All-in-One: Heterogeneous Interaction Modeling for Cold-Start Rating Prediction

Shuheng Fang*, Kangfei Zhao†, Yu Rong‡, Zhixun Li* and Jeffrey Xu Yu*
*The Chinese University of Hong Kong, {shfang, yu}@se.cuhk.edu.hk, lizhixun1217@gmail.com
†Beijing Institute of Technology, zkf1105@gmail.com
‡Alibaba DAMO Academy, yu.rong@hotmail.com

***Abstract*—Cold-start rating prediction is a fundamental problem in recommender systems that has been extensively studied. Many methods have been proposed that exploit explicit relations among existing data, such as collaborative filtering, social recommendations and heterogeneous information network, to alleviate the data insufficiency issue for cold-start users and items. However, the explicit relations constructed based on data between different entities may be unreliable and irrelevant, which limits the performance ceiling of a specific recommendation task. Motivated by this, in this paper, we propose a flexible framework dubbed heterogeneous interaction rating network (HIRE). HIRE does not solely rely on pre-defined interaction patterns or a manually constructed heterogeneous information network. Instead, we devise a Heterogeneous Interaction Module (HIM) to jointly model heterogeneous interactions and directly infer the important interactions via the observed data. In the experiments, we evaluate our framework under 3 cold-start settings on 3 real-world datasets. The experimental results show that HIRE outperforms other baselines by a large margin. Furthermore, we visualize the inferred interactions of HIRE to reveal the intuition behind our framework.**

***Index Terms*—Cold-start rating prediction, Heterogeneous interaction, Attention**

I. Introduction

Rating prediction is a fundamental task in recommender systems for estimating users' preference scores on items, which has widespread applications [1]–[5] across e-commerce, online education, and entertainment platforms. In cold-start scenarios, where new users/items have limited information for prediction, accurate rating prediction is crucial for effective user retention (in the case of cold users) and item promotion (in the case of cold items). The scarcity of ratings and interactions in cold-start scenarios poses significant challenges, as current recommendation models heavily rely on observed ratings and interactions to make predictions. To improve prediction accuracy, current solutions incorporates explicit additional associations and side information to enrich the features of cold users and items. These approaches can be generally categorized into three lines of research.

Collaborative filtering (CF) [6]–[12], the most traditional approach, assumes similar users and items exhibit similar ratings and exploits the user-item interactions to make the rating prediction. However, CF models are limited to process single-type interactions and fail to generalize well to cold users/items with scarce rating interactions. Social recommendation approaches [13]–[19] extend the scope of interactions and introduce social connections into the recommendation framework. These approaches are on the basis that users' preferences are highly influenced by their social communities and opinion leaders within social networks. However, the effectiveness of such approaches is constrained by the availability and reliability of social relations. Furthermore, recent research has explored the integration of complex interactions through Heterogeneous Information Networks (HINs), which incorporate entities beyond users and items, such as tags and points of interest (POIs). While HINs promise richer semantic information and enhance the representations of users and items, HIN-based approaches rely on pre-defined graph structures and are therefore vulnerable to the noise and mistakes introduced during graph construction. They also suffer from the issue of irrelevance, similar to social recommendation approaches. Fig. 1 illustrates the explicit interactions in the system and the challenge of effectively exploiting them.

† Corresponding author.

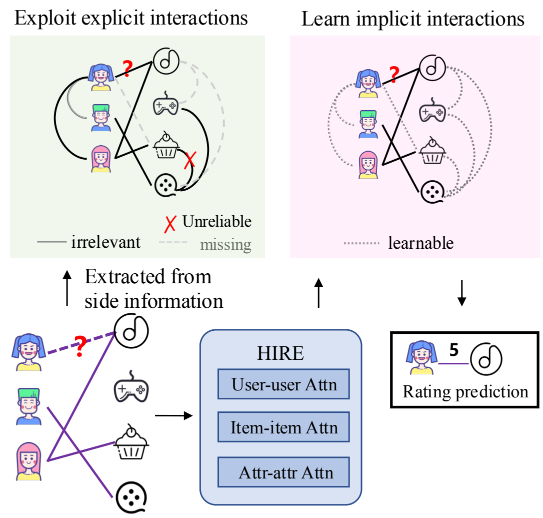


Fig. 1: An illustrative example: to predict the first girl's preference for music, relation between the girl and a boy seems to be irrelevant. The erroneous classification for cake (food) and movie (entertainment) is unreliable for rating prediction. HIRE directly models heterogeneous interactions in a data-driven way.

To address the issue of cold users/items, recent studies [20]–[22] employ meta learning to accommodate cold-start scenarios. MeLU [22] utilizes model-agnostic meta-learning (MAML) [23] to develop a personalized model for each user based on its preferences for a limited number of items. TaNP [20] develops a task-adaptive mechanism for learning the relevance of different users. MAMO [21] devises user preference memory to capture shared preferences. Despite these advances, these approaches capture only explicit interactions between users rating the same items, missing implicit relationships. In addition, their complicated optimization or memorization strategies in meta-learning-based recommendation will incur the barrier of implementations.

To summarize, the homogeneity, unreliability and irrelevancy introduced by the explicit interactions present significant obstacles for existing recommendation approaches, especially for the cold-start scenarios. This motivates us to reconsider *how a rating model can automatically infer heterogeneous interactions and how rating prediction can be supported in an end-to-end fashion.*

In this paper, we propose a deep learning framework named Heterogeneous Interaction Rating Network (HIRE) for rating prediction in cold-start scenarios. Distinguished from existing approaches we discussed, HIRE does not rely on side information in the data, but models the heterogeneous interactions for rating prediction in a fully data-driven way. Specifically, the core component of our HIRE model is the Heterogeneous Interaction Module (HIM), which jointly learns heterogeneous interactions from a prediction context of the cold users/items. HIM consists of three multi-head self-attention (MHSA) layers, the key building block of foundation models that exhibits powerful modeling capability of interactions in NLP tasks. However, in contrast to the usage of textual sequence, we leverage this interaction modeling capability of MHSA to capture the correlations between the users, items and attributes in the prediction context. The inherent metric learning idea of MHSA is beneficial for cold-start scenarios where the target users/items have only a few interactions. For ratings of user-item pairs to be predicted, to construct a prediction context, we design a neighborhood-based sampling strategy to select relevant users and items from the user-item rating bipartite graph. This provides an initial prior that is informative for predicting the ratings of cold users and items. The model is trained by learning the underlying heterogeneous interactions from multiple sampled prediction contexts to predict masked ratings. Our accurate prediction results demonstrate the promise of holistically modeling heterogeneous interactions using attention layers for cold-start recommendation tasks. The contributions of this paper are summarized as follows:

- We propose a novel learning framework, HIRE, for the cold-start rating prediction in recommendation systems. Instead of explicitly exploiting external interaction information, HIRE learns the interactions from heterogeneous sources in a holistic and data-driven fashion, and can deal with 3 typical cold-start scenarios, i.e., user cold-start, item cold-start and both user and item cold-start.
- We design a Heterogeneous Interaction Module (HIM), powered by multi-head self-attention. HIM learns the interaction in the perspective of users, items, and the multi-category attributes from a prediction context, which are composed of a set of users and items. We devise a sampling-based, simple yet effective context construction strategy and prove that HIM is permutation equivariant to the order of the users or items in the prediction context.
- We conduct substantial experimental studies on 3 real-world datasets in the 3 cold scenarios to verify the effectiveness of HIRE. Compared with 12 baselines, our HIRE outperforms others with higher Precision, NDCG and MAP by 0.21, 0.29 and 0.22 on average. We conduct case studies, and show that HIM provides potential interpretive ability for rating prediction.

II. Related Work

CF-based approaches. Collaborative Filtering (CF) is a classical approach in recommendation systems. In recent years, CF is enhanced by deep learning, where neural networks enable feature fusion from auxiliary information, e.g., user profiles and item descriptions. NCF [8] leverages multi-layer perceptron to capture non-linear feature interaction between user and item and can generalize matrix factorization. Wide&Deep [24] incorporates wide linear models and deep neural networks to fulfill memorization of sparse feature interaction and generalization to unseen user and item features. DeepFM [25] integrates factorization machine and neural networks for jointly modeling the low-order and high-order user-item feature interactions. AFN [26] introduces a logarithmic transformation layer with neural networks to model arbitrary-order feature interactions for rating prediction. The CF-based approaches only model the interactions of user and item features, and are not specifically tailored for cold-start recommendation.

Social recommendation approaches. Social recommendation relies on additional relationships [27]–[29] from social media to enhance feature interactions. TrustWalker [16] proposes a random walk model to combine the interactions in trust networks among users with item-based collaborative filtering. MCCP [12] designs the random walk process on user-item bipartite graph to simulate the preference propagation to alleviate the data sparsity problem for cold-start users. LOCABLE [17] exploits social contents from both local user friendship and global ranking of user reputation, i.e., PageRank. SoRec [18] jointly factorizes user-item rating matrix and user-user social relation matrix. GraphRec [15] is a GNN framework that jointly models the social influence in a user-user graph and the rating opinions in a user-item graph. To better learn user and item embeddings for recommendation, DiffNet [14] simulates the social influence propagation in a social network by an influence diffusion neural network model. Leveraging social information alleviates data scarcity and sparsity in cold-start scenarios. However, these approaches rely on explicit and external social information, which can be incomplete and inaccurate.

HIN-based approaches. Similar to social recommendation approaches, HIN-based approaches further utilize heterogeneous interactions from HINs. In early studies, meta-path or meta-graph based user and item similarity matrices are utilized for matrix factorization [30], [31] and collaborative filtering [32]. GraphHINGE [33] designs a neighborhood-based interaction model to enhance the user and item representations, where the neighbors are selected by meta-paths [34]. Similarly, NI-CTR [35] leverages neighborhood interactions in 4 types of HIN to assist the CTR prediction. To address cold-start problem, MetaHIN [36] exploits the rich semantics in HINs at the data level and meta-learning at the model level. HIN-based approaches need to collect and construct HINs with manually defined heterogeneous patterns, i.e., meta-paths or meta-graphs for any specific dataset. In contrast, our approach HIRE learns the heterogeneous interactions in a generic and data-driven way.

Cold-start recommendation. Meta-learning [23], [37]–[39] aims to learn a meta model that can rapidly adapt to new tasks with a few training samples. Recently, meta-learning is used for cold-start recommendation where predication for one user is treated as a task. MeLU [22] adopts model-agnostic meta learning (MAML) [23] to learn a personalized model for each user given her preference on a few items. MAMO [21] also adopts MAML to learn a global parameter initialization for all users. In addition, a feature-specific memory module and a task-specific memory module are used to guide parameter personalization and fast prediction of user preference. TaNP [20] learns a Neural Process [40], an encoder-decoder model architecture for meta-learning, where the decoder is equipped with a task-adaptive mechanism for task-specific adaptation. These proposed meta model only explicitly model the interaction of user-item pairs, where the potential interactions among users are derived from model adaptation.

Recently, Transformer-based methods [41]–[43] are proposed for capturing user-item interactions adaptively and globally. TransGNN [44] integrates Transformer and GNN layers to enhance the Transformer's performance. Following the scaling law of Transformers, Large language models (LLMs) have presented strong zero-shot generalization ability in cold-start recommendation [45]–[47]. However, LLMs for recommendation require user and item possessing rich semantic descriptions as extra knowledge. Furthermore, the performance of LLMs degenerates in numerical prediction tasks.

III. Preliminary

We formulate the problem of cold-start rating prediction, and introduce multi-head self-attention based on which HIRE is developed.

A. Problem Statement

A recommendation system has a set of users $\mathcal{U} = \{u_1, \cdots, u_M\}$ and a set of items $\mathcal{I} = \{i_1, \cdots, i_N\}$. The users and items are associated with categorical attributes, i.e., the user profiles and item descriptions. Here, we use $\boldsymbol{e}_u^k$ and $\boldsymbol{e}_i^k$ to denote the k-th categorical attribute of a user u and an item i, respectively, which are represented by one-hot encoding. We also use r_{ui} to denote the observed rating (preference) of user u to item i. Given a set of observed ratings, $\mathcal{R}$, rating prediction is to predict unknown rating for pairs of users and items.

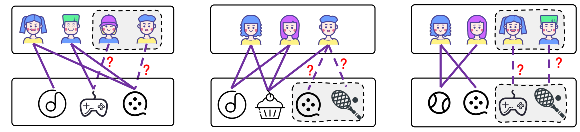

(a) User Cold-Start (b) Item Cold-Start (c) User&Item Cold-Start

Fig. 2: 3 cold-start scenarios for rating prediction: Entities in dotted boxes are the cold-start entities.

Cold-start rating prediction is to predict cold-start users or/and items. There are 3 scenarios:

- *User Cold-Start:* The user u^* is new arrival, i.e., $u^* \notin \mathcal{U}$, which has only a few rating interactions with existing items in $\mathcal{I}$.
- *Item Cold-Start:* The item i^* is new arrival, i.e., $i^* \notin \mathcal{I}$, which has only a few rating interactions with existing users in $\mathcal{U}$.
- *User&Item Cold-Start:* Both the user u^* and the item i^* are new arrivals, i.e., $u^* \notin \mathcal{U}$, $i^* \notin \mathcal{I}$. They have only a few rating interactions with existing and other possible new items/users in the system.

Fig. 2 shows an example of the 3 cold-start scenarios in a product rating system, where entities in the dotted boxes are cold-start entities. In Fig. 2(a), it shows a User cold-start scenario to predict two new users' ratings on three existing items, in Fig. 2(b), it shows an Item cold-start scenario to predict existing users' preference on two cold items (e.g., movie and tennis racket), and in Fig. 2(c), it shows a User&Item cold-start scenario to predict the ratings of two new users on 2 new items (e.g., game and tennis racket).

Given the user set $\mathcal{U}$, item set $\mathcal{I}$ and the observed rating $\mathcal{R}$, in this paper, we build a deep learning framework to support above cold-start rating prediction. It is important to note that by cold-start, the cold user and item to be predicted, as well as their associated ratings, are unavailable in the model training stage.

B. Multi-Head Self-Attention

Multi-head self-attention (MHSA) [48] is the building block of foundation models, such as BERT [49], GPT [50], Vision Transformer [51], achieving high performance in various natural language processing and computer vision tasks. Given a sequence of tokens $X = [x_1, \cdots, x_t] \in \mathbb{R}^{t \times d}$ as input, self-attention models the interactions among the tokens by computing the attention weights $A \in \mathbb{R}^{t \times t}$ as Eq. (1)-(2). Here, $W_Q \in \mathbb{R}^{d \times d^k}$, $W_K \in \mathbb{R}^{d \times d^k}$ are linear weights that map X to query matrix $Q \in \mathbb{R}^{t \times d^k}$ and key matrix $K \in \mathbb{R}^{t \times d^k}$,

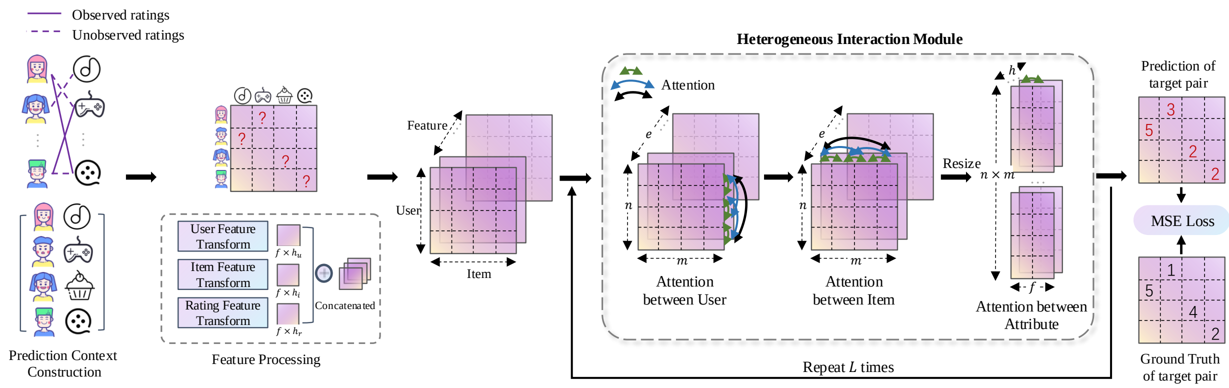


Fig. 3: The architecture of HIRE:For ratings of cold user/items to be predicted, HIRE first samples a prediction context. Then, the model constructs the initial embedding for the prediction context and learns the heterogeneous interactions via L HIMs. Finally, the output embedding of the last HIM is transformed into the predicted rating matrix.

respectively, A_{ij} indicate the similarity between token x_i and x_j, and d^k and d^v denote the dimensionality of the keys and values,

$$Q = XW_Q, K = XW_K, \tag{1}$$

$$A = \text{softmax}(\frac{QK^T}{\sqrt{d^k}}), \tag{2}$$

Self-attention (SA) fuses the embedding of the input tokens by taking A as the aggregation weights, where $W_V \in \mathbb{R}^{d \times d^v}$ is a linear weight matrix,

$$\text{SA}(X) = AXW_V. \tag{3}$$

MHSA computes l_a self-attention independently and concatenates the results as Eq. (4), where the weight $W_O \in \mathbb{R}^{l_a \cdot d^v \times d^o}$ maps the concatenated dimension $l_a \cdot d^v$ to the output dimension d^o and $\|$ is the concatenation operation,

$$\text{MHSA}(X) = [\text{SA}^{(1)}(X) \| \cdots \| \text{SA}^{(l_a)}(X)]W_O. \tag{4}$$

It is worth noting that MHSA is equivariant w.r.t. the permutation of the tokens. Given a permutation function $\Pi_{[1:t]}$ that shuffles the t tokens by an arbitrary order, we have:

$$\Pi_{[1:t]} \circ \text{MHSA}(X) = \text{MHSA}(\Pi_{[1:t]} \circ X). \tag{5}$$

Our model adopts MHSA as the unique building block, which indicates the power of deep attention layers in modeling heterogeneous relationships for recommendation tasks.

IV. Methodology

We propose a DL framework, Heterogeneous Interaction Rating Network (HIRE), for cold-start rating prediction. In this section, we present an overview of HIRE, followed by details on the input construction and the model architecture.

A. Overview

The gist of HIRE is to use a simple yet effective mechanism, MHSA, to holistically learn the interactions in three levels, i.e., users, items, and attributes, in an end-to-end, and data-driven fashion. The overall architecture of HIRE is illustrated in Fig. 3. To predict the rating of a user on an item, where either the user or item or both is cold-start, first we generate a prediction context which is composed of the target user/item, together with a set of pertinent users and items. These pertinent users and items are sampled from the user-item bipartite graph by leveraging the rating interaction. We envision that such users and items randomly selected by explicit rating relationships are informative for the cold-start user/item. Second, for a sampled prediction context, we construct a context matrix as the input of the model, which is a 3-dimensional tensor account for the users, items and attributes. The attributes contain individual attributes of users and items, and the observed ratings between the users and items in the context, where the ratings to be predicted are masked. The context matrix is transformed by L Heterogeneous Interaction Module (HIM) blocks, the main component of the HIRE model, followed by a decoder to generate the prediction matrix. In each HIM block, there are three MHSA layers that learn the interactions between different users, different items and different categories of attributes, respectively. Intuitively, as Fig. 4 shows, the three MHSA layers learn via message passing in three complete graphs, where the nodes are the users, items in the context, and the involved attributes, respectively. In the rest of this section, we will elaborate on the construction of the prediction context and the HIM in § IV-B and § IV-C, respectively.

From the perspective of the prediction task, what a HIRE model does is analogous to inductive matrix completion [52], [53], where Graph Neural Networks are used to model the interactions in a matrix. However, from the perspective of model design, HIRE is different from these GNN-based approaches. For one thing, GNN-based approaches only learn from a single type of rating relationship. For the other thing, the MHSA we use is a flexible specification of GNN layer [54]. As Eq. (2)-(3) shows, self-attention conducts the message passing on a complete graph with a learned 'soft' adjacency matrix instead of a fixed adjacency matrix.

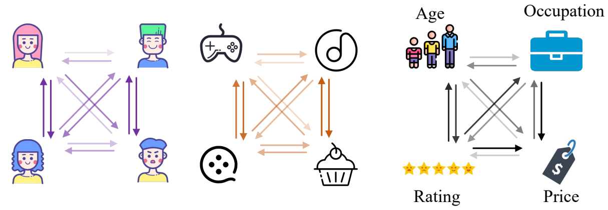


Fig. 4: Message Passing in complete graphs via learned interactions

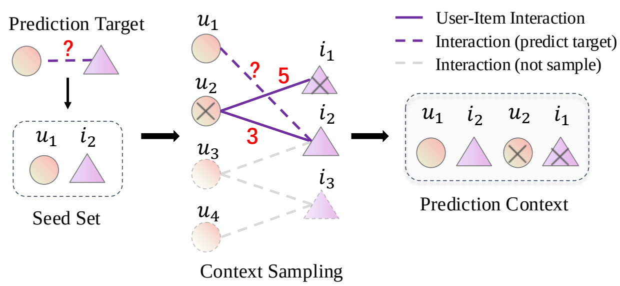


Fig. 5: An example of prediction context construction

B. Construction of Prediction Context

In this section, we delve into the methodology of constructing the prediction context for cold users/items, which encompasses rich information for modeling the heterogeneous interactions to cold users/items. For cold users/items, the prediction context comprises a set of users and items which are relevant to them, associated with their attributes and observed ratings. To determine the relevance of users and items, we employ a neighborhood-based sampling strategy to select the users and items from the user-item interaction bipartite graph. In a nutshell, given the limited budget of n users and m items in a prediction context, our sampling strategy preferentially samples those users/items that have rating interactions with the cold users/items. Concretely, we initialize a seed set by the prediction target, i.e., pairs of users and items involving cold users/items. Subsequently, the sampling begins at the seed set, by selecting the entities of the one-hop neighbors of the seed into the prediction context. If the number of the corresponding neighbor entities does not surpass the current budget, we put all the neighbor entities into the context. Otherwise, a subset of the neighborhood is sampled uniformly subject to the budget size. The neighborhood sampling iterates hop by hop until the budget is exhausted. In § VI-E, we also explore other sampling strategies to construct the context, such as random sampling and feature similarity sampling, where neighbourhood sampling achieves the best empirical results. Fig. 5 shows a toy example of prediction context construction.

Example 1: Given $n = 2, m = 2$, we aim to predict the rating of a cold user u_1 on an item i_2. Initially, the seed set is fixed as $\{u_1, i_2\}$, and an extra user and an extra item need to be sampled. In the bipartite rating graphs in Fig. 5, i_2 has two neighbors $\{u_2, u_3\}$, in which u_2 is sampled and the budget for users, n, is exhausted. As u_1 does not have any rated items, we can only sample an item from the neighbors of u_2, i.e., i_1. Finally, $\{u_1, u_2, i_1, i_2\}$ forms the prediction context.

After sampling n users and m items as the prediction context, we construct a 3-dimensional tensor as the initial input of the model. The tensor, denoted as $\boldsymbol{H} \in \mathbb{R}^{n\times m\times e}$, encodes the potential attributes or identities of the users and items and the observed ratings. Concretely, in Eq. (6), the k-th row and j-th column of $\boldsymbol{H}$, $\boldsymbol{H}[k, j, \cdot]$, is an e-dimensional vector that concatenates three parts of the features, i.e., the features of the user u_k, $\boldsymbol{x}_{u_k}$, the features of the item i_j, $\boldsymbol{x}_{i_j}$, the vector representation of the potential rating of u_k on i_j, $\boldsymbol{x}_r$.

$$\boldsymbol{H}[k, j, \cdot] \leftarrow [\boldsymbol{x}_{u_k} \| \boldsymbol{x}_{i_j} \| \boldsymbol{x}_r]. \tag{6}$$

The user and item features derive from linear transformation from the original attributes into hidden space. Let $\boldsymbol{e}_u^k$ denotes the one-hot embedding of the k-th categorical attribute of a user u. For the h_u attributes of users, there are h_u independent linear transformations $\{f_U^1, \cdots, f_U^{h_u}\}$ that transforms the one-hot embeddings into hidden features respectively, and the user feature $\boldsymbol{x}_u$ is the concatenation of those features as shown in Eq. (7). All the users in the recommendation system share the same set of linear feature transformations. And the item feature $\boldsymbol{x}_i$ is generated in a similar fashion as shown in Eq. (8).

$$\boldsymbol{x}_u \leftarrow [f_U^1(\boldsymbol{e}_u^1) \| f_U^2(\boldsymbol{e}_u^2) \| \ldots \| f_U^{h_u}(\boldsymbol{e}_u^{h_u})], \tag{7}$$

$$\boldsymbol{x}_i \leftarrow [f_I^1(\boldsymbol{e}_i^1) \| f_I^2(\boldsymbol{e}_i^2) \| \ldots \| f_I^{h_i}(\boldsymbol{e}_i^{h_i})]. \tag{8}$$

For the ratings, we also use a linear transformation f_R to map the concrete one-hot encoding of a rating r, $\boldsymbol{e}_r$, to the hidden representation $\boldsymbol{x}_r$ as shown in Eq. (9),

$$\boldsymbol{x}_r \leftarrow f(\boldsymbol{e}_r), \tag{9}$$

if the rating r is masked that will be predicted by the model, here, $\boldsymbol{e}_r$ and $\boldsymbol{x}_r$ are full-zero vectors. Suppose all the linear transformations f_U^k, f_I^k and f_R map all the one-hot embeddings into f-dimensional vectors. The dimension of the feature vector $\boldsymbol{H}[k, j, \cdot]$, e, equals to $(h_u + h_i + 1) \times f$. If a user/item does not possess an attribute, we set $\boldsymbol{e}_u$/$\boldsymbol{e}_i$ to the one-hot encoding of the ID of the user/item as its unique attribute, which is also transformed by a linear mapping f_U/f_I. The encoding of $\boldsymbol{H}$ can be easily extended to encode continuous attributes and ratings.

C. Model Details

We give the technical details of the heterogeneous interaction module (HIM) of HIRE. HIM fully exploits self-attention, a prevailing and powerful neural network layer, to learn implicit interactions. One HIM is composed of three different attention layers, i.e., attention between users, attention between items and attention between attributes, which are stacked layer by layer.

Modeling interactions between users (MBU). Given the initial embedding of the context $\boldsymbol{H} \in \mathbb{R}^{n\times m\times e}$, where n is the number of users, m is the number of items, e is the embedding dimension of attributes, we introduce an attention layer between users to model their implicit interactions. Specifically, we use $H_{i_j} = \boldsymbol{H}[\cdot, j, \cdot] \in \mathbb{R}^{n\times e}$ to denote the embedding view of item j in $\boldsymbol{H}$. An item embedding view,

H_{i_j}, is processed by a multi-head self-attention (MHSA) layer, where the attention weights between n users are computed, which is specific to item j. In Eq. (10), $\tilde{H}_{i_j} \in \mathbb{R}^{m\times e}$ is the fused item embedding view weighted by the attention weights among users. All the item embedding views are processed by a parameter-sharing MHSA in parallel, and the fused embeddings, $\{\tilde{H}_{i_j}|j \in [1, m]\}$, are stacked as the output embedding $\boldsymbol{H}^{(U)} \in \mathbb{R}^{n\times m\times e}$ (Eq. (11)),

$$\tilde{H}_{i_j} \leftarrow \text{MHSA}(H_{i_j}), \forall j \in [1, \cdots, m], \tag{10}$$

$$\boldsymbol{H}^{(U)} \leftarrow [\tilde{H}_{i_1} \| \tilde{H}_{i_2} \| \cdots \| \tilde{H}_{i_m}]. \tag{11}$$

Modeling interaction between items (MBI). To further model the interaction between items, we introduce an attention layer between items, given the stacked output embedding $\boldsymbol{H}^{(U)} \in \mathbb{R}^{n\times m\times e}$ as the input. Similarly, $H_{u_k} = \boldsymbol{H}^{(U)}[k, \cdot, \cdot] \in \mathbb{R}^{m\times e}$ denotes the embedding view of user k in $\boldsymbol{H}^{(U)}$. The attention weights between m items are computed through a MHSA layer, which is specific to a user embedding view, H_{u_k}, indicating the user-specific interactions of items. As Eq. (12) shows, $\tilde{H}_{u_k}$ is the fused user embedding view weighted by the attention weights among items. A parameter-sharing MHSA processes m independent user embedding views in parallel. The fused embeddings, $\{\tilde{H}_{u_k}|k \in [1, n]\}$, are stacked as the output embedding $\boldsymbol{H}^{(I)} \in \mathbb{R}^{n\times m\times e}$ (Eq. (13)),

$$\tilde{H}_{u_k} \leftarrow \text{MHSA}(H_{u_k}), \forall k \in [1, \cdots, n], \tag{12}$$

$$\boldsymbol{H}^{(I)} \leftarrow [\tilde{H}_{u_1} \| \tilde{H}_{u_2} \| \cdots \| \tilde{H}_{u_n}]. \tag{13}$$

Modeling interaction between attributes (MBA). HIM further learns the interaction between fine-grained attributes by computing the attention between user and item attributes. Taking the embedding $\boldsymbol{H}^{(I)} \in \mathbb{R}^{n\times m\times e}$ as input, we first reshape $\boldsymbol{H}^{(I)}$ to $\mathbb{R}^{n\times m\times h\times f}$ by spliting the feature dimension e into $h \times f$, where h is the number of categorical attributes and f is the embedding dimension for each attribute. We use $H_{u_k,i_j} = \boldsymbol{H}^{(I)}[k, j, \cdot, \cdot] \in \mathbb{R}^{h\times f}$ to denote the embedding view of a user-item pair (u_k, i_j) in $\boldsymbol{H}^{(I)}$. The embedding view H_{u_k,i_j} is processed by a MHSA layer, where the attention weights between h attributes are computed, indicating the interaction between attributes that is specific to a user-item pair (u_k, i_j). $\tilde{H}_{u_k,i_j} \in \mathbb{R}^{h\times f}$ in Eq. (14) is the fused user-item pair embedding view weighted by the attention weights between the h attributes. All the user-item pair embedding views are processed by a parameter-sharing MHSA in parallel, and the fused embeddings, $\{\tilde{H}_{u_k,i_j}|k \in [1, n], j \in [1, m]\}$, are stacked as the output embedding $\boldsymbol{H}^{(A)} \in \mathbb{R}^{n\times m\times h\times f}$ (Eq. (15)). The embedding $\boldsymbol{H}^{(A)}$ will be reshaped to $\mathbb{R}^{n\times m\times e}$ by flatting the feature dimension $h \times f$ back to e for future processing,

$$\tilde{H}_{u_k,i_j} \leftarrow \text{MHSA}(H_{u_k,i_j}), \forall k \in [1, \cdots, n], \forall j \in [1, \cdots, m], \tag{14}$$

$$\begin{aligned} \boldsymbol{H}^{(A)} \leftarrow [&\tilde{H}_{u_1,i_1} \| \tilde{H}_{u_1,i_2} \| \cdots \| \tilde{H}_{u_1,i_m} \\ &\tilde{H}_{u_2,i_1} \| \tilde{H}_{u_2,i_2} \| \cdots \| \tilde{H}_{u_2,i_m} \\ &\cdots \\ &\tilde{H}_{u_n,i_1} \| \tilde{H}_{u_n,i_2} \| \cdots \| \tilde{H}_{u_n,i_m}]. \end{aligned} \tag{15}$$

To summarize, HIM takes the sampled context $\boldsymbol{H} \in \mathbb{R}^{n\times m\times e}$ as input embedding, and model the interactions between users, items and attributes by three MHSA layer by layer. It finally generates embedding $\boldsymbol{H}^{(A)} \in \mathbb{R}^{n\times m\times e}$ which adaptively aggregates the features of users, items and attributes in the context by leveraging the attention weights. As Fig. 3 presents, the model HIRE is composed of L HIMs, where the output of the $(l-1)$-th HIM is fed into l-th HIM for modeling the higher order interactions.

Rating Prediction. Given $\boldsymbol{H}^{(A)} \in \mathbb{R}^{n\times m\times e}$ as the output of the L-th HIM, we use a decoder to predict the rating matrix $\hat{R} \in \mathbb{R}^{n\times m}$ of the sampled context as Eq. (16),

$$\hat{R} \leftarrow \alpha \cdot \text{Sigmoid}(g_\theta(\boldsymbol{H}^{(A)})). \tag{16}$$

Here, $g_\theta : \mathbb{R}^e \to \mathbb{R}^1$ is a linear transformation parameterized by θ and $\alpha \in \mathbb{R}$ is a scalar that rescales the range of the estimated ratings.

V. Model Training and Analysis

In this section, we further present the training of HIRE and analyze the complexity and inductive bias of the model.

A. Model Training

In a recommendation system, given a set of users $\mathcal{U}$, a set of items $\mathcal{I}$ and observed ratings $\mathcal{R}$, a HIRE model is trained by a set of prediction contexts, denoted as $\mathcal{D}$, which are sampled from $\mathcal{U}$, $\mathcal{I}$, and $\mathcal{R}$. Given a ratio of p ratings in a prediction context, the model is optimized to minimize the mean squared error (MSE) loss (Eq. (17)) for the $1-p$ of masked ratings,

$$\mathcal{L} = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{D}} \frac{1}{|\mathcal{Q}|} \sum_{r\in\mathcal{Q}} \|r - \hat{r}\|^2, \tag{17}$$

where $\hat{r} \in \mathbb{R}$ is a predicted rating in the output matrix $\hat{R}$ and $r \in \mathbb{R}$ is the corresponding ground truth rating. We use $\mathcal{Q}$ to denote the masked rating set.

Algorithm 1 presents the training process implemented by stochastic gradient descent. In each step, we randomly draw a mini-batch of prediction context, $\mathcal{B}$, from the training set $\mathcal{D}$ as line 4. Subsequently, in line 6-9, each context is transformed by the three attention layers in a HIM block one by one, and there are L HIM blocks in total. In line 10, the output of the L-th HIM is mapped to the predicted rating matrix $\hat{R}$. Finally, we compute the MSE Loss for the mini-batch $\mathcal{B}$ and update the model parameters via the back-propagation of gradient in line 11-12. The training will be terminated until the MSE loss converges.

In the test stage, when new user set $\mathcal{U}^*$ and item set $\mathcal{I}^*$ arrives, we can construct an embedding matrix $\boldsymbol{H}^*$. Ratings to be predicted are masked as 0. $\boldsymbol{H}^*$ is input into L HIM blocks following a mapping function to get predicted rating.

Example 2: Given the prediction context $\{u_1, u_2, i_1, i_2\}$ in Example 1, we suppose both users and items have 2 attributes. The context matrix $\boldsymbol{H}$ is a $2 \times 2 \times 80$ tensor, with each attribute and the rating are transformed into 16-dimensional vector by Eq. (7)-(9). Specifically, the embedded ratings contain the observed rating of (u_2, i_1) and (u_2, i_2), the target rating of (u_1, i_2) which is masked as 0, and the unobserved rating of (u_1, i_1) which is masked as -1. Taking $\boldsymbol{H}$ as the model input, MBU, MBI, MBA compute the embeddings $\boldsymbol{H}^{(U)}, \boldsymbol{H}^{(I)}, \boldsymbol{H}^{(A)} \in \mathbb{R}^{2\times 2\times 80}$, respectively. The process is repeated by L times. Finally, the last $\boldsymbol{H}^{(A)}$ is fed into a decoder in Eq. (16) to generate the rating matrix $\hat{R} \in \mathbb{R}^{2\times 2}$. The predicted rating of (u_1, i_2) corresponds to the element in the first row and second column of the matrix, i.e., $\hat{R}[0, 1]$.

B. Model Analysis

We analyze the time and space complexity of HIRE in brief. Suppose a prediction context is composed of n users and m items. $e = h \times f$ is the hidden dimension of the MHSA layer in HIM, where h is the number of attributes and f is the embedding dimension of each attribute. The linear feature transformation of HIRE (Eq. (7)-(9)) takes $\mathcal{O}(nmee_0)$ time, where e_0 is the maximum dimension of original feature. The computation of HIM dominates the complexity of HIRE. Specifically, the time complexities of the attention between users (Eq. (10)-(11)), attention between items (Eq. (12)-(13)) and attention between attributes (Eq. (14)-(15)) are $\mathcal{O}(n^2me)$, $\mathcal{O}(nm^2e)$, $\mathcal{O}(nmh^2f)$, respectively. The time complexity of the output layer (Eq. (16)) for rating prediction is $\mathcal{O}(nme)$. Assume $e_0 \ll e$ and the model is composed of K HIM blocks, the total time complexity is $\mathcal{O}(Knme(n + m + h))$. Regarding the space complexity, our neighborhood-based sampling strategy constructs a matrix which costs $\mathcal{O}(nme_0)$ space complexity for each task. For attention between users, items and attributes, the space complexity is $\mathcal{O}(n^2me)$, $\mathcal{O}(nm^2e)$ and $\mathcal{O}(nmh^2f)$, respectively. Thus the total space complexity for one context is $\mathcal{O}(Knme(n + m + h))$.

In addition, HIRE preserves an inherent inductive bias when exploiting sets of users and items for rating prediction, which is formulated as the property below.

Property 5.1. Given a set of users $\{u_1, \cdots, u_n\}$ and a set of items $\{i_1, \cdots, i_m\}$ as a prediction context $\boldsymbol{H}$ and the HIRE model $\mathcal{M}$, the predicted rating matrix, $\hat{R}$, is equivariant w.r.t. any permutation of the user set $\Pi_{[1:n]}$ and permutation of the item set $\Pi_{[1:m]}$,

$$\Pi_{[1:n]} \circ \Pi_{[1:m]} \circ \hat{R} = \mathcal{M}(\Pi_{[1:n]} \circ \Pi_{[1:m]} \circ \boldsymbol{H}). \tag{18}$$

The inductive bias is intuitive, since the MHSA is permutation equivariant., i.e.,

$$\Pi_{[1:n]} \circ \tilde{H}_{i_j} \leftarrow \text{MHSA}(\Pi_{[1:n]} \circ H_{i_j}), \tag{19}$$

$$\Pi_{[1:m]} \circ \tilde{H}_{u_k} \leftarrow \text{MHSA}(\Pi_{[1:m]} \circ H_{u_k}), \tag{20}$$

and matrices stack/concatenation naturally preserve permutation equivariance. This property ensures that given the sets of users and items in the prediction context, our model predicts deterministic ratings regardless their input order, which also shows that our approach can make use of MHSA designed for sequential inputs to solve the cold-start problem.

Algorithm 1: Training Algorithm of HIRE

```
Input : User set U, item set I, observed ratings R.
Initialize model parameters;
Generate a set of context matrices D = {H_1, ...} for training;
do
    Sample a mini-batch of context matrices B ~ D;
    for H ∈ B do
        for l → 1 to L do
            Compute H^(U) via Eq. (10)-(11);
            Compute H^(I) via Eq. (12)-(13);
            Compute H^(A) via Eq. (14)-(15);
        Compute the predicted rating matrix R̂ via Eq. (16);
    Compute the MSE Loss L via Eq. (17);
    Update model parameters by gradient descent;
while L converges
```

TABLE I: Profile of Datasets

Dataset	MovieLens-1M	Douban	Bookcrossing
# Users	6,040	23,822	278,858
# Items	3,706	185,574	271,379
# Ratings	1,000,209	1,387,216	1,149,780
User Attributes	Age, Occupation, Gender, Zip code	N/A	Age
Item Attributes	Rate, Genre, Director, Actor	N/A	Publication year
Range of Ratings	$1 \sim 5$	$1 \sim 5$	$1 \sim 10$

VI. Experimental Studies

In this section, we give the test setting (§ VI-A) and report our substantial experimental results in the following facets: ① Compare the effectiveness of HIRE with the state-of-the-art approaches under 3 cold-start scenarios (§ VI-B). ② Compare the test efficiency of HIRE with the baseline approaches (§ VI-C). ③ Investigate the sensitivity of HIRE regarding key hyper-parameter configurations (§ VI-D). ④ Study the influence of different attention layers via an ablation study (§ VI-E). ⑤ Conduct a case study for HIRE on a movie rating task. (§ VI-F).

A. Experimental Setup

Datasets: We use 3 widely used datasets to evaluate HIRE and the baseline approaches, whose profiles are summarized in Table I. MovieLens-1M [55] contains the ratings of users for movies, and the users and movies are associated with rich attributes. Following the previous following the previous literature [22], We randomly split 80% of users for training and validation, and 20% for test. movies released before 1997 are used for training and validation while the remaining are used for test, resulting in an approximate ratio of 8:2. Bookcrossing [56] is collected from Book-Crossing, containing ratings of users on books. Douban [57], [58] contains the ratings of users on musics, which are extracted from a rating website

TABLE II: Overall Performance in Warm States (%)

Datasets	Methods	TOP5 Precision	TOP5 NDCG	TOP5 MAP	TOP7 Precision	TOP7 NDCG	TOP7 MAP	TOP10 Precision	TOP10 NDCG	TOP10 MAP
MovieLens-1M	Popularity	$57.63_{(0.52)}$	$86.02_{(0.13)}$	$47.41_{(0.21)}$	$62.47_{(0.63)}$	$86.64_{(0.11)}$	$51.74_{(0.38)}$	$69.69_{(0.63)}$	$87.80_{(0.08)}$	$59.63_{(0.58)}$
	NeuMF	$49.92_{(0.12)}$	$73.01_{(0.11)}$	$40.83_{(0.10)}$	$52.25_{(0.19)}$	$74.97_{(0.15)}$	$42.26_{(0.16)}$	$56.01_{(0.20)}$	$76.62_{(0.17)}$	$45.71_{(0.21)}$
	TaNP	$58.73_{(0.12)}$	$87.24_{(0.06)}$	$48.66_{(0.04)}$	$61.79_{(0.09)}$	$87.55_{(0.05)}$	$50.95_{(0.20)}$	$67.21_{(0.17)}$	$88.27_{(0.02)}$	$56.56_{(0.24)}$
	HIRE	$\mathbf{78.23_{(0.57)}}$	$\mathbf{95.86_{(0.26)}}$	$\mathbf{73.06_{(0.68)}}$	$\mathbf{78.46_{(0.37)}}$	$\mathbf{96.58_{(0.24)}}$	$\mathbf{73.60_{(0.46)}}$	$\mathbf{78.51_{(0.13)}}$	$\mathbf{98.03_{(0.22)}}$	$\mathbf{73.90_{(0.16)}}$
Bookcrossing	Popularity	$35.92_{(0.19)}$	$83.02_{(0.12)}$	$25.58_{(0.23)}$	$41.05_{(0.28)}$	$83.83_{(0.10)}$	$27.75_{(0.31)}$	$50.05_{(0.28)}$	$85.47_{(0.07)}$	$34.36_{(0.28)}$
	NeuMF	$40.69_{(0.13)}$	$33.07_{(0.16)}$	$36.49_{(0.68)}$	$53.02_{(0.12)}$	$34.34_{(0.19)}$	$48.83_{(0.09)}$	$65.96_{(0.07)}$	$37.33_{(0.17)}$	$62.01_{(0.05)}$
	TaNP	$51.47_{(0.55)}$	$88.33_{(0.19)}$	$44.11_{(0.59)}$	$55.15_{(0.54)}$	$88.89_{(0.16)}$	$46.53_{(0.50)}$	$61.84_{(0.27)}$	$89.89_{(0.12)}$	$52.02_{(0.60)}$
	HIRE	$\mathbf{60.86_{(1.22)}}$	$\mathbf{91.09_{(0.44)}}$	$\mathbf{55.00_{(1.53)}}$	$\mathbf{65.51_{(0.70)}}$	$\mathbf{92.42_{(0.44)}}$	$\mathbf{58.54_{(0.82)}}$	$\mathbf{70.40_{(0.21)}}$	$\mathbf{94.47_{(0.42)}}$	$\mathbf{63.85_{(0.45)}}$
Douban	Popularity	$57.28_{(0.18)}$	$87.61_{(0.01)}$	$47.54_{(0.13)}$	$63.74_{(0.22)}$	$88.51_{(0.03)}$	$53.50_{(0.25)}$	$73.66_{(0.03)}$	$90.14_{(0.01)}$	$64.50_{(0.06)}$
	NeuMF	$35.45_{(0.05)}$	$43.32_{(0.04)}$	$28.29_{(0.02)}$	$41.90_{(0.05)}$	$44.30_{(0.04)}$	$34.44_{(0.02)}$	$53.81_{(0.05)}$	$44.75_{(0.05)}$	$47.23_{(0.04)}$
	TaNP	$57.26_{(1.02)}$	$86.63_{(0.22)}$	$46.34_{(0.82)}$	$66.15_{(0.94)}$	$88.17_{(0.24)}$	$55.10_{(0.91)}$	$\mathbf{78.94_{(1.10)}}$	$90.91_{(0.30)}$	$70.81_{(1.37)}$
	HIRE	$\mathbf{64.05_{(0.41)}}$	$\mathbf{90.07_{(0.23)}}$	$\mathbf{58.16_{(0.57)}}$	$\mathbf{69.39_{(0.35)}}$	$\mathbf{91.33_{(0.21)}}$	$\mathbf{62.73_{(0.55)}}$	$76.70_{(0.20)}$	$\mathbf{93.88_{(0.18)}}$	$\mathbf{71.10_{(0.35)}}$

TABLE III: Overall Performance in 3 Cold-Start Scenarios on MovieLens-1M (%)

Scenarios	Methods	TOP5 Precision	TOP5 NDCG	TOP5 MAP	TOP7 Precision	TOP7 NDCG	TOP7 MAP	TOP10 Precision	TOP10 NDCG	TOP10 MAP
UC	Popularity	$\underline{58.20_{(0.11)}}$	$85.40_{(0.06)}$	$\underline{47.38_{(0.16)}}$	$\underline{64.07_{(0.13)}}$	$86.40_{(0.05)}$	$\underline{52.95_{(0.17)}}$	$\underline{72.89_{(0.15)}}$	$88.00_{(0.06)}$	$\underline{63.38_{(0.20)}}$
	TransGNN	$55.20_{(1.80)}$	$56.86_{(1.69)}$	$41.29_{(2.11)}$	$56.39_{(1.80)}$	$57.33_{(2.06)}$	$41.40_{(2.79)}$	$58.12_{(0.48)}$	$57.50_{(1.57)}$	$43.40_{(1.48)}$
	NeuMF	$47.02_{(0.66)}$	$67.13_{(0.52)}$	$37.13_{(0.68)}$	$50.26_{(0.67)}$	$68.99_{(0.45)}$	$39.47_{(0.63)}$	$54.78_{(0.65)}$	$70.73_{(0.34)}$	$43.91_{(0.68)}$
	Wide&Deep	$51.89_{(0.83)}$	$83.85_{(0.15)}$	$41.57_{(1.02)}$	$55.61_{(0.47)}$	$83.96_{(0.21)}$	$44.20_{(0.58)}$	$62.48_{(0.66)}$	$84.71_{(0.15)}$	$50.81_{(0.93)}$
	DeepFM	$51.69_{(0.92)}$	$83.67_{(0.79)}$	$41.23_{(0.93)}$	$55.36_{(0.53)}$	$83.85_{(0.73)}$	$43.84_{(0.62)}$	$61.90_{(0.25)}$	$84.51_{(0.63)}$	$50.19_{(0.28)}$
	AFN	$50.84_{(0.57)}$	$82.94_{(0.32)}$	$39.98_{(0.56)}$	$55.50_{(0.33)}$	$83.33_{(0.29)}$	$43.40_{(0.33)}$	$62.54_{(0.26)}$	$84.23_{(0.18)}$	$50.28_{(0.43)}$
	GraphHINGE	$51.80_{(0.75)}$	$77.75_{(0.69)}$	$40.76_{(0.71)}$	$58.19_{(0.86)}$	$78.06_{(0.82)}$	$46.55_{(0.86)}$	$64.62_{(0.76)}$	$78.26_{(0.79)}$	$53.03_{(0.94)}$
	MetaHIN	$43.92_{(2.38)}$	$80.05_{(1.07)}$	$35.79_{(2.45)}$	$48.92_{(1.62)}$	$81.09_{(0.92)}$	$38.81_{(1.79)}$	$56.44_{(1.48)}$	$82.58_{(0.91)}$	$45.72_{(1.91)}$
	MAMO	$46.63_{(0.53)}$	$59.05_{(0.79)}$	$34.05_{(0.70)}$	$52.94_{(0.38)}$	$61.19_{(0.62)}$	$38.69_{(0.32)}$	$64.66_{(0.63)}$	$65.31_{(0.32)}$	$50.86_{(0.40)}$
	TaNP	$57.15_{(1.02)}$	$\underline{87.18_{(0.41)}}$	$47.28_{(0.99)}$	$59.86_{(0.44)}$	$\underline{87.31_{(0.26)}}$	$48.99_{(0.69)}$	$65.91_{(0.49)}$	$\underline{88.07_{(0.20)}}$	$54.95_{(0.54)}$
	MeLU	$50.93_{(0.66)}$	$62.54_{(1.73)}$	$40.11_{(1.49)}$	$57.87_{(0.18)}$	$64.99_{(1.29)}$	$46.22_{(1.07)}$	$68.82_{(1.49)}$	$69.74_{(1.39)}$	$57.67_{(1.67)}$
	HIRE	$\mathbf{69.99_{(1.84)}}$	$\mathbf{91.69_{(1.45)}}$	$\mathbf{64.54_{(3.09)}}$	$\mathbf{72.59_{(0.42)}}$	$\mathbf{92.54_{(1.76)}}$	$\mathbf{66.52_{(1.44)}}$	$\mathbf{76.56_{(2.46)}}$	$\mathbf{94.32_{(2.23)}}$	$\mathbf{70.65_{(2.11)}}$
IC	Popularity	$57.08_{(1.24)}$	$\underline{86.14_{(0.46)}}$	$49.13_{(1.15)}$	$60.78_{(0.89)}$	$\underline{86.26_{(0.39)}}$	$51.53_{(0.99)}$	$67.57_{(0.83)}$	$\underline{86.94_{(0.37)}}$	$55.99_{(1.03)}$
	TransGNN	$50.63_{(2.36)}$	$50.78_{(2.49)}$	$37.96_{(2.51)}$	$56.39_{(2.01)}$	$53.31_{(2.22)}$	$38.36_{(2.27)}$	$57.50_{(0.83)}$	$54.67_{(1.26)}$	$39.40_{(1.23)}$
	NeuMF	$\underline{57.26_{(0.62)}}$	$72.37_{(0.55)}$	$\underline{49.82_{(0.61)}}$	$\underline{60.97_{(0.62)}}$	$73.53_{(0.48)}$	$\underline{53.56_{(0.62)}}$	$66.09_{(0.63)}$	$75.03_{(0.45)}$	$\underline{59.33_{(0.61)}}$
	Wide&Deep	$30.06_{(1.11)}$	$51.96_{(6.14)}$	$19.25_{(1.74)}$	$37.51_{(1.96)}$	$54.07_{(6.03)}$	$23.73_{(3.28)}$	$46.81_{(3.01)}$	$56.35_{(5.83)}$	$30.90_{(5.33)}$
	DeepFM	$30.91_{(0.15)}$	$53.09_{(6.49)}$	$20.12_{(0.90)}$	$37.95_{(1.43)}$	$54.86_{(6.09)}$	$24.51_{(2.47)}$	$46.72_{(3.42)}$	$56.89_{(5.37)}$	$31.34_{(5.06)}$
	AFN	$29.89_{(2.56)}$	$48.55_{(8.78)}$	$18.91_{(0.57)}$	$37.05_{(1.26)}$	$51.11_{(8.56)}$	$23.08_{(0.89)}$	$45.92_{(1.28)}$	$53.87_{(7.80)}$	$29.65_{(3.38)}$
	GraphHINGE	$14.28_{(0.37)}$	$17.79_{(0.72)}$	$5.67_{(0.17)}$	$20.87_{(0.58)}$	$22.14_{(0.83)}$	$7.69_{(0.27)}$	$29.71_{(0.50)}$	$26.70_{(0.92)}$	$11.48_{(0.19)}$
	MetaHIN	$43.69_{(1.44)}$	$79.41_{(0.68)}$	$35.41_{(1.59)}$	$49.34_{(0.98)}$	$80.44_{(0.54)}$	$39.17_{(1.22)}$	$56.24_{(0.74)}$	$81.96_{(0.52)}$	$44.89_{(1.06)}$
	MAMO	$46.87_{(1.32)}$	$59.42_{(0.90)}$	$34.39_{(0.80)}$	$54.25_{(0.23)}$	$61.94_{(0.64)}$	$40.06_{(0.09)}$	$65.15_{(0.40)}$	$65.90_{(0.44)}$	$51.52_{(0.85)}$
	TaNP	$40.68_{(1.22)}$	$75.64_{(0.60)}$	$27.20_{(1.52)}$	$48.82_{(1.55)}$	$77.41_{(0.65)}$	$33.39_{(2.02)}$	$59.38_{(2.07)}$	$79.82_{(0.77)}$	$43.83_{(2.90)}$
	MeLU	$48.93_{(0.79)}$	$59.20_{(1.26)}$	$36.66_{(1.72)}$	$57.08_{(0.67)}$	$62.44_{(0.58)}$	$43.97_{(0.35)}$	$\underline{68.60_{(1.78)}}$	$67.61_{(0.39)}$	$56.65_{(1.39)}$
	HIRE	$\mathbf{59.89_{(1.61)}}$	$\mathbf{86.40_{(1.62)}}$	$\mathbf{53.04_{(3.07)}}$	$\mathbf{64.44_{(1.16)}}$	$\mathbf{87.50_{(1.88)}}$	$\mathbf{56.72_{(1.86)}}$	$\mathbf{71.25_{(2.84)}}$	$\mathbf{89.87_{(2.22)}}$	$\mathbf{63.72_{(2.60)}}$
U&I C	Popularity	$55.24_{(1.71)}$	$\underline{86.10_{(0.42)}}$	$47.13_{(1.86)}$	$59.69_{(2.29)}$	$\underline{86.26_{(0.68)}}$	$50.54_{(2.55)}$	$\underline{74.01_{(1.70)}}$	$\underline{86.87_{(0.60)}}$	$60.16_{(2.08)}$
	TransGNN	$54.38_{(1.24)}$	$55.07_{(1.45)}$	$42.29_{(2.18)}$	$55.21_{(1.12)}$	$55.16_{(1.23)}$	$42.65_{(1.63)}$	$55.63_{(0.47)}$	$55.68_{(0.62)}$	$46.24_{(0.90)}$
	NeuMF	$\underline{55.99_{(0.61)}}$	$68.85_{(0.57)}$	$\underline{48.50_{(0.64)}}$	$\underline{60.65_{(0.53)}}$	$69.36_{(0.62)}$	$\underline{53.26_{(0.54)}}$	$66.82_{(0.36)}$	$70.59_{(0.62)}$	$\underline{60.34_{(0.39)}}$
	Wide&Deep	$29.52_{(1.28)}$	$51.13_{(6.10)}$	$18.57_{(1.79)}$	$37.45_{(2.41)}$	$53.45_{(5.86)}$	$23.37_{(3.53)}$	$46.63_{(3.48)}$	$55.77_{(5.59)}$	$30.39_{(5.78)}$
	DeepFM	$30.99_{(0.73)}$	$52.86_{(6.74)}$	$19.71_{(0.58)}$	$38.70_{(0.74)}$	$54.67_{(6.38)}$	$24.74_{(2.20)}$	$46.91_{(3.30)}$	$56.51_{(5.57)}$	$31.21_{(5.23)}$
	AFN	$29.18_{(2.90)}$	$47.49_{(9.16)}$	$18.28_{(0.98)}$	$36.74_{(1.83)}$	$50.16_{(9.11)}$	$22.85_{(0.43)}$	$45.84_{(1.16)}$	$53.17_{(8.18)}$	$29.12_{(3.47)}$
	GraphHINGE	$9.92_{(0.45)}$	$11.31_{(0.62)}$	$3.35_{(0.18)}$	$17.00_{(0.32)}$	$15.67_{(0.55)}$	$5.52_{(0.15)}$	$29.13_{(0.34)}$	$20.78_{(0.43)}$	$12.48_{(0.42)}$
	MetaHIN	$43.92_{(2.38)}$	$80.05_{(1.07)}$	$35.79_{(2.45)}$	$48.92_{(1.62)}$	$81.09_{(0.92)}$	$38.81_{(1.79)}$	$56.44_{(1.48)}$	$82.58_{(0.91)}$	$45.72_{(1.91)}$
	MAMO	$47.89_{(0.53)}$	$60.46_{(0.01)}$	$35.50_{(0.20)}$	$53.91_{(0.62)}$	$62.51_{(0.18)}$	$39.71_{(0.26)}$	$64.80_{(0.13)}$	$66.14_{(0.28)}$	$51.20_{(0.69)}$
	TaNP	$41.14_{(1.12)}$	$76.63_{(0.51)}$	$28.13_{(0.93)}$	$49.29_{(1.00)}$	$78.20_{(0.42)}$	$34.41_{(0.66)}$	$58.67_{(0.74)}$	$80.16_{(0.24)}$	$43.35_{(0.96)}$
	MeLU	$46.80_{(1.03)}$	$56.92_{(0.60)}$	$33.93_{(0.55)}$	$56.16_{(1.26)}$	$60.87_{(0.72)}$	$42.10_{(1.28)}$	$68.40_{(1.79)}$	$66.50_{(0.94)}$	$55.56_{(2.32)}$
	HIRE	$\mathbf{60.30_{(1.75)}}$	$\mathbf{86.93_{(1.80)}}$	$\mathbf{53.62_{(3.19)}}$	$\mathbf{66.27_{(0.73)}}$	$\mathbf{88.37_{(2.07)}}$	$\mathbf{59.04_{(1.16)}}$	$\mathbf{74.19_{(4.29)}}$	$\mathbf{91.40_{(2.33)}}$	$\mathbf{67.64_{(4.33)}}$

Douban. This dataset also contains friendship relations between users. Since users/items do not have attributes, we use the embeddings of the user/item ID as the attributes, which are generated by learnable linear weights. For Bookcrossing and Douban, we randomly split 70% of users/items into the train and validation set and 30% into the test set, respectively. For all the datasets, train and validate sets are further split randomly by a ratio of 7:1. The test users/items are regarded as cold users/items where 10% of the interactions, i.e., no more than 3 interactions, in the datasets can be exploited by the recommendation models and the remaining 90% are used for model evaluation.

Baselines: To comprehensively evaluate HIRE, we compare with the following 12 baselines which fell into 6 categories: ❶ Basic Strategy: Popularity predicts ratings by summing the number of interactions and normalizing for each item. ❷ CF-based baselines: NeuMF [8], Wide&Deep [24], DeepFM [25] and AFN [26] adopt different neural network layers to model the interactions from user-item pairs. ❸ Social recommendation: GraphRec [15] models the users and items respectively by aggregating their local interactions via Graph Neural Network. Aggregation of the social relation among users is used to

TABLE IV: Overall Performance in 3 Cold-Start Scenarios on Bookcrossing (%)

Scenarios	Methods	TOP5 Precision	TOP5 NDCG	TOP5 MAP	TOP7 Precision	TOP7 NDCG	TOP7 MAP	TOP10 Precision	TOP10 NDCG	TOP10 MAP
UC	Popularity	37.68 (1.07)	83.77 (0.39)	26.33 (1.55)	42.33 (1.74)	84.14 (0.19)	27.51 (1.69)	48.01 (0.14)	86.10 (0.21)	31.29 (0.95)
	TransGNN	38.75 (1.76)	41.43 (1.72)	32.11 (1.56)	41.07 (2.52)	43.44 (2.27)	34.34 (2.35)	45.01 (0.07)	46.24 (0.67)	36.83 (1.53)
	NeuMF	33.28 (1.06)	38.87 (1.10)	26.57 (1.67)	41.76 (1.12)	39.62 (1.32)	33.88 (2.03)	51.65 (1.55)	41.78 (1.27)	42.14 (2.71)
	Wide&Deep	28.52 (1.61)	54.08 (1.97)	21.61 (0.08)	37.59 (2.31)	55.28 (2.73)	25.17 (1.96)	47.46 (6.78)	56.66 (4.16)	32.58 (5.04)
	DeepFM	29.56 (0.13)	51.16 (0.65)	18.70 (0.79)	35.11 (1.25)	51.43 (0.57)	21.20 (0.70)	43.21 (2.78)	52.15 (0.18)	27.87 (2.33)
	AFN	22.05 (7.14)	49.70 (3.54)	14.62 (6.44)	31.95 (4.14)	50.63 (2.64)	18.56 (4.52)	42.10 (0.66)	52.38 (0.80)	25.73 (2.77)
	MAMO	40.16 (1.15)	27.52 (1.94)	30.62 (1.23)	54.44 (0.70)	32.44 (1.84)	44.19 (0.91)	71.19 (0.46)	38.95 (1.50)	64.17 (0.65)
	TaNP	41.18 (0.30)	85.04 (1.45)	33.38 (0.39)	45.98 (0.54)	86.01 (1.10)	36.26 (1.31)	52.88 (2.10)	87.12 (0.72)	41.60 (3.29)
	MeLU	46.51 (4.16)	58.60 (7.09)	35.34 (4.40)	52.82 (7.79)	60.79 (7.37)	40.79 (6.62)	67.01 (4.42)	64.66 (7.98)	55.58 (4.92)
	HIRE	**57.13 (5.30)**	**89.31 (2.91)**	**50.79 (7.51)**	**64.05 (2.57)**	**90.58 (2.67)**	**56.40 (5.44)**	**71.91 (2.84)**	**92.95 (2.34)**	**64.75 (1.51)**
IC	Popularity	42.19 (6.50)	84.29 (0.84)	30.90 (5.72)	44.37 (4.29)	85.05 (0.35)	32.51 (4.52)	48.04 (3.13)	85.69 (0.06)	33.29 (3.94)
	TransGNN	27.50 (3.53)	27.67 (3.78)	26.19 (1.76)	28.03 (0.75)	28.39 (1.33)	26.25 (0.84)	29.28 (1.01)	29.30 (1.07)	26.50 (0.58)
	NeuMF	40.70 (0.66)	36.32 (4.07)	32.82 (4.14)	51.42 (2.25)	38.13 (4.22)	43.03 (7.73)	59.30 (8.68)	41.03 (4.30)	57.29 (2.00)
	Wide&Deep	50.07 (7.86)	80.14 (0.95)	38.14 (9.35)	54.51 (8.85)	80.59 (0.96)	39.53 (8.74)	58.70 (9.14)	80.76 (0.39)	43.93 (9.28)
	DeepFM	52.46 (8.13)	81.10 (1.62)	40.92 (7.72)	55.50 (2.12)	81.56 (1.81)	41.21 (9.28)	59.50 (2.13)	82.00 (2.45)	44.37 (9.95)
	AFN	49.15 (7.10)	80.18 (1.78)	40.40 (9.08)	53.95 (9.98)	80.83 (1.87)	42.00 (1.41)	56.79 (6.60)	81.29 (1.87)	43.99 (8.91)
	MAMO	41.29 (0.62)	28.10 (0.49)	32.46 (0.51)	54.62 (0.41)	32.40 (0.36)	44.98 (0.71)	71.60 (0.36)	39.42 (0.86)	64.18 (0.19)
	TaNP	41.16 (3.15)	85.45 (1.32)	31.25 (1.52)	45.06 (1.74)	86.18 (1.29)	32.87 (2.90)	51.94 (4.35)	87.13 (1.22)	38.48 (5.04)
	MeLU	49.25 (5.77)	61.59 (2.89)	37.64 (5.17)	52.04 (8.20)	62.58 (2.45)	39.15 (6.39)	64.15 (8.72)	65.65 (3.62)	53.41 (7.89)
	HIRE	**58.37 (3.38)**	**89.25 (0.56)**	**51.74 (3.27)**	**64.18 (2.44)**	**90.57 (0.57)**	**56.49 (2.06)**	**72.20 (4.71)**	**92.62 (0.65)**	**65.19 (4.97)**
U&I C	Popularity	26.47 (1.39)	84.46 (2.53)	16.55 (1.57)	29.57 (1.58)	84.63 (1.89)	18.02 (1.60)	34.48 (1.03)	85.13 (1.19)	21.51 (1.35)
	TransGNN	34.00 (2.82)	33.07 (4.58)	26.50 (2.68)	36.57 (0.81)	35.30 (2.42)	27.02 (1.72)	38.01 (5.65)	36.24 (1.73)	27.26 (1.05)
	NeuMF	38.29 (0.97)	42.21 (3.29)	29.76 (1.05)	43.42 (1.20)	43.04 (3.45)	32.74 (2.29)	49.22 (0.74)	43.64 (2.05)	36.61 (1.70)
	Wide&Deep	40.37 (8.41)	69.67 (3.77)	33.04 (6.96)	50.23 (6.82)	71.72 (8.40)	36.59 (6.91)	56.49 (6.54)	72.04 (7.90)	41.26 (8.67)
	DeepFM	39.27 (7.71)	68.48 (7.31)	30.18 (7.10)	47.50 (2.94)	69.00 (8.38)	36.49 (4.29)	54.14 (9.65)	72.66 (5.03)	38.81 (9.81)
	AFN	34.76 (3.96)	60.94 (2.88)	26.90 (2.93)	35.05 (1.93)	62.98 (4.68)	27.00 (0.09)	39.88 (3.17)	64.58 (4.79)	28.39 (2.89)
	MAMO	41.00 (0.87)	32.56 (6.32)	30.26 (3.76)	51.67 (7.63)	36.99 (5.34)	41.01 (9.34)	71.78 (1.54)	44.85 (7.24)	63.80 (3.91)
	TaNP	51.14 (1.74)	86.20 (7.90)	43.65 (0.83)	51.53 (2.14)	87.74 (4.47)	44.62 (4.77)	54.84 (6.76)	87.80 (2.50)	45.31 (8.56)
	MeLU	43.35 (4.36)	54.65 (1.46)	33.49 (2.93)	49.92 (9.28)	57.16 (0.80)	37.84 (7.88)	63.73 (9.18)	59.93 (3.51)	53.50 (9.99)
	HIRE	**60.77 (3.33)**	**90.60 (0.25)**	**55.29 (3.22)**	**69.03 (4.30)**	**92.09 (5.10)**	**62.84 (4.05)**	**76.36 (6.89)**	**93.92 (0.64)**	**70.15 (8.12)**

enhance user embeddings. We only compare GraphRec on Douban where the social relationship between users is available. ❹ HIN-based baselines: GraphHINGE [33] aggregates rich interactive patterns from an HIN. MetaHIN [36] incorporates multifaceted semantic contexts induced by meta-paths. We test MetaHIN and GraphHINGE only on MovieLens-1M which provides sufficient attributes for constructing an HIN. Specifically, we define user, movie, genre, occupation, and age as different node types and create links between nodes, such as movie-genre interaction, user-age interaction, and user-movie interaction. For the other two datasets Bookcrossing and Douban, due to lack of sufficient attributes of user/item, the two baselines cannot be applied to them. ❺ Meta-learning baselines: MAMO [21], MeLU [22] and TaNP [20] adopt meta-learning for cold-start recommendation in a multi-task fashion. MAMO and MeLU support all the 3 cold-start scenarios while TaNP only supports user cold-start originally. Here, we extend the task settings of TaNP to item cold-start and user & item cold-start. ❻ Transformer-based baselines: TransGNN [44] integrates transformer and GNN layers to mutually enhance their capabilities.

Implementation details: We introduce the model and training details of HIRE as well as the baselines. For HIRE, the model is equipped with 3 HIM blocks in which each MHSA layer has 8 heads and the hidden dimension of each head is set to 16. The model is trained by a LAMB optimizer [59] with $\beta = (0.9, 0.999)$ and $\epsilon = 10^{-6}$, and a Lookahead [60] wrapper with slow update rate $\alpha = 0.5$ and $k = 6$ steps between updates. We use a flat-then-anneal learning rate scheduler which flats at the base learning rate for 70% of steps, and then anneals following a cosine schedule to 0 by the end of training. We set the base learning rate to 10^{-3} and the gradient clip threshold to 1.0. The optimizer and the scheduler are widely used for training language models composed of deep MHSA layers. Regarding the prediction contexts, we set the number of users and items in each context to 32 by default. In the training and test, 90% of observed ratings are masked for prediction and the remaining 10% are associated with the context input. The learning framework of HIRE is built on PyTorch.

For the baseline approaches, we use their released source code and keep the hyper-parameters as their default settings. All the models are trained until convergence. For the four CF-based approaches, GraphRec and GraphHINGE, to achieve a fair comparison, we use all the observed user-item ratings in the sampled training contexts, together with the 10% unmasked user-item ratings in the test context as the ground truth to train the models. We use the prediction context to test TransGNN, where a few number of user-item interaction is used for training and the remaining is used for test. For meta-learning approaches, in each task, 10% of the observed ratings are used as the ground truth in the support set and the remaining 90% are used in the query set. Training and testing are conducted on one Tesla V100 with 16GB memory.

Evaluation Metrics: We adopt 3 commonly used metrics following the previous work [20] to evaluate the recommendation performance in our testing, including: Precision, Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain [61] (NDCG). Top k actual rating values sorted by predicted rating values are used to calculate the above metrics, where $k \in \{5, 7, 10\}$.

B. Comparison Results

We investigate the overall performance of HIRE in 3 cold-start scenarios, i.e., user cold-start (UC), item cold-start (IC), and user & item cold-start (U&I C), in comparison with the 12 baseline approaches. We also investigate whether HIRE can effectively handle regular cases by using data from warm states and we compare HIRE against several competitive baselines. This means that the test users and items may also appear in the training dataset and have a substantial history of interactions. The results are reported in Table II. Table III, IV and V list the test results on 3 datasets, respectively, where the first-best (yellow) and the second-best (underlined) scores are highlighted. In general, HIRE achieves the best test accuracy in most cases. The Precision, NDCG and MAP of HIRE succeed all the baselines 0.21, 0.29, 0.22 on average, respectively. The superiority of HIRE is reflected in improving NDCG while keeping high Precision and MAP.

In Table II, HIRE outperforms these baselines, demonstrating its capability in handling regular warm cases effectively. Furthermore, since in regular cases, the prediction context involves a substantial amount of historical interactions for the test users/items, HIRE achieves better performance in regular cases than in the cold-start scenarios in most cases.

Popularity achieves a very competitive performance on MovieLens-1M and Douban, since the correlation between the number of interactions and the rating is positive in these datasets. TransGNN does not perform well in cold-start scenarios since it relies heavily on user-item interactions. In contrast to neural CF-based approaches, NeuMF, Wide&Deep, DeepFM and AFN, HIRE outperforms them on all the datasets significantly. The CF-based approaches only model a single type of user-item interaction via observed ratings, which is difficult to generalize to new users/items with only a few interactions. This reflects modeling heterogeneous interactions is beneficial for cold-start prediction. The meta-learning approaches, MAMO, TaNP and MeLU, outperform the CF-based approaches by a large margin. The performance of these 3 approaches dominates the second best result, and is competitive to our HIRE in some cases. The meta-learning approaches focus on exploiting parameter sharing and adaption to address cold-start problem. Their complicated adaption module and learning strategy may lead to a difficult learning procedure. HIRE deals with the cold-start problem in a different way that relies on the interaction learned from the data. It concentrates on using a relatively simple neural network framework and learning algorithm to model heterogeneous interactions in a data-driven fashion.

On MovieLens-1M (Table III), we compared HIN-based baseline GraphHINGE and MetaHIN. In the user cold-start scenario, GraphHINGE gains better performance than CF-based approaches, indicating that metapath-guided neighborhood can capture the complex and high-order semantics in the HIN. In item cold-start and user & item cold-start scenarios, MetaHIN consistently outperforms GraphHINGE, as indicated by its higher NDCG scores. However, these two approaches fail to outperform HIRE and TaNP. It validates again that manually defined heterogeneous patterns may not contribute to estimating preference, while the learned heterogeneous interactions by HIRE tend to be more reliable. We compare the social recommendation baseline, GraphRec, on Douban (Table V). GraphRec performs better in the user cold-start scenario, and can even surpasses HIRE. GraphRec uses Graph Neural Network to integrate the rating interactions and social relations, which generates better user representations. However, we find that GraphRec is less effective in scenarios with cold items, where the social relations between users are less helpful for cold items.

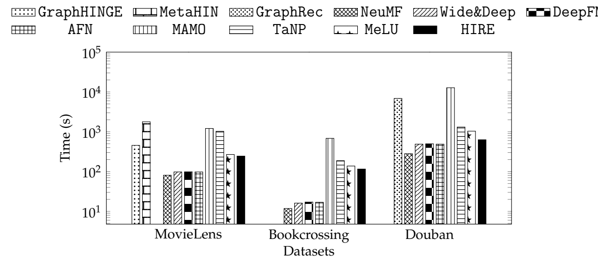


Fig. 6: Total Test Time (in second)

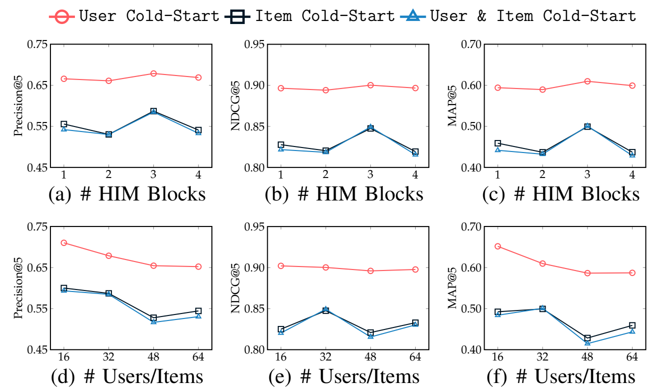


Fig. 7: Sensitivity Analysis on MovieLens-1M

C. Efficiency

We compare the test time of HIRE with the baseline approaches, where the total test time for the 3 datasets on a single Tesla V100 GPU is presented in Fig. 6. Recall that the HIN-based approach GraphHINGE is only applicable for MovieLens-1M and the social recommendation approach GraphRec is only applicable for Douban. Here, we only compare for the user cold-start since the test time for the 3 cold-start scenarios is similar. The CF-based approaches, NeuMF, Wide&Deep, AFN and DeepFM are the most time-efficiency approaches, since they only take a pair of user and item with their potential features as input, and the model architectures

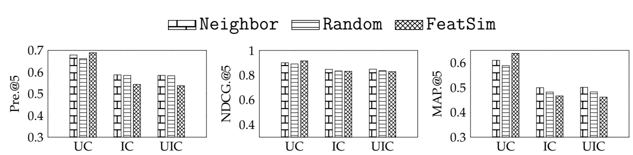
TABLE V: Overall Performance in 3 Cold-Start Scenarios on Douban (%)

Scenarios	Methods	TOP5 Precision	TOP5 NDCG	TOP5 MAP	TOP7 Precision	TOP7 NDCG	TOP7 MAP	TOP10 Precision	TOP10 NDCG	TOP10 MAP
UC	Popularity	$51.43_{(0.25)}$	$86.58_{(0.06)}$	$40.01_{(0.29)}$	$56.03_{(0.27)}$	$87.24_{(0.06)}$	$43.12_{(0.34)}$	$64.46_{(0.23)}$	$88.52_{(0.06)}$	$51.20_{(0.37)}$
	TransGNN	$38.54_{(0.90)}$	$38.45_{(1.35)}$	$21.76_{(1.24)}$	$38.69_{(2.10)}$	$38.51_{(1.53)}$	$23.55_{(1.51)}$	$38.96_{(1.30)}$	$38.64_{(1.80)}$	$25.60_{(2.10)}$
	NeuMF	$44.43_{(1.60)}$	$33.34_{(1.99)}$	$40.56_{(1.64)}$	$58.72_{(1.24)}$	$34.71_{(1.91)}$	$56.23_{(1.30)}$	$72.61_{(0.71)}$	$37.65_{(1.68)}$	$71.13_{(0.80)}$
	Wide&Deep	$54.42_{(1.05)}$	$73.76_{(0.70)}$	$44.43_{(1.75)}$	$55.66_{(0.31)}$	$75.95_{(0.81)}$	$45.01_{(1.20)}$	$58.26_{(0.32)}$	$77.25_{(0.83)}$	$47.83_{(0.68)}$
	DeepFM	$51.32_{(1.78)}$	$67.07_{(2.86)}$	$40.15_{(1.69)}$	$51.33_{(2.39)}$	$70.16_{(2.93)}$	$41.15_{(2.80)}$	$52.81_{(1.47)}$	$72.61_{(2.42)}$	$43.12_{(1.93)}$
	AFN	$59.18_{(1.19)}$	$78.75_{(1.98)}$	$49.19_{(1.98)}$	$60.88_{(0.83)}$	$79.81_{(1.82)}$	$49.92_{(1.57)}$	$64.42_{(1.08)}$	$80.41_{(1.71)}$	$54.07_{(0.38)}$
	GraphRec	$60.65_{(2.79)}$	$50.73_{(3.66)}$	$54.77_{(3.26)}$	$\mathbf{75.49}_{(\mathbf{2.38})}$	$56.05_{(3.73)}$	$\mathbf{72.20}_{(\mathbf{2.67})}$	$\mathbf{87.71}_{(\mathbf{1.72})}$	$62.06_{(3.56)}$	$\mathbf{86.31}_{(\mathbf{1.89})}$
	MAMO	$60.98_{(0.07)}$	$73.56_{(0.05)}$	$51.01_{(0.05)}$	$67.96_{(0.05)}$	$75.94_{(0.02)}$	$57.41_{(0.04)}$	$80.06_{(0.03)}$	$80.60_{(0.02)}$	$72.09_{(0.04)}$
	TaNP	$64.08_{(3.40)}$	$90.20_{(1.04)}$	$54.65_{(4.47)}$	$69.06_{(3.08)}$	$91.00_{(0.94)}$	$59.19_{(4.22)}$	$78.24_{(3.86)}$	$92.59_{(1.04)}$	$70.44_{(5.30)}$
	MeLU	$45.42_{(4.80)}$	$64.52_{(1.81)}$	$34.63_{(4.57)}$	$50.54_{(7.06)}$	$66.23_{(2.69)}$	$38.15_{(6.93)}$	$59.53_{(7.98)}$	$69.30_{(3.52)}$	$47.23_{(8.85)}$
	HIRE	$\mathbf{71.52}_{(\mathbf{3.91})}$	$\mathbf{92.69}_{(\mathbf{1.31})}$	$\mathbf{65.95}_{(\mathbf{5.74})}$	$74.59_{(2.48)}$	$\mathbf{93.28}_{(\mathbf{1.22})}$	$68.37_{(4.20)}$	$80.46_{(1.30)}$	$\mathbf{94.79}_{(\mathbf{1.11})}$	$75.21_{(1.08)}$
IC	Popularity	$47.35_{(0.40)}$	$84.35_{(0.22)}$	$36.10_{(0.45)}$	$54.63_{(0.08)}$	$85.44_{(0.12)}$	$42.06_{(0.30)}$	$64.85_{(0.09)}$	$87.12_{(0.09)}$	$51.76_{(0.14)}$
	TransGNN	$54.67_{(2.30)}$	$50.73_{(2.01)}$	$44.49_{(0.84)}$	$59.42_{(0.99)}$	$55.10_{(0.59)}$	$47.38_{(0.37)}$	$61.33_{(1.15)}$	$57.51_{(0.47)}$	$49.39_{(0.25)}$
	NeuMF	$39.19_{(4.42)}$	$43.05_{(4.28)}$	$30.50_{(1.83)}$	$46.88_{(1.67)}$	$44.23_{(4.13)}$	$36.93_{(2.12)}$	$56.04_{(0.88)}$	$46.44_{(3.95)}$	$45.36_{(3.09)}$
	Wide&Deep	$22.85_{(0.80)}$	$44.27_{(1.78)}$	$17.87_{(0.59)}$	$25.00_{(1.53)}$	$44.53_{(1.70)}$	$18.78_{(1.27)}$	$29.37_{(1.30)}$	$45.33_{(1.72)}$	$21.68_{(1.05)}$
	DeepFM	$23.90_{(0.23)}$	$46.33_{(0.21)}$	$18.56_{(0.29)}$	$25.74_{(0.19)}$	$46.65_{(1.51)}$	$18.81_{(0.53)}$	$29.70_{(0.68)}$	$47.31_{(1.48)}$	$21.30_{(0.74)}$
	AFN	$26.00_{(2.71)}$	$49.15_{(2.67)}$	$20.44_{(2.17)}$	$28.75_{(2.10)}$	$49.93_{(2.38)}$	$21.60_{(1.75)}$	$32.98_{(2.59)}$	$50.56_{(2.65)}$	$24.05_{(1.83)}$
	GraphRec	$34.60_{(2.98)}$	$39.73_{(3.60)}$	$28.47_{(3.93)}$	$43.28_{(3.28)}$	$42.73_{(4.09)}$	$35.97_{(4.36)}$	$53.88_{(2.70)}$	$46.39_{(4.24)}$	$45.08_{(4.69)}$
	MAMO	$59.80_{(0.11)}$	$72.50_{(0.05)}$	$49.86_{(0.10)}$	$\mathbf{66.96}_{(\mathbf{0.11})}$	$74.95_{(0.04)}$	$56.39_{(0.12)}$	$\mathbf{79.43}_{(\mathbf{0.06})}$	$79.78_{(0.03)}$	$\mathbf{71.20}_{(\mathbf{0.06})}$
	TaNP	$49.45_{(3.03)}$	$85.02_{(0.90)}$	$38.08_{(3.26)}$	$54.79_{(3.96)}$	$85.90_{(1.00)}$	$42.30_{(4.62)}$	$63.28_{(5.95)}$	$87.40_{(1.35)}$	$51.07_{(7.50)}$
	MeLU	$50.87_{(3.63)}$	$66.50_{(2.21)}$	$38.76_{(3.35)}$	$56.19_{(6.85)}$	$68.59_{(3.71)}$	$43.54_{(8.22)}$	$65.12_{(9.02)}$	$71.79_{(4.93)}$	$57.70_{(5.76)}$
	HIRE	$\mathbf{61.28}_{(\mathbf{4.67})}$	$\mathbf{89.26}_{(\mathbf{1.59})}$	$\mathbf{54.96}_{(\mathbf{6.79})}$	$64.63_{(3.07)}$	$\mathbf{89.83}_{(\mathbf{1.47})}$	$\mathbf{57.08}_{(\mathbf{5.13})}$	$70.66_{(0.67)}$	$\mathbf{91.22}_{(\mathbf{1.36})}$	$63.05_{(1.96)}$
U&I C	Popularity	$47.54_{(0.39)}$	$87.82_{(0.19)}$	$42.24_{(0.57)}$	$54.10_{(0.26)}$	$87.95_{(0.05)}$	$46.47_{(0.27)}$	$70.82_{(0.05)}$	$90.14_{(0.06)}$	$62.91_{(1.12)}$
	TransGNN	$27.33_{(1.15)}$	$30.34_{(1.50)}$	$25.07_{(0.71)}$	$30.47_{(1.65)}$	$32.91_{(1.88)}$	$27.33_{(1.01)}$	$30.67_{(2.30)}$	$33.62_{(2.26)}$	$29.11_{(1.02)}$
	NeuMF	$27.63_{(0.55)}$	$35.97_{(1.91)}$	$22.02_{(0.63)}$	$28.57_{(1.30)}$	$36.16_{(1.18)}$	$22.37_{(0.99)}$	$33.40_{(2.35)}$	$38.98_{(0.98)}$	$24.78_{(0.79)}$
	Wide&Deep	$8.42_{(2.72)}$	$13.03_{(1.53)}$	$7.30_{(1.67)}$	$8.48_{(2.56)}$	$14.35_{(1.18)}$	$7.31_{(1.44)}$	$8.57_{(2.02)}$	$16.15_{(1.43)}$	$8.32_{(1.77)}$
	DeepFM	$5.78_{(3.82)}$	$11.03_{(6.67)}$	$4.82_{(2.83)}$	$5.96_{(3.30)}$	$12.48_{(6.63)}$	$5.21_{(2.92)}$	$7.52_{(0.36)}$	$14.50_{(1.82)}$	$6.12_{(0.84)}$
	AFN	$5.26_{(0.98)}$	$11.23_{(1.82)}$	$4.76_{(0.26)}$	$5.88_{(1.67)}$	$12.86_{(1.92)}$	$5.27_{(1.26)}$	$6.39_{(1.00)}$	$14.84_{(1.95)}$	$5.52_{(1.15)}$
	GraphRec	$35.68_{(2.83)}$	$39.00_{(1.68)}$	$26.24_{(.01.78)}$	$40.66_{(2.71)}$	$40.43_{(1.65)}$	$28.85_{(2.34)}$	$47.00_{(1.81)}$	$42.76_{(1.28)}$	$33.26_{(1.49)}$
	MAMO	$60.09_{(0.12)}$	$72.78_{(0.04)}$	$50.37_{(0.11)}$	$67.50_{(0.16)}$	$75.31_{(0.05)}$	$57.21_{(0.16)}$	$\mathbf{79.76}_{(\mathbf{0.06})}$	$80.07_{(0.03)}$	$\mathbf{71.73}_{(\mathbf{0.08})}$
	TaNP	$60.67_{(2.44)}$	$87.34_{(0.40)}$	$49.82_{(2.17)}$	$\mathbf{69.62}_{(\mathbf{0.89})}$	$89.03_{(0.30)}$	$58.05_{(1.66)}$	$77.55_{(1.74)}$	$91.00_{(0.39)}$	$66.90_{(1.71)}$
	MeLU	$50.32_{(1.73)}$	$67.37_{(0.70)}$	$39.34_{(1.11)}$	$55.44_{(2.73)}$	$68.75_{(1.16)}$	$43.55_{(2.28)}$	$67.17_{(1.25)}$	$72.94_{(0.78)}$	$56.26_{(1.36)}$
	HIRE	$\mathbf{62.66}_{(\mathbf{1.95})}$	$\mathbf{89.02}_{(\mathbf{2.14})}$	$\mathbf{54.16}_{(\mathbf{5.86})}$	$69.53_{(1.69)}$	$\mathbf{90.17}_{(\mathbf{1.94})}$	$\mathbf{61.48}_{(\mathbf{3.24})}$	$78.33_{(2.51)}$	$\mathbf{92.47}_{(\mathbf{1.66})}$	$71.06_{(2.00)}$

TABLE VI: Ablation Study for Different Attention Layers on MovieLens-1M (%)

Blocks	User Cold-Start			Item Cold-Start			User & Item Cold-Start		
	Pre.@5	NDCG@5	MAP@5	Pre.@5	NDCG@5	MAP@5	Pre.@5	NDCG@5	MAP@5
wo/ Item & Attribute	44.65	78.58	32.32	43.92	76.00	31.77	46.63	77.00	34.40
wo/ User & Attribute	65.52	89.26	58.38	52.68	81.74	43.01	52.27	81.38	42.39
wo/ User & Item	67.52	89.86	60.40	51.63	81.28	42.02	50.67	80.79	40.73
wo/ User	65.90	89.25	58.85	52.72	81.16	42.23	52.39	81.11	42.13
wo/ Item	44.61	78.66	32.38	44.14	76.10	31.93	46.87	77.00	34.47
wo/ A	44.77	78.65	32.42	44.13	76.11	32.00	46.71	76.99	34.42
wo/ Attribute	**69.99**	**91.69**	**64.54**	**59.89**	**86.40**	**53.04**	**60.30**	**86.93**	**53.62**

are relatively simple. Our approach, HIRE, spends a longer time for prediction than the CF-based approaches due to the computational cost of multi-layer MHSA. However, the prediction of HIRE is faster than that of the other 6 approaches on average. The meta-learning approaches, MAMO, MetaHIN, TaNP and MeLU, leverage extra modules for model adaption. GraphHINGE needs to sample and model the neighborhood interactions from an HIN, which results in the second longest time in MovieLens-1M. GraphRec spends the second longest time for prediction since it deploys an additional Graph Neural Network to aggregate social relations. MAMO is significantly slower than HIRE, with about one order of magnitude. This is because MAMO is built upon the MAML algorithm and utilizes two specific memory modules, which have high time and space complexities. In a nutshell, HIRE outperforms other methods by achieving the best overall effectiveness while maintaining a competitive prediction efficiency.



(a) Precision (b) NDCG (c) MAP

Fig. 8: Impact of sampling methods on MovieLens-1M

D. Sensitivity Analysis

We investigate the parameter sensitivity of HIRE with respect to two key hyper-parameter configurations: (1) the number of the Heterogeneous Interaction Modules (HIM) in

HIRE and (2) the number of users/items sampled in one prediction context. Due to space limitation, we present the evaluation results for the metrics at $k = 5$ considering it can be regarded as a lower bound for all the metrics. First, we train HIRE model variants by varying the number of HIMs in $\{1, 2, 3, 4\}$ on MovieLens-1M, where all the other hyper-parameters are fixed as default. Fig. 7(a), 7(b) and 7(c) show the test accuracy in the 3 cold-start scenarios. Here, HIRE with 3 HIMs achieves the best performance, which is consistent to the different cold-start scenarios. As the number of HIMs increases, the model is able to capture increasingly high-order and complex interactions. However, more HIMs such as 4 will incur the risk of overfiting and lead to performance degradation in practice. In contrast, for Bookcrossing and Douban, in our extra experiments, we observe that 2 and 4 HIM blocks achieve the best performance, respectively. Thereby, different datasets may need different configurations to reach the best result. Another observation is that HIRE in user cold-start is less sensitive to the number of HIMs, compared with item and user & item cold-start.

In addition, we also compare the accuracy of HIRE by varying the number of sampled users/items in the training and test context in $\{16, 32, 48, 64\}$, where the results are shown in Fig. 7(d), 7(e) and 7(f). As the number of users/items increases, the performance of HIRE may not change monotonically. More users/items in the context matrices may provide richer information for the prediction, e.g., in the cases of 64 users/items, or bring noise information, e.g., in the cases of 48 users/items, which hurts the performance. For MovieLens-1M, setting the number of users/items to 32 yields the most impressive scores, whereas in our extra testing on Bookcrossing and Douban, the setting of 64 yields the best scores. We speculate that the reason would be the latter two datasets have less attributes so that they need more users/items in the prediction context.

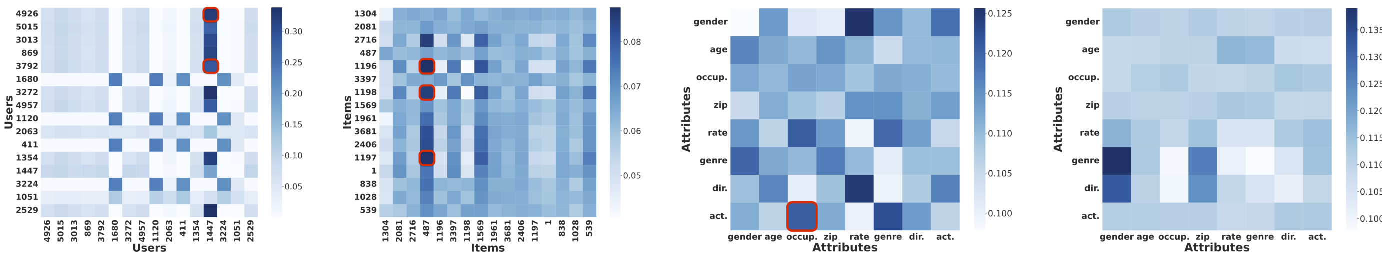
E. Ablation Study

Impacts of the three types of attention layers. We conduct an ablation study to investigate the effectiveness of the three types of attention layers, attention between users, items and attributes, in HIRE. Here, we train 6 model variants on MovieLens-1M, where single types or combinations of two types of layers are removed for the model. Table VI lists the test performance of the model variants in 3 cold-start scenarios. The full model with the three types of attention layers achieves the overall best performance, indicating that the heterogeneous interactions from three different perspectives collaboratively contribute to the rating prediction. In addition, we find that the attention between items (or attributes) plays a more important role than attention between users in rating prediction, by comparing model variants wo/ Item (or wo/ Attribute) with model variant wo/ User. For MovieLens-1M, we speculate that the similarity between the movies and the similarity between the proprieties of users and movies influences the rating to a larger degree. Meanwhile, another possible reason would be the interactions between users may not be reliable. That is because there is a counter-intuitive observation on that the model variant wo/ Item (or wo/ Attribute) performs worse than the model variant wo/ User & Item (or wo/ User & Attribute). And the model with only attention between users, i.e., wo/ Item & Attribute, leaves the worst performance among all the model variants. The attention between users may learn irrelevant interactions, where deploying this layer alone or combining it with another single type of attention layer incurs extra noise for the prediction. Furthermore, in our experiments on Bookcrossing and Douban datasets, the full model consistently outperforms other variants.

Impacts of sampling methods. To study our neighborhood-based sampling strategy for context construction, we compare it with the random sampling strategy and feature similarity sampling strategy as an ablation. For the feature similarity sampling strategy, we compute the cosine similarity of attributes between target users (items) and other users (items). The users (items) with higher scores will be sampled. Fig. 8 shows the test result of HIRE on MovieLens-1M, by fixing the number of users and items as 32. Here, our neighborhood-based sampling strategy is better than random sampling in all cases by 1.26% on average. On the other two datasets, we also achieve similar results, further confirming the effectiveness of the neighborhood-based sampling strategy. The reason would be that compared with a fully randomized batch of users/items whose correlations are unknown, neighborhood-based sampling strategy is able to select more relevant neighbor users to construct the prediction context. That promotes HIRE to make more accurate predictions. Feature similarity sampling strategy tends to perform better than neighborhood-based in the user cold-start scenario. The reason would be that relevant users can be sampled via computing their feature similarity. For cold items, feature similarity is unable to select the most relevant items, which results in poor performance in other two scenarios.

F. Case Study

We conduct a case study to investigate whether HIM learns relevant and reliable interactions between users by the 3 layers MBU, MBI and MBA. We visualize the attention matrices for the prediction on MovieLens-1M in Fig. 9. Here, the darker the cell, the higher the corresponding weight, and the stronger the implicit interaction learned by HIM. First, Fig. 9(a) shows the attention weights in MBU layer among 16 users for item $1,304$, which is an action and comedy movie, 'Butch Cassidy and the Sundance Kid'. In Fig. 9(a), as the highlighted red rectangle, user $1,447$ is deeply affected by user $4,926$ and $3,792$. We find that the three users share the same preference on item $1,304$. With the learned user interactions, HIRE predicts the ratings of the three users on item $1,304$ as 3.41, 4.13 and 4.20, which is highly consistent to the ground-truth ratings, $3, 4, 4$, respectively. That indicates the interaction between users has a strong correlation to their ratings. Second, Fig. 9(b) shows the attention weights in MBI layer among 16 items for user $4,926$, who is a female student under 18 years old. Similarly, we observe that item 487 is



(a) Weights of MBU (b) Weights of MBI (c) Weights of MBA (high rating) (d) Weights of MBA (low rating)

Fig. 9: Visualizations of the Attention Weights in HIRE

strongly affected by items $1,196$, $1,197$ and $1,198$. With the learned interaction between items, HIRE predicts the ratings of that user on item 487 as 3.42, and on items $1,196$, $1,197$ and $1,198$ as $3.88, 3.95, 3.86$, respectively. That is also consistent to the ground-truth ratings of 3 for item 487 and 4 for the other three items. The results also demonstrate that the learned implicit interactions among items guide the model to predict similar ratings for interacted items. Finally, Fig. 9(c) and (d) present the attention weights among the 8 attributes in the MBA layer, for two user-item pairs, user 4,926 and item 1,304, user 1,051 and item 2,081, respectively. The former pair has a high rating of 4. User 1051 is a $18-24$ year old male technician/engineer and item 2081 is a comedy 'Trees Lounge'. The user 1051 gave a low rating 2 to movie $2,081$. It is reasonable according to the occupation of user and genre of movie. As we can see, high rating pair has more attribute interactions. Actor attribute has interactions with occupation in Fig. 9(c). This means that the actor in the movie may be liked by people in a specific occupation. The latter one has less interactions between movie attributes and user attributes. It may because user has no interest to this movie. It is worth mentioning that the weight matrices are unsymmetrical due to the computation of attention (Eq. (2)), which indicates the interactions are single direction.

VII. Conclusion

In this paper, we propose a fully data-driven recommender system HIRE that can model heterogeneous interactions for cold-start rating prediction. For target cold user/items, we adopt a neighborhood-based sampling strategy to sample prediction context from the user-item bipartite rating graph, and explore reliable interactions in the context. Specifically, we devise a Heterogeneous Interaction Module to jointly model the interactions in three levels, i.e., users, items and attributes, respectively. Comprehensive experiments on 3 real-world datasets validate the effectiveness of our model. HIRE outperforms baselines with higher precision, NDCG and MAP by 0.21, 0.29 and 0.22 on average. The source code is publicly available at https://github.com/FangShuheng/HIRE.

Acknowledgments

This work was supported by the Research Grants Council of Hong Kong, China, No. 14203618, No. 14202919 and No. 14205520. Kangfei Zhao is supported by National Key Research and Development Plan, No. 2023YFF0725101.

References

[1] F. Lyu, X. Tang, H. Guo, R. Tang, X. He, R. Zhang, and X. Liu, "Memorize, factorize, or be naive: Learning optimal feature interaction methods for ctr prediction," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 2022, pp. 1450–1462.

[2] W. Guo, C. Zhang, Z. He, J. Qin, H. Guo, B. Chen, R. Tang, X. He, and R. Zhang, "Miss: Multi-interest self-supervised learning framework for click-through rate prediction," in *2022 IEEE 38th international conference on data engineering (ICDE)*, 2022, pp. 727–740.

[3] T. Qian, Y. Liang, Q. Li, and H. Xiong, "Attribute graph neural networks for strict cold start recommendation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 8, pp. 3597–3610, 2020.

[4] H. Li, Y. Wang, Z. Lyu, and J. Shi, "Multi-task learning for recommendation over heterogeneous information network," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 2, pp. 789–802, 2020.

[5] J. Cao, J. Sheng, X. Cong, T. Liu, and B. Wang, "Cross-domain recommendation to cold-start users via variational information bottleneck," in *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, 2022, pp. 2209–2223.

[6] G. Linden, B. Smith, and J. York, "Amazon. com recommendations: Item-to-item collaborative filtering," *IEEE Internet computing*, vol. 7, no. 1, pp. 76–80, 2003.

[7] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the 24th international conference on World Wide Web*, 2015, pp. 111–112.

[8] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T.-S. Chua, "Neural collaborative filtering," in *Proceedings of the 26th international conference on world wide web*, 2017, pp. 173–182.

[9] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[10] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "Bpr: Bayesian personalized ranking from implicit feedback," 2012.

[11] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 285–295.

[12] Y. Rong, X. Wen, and H. Cheng, "A monte carlo algorithm for cold start recommendation," in *Proceedings of the 23rd international conference on World wide web*, 2014, pp. 327–336.

[13] J. Tang, X. Hu, and H. Liu, "Social recommendation: a review," *Soc. Netw. Anal. Min.*, vol. 3, no. 4, pp. 1113–1133, 2013.

[14] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, "A neural influence diffusion model for social recommendation," in *Proceedings of the 42nd international ACM SIGIR conference on research and development in information retrieval*, 2019, pp. 235–244.

[15] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The world wide web conference*, 2019, pp. 417–426.

[16] M. Jamali and M. Ester, "Trustwalker: a random walk model for combining trust-based and item-based recommendation," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 397–406.

[17] J. Tang, X. Hu, H. Gao, and H. Liu, "Exploiting local and global social context for recommendation." in *IJCAI*, vol. 13, 2013, pp. 2712–2718.

[18] H. Ma, H. Yang, M. R. Lyu, and I. King, "Sorec: social recommendation using probabilistic matrix factorization," in *Proceedings of the 17th ACM conference on Information and knowledge management*, 2008, pp. 931–940.

[19] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proceedings of the fourth ACM conference on Recommender systems*, 2010, pp. 135–142.

[20] X. Lin, J. Wu, C. Zhou, S. Pan, Y. Cao, and B. Wang, "Task-adaptive neural process for user cold-start recommendation," in *Proceedings of the Web Conference 2021*, 2021, pp. 1306–1316.

[21] M. Dong, F. Yuan, L. Yao, X. Xu, and L. Zhu, "Mamo: Memory-augmented meta-optimization for cold-start recommendation," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 688–697.

[22] H. Lee, J. Im, S. Jang, H. Cho, and S. Chung, "Melu: Meta-learned user preference estimator for cold-start recommendation," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 1073–1082.

[23] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *International conference on machine learning*. PMLR, 2017, pp. 1126–1135.

[24] H.-T. Cheng, L. Koc, J. Harmsen, T. Shaked, T. Chandra, H. Aradhye, G. Anderson, G. Corrado, W. Chai, M. Ispir *et al.*, "Wide & deep learning for recommender systems," in *Proceedings of the 1st workshop on deep learning for recommender systems*, 2016, pp. 7–10.

[25] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "Deepfm: a factorization-machine based neural network for ctr prediction," 2017.

[26] W. Cheng, Y. Shen, and L. Huang, "Adaptive factorization network: Learning adaptive-order feature interactions," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 3609–3616.

[27] S. Fang, K. Zhao, G. Li, and J. X. Yu, "Community search: a meta-learning approach," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023, pp. 2358–2371.

[28] S. Fang, K. Zhao, Y. Rong, Z. Li, and J. X. Yu, "Inductive attributed community search: To learn communities across graphs," *Proceedings of the VLDB Endowment*, vol. 17, no. 10, pp. 2576–2589, 2024.

[29] H. Zhang, C. Shen, X. Sun, J. Tan, Y. Rong, C. Piao, H. Cheng, and L. Yi, "Adaptive coordinators and prompts on heterogeneous graphs for cross-domain recommendations," 2024. [Online]. Available: https://arxiv.org/abs/2410.11719

[30] X. Yu, X. Ren, Y. Sun, B. Sturt, U. Khandelwal, Q. Gu, B. Norick, and J. Han, "Recommendation in heterogeneous information networks with implicit user feedback," in *Proceedings of the 7th ACM conference on Recommender systems*, 2013, pp. 347–350.

[31] H. Zhao, Q. Yao, J. Li, Y. Song, and D. L. Lee, "Meta-graph based recommendation fusion over heterogeneous information networks," in *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, 2017, pp. 635–644.

[32] C. Luo, W. Pang, Z. Wang, and C. Lin, "Hete-cf: Social-based collaborative filtering recommendation using heterogeneous relations," in *2014 IEEE International Conference on Data Mining*, 2014, pp. 917–922.

[33] J. Jin, J. Qin, Y. Fang, K. Du, W. Zhang, Y. Yu, Z. Zhang, and A. J. Smola, "An efficient neighborhood-based interaction model for recommendation on heterogeneous graph," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 75–84.

[34] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "Pathsim: Meta path-based top-k similarity search in heterogeneous information networks," *Proceedings of the VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[35] E. Min, Y. Rong, T. Xu, Y. Bian, D. Luo, K. Lin, J. Huang, S. Ananiadou, and P. Zhao, "Neighbour interaction based click-through rate prediction via graph-masked transformer," in *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2022, pp. 353–362.

[36] Y. Lu, Y. Fang, and C. Shi, "Meta-learning on heterogeneous information networks for cold-start recommendation," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 1563–1573.

[37] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," *Advances in neural information processing systems*, vol. 30, 2017.

[38] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *International conference on learning representations*, 2016.

[39] A. Graves, G. Wayne, and I. Danihelka, "Neural turing machines," 2014.

[40] M. Garnelo, J. Schwarz, D. Rosenbaum, F. Viola, D. J. Rezende, S. Eslami, and Y. W. Teh, "Neural processes," 2018.

[41] J. Jiang, P. Zhang, Y. Luo, C. Li, J. B. Kim, K. Zhang, S. Wang, X. Xie, and S. Kim, "Adamct: adaptive mixture of cnn-transformer for sequential recommendation," in *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023, pp. 976–986.

[42] K. Kang, J. Park, W. Kim, H. Choe, and J. Choo, "Recommender system using sequential and global preference via attention mechanism and topic modeling," in *Proceedings of the 28th ACM international conference on information and knowledge management*, 2019, pp. 1543–1552.

[43] W.-C. Kang and J. McAuley, "Self-attentive sequential recommendation," in *2018 IEEE international conference on data mining (ICDM)*. IEEE, 2018, pp. 197–206.

[44] P. Zhang, Y. Yan, X. Zhang, C. Li, S. Wang, F. Huang, and S. Kim, "Transgnn: Harnessing the collaborative power of transformers and graph neural networks for recommender systems," in *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 1285–1295.

[45] L. Wu, Z. Zheng, Z. Qiu, H. Wang, H. Gu, T. Shen, C. Qin, C. Zhu, H. Zhu, Q. Liu *et al.*, "A survey on large language models for recommendation," 2023.

[46] J. Wang, H. Lu, J. Caverlee, E. H. Chi, and M. Chen, "Large language models as data augmenters for cold-start item recommendation," in *Companion Proceedings of the ACM on Web Conference 2024*, 2024, pp. 726–729.

[47] S. Sanner, K. Balog, F. Radlinski, B. Wedin, and L. Dixon, "Large language models are competitive near cold-start recommenders for language-and item-based preferences," in *Proceedings of the 17th ACM conference on recommender systems*, 2023, pp. 890–896.

[48] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[49] J. D. M.-W. C. Kenton and L. K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of naacL-HLT*, vol. 1, 2019, p. 2.

[50] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[51] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," 2020.

[52] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, 2018, pp. 974–983.

[53] M. Zhang and Y. Chen, "Inductive matrix completion based on graph neural networks," in *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*, 2020.

[54] E. Min, R. Chen, Y. Bian, T. Xu, K. Zhao, W. Huang, P. Zhao, J. Huang, S. Ananiadou, and Y. Rong, "Transformer for graphs: An overview from architecture perspective," 2022.

[55] F. M. Harper and J. A. Konstan, "The movielens datasets: History and context," *Acm transactions on interactive intelligent systems (tiis)*, vol. 5, no. 4, pp. 1–19, 2015.

[56] C.-N. Ziegler, S. M. McNee, J. A. Konstan, and G. Lausen, "Improving recommendation lists through topic diversification," in *Proceedings of the 14th international conference on World Wide Web*, 2005, pp. 22–32.

[57] E. Zhong, W. Fan, J. Wang, L. Xiao, and Y. Li, "Comsoc: Adaptive transfer of user behaviors over composite social network," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '12, 2012.

[58] E. Zhong, W. Fan, and Q. Yang, "User behavior learning and transfer in composite social networks," *ACM Trans. Knowl. Discov. Data*, vol. 8, no. 1, pp. 6:1–6:32, 2014.

[59] Y. You, J. Li, S. Reddi, J. Hseu, S. Kumar, S. Bhojanapalli, X. Song, J. Demmel, K. Keutzer, and C.-J. Hsieh, "Large batch optimization for deep learning: Training bert in 76 minutes," 2019.

[60] M. Zhang, J. Lucas, J. Ba, and G. E. Hinton, "Lookahead optimizer: k steps forward, 1 step back," vol. 32, 2019.
[61] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of ir techniques," *ACM Transactions on Information Systems (TOIS)*, vol. 20, no. 4, pp. 422–446, 2002.