly transferring users' cross-domain behaviors from multiple sources domains to the target domain is not a good approach. It will introduce interference information and degenerate the user representations. To extract correlative preference from users' cross-domain behaviors for transfer, we design a dynamic item transfer module, namely Hyper-I. It dynamically adjusts the source item representations during transfer to be more relevant to a given target domain, that contributes to capturing correlative user preferences from source domains.

Take domain $\mathcal{D}_t$ as target domain, and the others as source domains. For each source domain $\mathcal{D}_s$, before feeding item node hidden representation $h_i^{(l)}$ into message passing layers that acquire user node representation by aggregating information from neighboring items, we adjust the item representations to eliminate domain discrepancy. Specifically, for each user's interacted item under source domain $\mathcal{D}_s$, we seek out similar items from the target domain $\mathcal{D}_t$, and then construct a hyperedge (named **hyperedge-i**) to connect these item nodes. This hyperedge contains a two-level relationship. The first level is that the interacted source item is related to the picked target items. The second level is that the picked target items are also related to each other. We first introduce the method to seek out the related target items, and then we design a structure-aware hypergraph layer to adjust item representations.

Hyperedge Construction. For a given interacted item i in a source domain $\mathcal{D}_s$, we seek out a similar item set $\mathcal{S}_i^t$ from the target domain $\mathcal{D}_t$, and construct a hyperedge to connect the source item node i and the item nodes of picked item set $\mathcal{S}_i^t$. We offer two ways to get similar items: path-based and embedding-based.

- Path-based: Utilize co-occurrence relation among items. We have such an assumption: if there is a user u that clicked on both items i and j, then the two items are similar. Based on this assumption, we design a walk path $(i \to u^s \to u^t \to j)$ and sample k items from the item set $\mathcal{I}^t$ of target domain as similar items.
- Embedding-based: Path-based method is an intuitive way but it seriously relies on the interaction history of users. Embedding-based method makes use of the hidden representation of items $h_i^{(l-1)}$. It leverages the appropriate nearest neighbor algorithm to find the top-k similar items from the target domain, where the source item node i is query and $\mathcal{I}^t$ is candidate set.

Graphormer Layer. To perform message passing within the hyperedge, UniGNN [8] and AllSet [4] propose a message passing paradigm on the hypergraph. UniGNN rethinks the message-passing layer of the basic GNN as a two-stage aggregation process. In the first stage, for each hyperedge, use a permutation-invariant function to aggregate the information of the nodes within it. In the second stage, update each node with its incident hyperedges using another aggregating function. The method of AllSet is similar.

We claim that the above message-passing paradigm fails to model the two-level relationship within hyperedge-i. Instead, we employ attention mechanism [18] to adjust the item representation. Moreover, to effectively exploit the two-level relations and leverage the topology structure within the hyperedge-i, we introduce the distance matrix of the shortest path among the picked nodes (denoted as $\boldsymbol{B}$) into the attention layers, as introduced in [25]. Fig. 2 illustrates the details of this module. Specifically,

$$h_i^{(l-1)} \leftarrow \mathrm{GH}_{\mathrm{HyperI}}\left(\mathrm{Concat}\left(h_i^{(l-1)}, \left\{h_j^{(l-1)} \mid | j \in \mathcal{S}_i^t\right\}\right)\right)[0] \tag{6}$$

where $\mathrm{GH}_{\mathrm{HyperI}}(\cdot)$ is the graphormer layer for Hyper-I module:

$$\begin{aligned} \mathrm{GH}_{\mathrm{HyperI}}(H_{\mathrm{I}}) &= \mathrm{Concat}\left(\mathrm{Attn}_{\mathrm{I},1}(H_{\mathrm{I}}), \cdots, \mathrm{Attn}_{\mathrm{I},P}(H_{\mathrm{I}})\right) W_{\mathrm{I}}^{O}, \\ \mathrm{Attn}_{\mathrm{I},p}(H_{\mathrm{I}}) &= \mathrm{softmax}\left(\frac{Q_{\mathrm{I},p} K_{\mathrm{I},p}^{\top}}{\sqrt{d_{h_i}/P}} + \Phi(\boldsymbol{B})\right) V_{\mathrm{I},p}, \\ Q_{\mathrm{I},p} = H_{\mathrm{I}} W_{\mathrm{I},p}^{Q}, &K_{\mathrm{I},p} = H_{\mathrm{I}} W_{\mathrm{I},p}^{K}, V_{\mathrm{I},p} = H_{\mathrm{I}} W_{\mathrm{I},p}^{V} \end{aligned} \tag{7}$$

here $\Phi()$ is a learnable function shared across all layers that maps the distance between every paired nodes to a scalar. W_*^Q, W_*^K, W_*^V, and W_*^O are training parameters.

3.3.2 *Hyper-U: Adaptive User Aggregation Module.*

After adjusting item representations with Hyper-I, we acquire the representations of separate user nodes by aggregating adjusted representation of their neighbor items. Each user corresponds with multiple nodes that characterize the user's domain-specific preference. Next step is to transfer correlative user preferences from source domains to the target and refine the user representation of target domain.

Noted that the preference transfer in MDR involves more than one source. The key point is how to aggregate users' scattered preferences in multiple domains and adequately exploit the high-order connections among them. Here we integrate a hyperedge-based module: Hyper-U, to realize adaptive user aggregation.

Hyperedge Construction. We utilize hyperedge to connect nodes that belong to the same user, and we name this hyperedge as **hyperedge-u**. Within hyperedge-u, each separate node representation characterizes user's interest preference under a specific domain. The hyperedge-u connects these separate user nodes and bridges the information propagation across domains, thus realizing adaptive preference transfer. Moreover, benefiting from that hyperedge connects plural nodes, hyperedge-u can further exploit the high-order (more than pairwise) connections among multiple domains.

Multi-head Attention Layer. We design a new message passing layer for the hyperedge-u to replace the original layer. For the l-th layer, we first acquire user's separate representations under multiple domains, denoted as $[h_{u^1}^{(l)}, h_{u^2}^{(l)}, \cdots, h_{u^T}^{(l)}]$. Hyper-U module take these separate representations as input, and then refine these representations by aggregating users' scattered preferences and transferring knowledge from other domains. Considering the domain discrepancy and diversity of users' multi-domain behaviors, we employ self-attention mechanism in the Hyper-U module to adaptively fuse users' cross-domain interest representations. To refine representation for domain t after the Hyper-U module,

$$\left[h_{u^1}^{(l)}, h_{u^2}^{(l)}, \cdots, h_{u^T}^{(l)}\right] \leftarrow \mathrm{MHA}_{\mathrm{HyperU}}\left(\left[h_{u^1}^{(l)}, h_{u^2}^{(l)}, \cdots, h_{u^T}^{(l)}\right]\right), \tag{8}$$

where $\mathrm{MHA}_{\mathrm{HyperU}}(\cdot)$ denotes the multi-head attention layer:

$$\begin{aligned} \mathrm{MHA}_{\mathrm{HyperU}}(H_{\mathrm{U}}) &= \mathrm{Concat}\left(\mathrm{Attn}_{\mathrm{U},1}(H_{\mathrm{U}}), \cdots, \mathrm{Attn}_{\mathrm{I},P}(H_{\mathrm{U}})\right) W_{\mathrm{U}}^{O}, \\ \mathrm{Attn}_{\mathrm{U},p}(H_{\mathrm{U}}) &= \mathrm{softmax}\left(\frac{Q_{\mathrm{U},p} K_{\mathrm{U},p}^{\top}}{\sqrt{d_{h_u}/P}}\right) V_{\mathrm{U},p}, \\ Q_{\mathrm{U},p} = H_{\mathrm{U}} W_{\mathrm{U},p}^{Q}, &K_{\mathrm{U},p} = H_{\mathrm{U}} W_{\mathrm{U},p}^{K}, V_{\mathrm{U},p} = H_{\mathrm{U}} W_{\mathrm{U},p}^{V} \end{aligned} \tag{9}$$

here W_*^Q, W_*^K, W_*^V, and W_*^O are trainable parameters. The multi-head attention layer takes users' separate nodes representations as input and exploits the high-order connections with the self-attention mechanism. For each domain, the corresponding node can adaptively refine its preference representation by extracting the correlative information from other domains.

3.4 Model Optimization and Time Complexity

These two hyperedge-base modules: dynamic item transfer module (Hyper-I) and adaptive user aggregation module (Hyper-U), compose a hierarchical hypergraph neural network. It realizes correlative preference transfer and exploits the high-order connection among users' multi-domain behaviors.

For model optimization, we mix the multi-domain data and randomly select a sample (u^m, i) from domain $\mathcal{D}_m$ for each training step. Domain $\mathcal{D}_m$ is taken as the target domain and the others are source domains. Moreover, we employ a contrastive loss InfoNCE [17] to learn more effective representations, which maximizes the agreements between positive pairs. Formally,

$$\mathcal{L}(u, i \mid \mathcal{D}_m) = -\log \frac{\exp(sim(z_u^m, z_i)/\tau)}{\sum_{i_-} \exp(sim(z_u^m, z_{i_-})/\tau)} \tag{10}$$

where $sim(\cdot)$ stands for similarity measure function and we use inner product. (u^m, i_-) is a randomly sampled negative pair that $r_{u,i_-}^m = 0$, and τ is the temperature hyperparameter.

The time complexity of Hyper-U module is $O(T^2 d + T d^2)$, where T is domain number and d is embedding dim. For Hyper-I module, the time complexity is $O(Tn(k^2 d + k d^2))$, where n is the number of sampled neighbors and k is the size of similar item set. The main limitation of $\mathsf{H}^3\mathsf{Trans}$ is computation cost and memory cost (incorporating hyperedges). Compared to the baselines that trains models for multiple domains in parallel, $\mathsf{H}^3\mathsf{Trans}$ unifies all domain data and training time increases. In future work, we shall focus on efficient algorithms, i.e., reducing memory cost via hyperedge dropout and reducing time complexity via accelerating self-attention.

4 EXPERIMENTS

In this section, we conduct both offline and online experiments to validate the effectiveness of our method. And the experiments are intended to answer the following research questions:

- **RQ1**: How does our proposed method perform when compared with other state-of-the-art GNN-based methods?
- **RQ2**: How do the different components (e.g., unified multi-domain graph, adaptive user aggregation module, dynamic item transfer module) contribute to the model performance?
- **RQ3**: Does our method help alleviate the behavior sparseness issue and improve recommendation performance for the relatively inactive users (with fewer interaction items)?

Table 1: Dataset Statistics

Product Dataset				Public Amazon Dataset			
Domains	#user	#item	#click	Domains	#user	#item	#click
MDR-A	84.6M	6.3M	3.1B	Books	1.67M	0.99M	26.8M
MDR-B	34.0M	1.4M	0.6B	Music	0.11M	0.12M	1.5M
MDR-C	24.7M	0.5M	0.3B	Movie	0.23M	0.08M	3.1M
MDR-D	29.1M	0.6M	0.2B	-	-	-	-

- **RQ4**: Does $\mathsf{H}^3\mathsf{Trans}$ achieve improvement when deployed to our advertising system?

4.1 Experimental Settings

4.1.1 ***Datasets***. We conduct extensive offline experiments on both the public dataset and the product dataset.

Public Dataset: Amazon dataset [16] is a popular dataset to conduct experiments of multi-domain recommendation. The dataset provides dozens of domains and the frequently-used domains are Books, Movies and TV (Movie), and CDS and Vinyl (Music). Following existing research, we take binarize the ratings to 1 and 0 (the ratings higher or equal to 4 as positive and others as negative.) Besides, we filter the users and items with less than 5 interactions.

Product Dataset: The product dataset is collected from four real-world scenarios from an industry advertising platform, named MDR-A, MDR-B, MDR-C, and MDR-D. These four sub-datasets share the same user set and have overlapped items. Each subset consists of users' interacted items. We additionally filter the datasets to retain users/items with at least 5 interactions. Table 1 lists the statistics of both the product dataset and the public amazon dataset.

4.1.2 ***Compared methods***. We compare $\mathsf{H}^3\mathsf{Trans}$ with following strong baselines. Except for the base model, all baselines attempt to transfer information from other domains in different ways.

- **Base**. Base method constructs a user-item bipartite graph and trains models individually for each domain with its user behavior data.
- **PPGN**. PPGN [31] fuses the interaction information of multiple domains into a graph and shares the features of users learned from the joint interaction graph. Notes that one user only has one node within the joint graph.
- **MGNN**. MGNN [29] integrates users' multi-domain behaviors and constructs the unified multi-domain graph. Nodes belonging to the same user share the same attribute. MGNN learns domain-specific representation for user nodes.
- **PCRec**. PCRec [20] adopts a pre-training and fine-tuning diagram to transfer knowledge from the source domain to the target. Here we first pre-train a graph model on the joint graph and then fine-tune it on each domain.
- **BiTGCF**. BiTGCF [13] is proposed for dual-target recommendation. It connects common users of both domains as bridge and designs a feature transfer layer to realize the two-way transfer of knowledge across two domains. Here we randomly pick two domains to realize the combination layer.
- **BiTGCF+**. BiTGCF+ is an extended version of BiTGCF. Here we modify the feature transfer layer and extend it to multi-domain recommendation.

4.1.3 ***Evaluation Protocol***. We adopt the widely used leave-one-out evaluation method. Specifically, we take the last interaction from each user's interaction history as the test set, and the remaining are utilized for training. For users in the testing set, we follow the all-ranking protocol [22] to evaluate the top-K recommendation performance. For product dataset, we report the average HitRate@K (HR@K) and Mean Reciprocal Rank (MRR) on each domain. For public dataset, we report the HR@K and NDCG@K as these two metrics are more popular of public experiments.

4.1.4 ***Implementation Details***. We provide the implementation details of our proposed model and baselines. For fair comparison, each of graph neural network models has two layers, and the hidden embedding dimensions are set as [128, 64]. We sample $k = 20$ related items to build *hyperedge-i* in Hyper-I module. For model training, we set batch size $N = 512$ and adopt adam optimizer [11], where the learning rate is set to 0.01.

4.2 Performance Comparison (RQ1)

Table 2 and Table 3 present the experimental results of $\mathsf{H}^3\mathsf{Trans}$ compared with other baselines. From these two tables, we have the following observations.

- Base method performs poorly on all domains, which indicates that individually training model for each domain limits the recommendation performance in multi-domain recommendation.
- PPGN mixes the multi-domain data and constructs a joint graph for model training. As a result, it achieves large improvement in most domains. But it still has negative effects on some domains such as MDR-B, because different domains share the same user representation and neglect the user's domain-specific preferences. The user representation is dominated by the data-rich domain.
- MGNN takes account of both the common feature and the domain-specific feature for different domains. which brings improvement to the recommendation service. Note that common feature is only acquired by the shared node attributes. The information transfer among domains is limited.
- PCRec performs transfer learning by adopting the pre-training and fine-tuning diagram. Pre-training on the joint graph helps learn users' common preferences among domains. Then fine-tuning on domain's individual graph make the user node representation more preferable for each domain. However, fine-tuning is more time- and space-consuming for multi-domain recommendation.
- BiTGCF and BiTGCF+ are two competitive baselines in our experiments. BiTGCF leverages a combination layer to realize the two-way transfer across domains. Here we extend the feature transfer layer of BiTGCF to multiple domains as BiTGCF+. We can see that BitGCF+ achieves larger improvement than BitGCF because it introduces more domains to perform multi-domain recommendation. But the improvement is still limited because we just simply sum user's multi-domain representations and neglect the high-order connections among them.
- $\mathsf{H}^3\mathsf{Trans}$ achieves the best performance with significant improvement on all metrics of all domains. This indicates that $\mathsf{H}^3\mathsf{Trans}$ benefits from learning the high-order connections among multiple domains extracted by Hyper-U module and transferring correlative information via Hyper-I. The high-quality representations learned from the hypergraph enhance the recommendation performance in all domains.

Table 2: Main results on product dataset

Method	MDR-A			MDR-B			MDR-C			MDR-D		
	Mrr	HR@20	HR@50	Mrr	HR@20	HR@50	Mrr	HR@20	HR@50	Mrr	HR@20	HR@50
Base	0.0368	2.37%	6.46%	0.0625	4.87%	12.60%	0.0640	4.78%	13.20%	0.0753	5.11%	12.65%
PPGN (Mix)	0.0481	2.98%	8.47%	0.0603	4.22%	11.61%	0.1017	8.58%	18.99%	0.1131	7.60%	17.59%
MGNN	0.0544	3.68%	8.11%	0.0699	5.34%	14.28%	0.1079	12.22%	21.34%	0.1428	10.67%	21.81%
PCRec	0.0635	4.38%	9.71%	0.0845	7.31%	16.63%	0.1546	14.71%	25.99%	0.1738	15.16%	26.59%
BiTGCF	0.0663	4.59%	10.61%	0.0986	8.66%	18.46%	0.1591	15.48%	26.49%	0.1577	13.73%	23.66%
BiTGCF+	0.0750	5.08%	12.31%	0.1237	9.87%	20.71%	0.1713	16.15%	28.63%	0.1685	14.76%	25.85%
H^3Trans	**0.1171**	**7.20%**	**16.79%**	**0.1686**	**14.29%**	**28.65%**	**0.2084**	**18.78%**	**34.89%**	**0.2158**	**18.69%**	**32.73%**

Table 3: Main results on public amazon dataset

Method	Books		Music		Movie	
	NDCG	HR@20	NDCG	HR@20	NDCG	HR@20
Base	0.0270	4.71%	0.0631	13.39%	0.0433	10.45%
PPGN	0.0289	4.96%	0.0660	13.93%	0.0473	11.23%
MGNN	0.0311	5.12%	0.0672	14.14%	0.0465	11.03%
PCRec	0.0331	5.31%	0.0742	15.67%	0.0489	11.52%
BitGCF	0.0359	5.57%	0.0694	14.65%	0.0495	11.78%
BitGCF+	0.0381	5.78%	0.0719	15.29%	0.0509	12.02%
H^3Trans	**0.0399**	**5.97%**	**0.0761**	**16.01%**	**0.0524**	**12.33%**

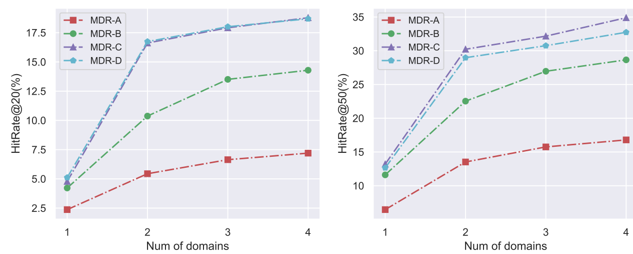

Figure 3: Performance comparison over different number of domains in MDR

4.3 Ablation Study (RQ2)

For further analysis, we compare different variants of H^3Trans on the product dataset for ablation study, and the results are listed in table 4. Vanilla is a basic graph model trained on the unified multi-domain graph. User nodes learn the common interest only through the shared node attributes.

4.3.1 ***Effect of Hyper-U module:*** HU adds the Hyper-U module but without the attention mechanism based layer. It only utilizes a vanilla combination layer to combine users' separate representations from multiple domains. HU+ integrates our self-attention mechanism based message passing layer into HU. From the table, we can see that aggregating users' scattered preferences and modeling the high-order connections among multiple domains could help refine the user representation for each separate domain. And the self-attention mechanism contributes to further improving the representation quality, because the attention layer adaptively extracts correlative knowledge from source domains.

4.3.2 ***Effect of Hyper-I module:*** PHI and EHI are two models that additionally integrate the Hyper-I module, and equipped with path-based or embed-based method to seek out similar items respectively. Table 4 shows that these two methods perform better than HU+, which indicates that the dynamic item transfer module could eliminate the domain discrepancy and adjust the latent item representation more correlative to the target domain without interference information. Besides, EHI achieves a marginal improvement than PHI, that shows embed-based method is a little better than path-based method. EHI+ is the best variant of our model, which further employs the graphormer layer to exploit the structure information within the hyperedge-i. It consistently shows around 1% on HR@20 and 2% on HR@50.

The itemset size of each domain ranges from tens-of-thousands to millions, while the size of selected correlative itemset is K. The value of K is a key hyperparameter: A too small value brings unstable training. A too large value increases computation cost, and different source items usually retrieve similar itemsets that lacks of discriminatory information.

4.3.3 ***Effect of multiple domains:*** Multi-domain recommendation jointly optimizes the recommendation performance of all domains. Intuitively, with more domains, we can access more users' behaviors to better characterize users' interest. Here we analyze the effect when introducing different numbers of domains to perform multi-domain recommendation. The results are reported in figure 3. We can see that it indeed achieves better performance when introducing more domains, because we can transfer knowledge from more source domains, and H^3Trans help exploit the high-order connections among them. Additionally, the marginal improvement decreases as more domains are introduced.

4.4 Alleviating Behavior Sparseness (RQ3)

As stated before, GNN-based methods suffer from the behavior sparseness issue, and here we conduct a detailed analysis to test the improvement on behavior-sparse users. Specifically, we split the users into four groups G1, G2, G3, G4, and G5 in the order of increasing number of interactions. The larger the GroupID is, the more behaviors the users have collected. Figure 4 reports the percentage increase compared with the Base model. We can find that the improvement achieved in the first three groups is more significant than that of the last two. We conclude that H^3Trans help improve more for relatively inactive users (with fewer user-item interactions), indicating that H^3Trans alleviates the sparseness of user behaviors by transferring knowledge from other domains.

Table 4: Ablation study on product dataset. Methods refer to different variants of H^{3}Trans.

Method	MDR-A			MDR-B			MDR-C			MDR-D		
	Mrr	HR@20	HR@50	Mrr	HR@20	HR@50	Mrr	HR@20	HR@50	Mrr	HR@20	HR@50
Vanilla	0.0544	3.68%	8.11%	0.0699	5.34%	14.28%	0.1079	12.22%	21.34%	0.1428	10.67%	21.81%
HU	0.0750	5.08%	12.31%	0.1237	9.87%	20.71%	0.1712	16.15%	28.63%	0.1685	14.76%	25.85%
HU+	0.0894	5.56%	13.68%	0.1383	10.53%	23.08%	0.1848	17.01%	29.82%	0.1846	16.38%	28.48%
PHI	0.1016	6.35%	15.22%	0.1509	11.96%	24.52%	0.1887	17.58%	30.92%	0.1913	17.21%	29.80%
EHI	0.1051	6.53%	15.68%	0.1581	12.34%	25.54%	0.1958	17.93%	31.64%	0.1937	17.84%	30.62%
EHI+	**0.1171**	**7.20%**	**16.79%**	**0.1686**	**14.29%**	**28.65%**	**0.2084**	**18.78%**	**34.89%**	**0.2158**	**18.69%**	**32.73%**

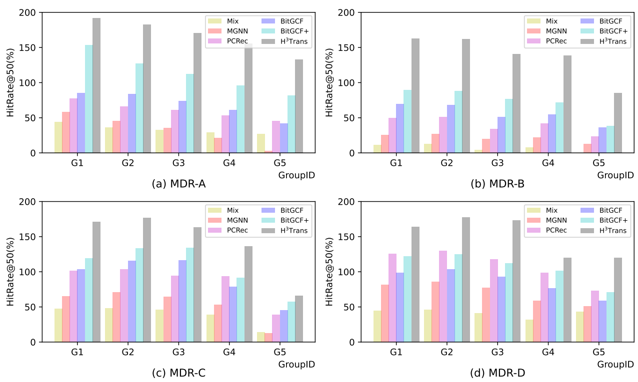

Figure 4: Performance comparison over different user groups (percentage increase relative to Base model)

4.5 Online Experiment (RQ4)

We have deployed H^{3}Trans online to the retrieval module of our advertising system for an emerging scenario, and conducted online A/B test for one week. For fair comparison, we follow the same configuration with the best retrieval model deployed online [20]. The online metrics include CTR, conversion rate (CVR), gross merchandise volume (GMV) and return on investment (ROI).

We observe that H^{3}Trans achieves **+2.8%** lift on CTR, **+10.9%** lift on CVR, **+6.7%** lift on GMV and **+7.3%** lift on ROI, and the daily improvement over baseline is stable. The uplift is mainly from users having lowest activity level, verifying that H^{3}Trans learns high-quality embeddings for inactive users through preference transfer. Therefore H^{3}Trans improves the important online metrics and promotes the performance to our system.

5 RELATED WORK

5.1 Multi-domain Recommendation

Multi-domain recommendation aims to improve recommendations performance of all domains by transferring knowledge from related domains. MCF [30] and ICAN [24] consider multiple collaborative filtering tasks in different domains simultaneously and exploit the relationships between domains. Ma et al. [15] further introduce cross-media content information. Some works focus on the users' multiple behaviors. MBGCN [10] and MGNN [29] propose a multi-behavior graph convolutional network to capture behaviors' different influences on target behavior. Furthermore, by considering each domain as a task, multi-task approaches can be directly applied in MDR. For general MDR, MMoE [14] models the tradeoffs between domain-specific objectives and inter-domain relationships with a new multi-gate expert strategy.

5.2 GNNs for Cross-domain Recommendation

Inspired by the success of graph neural networks[6, 12], researchers have taken efforts to exploit the user-item interaction behavior graph. GNN-based methods [7, 22, 26] suffer from the sparseness of user behaviors, and some researchers have exploited to alleviate it by transferring information from other domains [13, 20, 31]. PPGN [31] fuses the interaction information of two domains into a graph and learns shared features for users. Wang et al. [20] propose a pre-training and fine-tuning diagram to transfer information to the target domain. Liu et al. [13] realizes the two-way transfer of knowledge across two domains with a bi-directional feature transfer module. Zhu et al. [32] propose a graphical and attentional model to combine the embeddings of common users from both domains, thus enhancing the quality of user embeddings and improving the recommendation performance on each domain. However, they fail to model high-order connections among more domains.

5.3 Hypergraph Learning for Recommendation

Hypergraph, as a more general topological structure to model high-order connections, has been exploited in recommendation [2, 9, 21, 23, 27, 28]. Xia et al. [23] models session-based data as a hypergraph and then propose a hypergraph convolutional network for session-based recommendation. Yu et al. [27] propose a multi-channel hypergraph convolutional network to enhance social recommendation by leveraging high-order user connections. Zhang et al. [28] incorporate the complex tuple-wise correlations into a hypergraph and propose a self-supervised hypergraph learning framework for group recommendation. Our work is the first to investigate hypergraph learning in multi-domain recommendation, which can exploit the high-order connections among multiple domain and realize correlative preference transfer.

6 CONCLUSION

In this paper, we propose an correlative preference transfer framework with hierarchical hypergraph network (H^{3}Trans) to improve multi-domain recommendations. H^{3}Trans constructs a unified multi-domain graph and integrates two hyperedge-based module: adaptive user aggregation and dynamic item transfer. H^{3}Trans not only exploits high-order connections among users' scattered preferences in multiple domain, but also transfers correlative user preference to alleviate the behavior sparseness of each single domain. Extensive experiments demonstrate the superiority of our method.

REFERENCES

[1] Fengwen Chen, Shirui Pan, Jing Jiang, Huan Huo, and Guodong Long. 2019. DAGCN: dual attention graph convolutional networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[2] Xu Chen, Kun Xiong, Yongfeng Zhang, Long Xia, Dawei Yin, and Jimmy Xiangji Huang. 2020. Neural feature-aware recommendation with signed hypergraph convolutional network. *ACM Transactions on Information Systems (TOIS)* (2020).

[3] Yuting Chen, Yanshi Wang, Yabo Ni, An-Xiang Zeng, and Lanfen Lin. 2020. Scenario-aware and Mutual-based approach for Multi-scenario Recommendation in E-Commerce. In *2020 International Conference on Data Mining Workshops (ICDMW)*. IEEE, 127–135.

[4] Eli Chien, Chao Pan, Jianhao Peng, and Olgica Milenkovic. 2022. You are AllSet: A Multiset Function Framework for Hypergraph Neural Networks. *ArXiv* (2022).

[5] Yulong Gu, Wentian Bao, Dan Ou, Xiang Li, Baoliang Cui, Biyu Ma, Haikuan Huang, Qingwen Liu, and Xiaoyi Zeng. 2021. Self-Supervised Learning on Users' Spontaneous Behaviors for Multi-Scenario Ranking in E-commerce. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.

[6] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems* 30 (2017).

[7] 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*.

[8] Jing Huang and Jie Yang. 2021. UniGNN: a Unified Framework for Graph and Hypergraph Neural Networks. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI-21)*.

[9] Shuyi Ji, Yifan Feng, Rongrong Ji, Xibin Zhao, Wanwan Tang, and Yue Gao. 2020. Dual channel hypergraph collaborative filtering. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*.

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

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

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

[13] Meng Liu, Jianjun Li, Guohui Li, and Peng Pan. 2020. Cross Domain Recommendation via Bi-Directional Transfer Graph Collaborative Filtering Networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.

[14] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1930–1939.

[15] Weizhi Ma, Min Zhang, Chenyang Wang, Cheng Luo, Yiqun Liu, and Shaoping Ma. 2018. Your Tweets Reveal What You Like: Introducing Cross-media Content Information into Multi-domain Recommendation.. In *IJCAI*. 3484–3490.

[16] Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. 188–197.

[17] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv e-prints* (2018).

[18] 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).

[19] Bei Wang, Chenrui Zhang, Hao Zhang, Xiaoqing Lyu, and Zhi Tang. 2020. Dual Autoencoder Network with Swap Reconstruction for Cold-Start Recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2249–2252.

[20] Chen Wang, Yueqing Liang, Zhiwei Liu, Tao Zhang, and Philip S. Yu. 2021. Pre-training Graph Neural Network for Cross Domain Recommendation. *CoRR* abs/2111.08268 (2021). arXiv:2111.08268

[21] Jianling Wang, Kaize Ding, Liangjie Hong, Huan Liu, and James Caverlee. 2020. Next-item recommendation with sequential hypergraphs. In *Proceedings of the 43rd international ACM SIGIR conference on research and development in information retrieval*.

[22] 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.

[23] Xin Xia, Hongzhi Yin, Junliang Yu, Qinyong Wang, Lizhen Cui, and Xiangliang Zhang. 2021. Self-supervised hypergraph convolutional networks for session-based recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

[24] Ruobing Xie, Zhijie Qiu, Jun Rao, Yi Liu, Bo Zhang, and Leyu Lin. 2020. Internal and Contextual Attention Network for Cold-start Multi-channel Matching in Recommendation.. In *IJCAI*. 2732–2738.

[25] Chengxuan Ying, Tianle Cai, Shengjie Luo, Shuxin Zheng, Guolin Ke, Di He, Yanming Shen, and Tie-Yan Liu. 2021. Do transformers really perform badly for graph representation? *Advances in Neural Information Processing Systems* 34 (2021), 28877–28888.

[26] 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.

[27] Junliang Yu, Hongzhi Yin, Jundong Li, Qinyong Wang, Nguyen Quoc Viet Hung, and Xiangliang Zhang. 2021. Self-supervised multi-channel hypergraph convolutional network for social recommendation. In *Proceedings of the Web Conference 2021*.

[28] Junwei Zhang, Min Gao, Junliang Yu, Lei Guo, Jundong Li, and Hongzhi Yin. 2021. Double-Scale Self-Supervised Hypergraph Learning for Group Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*.

[29] Weifeng Zhang, Jingwen Mao, Yi Cao, and Congfu Xu. 2020. Multiplex Graph Neural Networks for Multi-Behavior Recommendation. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*.

[30] Yu Zhang, Bin Cao, and Dit-Yan Yeung. 2010. Multi-domain collaborative filtering. In *Proceedings of the Twenty-Sixth Conference on Uncertainty in Artificial Intelligence*. 725–732.

[31] Cheng Zhao, Chenliang Li, and Cong Fu. 2019. Cross-Domain Recommendation via Preference Propagation GraphNet. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*.

[32] Feng Zhu, Yan Wang, Chaochao Chen, Guanfeng Liu, and Xiaolin Zheng. 2020. A Graphical and Attentional Framework for Dual-Target Cross-Domain Recommendation.. In *IJCAI*. 3001–3008.