Curriculum Pre-Training Heterogeneous Subgraph Transformer for Top-N Recommendation

HUI WANG*, Renmin University of China, China
KUN ZHOU*, Renmin University of China, China
WAYNE XIN ZHAO, Renmin University of China, China Beijing Academy of Artificial Intelligence, China
JINGYUAN WANG, Beihang University, China Peng Cheng Laboratory, China
JI-RONG WEN, Renmin University of China, China

To characterize complex and heterogeneous side information in recommender systems, heterogeneous information network (HIN) has shown superior performance and attracted much research attention. In HIN, the rich entities, relations and paths can be utilized to model the correlations of users and items, such a task setting is often called *HIN-based recommendation*. Although HIN provides a general approach to modeling rich side information, it lacks special consideration on the goal of the recommendation task. The aggregated context from the heterogeneous graph is likely to incorporate irrelevant information, and the learned representations are not specifically optimized according to the recommendation task. Therefore, there is a need to rethink how to leverage the useful information from HIN to accomplish the recommendation task.

To address the above issues, we propose a Curriculum pre-training based HEterogeneous Subgraph Transformer (called *CHEST*) with new *data characterization*, *representation model* and *learning algorithm*. Specifically, we consider extracting useful information from HIN to compose the interaction-specific heterogeneous subgraph, containing highly relevant context information for recommendation. Then, we capture the rich semantics (*e.g.,* graph structure and path semantics) within the subgraph via a heterogeneous subgraph Transformer, where we encode the subgraph into multi-slot sequence representations. Besides, we design a curriculum pre-training strategy to provide an elementary-to-advanced learning process. The elementary course focuses on capturing local context information within the subgraph, and the advanced course aims to learn global context information. In this way, we gradually capture useful semantic information from HIN for modeling user-item interactions. Extensive experiments conducted on four real-world datasets demonstrate the superiority of our proposed method over a number of competitive baselines, especially when only limited training data is available.

CCS Concepts: • **Information systems** → **Recommender systems**.

*Both authors contributed equally to this work.

This work was partially supported by National Natural Science Foundation of China under Grant No. 61872369 and 82161148011, Beijing Natural Science Foundation under Grant No. 4222027, and Beijing Outstanding Young Scientist Program under Grant No. BJJWZYJH012019100020098. This work is also supported by Beijing Academy of Artificial Intelligence (BAAI). Xin Zhao is the corresponding author.
Authors' addresses: Hui Wang, School of Information, Beijing Key Laboratory of Big Data Management and Analysis Methods, Renmin University of China, Beijing, 100872, China, hui.wang@ruc.edu.cn; Kun Zhou, School of Information, Beijing Key Laboratory of Big Data Management and Analysis Methods, Renmin University of China, Beijing, 100872, China, francis_kun_zhou@163.com; Wayne Xin Zhao, Gaoling School of Artificial Intelligence, Beijing Key Laboratory of Big Data Management and Analysis Methods, Renmin University of China, Beijing, 100872, China and Beijing Academy of Artificial Intelligence, Beijing, 100874, China, batmanfly@gmail.com; Jingyuan Wang, School of Computer Science and Engineering, Laboratory for Low-carbon Intelligent Governance, Beihang University, Beijing, 100191, China and Peng Cheng Laboratory, Shenzhen, 518055, China, jywang@buaa.edu.cn; Ji-Rong Wen, Gaoling School of Artificial Intelligence, School of Information, Beijing Key Laboratory of Big Data Management and Analysis Methods, Renmin University of China, Beijing, 100872, China, jrwen@ruc.edu.cn.

1046-8188/2022/7-ART $15.00
https://doi.org/10.1145/3528667

Additional Key Words and Phrases: Curriculum Pre-training, Heterogeneous Information Network, Recommender Systems

1 INTRODUCTION

Online consumption (*e.g.,* purchasing goods and watching movies) has become increasingly popular with the rapid development of Internet services, and users repeatedly encounter the resource selection problem because of information overload [19]. To address such problems, recommender systems (RS) have become an important tool in online platforms [54], which model users' preferences on items based on their past interactions. Due to the complexity of user behavior, recent works utilize various kinds of auxiliary data to improve recommender systems, such as item attributes and user profiles. These auxiliary data can be considered as important context to understand user-item interaction, hence it is essential to effectively utilize such context data to improve the recommendation performance [18, 36, 40]. As a promising approach, heterogeneous information network (HIN) [6, 53], consisting of multiple types of nodes and edges, has been widely applied to model the rich context information in recommender systems [45, 55]. The recommendation task framed in the HIN setting is usually referred to as *HIN-based recommendation* [13, 14, 20].

For HIN-based recommendation, the most essential problem is how to effectively leverage the rich information in HIN for recommendation task. A variety of approaches have been proposed to solve this problem, which roughly falls into two categories, namely path-based methods and graph representation learning methods. Since users and items are connected by paths in HIN, path-based approaches [14, 27] mainly focus on sampling paths from HIN and modeling path-level semantics to characterize the user-item interaction relation. As a widely-used schema, meta-path [45] has been used to extract features for depicting the user-item association. By modeling path-based features via similarity factorization [58] or co-attention model [14], it is able to improve the recommendation performance. On the other hand, graph representation learning methods [15, 22, 47] consider aggregating features from neighbor nodes in the HIN, and leverage the graph structure information to learn the data representations [50, 57]. These methods focus on learning the structural information (*e.g.,* edges) in the graph without considering the downstream tasks, and the user-item association is typically predicted using the representations of the user and the item.

Although existing methods have shown effective to some extent, there are two major challenges that have not been well addressed in HIN-based recommendation. First, HIN characterizes complex, heterogeneous data relations, hence it is difficult to extract sufficient contextual semantics and meanwhile avoid incorporating task-irrelevant information from HIN. Existing approaches either select limited context information from specially designed strategies (*e.g.,* path-based methods) [14, 27], or consider the global view that may incorporate noisy information from task-irrelevant nodes and edges (*e.g.,* graph representation learning methods) [17]. There is a need to consider both *relevance* and *sufficiency* in leveraging HIN information for recommendation. Second, HIN is in essence a general data characterization way, and it is difficult to design suitable learning strategies to derive task-specific data representations for HIN. Existing methods either fully rely on the downstream recommendation task (easy to overfit on training data) [13, 14], or employ task-insensitive pre-training strategies (unaware of the final task goal) [7]. There is a need for a more principled learning algorithm that can more effectively control the learning process with the guidance of the task goal.

To solve the aforementioned issues, we concentrate on user-item interaction to design a systematic approach for HIN-based recommendation. Firstly, we design a more suitable data characterization by introducing *interaction-specific heterogeneous subgraph*, with both sufficient and relevant context information for recommendation. Then, we further develop a *heterogeneous subgraph Transformer* that captures rich semantics from interaction-specific subgraphs for the recommendation task. Furthermore, we propose a *curriculum pre-training* strategy consisting of elementary and advanced courses (*i.e.,* pre-training tasks) to gradually learn from both local and global contexts

in the subgraph tailored to the recommendation task. The above three aspects jointly ensure that our approach can leverage HIN information for recommendation more effectively.

To this end, in this paper, we propose a Curriculum pre-training HEterogeneous Subgraph Transformer (called *CHEST*) for HIN-based recommendation. First, we construct the interaction-specific heterogeneous subgraph consisting of high-quality paths (derived from meta-paths) that connect a user-item pair, which are extracted from HIN but specifically for recommendation task. Then, we propose a heterogeneous subgraph Transformer to encode the subgraphs with multi-slot sequence representations. It consists of a composite embedding layer to map useful contextual information of nodes (*i.e.,* node ID, node type, position in sampled paths, and precursors in the subgraph) into dense embedding vectors and a self-attention layer to aggregate node and subgraph representations. Finally, we devise the curriculum pre-training algorithm with both elementary and advanced courses to gradually learn useful information from the interaction subgraph. The elementary course consists of three pre-training tasks related to node, edge and meta-path, focusing on local context information within the subgraph. The advanced course is a subgraph contrastive learning task, focusing on global context information at the subgraph level for user-item interaction.

To demonstrate the effectiveness of our approach, we conduct extensive experiments on four real-world datasets. It shows that our model is able to outperform all baseline models, including path-based methods and graph representation learning methods. In addition, we perform a series of detailed analyses. We find that our model is robust to the data sparsity problem to some extent, and the learned embeddings obtained by curriculum learning can form meaningful and coherent clusters in the representation space.

Our main contributions are summarized as follows.

- We construct the interaction-specific heterogeneous subgraph to extract useful semantics from HIN related to the correlations between users and items and design the heterogeneous subgraph Transformer to capture useful contextual information from the subgraphs for recommendation task.
- We devise the curriculum pre-training strategy to learn local and global context information within the interaction-specific heterogeneous subgraph, which gradually learns useful evidence for user-item interaction to improve the recommendation task.
- Extensive experiments conducted on four real-world datasets demonstrate the effectiveness of our proposed approach against a number of competitive baselines, especially when only limited training data is available.

We organize the following content as follows: Section 2 discusses the related work of HIN-based recommendation, graph pre-training and curriculum learning. Section 3 and Section 4 introduce the preliminaries and the proposed approach, respectively. We present the experiments in Section 5. Section 6 concludes this research.

2 RELATED WORK

Our work is closely related to the studies on HIN-based recommendation, graph pre-training and curriculum learning.

2.1 HIN-based Recommendation

In the literature on recommender systems, early works mainly adopt collaborative filtering (CF) methods to utilize historical interactions for recommendation [11, 24], where matrix factorization approach [25] and factorization machine [40] have shown effectiveness and efficiency in many applications. Since these methods usually suffer from the cold start problem, many works [63, 64] attempt to leverage additional information to improve recommendation performance, including social relation [33], item reviews [28] and knowledge graph [36, 49].

To effectively utilize the additional information, some works focus on using heterogeneous information network (HIN) [14, 37, 44] in recommender systems. In this way, objects are of different types and edges among objects represent different relations, which naturally characterize complex objects and rich relations.

A mainstream approach is the path-based methods [13, 14, 27], where the semantic associations between two nodes are reflected by the paths that connect them. Various methods are proposed to characterize path-level semantics for recommendation [55]. Early works [45] propose several path-based similarity measures to evaluate the similarity of objects in heterogeneous information networks, which can also be applied in recommendation task. Furthermore, the concept of meta-path is introduced into hybrid recommender systems [56]. Luo et al. [32] propose a collaborative filtering-based social recommendation method using heterogeneous relations. Hu et al. [14] leverage the path-based contextual information to capture user-item correlations.

In recent years, graph representation learning [50, 57] has been introduced to model HINs for improving various downstream applications, including the recommendation task. Typical works adopt graph neural network (GNN) to aggregate the heterogeneous information from adjacent nodes and utilize objectives of general purpose to learn node or graph representations [59]. Zhang et al. [57] propose heterogeneous graph neural network to aggregate feature information of sampled neighboring nodes, and leverage graph context loss to train the model. Wang et al. [50] utilize graph attention network to aggregate features from meta-path based neighbors in a hierarchical manner, which mainly focuses on the node classification task. Wang et al. [52] learn disentangled user/item representations from different aspects in a HIN, which leverages meta relations to decompose high-order connectivity between node pairs. Compared with these studies, our approach combines the merits of path-based methods and graph representation learning methods to learn recommendation-specific data representations.

2.2 Graph Pre-training

Inspired by the success of pre-training methods in computer vision (CV) [42] and natural language processing (NLP) [4], the pre-training technique has been recently applied to graph datasets for improving GNNs [16, 31]. The purpose of pre-training graph neural networks is to learn the parameters of the model for producing general graph representations, which can be further fine-tuned on different downstream tasks. It has been shown that pre-training methods have the potential to address scarce labeled data [12] and out-of-distribution prediction [15].

As an effective unsupervised pre-training strategy, mutual information maximization [23] has been utilized to capture the correlations within the graph (*e.g.,* nodes, edges and subgraphs) [48]. Velickovic et al. [48] propose a graph information maximization method to learn node representations, which can better capture global structural properties of the graph. Ren et al. [39] explore mutual information maximization for heterogeneous graph representation learning, which focuses on learning high-level representations based on meta-path. Hu et al. [15] pre-train an expressive GNN at the level of individual nodes as well as the entire graph, so that the GNN can learn useful local and global representations simultaneously. Lu et al. [31] further attempt to learn how to fine-tune models during the pre-training stage, and design a dual adaptation mechanism to encode both local and global information as the transferable prior knowledge.

Besides, contrastive learning and graph generation strategies are also utilized to pre-train GNNs. Qiu et al. [38] utilize contrastive learning to capture the universal network topological properties across multiple graphs, which empowers graph neural networks to learn the intrinsic and transferable structural representations. Zhu et al. [65] develop a framework for unsupervised graph representation learning by leveraging contrastive learning with augmentations, which can produce graph representations of better generalizability, transferability and robustness. Hu et al. [16] introduce a self-supervised attributed graph generation task to pre-train the GNN so that it can capture the structural and semantic properties of the graph.

Generally, most of these methods aim to learn general node representations based on the whole graph. As a comparison, we propose a curriculum pre-training strategy to learn recommendation-specific representations, which is able to gradually extract useful information from HIN for recommendation task.

2.3 Curriculum Learning

Inspired by the human learning process, curriculum learning [1] is proposed as a learning paradigm that starts from simple patterns and gradually increases to more complex patterns. Several studies [9, 29] have shown that this training approach results in better generalization and speeds up the convergence.

Most of the works [3, 9, 29] on curriculum learning focus on feeding training instances to the model from easy to hard. Guo et al. [9] utilize curriculum learning in the image classification task and show its effectiveness. Liu et al. [29] employ two practical measurements to automatically measure the difficulty of question-answer pairs and improve the performance of question answering. Chu et al. [3] combine curriculum learning and contrastive learning to pre-train graph-level representations, which samples negatives from easier to harder for contrastive learning. Recently, some studies [30, 43] also explore the curriculum learning strategies at the task level, and show that a group of well-designed curriculums are helpful to improve the generalization capacity and convergence rate of various models. Guo et al. [8] train the model with sequentially increased degrees of parallelism to train the model from easy to hard, which achieves significant accuracy improvements over previous non-autoregressive neural machine translation methods. Sarafianos et al. [43] group individual tasks into hierarchical clusters based on their correlation and utilize curriculum learning to transmit the acquired knowledge between clusters.

In this work, we design a curriculum pre-training strategy to gradually learn both local and global contexts from the subgraph, which helps our model to leverage HIN information for recommendation more effectively.

3 PROBLEM FORMULATION

A heterogeneous information network (HIN) is a special kind of information network, which contains multiple types of objects and multiple types of edges. In this work, we consider the recommendation task in the setting of heterogeneous information network.

Definition 1. ***Heterogeneous Information Network (HIN).*** A HIN [6, 53] is defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, in which $\mathcal{V}$ and $\mathcal{E}$ are the sets of nodes and edges, respectively. Each node v and edge e are associated with their type mapping functions $\phi : \mathcal{V} \longrightarrow \mathcal{A}$ and $\varphi : \mathcal{E} \longrightarrow \mathcal{R}$, respectively, where $\mathcal{A}$ and $\mathcal{R}$ denote the sets of pre-defined node and edge types, where $|\mathcal{A}| + |\mathcal{R}| > 2$.

Recently, HIN has become a mainstream approach to modeling various complex interaction systems [62]. Especially, it has been adopted in recommender systems for characterizing complex and heterogeneous recommendation settings. Based on the above preliminaries, we define our task as follows.

Definition 2. ***HIN-based Recommendation.*** In a recommender system, various kinds of information can be modeled by a HIN $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. On recommendation-oriented HINs, two kinds of entities (*i.e.,* users and items) together with the relations between them (*i.e.,* rating relation) are our focus. Let $\mathcal{U} \subset \mathcal{V}$ and $\mathcal{I} \subset \mathcal{V}$ denote the sets of users and items respectively, for each user $u \in \mathcal{U}$, our task is to recommend a ranked list of items that are of interest to u based on her/his historical record $\mathcal{I}_u$, where $\mathcal{I}_u \subset \mathcal{I}$ denotes the set of items that u has interacted with before.

In HIN, two objects can be connected via different semantic patterns, which are defined as *meta-paths* [45].

Definition 3. ***Meta-path.*** A meta-path is defined as a path in the form of $o_1 \xrightarrow{r_1} o_2 \xrightarrow{r_2} \ldots \xrightarrow{r_l} o_{l+1}$ (abbreviated as $o_1 o_2 \ldots o_{l+1}$), which describes a composite relation $r_1 \circ r_2 \circ \cdots \circ r_l$ between object o_1 and o_{l+1}, where "$\circ$" denotes the composition operator on relations.

For a meta-path ρ, there exist multiple specific paths following the meta-path, which are called *path instances* denoted by p. For example, in Figure 1, user u_1 can be connected to item i_1 through the paths u_1-i_2-u_2-i_1, u_1-i_3-a_1-i_1 and u_1-i_3-a_2-i_1, which correspond to meta-paths "$UIUI$" or "$UIAI$". These paths reflect potential associations between two nodes in HIN. In our task, we mainly focus on the meta-paths starting with a user node and ending with an item node.

Table 1. Notations and explanations

Notation	Explanation
$\mathcal{G}$	heterogeneous information network
$\mathcal{G}_{u,i}$	a heterogeneous subgraph connecting user-item pair $\langle u, i \rangle$
$\mathcal{V}$	the set of nodes
$\mathcal{E}$	the set of edges
$\mathcal{A}$	the set of pre-defined entity types
$\mathcal{R}$	the set of pre-defined edge types
$\mathcal{U}$	the set of users
$\mathcal{I}$	the set of items
$\mathcal{I}_u$	the set of items that u has interacted with before
$\mathcal{S}$	the set of slots
$\mathcal{P}$	the set of meta-paths
u	a user
r	a relation
o	an object
a	an attribute
ρ	a meta-path
p	a path instance
i	an item
$i^{'}$	a random sampled negative item
v	a node
C_{v_t}	the surrounding context for v_t in a heterogeneous subgraph
$\texttt{Pr}(\rho \vert u, i)$	the preference score of user u and item i
σ	the sigmoid function
M_V, M_A, M_S, M_P	the embedding matrices of node ID, node type, slot and precursor
E_V, E_A, E_S, E_P	the embedding matrices of node ID, node type, slot and precursor for a heterogeneous subgraph
E	the composite embedding matrix of a heterogeneous subgraph
W^O, W_i^Q, W_i^K, W_i^V	learnable parameter matrices in multi-head self-attention layer
W_1, W_2	learnable parameter matrices in point-wise feed-forward network
W_N, W_E	learnable parameter matrices for masked node/edge prediction task
b_1, b_2	learnable parameter vectors
F^l	the input of the l-th layer
F_u^L, F_i^L	the representations of user u and item i from the last self-attention layer
e_v	the node ID embedding of node v
$z_{\mathcal{G}}$	the subgraph representation
$head_i$	the output of the i-th head of self-attention layer
d	the embedding dimension
L	the number of layers in the Transformer model
n	the number of nodes in the subgraph
h	the number of heads in the multi-head self-attention layer
τ	the hyper-parameter for softmax temperature

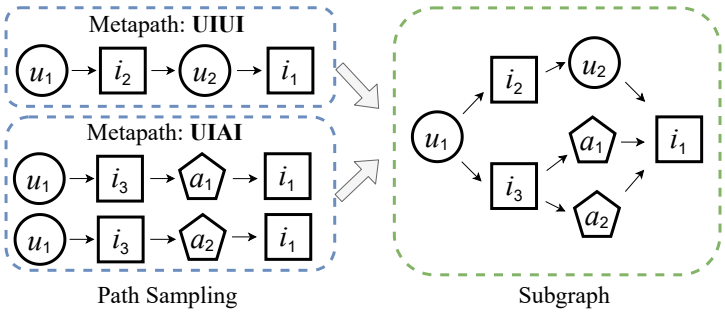


Fig. 1. The illustration of the interaction-specific heterogeneous subgraph. The subgraph is constructed by path instances based on meta-paths.

Next, we will present a new curriculum pre-training based heterogeneous subgraph Transformer for this task, which is able to effectively leverage the information reflected in HINs. The notations that we will use throughout the article are summarized in Table 1.

4 APPROACH

In this paper, we propose a novel Curriculum pre-training based HEterogeneous Subgraph Transformer (called as *CHEST*) to effectively utilize HIN information for improving the recommendation performance. Tailored to the recommendation task, we first construct an interaction-specific heterogeneous subgraph to extract useful contextual information from HIN for the user-item pair, and then design a heterogeneous subgraph Transformer to encode this subgraph. Finally, we introduce a curriculum pre-training strategy to learn recommendation-specific representations. Figure 2 presents the overall illustration of the proposed CHEST approach. Next, we describe each part in detail.

4.1 Constructing Interaction-Specific Heterogeneous Subgraph

In our task, it is essential to leverage useful semantics from HIN to capture the connections between users and items for effective recommendation. Different from prior studies [44, 50], we collect the most relevant paths that connect the two nodes. Then, these paths (including nodes and edges) compose a heterogeneous subgraph for the user-item pair $\langle u, i \rangle$, denoted by $\mathcal{G}_{u,i}$. We expect such a subgraph to contain most of the relevant context information for a specific user-item interaction.

To derive relevant and reliable paths between two nodes, following existing works [14, 27], we pre-define multiple meta-paths to guide the selection of paths. Specifically, we first use metapath2vec [5] to learn the latent vectors for all nodes. Then, given the user-item pair, we start from the user node to find the path instance connecting them. At each step, we obtain the "priority" scores by computing the embedding similarity between the current node and its neighbors, and then sample the next-hop node from the neighboring nodes according to the "priority" score. For each meta-path, we sample $2 \times K$ path instances as candidates. Finally, we rank these candidate paths based on the average cosine similarity between the latent vectors of two consecutive nodes on it, and only keep top-K path instances with the highest average similarities for each meta-path to compose the interaction-specific heterogeneous subgraph.

In Figure 1, we present an example for our interaction-specific heterogeneous subgraph for user u_1 and item i_1, where we consider two types of meta-paths "*UIUI*" or "*UIAI*". For each meta-path, we obtain the corresponding path instances from the HIN by the "priority"-based sampling strategy. In detail, we acquire the paths u_1-i_2-u_2-i_1, u_1-i_3-a_1-i_1 and u_1-i_3-a_2-i_1 connecting the user-item pair $\langle u_1, i_1 \rangle$ according to meta-paths "*UIUI*" and "*UIAI*",

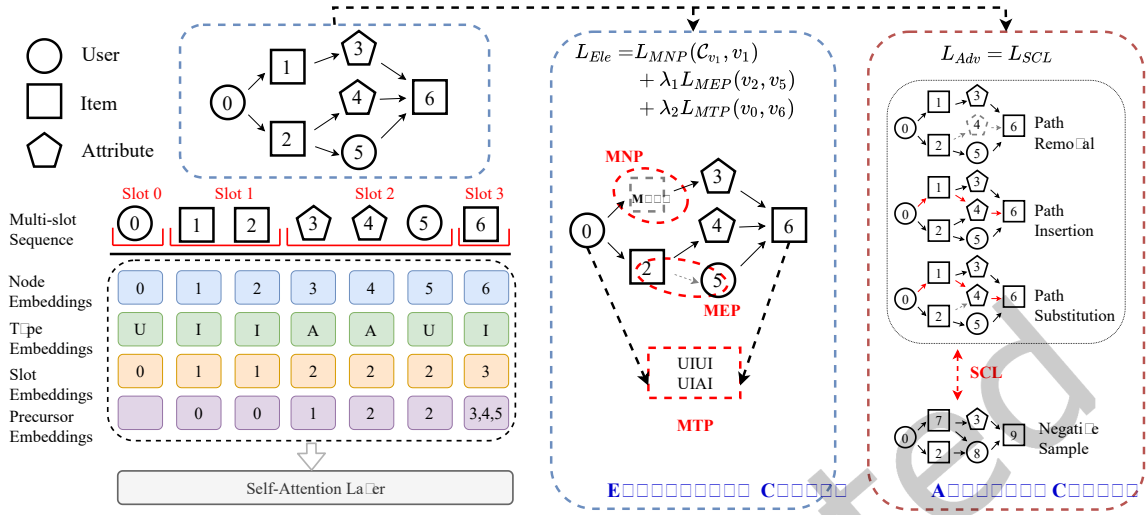


Fig. 2. The overview of our proposed Transformer model and curriculum pre-training strategy. The elementary courses consist of three pre-training tasks: (1) Masked Node Prediction (MNP), (2) Masked Edge Prediction (MEP) and (3) Meta-path Type Prediction (MTP). And the advanced course is the Subgraph Contrastive Learning (SCL) task.

respectively. Finally, we re-connect all the nodes with the edges in these paths and produce the interaction-specific heterogeneous subgraph as the right part of Figure 1. With heterogeneous subgraphs, we can explicitly keep the semantics of multiple meta-paths and model the correlations among nodes across different paths. It is more efficient to aggregate neighboring node information within a compact, relevant subgraph than the entire graph [17, 57], since most irrelevant nodes in HIN are excluded through the "priority"-based sampling strategy.

4.2 Heterogeneous Subgraph Transformer

Given the interaction-specific heterogeneous subgraph for a special user-item pair, we design the heterogeneous subgraph Transformer to capture useful semantics within it, which consists of an embedding layer and multiple self-attention layers.

4.2.1 Embedding Layer. Unlike the embedding mechanism in BERT [4] for sequences, we need to effectively model the nodes and edges in the subgraph. To preserve the rich structure semantics in subgraphs, a key point is how to model the position information (*i.e.,* location) of a node and its links with other nodes in the subgraph. For this purpose, we first assign a slot index to a node according to the relative position *w.r.t.* the target user node in sampled paths. To be specific, the slot index of the starting user node is assigned to zero, and the index of the other node is set as its minimum distance with the starting node among multiple involved paths in a subgraph. In this way, each node is placed according to its slot index and the original subgraph will be converted into a multi-slot sequence. To model the edges in subgraphs, we further incorporate a precursor index to trace the precursor in paths for a node. To facilitate the multi-slot sequence representation, we incorporate four types of node embeddings to preserve the characteristics of the subgraph :

• *Node ID Embedding*: For each node v in the heterogeneous subgraph $\mathcal{G}_{u,i}$, we maintain an ID embedding matrix $\mathbf{M}_V \in \mathbb{R}^{|\mathcal{V}| \times d}$, which projects the high-dimensional one-hot ID representation of a node into low-dimensional dense representation.

• *Node Type Embedding*: In HIN, each node is associated with a specific node type. Therefore, we also maintain a node type embedding matrix $\mathbf{M}_A \in \mathbb{R}^{|\mathcal{A}|\times d}$ to project the one-hot node type representation into dense representation.

• *Slot Embedding*: The interaction-specific heterogeneous subgraph is composed of multiple paths that connect the user (starting node) and the item node (ending node). In these paths, the distance between two nodes can reflect their semantic relationship. As to the starting user node, its distance with an other node (slot index) is able to depict the user's preference on it, which is beneficial for the personalized recommendation. Therefore, we consider designing the slot embeddings to represent the above characteristics. Since we have assigned a slot index for each node according to the relative position in the involved paths, we use a slot embedding matrix $\mathbf{M}_S \in \mathbb{R}^{|\mathcal{S}|\times d}$ to project the slot index of nodes into corresponding representations, where $|\mathcal{S}|$ is the number of slots in the subgraph.

• *Precursor Embedding*: Although the slot embedding has modeled the relative distance from the starting user node, the adjacent relations between two consecutive nodes in the subgraph have not been represented. Hence, we further add precursor indices to record the preceding nodes for each node in the subgraph. We maintain a precursor embedding matrix $\mathbf{M}_P \in \mathbb{R}^{n\times d}$ to project the precursor indices of each node into embeddings, where n is the maximum number of nodes in the subgraph. Since a node may have multiple precursors, we average the embeddings of the precursor indices as a single vector.

Based on the above embeddings, we aggregate them together to produce the subgraph representation in a multi-slot sequence form. Formally, the representation of nodes in the subgraph is a embedding matrix $\mathbf{E} \in \mathbb{R}^{|\mathcal{N}|\times d}$, which is composed of four parts:

$$\mathbf{E} = \mathbf{E}_V + \mathbf{E}_A + \mathbf{E}_S + \mathbf{E}_P, \tag{1}$$

where the four matrices $\mathbf{E}_V$, $\mathbf{E}_A$, $\mathbf{E}_S$ and $\mathbf{E}_P$ denote the node ID embedding, node type embedding, slot embedding and precursor embedding, respectively. These embeddings are obtained by the look-up operation from $\mathbf{M}_V$, $\mathbf{M}_A$, $\mathbf{M}_S$ and $\mathbf{M}_P$, respectively. It is worth noting that through the above representations, the heterogeneous (*e.g.,* node type), path-level (*e.g.,* position in the path) and graph-structure (*e.g.,* edges in the subgraph) information from subgraph $\mathcal{G}_{u,i}$ have been encoded in the composite embedding matrix $\mathbf{E}$.

4.2.2 Self-Attention Layer. Similar to the architecture of Transformer [46], based on the embedding layer, we develop the subgraph encoder by stacking multiple self-attention layers. A self-attention layer generally consists of two sub-layers, *i.e.,* a multi-head self-attention layer and a point-wise feed-forward network. Specifically, the multi-head self-attention is defined as:

$$\text{MHAttn}(\mathbf{F}^l) = [head_1, head_2, ..., head_h]\mathbf{W}^O, \tag{2}$$

$$head_i = \text{Attention}(\mathbf{F}^l\mathbf{W}_i^Q, \mathbf{F}^l\mathbf{W}_i^K, \mathbf{F}^l\mathbf{W}_i^V), \tag{3}$$

where the $\mathbf{F}^l$ is the input for the l-th layer, when $l = 0$, we set $\mathbf{F}^0 = \mathbf{E}$, and the projection matrix $\mathbf{W}_i^Q \in \mathbb{R}^{d\times d/h}$, $\mathbf{W}_i^K \in \mathbb{R}^{d\times d/h}$, $\mathbf{W}_i^V \in \mathbb{R}^{d\times d/h}$ and $\mathbf{W}^O \in \mathbb{R}^{d\times d}$ are the corresponding learnable parameters for each attention head. The attention function is implemented by scaled dot-product operation:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^\top}{\sqrt{d/h}})\mathbf{V}, \tag{4}$$

where $\mathbf{Q} = \mathbf{F}^l\mathbf{W}_i^Q$, $\mathbf{K} = \mathbf{F}^l\mathbf{W}_i^K$, and $\mathbf{V} = \mathbf{F}^l\mathbf{W}_i^V$ are the linear transformations of the input embedding matrix, and $\sqrt{d/h}$ is the scale factor to avoid large values of the inner product.

After the multi-head attention layer, we endow the non-linearity of the self-attention layer by applying a point-wise feed-forward network. The computation is defined as:

$$\mathbf{F}^l = [\text{FFN}(\mathbf{F}_1^l)^\top; \cdots; \text{FFN}(\mathbf{F}_n^l)^\top], \tag{5}$$

$$\text{FFN}(x) = (\text{ReLU}(x\mathbf{W}_1 + \mathbf{b}_1))\mathbf{W}_2 + \mathbf{b}_2, \tag{6}$$

where $\mathbf{W}_1$,$\mathbf{b}_1$,$\mathbf{W}_2$,$\mathbf{b}_2$ are trainable parameters.

Finally, we can compute the representation for the interaction-specific heterogeneous subgraph $\mathcal{G}_{u,i}$ based on the representations at the final self-attention layer as:

$$\mathbf{z}_{\mathcal{G}_{u,i}} = \text{MLP}(\mathbf{F}_u^L \oplus \mathbf{F}_i^L), \tag{7}$$

where "$\oplus$" denotes the vector concatenation operation, $\mathbf{F}_u^L$ and $\mathbf{F}_i^L$ are the representations of user u and item i from the last self-attention layer, which represent the starting user u and ending item i in the subgraph, and L is the number of self-attention blocks.

4.3 Curriculum Pre-training

With the above model architecture, we focus on developing an effective representation learning approach that is special for HIN-based recommendation. Considering that HIN encodes complex and heterogeneous data relations, our idea is to gradually extract and learn useful information from *local* (*e.g.,* node-level) to *global* (*i.e.,* subgraph-level) context from interaction-specific heterogeneous subgraphs. Such an idea can be in essence characterized by *curriculum learning* [1], which starts from simple tasks or instances and gradually transforms to more complex ones [30, 43]. Based on this idea, we develop a novel curriculum pre-training strategy that designs both elementary and advanced courses (*i.e.,* pre-training tasks) with *increasing difficulty levels*.

4.3.1 Elementary Course. The elementary course aims to learn local context information from interaction-specific heterogeneous subgraphs. We propose to train the proposed heterogeneous subgraph Transformer model with three new tasks, namely masked node prediction, masked edge prediction and meta-path type prediction. The first two tasks focus on enhancing the node-level representations, while the meta-path type prediction task is designed for capturing path-level semantics for user-item interactions.

• *Masked Node Prediction*: This task is to infer a masked node based on its surrounding context in a heterogeneous subgraph. Following the *Cloze* task in BERT [4], we randomly mask a proportion of nodes in a heterogeneous subgraph and then predict the masked nodes based on the remaining contexts. As for the representation of the masked node, similar to the mask operation in BERT, we only remove its node id embedding but keep other embeddings. Besides, for all the nodes with the masked node as the precursor, we also remove the masked node from their precursor embeddings. Such an operation is able to prevent the masked node from affecting the representations of other nodes. Assume that we mask node v_t in a multi-slot sequence $\{v_1, \cdots, v_t, \cdots, v_n\}$. We treat the rest sequence $\{v_1, \cdots, \text{MASK}, \cdots, v_n\}$ as the surrounding context for v_t, denoted by C_{v_t}. Given the surrounding context C_{v_t} and the masked node v_t, we minimize the Masked Node Prediction (MNP) loss by:

$$L_{MNP}(C_{v_t}, v_t) = -\log\Big(\sigma(\mathbf{F}_t^\top \mathbf{W}_N \mathbf{e}_{v_t}) - \sigma(\mathbf{F}_t^\top \mathbf{W}_N \mathbf{e}_{\tilde{v}})\Big), \tag{8}$$

where $\tilde{v}$ denotes an irrelevant node, $\mathbf{e}_{v_t}$ and $\mathbf{e}_{\tilde{v}}$ denote the node ID embedding for v_t and $\tilde{v}$ respectively, $\mathbf{W}_N \in \mathbb{R}^{d\times d}$ is a parameter matrix to learn and $\mathbf{F}_t$ is the learned representation for the t-th position using our subgraph encoder as in Eq. 5.

• *Masked Edge Prediction*: The masked edge prediction task is to recover the masked edge of two adjacent nodes based on the surrounding context. Similar to masked node prediction, we randomly mask a proportion of edges in the input (*i.e.,* removing the precursor index) and then predict the masked edges based on the surrounding

contexts. In practice, if we mask the edge $\langle v_j, v_k \rangle$, we only need to remove v_j from the precursors of v_k. Formally, the Masked Edge Prediction (MEP) loss for the edge $\langle v_j, v_k \rangle$ can be given as:

$$L_{MEP}(v_j, v_k) = -\log\Big(\sigma(\mathbf{F}_j^\top \mathbf{W}_E \mathbf{F}_k) - \sigma(\mathbf{F}_j^\top \mathbf{W}_E \mathbf{F}_{k'})\Big), \tag{9}$$

where $v_{k'}$ is a sampled node that is not adjacent to v_j, $\mathbf{W}_E \in \mathbb{R}^{d \times d}$ is a parameter matrix to learn, $\mathbf{F}_j$, $\mathbf{F}_k$ and $\mathbf{F}_{k'}$ are the learned representations for the corresponding positions obtained in the same way as Eq. 5.

• *Meta-path Type Prediction*: Since the user-item interaction subgraph is composed of multiple paths, path-level semantics encode important evidence to explain the underlying reasons why a specific user-item interaction occurs [13, 14]. We would like to directly capture the semantics from meta-paths for improving the path semantic information in representations. Specifically, we consider meta-path type prediction as a classification task and introduce the Meta-path Type Prediction (MTP) loss as:

$$L_{MTP}(u, i) = -\sum_{\rho \in \mathcal{P}} \Big(y_{u,i,\rho} \cdot \log \Pr(\rho|u, i) + (1 - y_{u,i,\rho}) \cdot \log(1 - \Pr(\rho|u, i))\Big), \tag{10}$$

where $y_{u,i,\rho}$ is a binary label indicating whether there exists a path from the meta-path ρ between u and i, $\mathcal{P}$ is the meta-path set, and $\Pr(\rho|u, i)$ is the probability that the user and item are connected by the meta-path ρ, which is defined as:

$$\Pr(\rho|u, i) = \sigma(\text{MLP}(\mathbf{F}_u^L \oplus \mathbf{F}_i^L)), \tag{11}$$

where $\text{MLP}(\cdot)$ is a multi-layer perceptron and σ is the sigmoid function.

4.3.2 Advanced Course. Although the above pre-training tasks have captured local context information (*e.g.,* node, edge and path) from the heterogenous subgraph, the global correlations at the subgraph level cannot be effectively learned by the elementary course. To characterize the overall effect of global contexts on recommendation, we devise an advanced course to train the heterogeneous subgraph Transformer with a *Subgraph Contrastive Learning (SCL)* task. Based on the original subgraph, the core idea is to augment a number of interaction-specific subgraphs. Then, we apply contrastive learning [2, 10] to further capture subgraph-level evidence for modeling user-item interaction. Here, we consider three path-based subgraph augmentation strategies:

• *Path Removal*: It augments new subgraphs by randomly removing a small portion of paths from the original user-item interaction subgraph, which is expected to make the learned representations less sensitive to structural perturbation and improve their robustness.

• *Path Insertion*: It introduces a small proportion of new paths into the original subgraph. In this way, the edges in these new paths will be inserted into the subgraph, which can also improve the robustness of our model to resist noisy graph structure information.

• *Path Substitution*: It can be considered as the combination of the path removal and path insertion strategies, where we substitute a proportion of paths with new paths. In this way, we further enlarge the difference between the augmented subgraphs and the original subgraph, and enforce the model to capture more fundamental semantics for user-item interactions.

Given the target user-item subgraph $\mathcal{G}_{u,i}$ (focusing on user u and i), we first augment a new subgraph with the above subgraph augmentation strategies, and consider them as *positive subgraph*, denoted by $\mathcal{G}_{u,i}^+$. While, we consider the subgraphs connecting the same user u with other items as *negative subgraphs*, denoted by $\{\mathcal{G}_{u,i}^-\}$. Following a standard constative learning approach [2], we maximize the difference of augmented positive subgraph and negative subgraphs, *w.r.t.* the original subgraph:

$$L_{SCL}(\mathcal{G}, \mathcal{G}^+, \{\mathcal{G}^-\}) = -\log \frac{\exp\Big(\text{sim}(\mathbf{z}_{\mathcal{G}}, \mathbf{z}_{\mathcal{G}^+})/\tau\Big)}{\exp\Big(\text{sim}(\mathbf{z}_{\mathcal{G}}, \mathbf{z}_{\mathcal{G}^+})/\tau\Big) + \sum_{\mathcal{G}^-} \exp\Big(\text{sim}(\mathbf{z}_{\mathcal{G}}, \mathbf{z}_{\mathcal{G}^-})/\tau\Big)}, \tag{12}$$

where $z_{\mathcal{G}}$, $z_{\mathcal{G}^+}$ and $z_{\mathcal{G}^-}$ are the produced subgraph representations from the heterogeneous subgraph Transformer (Eq. 7) for the original subgraph, augmented positive subgraph and augmented negative subgraph (we omit u and i in subscripts for simplicity), respectively, $\texttt{sim}(\mathbf{x}, \mathbf{y})$ denotes the cosine similarity function, and τ is a hyper-parameter for softmax temperature.

This constative learning loss enforces the model to learn subgraph-level semantics for user-item interaction. By combining with the elementary course, both local and global context information can be captured in final learned representations. In particular, we schedule the pre-training tasks from two courses in an "*easy-to-difficult*" order, which is necessary to model complex data relations in HIN.

4.4 Learning and Discussion

In this part, we present the learning and related discussions of our approach for HIN-based recommendation.

4.4.1 Learning. The entire procedure of our approach consists of two major stages, namely curriculum pre-training and fine-tuning stages. At the curriculum pre-training stage, we first pre-train our model on the elementary course, consisting of three pre-training objectives to learn local context information in the subgraph, then pre-train on the advanced course to learn global context information from HIN. At the fine-tuning stage, we utilize the pre-trained parameters to initialize the parameters, and then adopt the recommendation task to train our model. Given user u and item i, the preference score is calculated by:

$$\Pr(u, i) = \sigma(\mathbf{z}_{\mathcal{G}_{u,i}}), \tag{13}$$

where $\sigma(.)$ is the sigmoid function and $\mathbf{z}_{\mathcal{G}_{u,i}}$ is the representation for $\mathcal{G}_{u,i}$ defined in Eq. 7. We adopt the binary cross-entropy loss as the final objective:

$$L_{rec}(u, i) = -\log \Pr(u, i) - \log(1 - \Pr(u, i')), \tag{14}$$

where we pair each ground-truth item i with one (or several) negative item i' that is randomly sampled. The detailed learning process is shown in Algorithm 1.

4.4.2 Time complexity. In recommender systems, the online service time is more important to consider than offline training time. Once our model has been learned (after pre-training and fine-tuning), online service time mainly includes the cost of evaluating all the candidate items according to Eq. 13 and the cost of selecting top items, which is similar to previous neural collaborative filtering methods [11]. A major preprocessing cost lies in the construction of heterogeneous subgraphs for possible user-item pairs. As discussed before, we can pre-compute the priority scores of neighbors for all the nodes. Based on priority scores, we can sample a high-quality path instance in a time roughly as $O(\bar{L})$ using pre-built efficient data structures such as alias table [26] (taking time $O(1)$ to sample from categorical distributions), where $\bar{L}$ is the average path length. In practice, the number of meta-paths and the number of paths in a subgraph are usually set to small values, so that the number of nodes in a subgraph can be bounded below a reasonable value (*e.g.,* 50). In this way, our pre-training and fine-tuning costs are similar to train/pre-train Transformer architecture [4, 46] over sequence data, which can be efficient if we use very few self-attention layers or parallelize the computation.

4.4.3 Discussion. Compared with existing work for HIN-based recommendation, our approach has two major differences. In our approach, data characterization, representation model and learning algorithm are specially designed for user-item interaction based on HIN. As for *data characterization*, we introduce interaction-specific heterogeneous subgraph to reduce the incorporation of irrelevant information. Based on such a subgraph structure, we further propose a novel heterogeneous subgraph Transformer as the *representation model*, which can effectively model the subgraph semantics. Furthermore, we propose a novel *learning algorithm* by designing a curriculum pre-training approach, in which elementary and advanced courses are organized to gradually extract local and

Algorithm 1: The overall training process for the CHEST model.

Input: The heterogeneous information network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, pre-defined meta-paths $\mathcal{P}$, the user set $\mathcal{U}$, the item set $\mathcal{I}$, the user-item historical records $\mathcal{D} = \langle u, i \rangle$
Output: The learned node embedding matrix $\mathbf{E}$, the learned parameters of the self-attention layer Θ

1 Use metapath2vec to learn latent vectors of all the nodes in $\mathcal{G}$.
2 **for** $j = 1 \rightarrow |\mathcal{U}|$ **do**
3 **for** $k = 1 \rightarrow |\mathcal{I}|$ **do**
4 **for** $l = 1 \rightarrow |\mathcal{P}|$ **do**
5 Collect the top-K path instances with the highest average similarities corresponding to the meta-path ρ_l that connect the user node u_j and item nodes i_k.
6 **end**
7 Merge the collected path instances into a subgraph $\mathcal{G}_{u_j, i_k}$.
8 **end**
9 **end**
10 Randomly initialize $\mathbf{E}$ and Θ.
11 // Pre-training parameters on the elementary course.
12 **for** $j = 1 \rightarrow |\mathcal{U}|$ **do**
13 **for** $k = 1 \rightarrow |\mathcal{I}|$ **do**
14 Transform the subgraph $\mathcal{G}_{u_j, i_k}$ for the user-item pair into multi-slot sequence.
15 Acquire ID embeddings $\mathbf{E}_V$, node type embeddings $\mathbf{E}_A$, slot embeddings $\mathbf{E}_S$ and precursor embeddings $\mathbf{E}_P$ for the nodes in the subgraph $\mathcal{G}_{u_j, i_k}$.
16 Acquire the composite embedding matrix $\mathbf{E}$ using Eq. 1.
17 Acquire the subgraph representations $\mathbf{F}^L$ by multiple self-attention layers using Eq. 2, Eq. 3, Eq. 5, Eq. 6 and Eq. 7.
18 Pre-train the parameters $\mathbf{E}$ and Θ using Eq. 8, Eq. 9 and Eq. 10.
19 **end**
20 **end**
21 // Pre-training parameters on the advanced course.
22 **for** $j = 1 \rightarrow |\mathcal{U}|$ **do**
23 **for** $k = 1 \rightarrow |\mathcal{I}|$ **do**
24 Randomly select a subgraph augmentation strategy from the three path-based subgraph augmentation strategies (Path Removal, Path Insertion and Path Substitution).
25 Augment a new positive subgraph $\mathcal{G}^+_{u,i}$ for each subgraph $\mathcal{G}_{u,i}$.
26 Generate the representations of the subgraphs using the operations from line 14 to line 17.
27 Pre-train the parameters $\mathbf{E}$ and Θ using Eq. 12.
28 **end**
29 **end**
30 // Fine-tuning parameters on the recommendation task.
31 **for** $t = 1 \rightarrow |\mathcal{D}|$ **do**
32 Encode the subgraph $\mathcal{G}_{u_j, i_k}$ using the operations from line 14 to line 17.
33 Compute $\Pr(u, i)$ using Eq. 13.
34 Fine-tune the parameters $\mathbf{E}$ and Θ using Eq. 14.
35 **end**
36 **return** $\mathbf{E}$ and Θ.

global context information from HIN to recommendation task. The three aspects jointly ensure that our approach can better extract and leverage relevant contextual information from HIN for modeling user-item interaction.

Our work is related to two categories of models, namely path-based methods [14, 20, 27] and graph representation learning methods [17, 50, 57]. The former category separately models the sampled paths, so that graph structure or cross-path node correlation can not be explicitly captured. Besides, these path-based methods rely on the recommendation task to learn the representations, which may suffer from data sparsity problem and cause overfitting. As a comparison, our approach constructs an interaction-specific heterogeneous subgraph based on high-quality paths, which is able to capture richer semantics from the subgraph structure. In addition, we propose an elementary-to-advanced curriculum pre-training strategy to gradually learn from both local and global contexts in the subgraph, which is able to learn more effective representations. Graph representation learning methods aggregate information from neighboring nodes in the entire HIN and then learn the representations via task-insensitive objectives. In this way, noisy or irrelevant information is likely to be incorporated into the learned representations, without consider the goal of the recommendation task. In our approach, the interaction-specific heterogeneous subgraph is utilized to characterize high-quality context information, which effectively reduces the influence of irrelevant nodes and edges. Then, we design a curriculum pre-training strategy based on the heterogeneous subgraph to learn the user-item association, which is more suitable for the recommendation task.

5 EXPERIMENT

In this section, we first set up the experiments, and then present the results and detailed analyses.

5.1 Experimental Setup

5.1.1 Datasets. We conduct experiments on four widely-used datasets from different domains, namely Movielens, Douban, Yelp and AMiner (a sub-dataset of collaboration network), where movies, movies, businesses and papers are considered as items for recommendation, respectively. For reproducible comparison, we reuse the preprocessed results and the selected meta-paths released in [13, 20][1]. We treat a rating as an interaction record, indicating whether a user has rated an item or not. The detailed statistics of these datasets after preprocessing are summarized in Table 2, where we report the statistics by different edge relations. The first row of each dataset corresponds to the number of users, items and interactions, while the other rows correspond to the statistics of other relations. The selected meta-paths for each dataset are in the last column.

5.1.2 Evaluation Metrics. We use three commonly used metrics to evaluate the performance of our proposed model.

• Hit Rate: Hit rate (HR) measures the percentage that recommended items contain at least one correct item interacted by the user, which does not consider the actual rank of the items and has been widely used in previous works and is defined as:

$$\text{HR@}k = \frac{1}{|\mathcal{U}|} \sum_{u \in \mathcal{U}} \mathbb{I}(|\hat{\mathcal{I}}_{u,k} \cap \mathcal{I}_u| > 0), \tag{15}$$

where $\hat{\mathcal{I}}_{u,k}$ denotes the set of top-k recommended items for user u and $\mathcal{I}_u$ is the set of testing items for user u, and $\mathbb{I}(\cdot)$ is an indicator function.

• Normalized Discounted Cumulative Gain: Normalized Discounted Cumulative Gain (NDCG) takes the positions of correct recommended items into consideration, which is important in settings where the order of

[1] https://github.com/librahu/HIN-Datasets-for-Recommendation-and-Network-Embedding

Table 2. Basic statistics of the four datasets.

Datasets	Relations	#Type A	#Type B	#A-B	Meta-path
Movielens	User-Movie	943	1,682	100,000	UMUM
	Movie-Movie	1,682	1,682	82,798	UMMM
	User-Occupation	943	21	943	UOUM
	Movie-Genre	1,682	18	2,861	UMGM
Douban	User-Movie	13,367	12,677	1,068,378	UMUM
	Movie-Actor	12,677	6,311	33,572	MAMA
	Movie-Director	12,677	2,449	11,276	MDMD
	Movie-Type	12,677	38	27,668	MTMT
Yelp	User-Business	16,239	14,284	198,397	UBUB
	User-User	16,239	16,239	158,590	UUUB
	Business-City	14,284	47	14,267	UBCiB
	Business-Category	14,284	511	40,009	UBCaB
AMiner	Author-Paper	31,664	103,470	196,170	APAP
	Paper-Conference	103,470	101	103,470	APCP
	Paper-Label	103,470	10	103,190	APLP
	Paper-Year	103,470	46	103,470	APYP

recommendations matters and is defined as:

$$\text{NDCG@}k = \frac{\text{DCG@}k}{\text{iDCG}},$$
$$\text{DCG@}k = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}\sum_{j=1}^{k}\frac{\mathbb{I}(\hat{\mathcal{I}}_{u,j}\in\mathcal{I}_u)}{\log_2(j+1)} \tag{16}$$

where $\hat{\mathcal{I}}_{u,j}$ denotes the j-th recommended item for the user u, and IDCG denots the ideal discounted cumulative gain, which is a normalization constant and is the maximum possible value of DCG@k.

• Mean Reciprocal Rank: Mean Reciprocal Rank (MRR) is associated with rank models where the user only wishes to see one relevant item, which is defined as:

$$\text{MRR} = \frac{1}{|\mathcal{U}|}\sum_{u\in\mathcal{U}}(\frac{1}{r_u}), \tag{17}$$

where r_u denotes the position of the highest-ranked relevant item for the user u.

We report results on HR@{5, 10, 20}, NDCG@{5, 10, 20} and MRR. Following [60, 61, 63], we apply the *leave-one-out* strategy for evaluation. To avoid data leakage, for each user, we sort his/her interaction records by the interaction timestamps ascendingly. Besides, we hold out the last interaction as the test set, the interaction before the last one is used as the validation data, and the remaining data is used for training. Since it is time-consuming to rank all items for every user during evaluation, we pair the ground-truth item with 1000 randomly sampled negative items that the user has not interacted with. We calculate all metrics according to the ranking of the items and report the average score over all test users.

5.1.3 Baselines. We consider the following baselines:

- BPR [41] is a classic personalized ranking algorithm that optimizes the latent factor model with the pairwise ranking loss function via stochastic gradient descent.
- UltraGCN [35] is the state-of-the-art collaborative filtering model based on graph neural network, which simplifies the formulation of GCNs and skips infinite layers of message passing for more concise efficient recommendation
- DGCF [51] pays attention to modeling the finer granularity of user intents, which disentangle different intents from the single user-item interaction graph and yield disentangled representations for user and item.
- PF-HIN [7] designs a ranking-based breadth-first search strategy to generate node sequence and utilizes masked node prediction to pre-train the nodes' representations.
- GCC [38] is a recently proposed pre-training method for homogeneous graphs via contrastive learning. We fine-tune the pre-trained model released by the authors on our datasets.
- HAN [50] treats meta-paths as virtual edges to connect nodes and utilizes a hierarchical attention mechanism to capture both node-level and semantic-level information.
- HGT [17] introduces node- and edge-type dependent attention mechanism to model heterogeneous graph, which assigns different weights on neighbors during aggregation to capture the interactions among different types of nodes.
- MCRec [14] utilizes convolutional neural network to construct meta-path embeddings and further leverages the co-attention mechanism to model interactions among users, items and meta-paths.
- MTRec [27] introduces a multi-task learning framework for HIN-based recommender systems. It utilizes link prediction as an auxiliary task to improve the recommendation performance.
- HINGE [20] captures and aggregates the interactive patterns under different meta-paths between each pair of user and item nodes. It formulates the interaction modules via a convolutional framework and efficiently learns the parameters with fast Fourier transform.

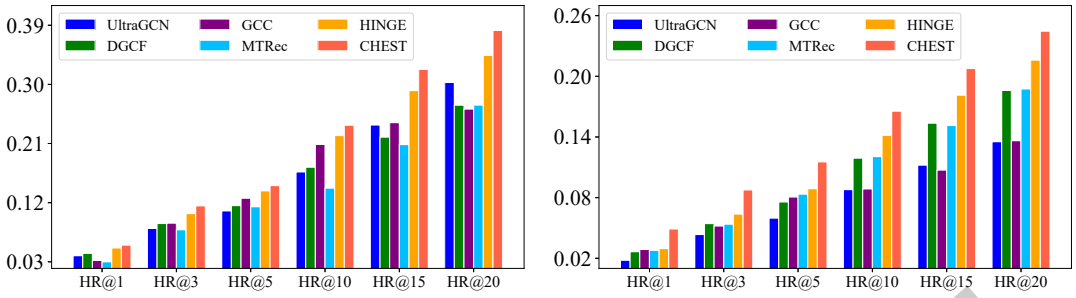
Our baselines can be roughly categorized into four groups: (1) BPR, UltraGCN and DGCF are classic or neural collaborative filtering methods, (2) PF-HIN and GCC are pre-training methods that utilize supervised signals to pre-train graph encoders for heterogeneous and homogeneous graphs respectively, (3) HGT and HAN are specially designed graph neural networks for modeling HIN, (4) MCRec, MTRec and HINGE are state-of-the-art HIN-based recommenders, which extract path instances based on meta-paths from HIN and then model the paths using neural networks.

5.1.4 Implementation Details. For all the baselines, we either adopt the original source code or implement the model by PyTorch. Specifically, in Table 3, models with "♢" are implemented by provided source code while those with "♡" are implemented by ourselves. For our model, we implement it by PyTorch. For all methods that use meta-paths, we use the same meta-paths as shown in the last column of Table 2 and sample five path instances for each meta-path. For MCRec, we also pre-learned the latent vectors for nodes to initialize parameters as the authors suggested.

In our model, we use two self-attention blocks each with two attention heads and set the embedding size as 64. We utilize the learned parameters at the pre-training stage to initialize the model parameters at the fine-tuning stage. In the pre-training stage, the mask proportions of nodes and edges are set as 0.3 and 0.2, and the weights for the three pre-training losses in the elementary course (*i.e.,* MNP, MEP and MTP) are set as 0.5, 1.0, and 0.2, and the softmax temperature in the advanced course is set to 0.1. We use the Adam optimizer [21] with learning rates of 0.001 and 0.0001 for pre-training and fine-tuning stages, respectively. For the baselines, all the models have some parameters to tune. We either follow the reported optimal parameter settings or optimize each model separately using the validation set.

Table 3. Performance comparison of different methods on HIN-based recommendation. The best and second best results are in bold and underlined fonts respectively. "†" indicates the statistical significance for $p < 0.01$ compared to the best baseline.

Datasets	Mothods	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20	MRR
Movielens	BPR$^{\heartsuit}$	0.0806	0.1474	0.2280	0.0519	0.0734	0.0929	0.0674
	UltraGCN$^{\diamond}$	0.0764	0.1209	0.1792	0.0506	0.0650	0.0797	0.0599
	DGCF $^{\diamond}$	0.0891	0.1410	0.2153	0.0577	0.0745	0.0939	0.0697
	PF-HIN$^{\heartsuit}$	0.0721	0.1249	0.1983	0.0508	0.0672	0.0854	0.0608
	GCC$^{\diamond}$	0.0859	0.1304	0.2015	0.0580	0.0726	0.0904	0.0701
	HAN$^{\diamond}$	0.0723	0.1256	0.2004	0.0495	0.0655	0.0843	0.0622
	HGT$^{\diamond}$	0.0774	0.1389	0.2195	0.0482	0.0679	0.0882	0.0627
	MCRec$^{\diamond}$	0.0712	0.1209	0.1875	0.0504	0.0632	0.0723	0.0597
	MTRec$^{\heartsuit}$	0.0734	0.1251	0.1983	0.0512	0.0650	0.0833	0.0608
	HINGE$^{\diamond}$	0.0901	0.1485	0.2344	0.0541	0.0708	0.0924	0.0628
	CHEST	**0.0933**†	**0.1616**†	**0.2524**†	**0.0611**†	**0.0802**†	**0.1073**†	**0.0759**†
Douban	BPR$^{\heartsuit}$	0.0944	0.1571	0.2371	0.0578	0.0779	0.0981	0.0672
	UltraGCN$^{\diamond}$	0.1075	0.1668	0.3029	0.0743	0.0931	0.1273	0.0875
	DGCF $^{\diamond}$	0.1155	0.1739	0.2683	0.0803	0.0990	0.1225	0.0921
	PF-HIN$^{\heartsuit}$	0.0967	0.1538	0.2440	0.0676	0.0845	0.1036	0.0721
	GCC$^{\diamond}$	0.0989	0.1886	0.2624	0.0910	0.1071	0.1208	0.0849
	HAN$^{\diamond}$	0.0789	0.1634	0.2438	0.0559	0.0795	0.1005	0.0687
	HGT$^{\diamond}$	0.0875	0.1692	0.2623	0.0542	0.0804	0.1039	0.0711
	MCRec$^{\diamond}$	0.0900	0.1386	0.2467	0.0524	0.0623	0.1010	0.0597
	MTRec$^{\heartsuit}$	0.1138	0.1421	0.2685	0.0538	0.0644	0.0960	0.0607
	HINGE$^{\diamond}$	0.1380	0.2221	0.3441	0.0882	0.1139	0.1362	0.0961
	CHEST	**0.1460** †	**0.2378**†	**0.3821**†	**0.0974**†	**0.1266**†	**0.1631**†	**0.1175**†
Yelp	BPR$^{\heartsuit}$	0.0517	0.0858	0.1394	0.0335	0.0444	0.0579	0.0421
	UltraGCN$^{\diamond}$	0.0597	0.0879	0.1353	0.0391	0.0482	0.0600	0.0447
	DGCF $^{\diamond}$	0.0757	0.1190	0.1860	0.0510	0.0649	0.0817	0.0611
	PF-HIN$^{\heartsuit}$	0.0615	0.0785	0.1478	0.0512	0.0576	0.0623	0.0459
	GCC$^{\diamond}$	0.0806	0.0886	0.1363	0.0572	0.0597	0.0641	0.0551
	HAN$^{\diamond}$	0.0675	0.0873	0.1683	0.0410	0.0510	0.0613	0.0420
	HGT$^{\diamond}$	0.0810	0.1274	0.1922	0.0504	0.0685	0.0898	0.0626
	MCRec$^{\diamond}$	0.0726	0.1186	0.1823	0.0498	0.0657	0.0745	0.0487
	MTRec$^{\heartsuit}$	0.0834	0.1206	0.1875	0.0514	0.0698	0.0875	0.0594
	HINGE$^{\diamond}$	0.0888	0.1416	0.2160	0.0593	0.0763	0.0949	0.0703
	CHEST	**0.1154**†	**0.1655**†	**0.2446**†	**0.0826**†	**0.0986**†	**0.1185**†	**0.0924**†
AMiner	BPR$^{\heartsuit}$	0.1001	0.1395	0.1834	0.0734	0.1064	0.1300	0.0865
	UltraGCN$^{\diamond}$	0.1125	0.1469	0.1987	0.0892	0.1151	0.1521	0.0986
	DGCF $^{\diamond}$	0.1187	0.1495	0.1855	0.0945	0.1281	0.1557	0.1017
	PF-HIN$^{\heartsuit}$	0.1223	0.1654	0.1840	0.0956	0.1378	0.1467	0.1132
	GCC$^{\diamond}$	0.1234	0.1674	0.1807	0.0925	0.1416	0.1549	0.1141
	HAN$^{\diamond}$	0.1189	0.1598	0.1778	0.0938	0.1342	0.1521	0.1152
	HGT$^{\diamond}$	0.1201	0.1637	0.1807	0.0967	0.1313	0.1406	0.1224
	MCRec$^{\diamond}$	0.1265	0.1731	0.1854	0.1028	0.1323	0.1532	0.1276
	MTRec$^{\heartsuit}$	0.1363	0.1872	0.1976	0.1068	0.1338	0.1665	0.1391
	HINGE$^{\diamond}$	0.1429	0.2106	0.2389	0.1209	0.1575	0.1771	0.1528
	CHEST	**0.1586**†	**0.2538**†	**0.3474**†	**0.1467**†	**0.1963**†	**0.2199**†	**0.1830**†



(a) Douban (b) Yelp

Fig. 3. Performance comparison *w.r.t.* different evaluation metrics on Douban and Yelp dataset.

5.2 Performance Comparison

Table 3 presents the performance comparison of different methods on the recommendation task.

As we can see, for three classic recommendation baselines, the following performance order is consistent across all datasets: DGCF>UltraGCN>BPR. A possible reason is that DGCF can capture diverse relationships in the interaction graph and disentangle user intents from single user-item interaction into different aspects. Besides, they all perform worse on the more sparse datasets (*i.e.,* Yelp and AMiner), because they are trained with limited user-item interactions and are likely to suffer from data sparsity problem.

Second, we observe that GCC performs better than PF-HIN. A major reason is that GCC is pre-trained on large-scale homogeneous graphs from various fields and has encoded transferable graph structure knowledge, which is useful for improving the recommendation task.

Third, HGT performs better than HAN in most cases. One possible reason is that HGT designs node- and edge-type dependent parameters to characterize the heterogeneous attention over each edge, which can capture dedicated representations for different types of nodes and edges. However, the two methods are general GNN embedding methods, which may not be aware of the recommendation goal and cannot perform better than UltraGCN and DGCF in some cases. For path-based baselines, the performance order is as follows: HINGE>MTRec>MCRec. MCRec and MTRec sample path instances through "priority"-based walking strategy. Besides, MTRec utilizes the self-attention mechanism to learn the semantics of meta-paths in HIN and designs a special auxiliary link prediction task for improving the recommendation performance. While, GCC and PF-HIN perform slightly worse than MTRec, which shows that task-agnostic graph pre-training methods cannot yield best performance for HIN-based recommendation.

Furthermore, in order to compare the performance of our method and some competitive baseline methods in more detail, we report more metrics (*i.e.,* HR@1, HR@3, HR@5, HR@10, HR@15, HR@20) on Douban and Yelp datasets in Figure 3.

As seen in Table 3 and Figure 3, our model CHEST performs consistently better than all the baselines by a large margin on four datasets. Different from these baselines, our heterogeneous subgraphs are specially sampled for user-item interaction, which is tailored to the recommendation task. Besides, our proposed heterogeneous subgraph Transformer is able to preserve graph structure and path-level semantics within the subgraph via special composite node embeddings. We further propose the curriculum pre-training strategy to learn effective representations for utilizing useful information in HIN for recommendation task. Comparing our approach with all the baseline models, it can be observed that the above strategies are very useful to improve the recommendation performance.

Table 4. Ablation study of our approach on composite node embeddings.

Datasets	Mothods	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20	MRR
Douban	CHEST	**0.1460**	**0.2378**	**0.3821**	**0.0974**	**0.1266**	**0.1631**	**0.1175**
	w/o Node Type	0.1423	0.2145	0.3512	0.0945	0.1213	0.1543	0.1122
	w/o Slot	0.1410	0.2098	0.3496	0.0924	0.1172	0.1510	0.1098
	w/o Precursor	0.1387	0.2035	0.3453	0.0886	0.1078	0.1452	0.1070
Yelp	CHEST	**0.1154**	**0.1655**	**0.2446**	**0.0826**	**0.0986**	**0.1185**	**0.0924**
	w/o Node Type	0.1123	0.1566	0.2334	0.0769	0.0906	0.1106	0.0848
	w/o Slot	0.1087	0.1553	0.2271	0.0774	0.0873	0.1087	0.0798
	w/o Precursor	0.1014	0.1547	0.2342	0.0735	0.0865	0.1054	0.0786

Table 5. Ablation study of our approach on pre-training tasks (P) and other curriculum settings (C).

Datasets		Method	HR@5	HR@10	HR@20	NDCG@5	NDCG@10	NDCG@20	MRR
Douban		CHEST	**0.1460**	**0.2378**	**0.3821**	**0.0974**	**0.1266**	**0.1631**	**0.1175**
	(P)	w/o MTP	0.1389	0.2134	0.3512	0.0967	0.1209	0.1552	0.1108
		w/o MEP	0.1437	0.2186	0.3486	0.0945	0.1242	0.1569	0.1137
		w/o MNP	0.1322	0.2047	0.3435	0.0859	0.1092	0.1440	0.0990
		w/o SCL	0.1381	0.2112	0.3345	0.0934	0.1169	0.1476	0.1063
	(C)	Multi-task	0.1356	0.2062	0.3420	0.0910	0.1136	0.1474	0.1045
		Reverse courses	0.1299	0.2037	0.3238	0.0871	0.1107	0.1407	0.1008
Yelp		CHEST	**0.1154**	**0.1655**	**0.2446**	**0.0826**	**0.0986**	**0.1185**	**0.0924**
	(P)	w/o MTP	0.1131	0.1637	0.2394	0.0794	0.0956	0.1145	0.0884
		w/o MEP	0.1116	0.1616	0.2316	0.0803	0.0964	0.1139	0.0895
		w/o MNP	0.1032	0.1525	0.2203	0.0725	0.0883	0.1053	0.0818
		w/o SCL	0.1076	0.1604	0.2307	0.0739	0.0909	0.1086	0.0827
	(C)	Multi-task	0.1120	0.1544	0.2312	0.0784	0.0905	0.1098	0.0844
		Reverse courses	0.1078	0.1536	0.2213	0.0739	0.0881	0.1060	0.0814

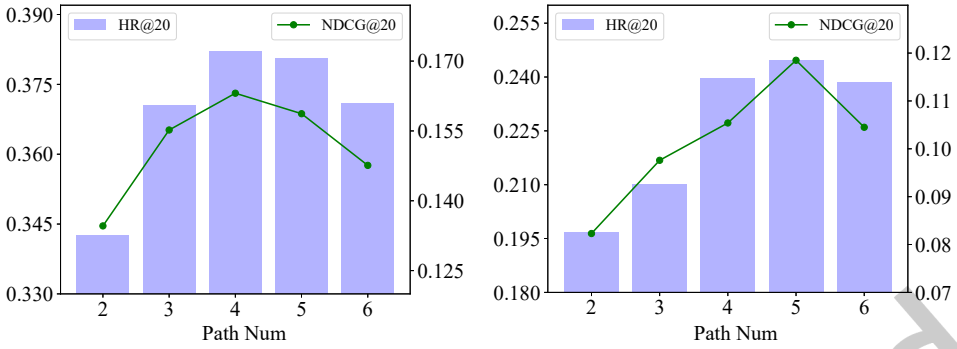
5.3 Detailed Analysis

In this section, we perform a series of detailed analyses on the performance of our model.

5.3.1 Ablation Study. In our proposed CHEST, we have incorporated four types of node embeddings and designed a curriculum pre-training strategy for HIN-based recommendation. In this part, we conduct comprehensive ablation studies on Douban and Yelp datasets to examine the effectiveness of these proposed components and techniques on the model performance.

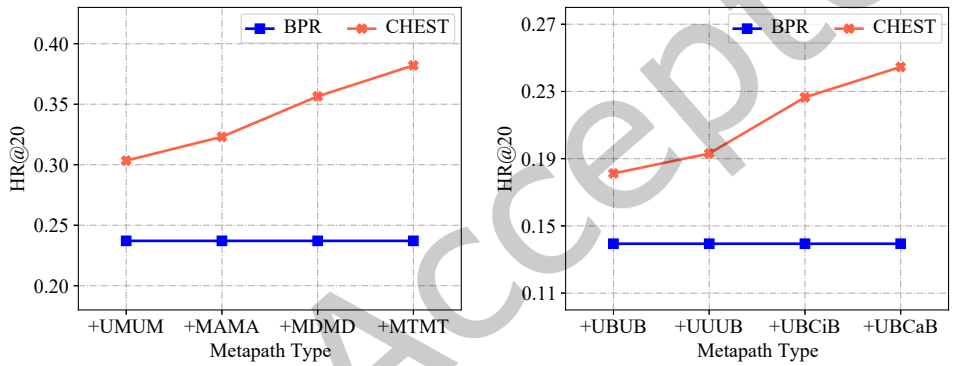
We first analyze the contribution of the composite embeddings. Besides node ID embeddings, we introduce node type embedding, slot embedding and precursor embedding to preserve the semantics of interaction-specific heterogeneous subgraphs in multi-slot sequence representations. The results of embedding ablation (ID embedding is reserved in all cases) are shown in Table 4. As we can see, all the embeddings are useful to improve the model performance. Especially, the precursor embedding seems more important than the other two embeddings, since it can preserve the graph structure semantics within the subgraph.

Next, we continue to conduct the ablation study to analyze the contribution of each pre-training task and other curriculum settings. As can be seen in Table 5, the performance drops when we remove one of the pre-training



(a) Douban (b) Yelp

Fig. 4. Performance (HR@20) comparison *w.r.t.* different #path K on Douban and Yelp dataset.

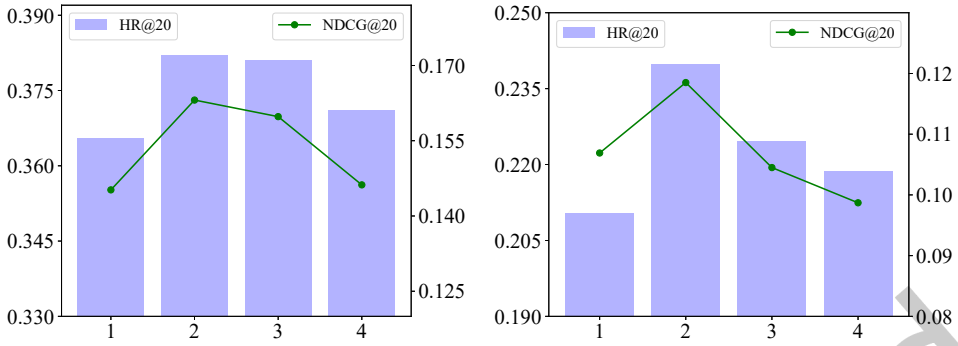


(a) Douban (b) Yelp

Fig. 5. Performance (HR@20) comparison *w.r.t.* different meta-paths types on Douban and Yelp dataset.

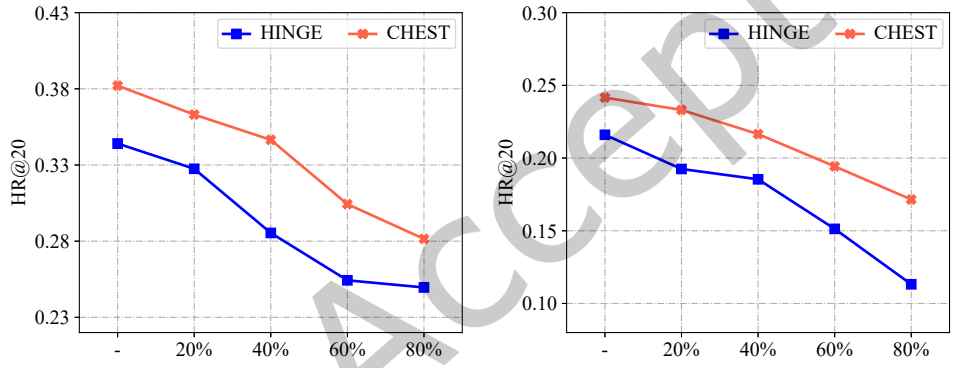
tasks, which shows that the above tasks are all beneficial to our model. Among them, the MNP (Masked Node Prediction) is more important than other pre-training tasks. One possible reason is that the correlations between the node and its surrounding context are important for recommendation task. Under the "*Multi-task*" setting, we pre-train the model on four pre-training tasks via multi-task learning, and the performance drops compared to the curriculum learning paradigm. The "*Reverse courses*" setting means reversing the learning order of the elementary course and the advanced course, which decreases the recommendation performance. These findings verify the rationality of our elementary-to-advanced curriculum learning setting.

5.3.2 Subgraph Construction. To construct the interaction-specific heterogeneous subgraph, we keep top-K path instances with the highest average similarities for each meta-path. We study the effectiveness of different K on the model performance. As we can see in Figure 4, CHEST could achieve good results using only two path instances for each meta-path, which indicates that "priority"-based walking strategy is able to sample high-quality path instances. But when the K is too large, the results drop a bit. One possible reason is that we introduce some noisy paths into the subgraph.



(a) Douban (b) Yelp

Fig. 6. Performance (HR@20) tuning *w.r.t.* different number of Transformer layers on Douban and Yelp dataset.



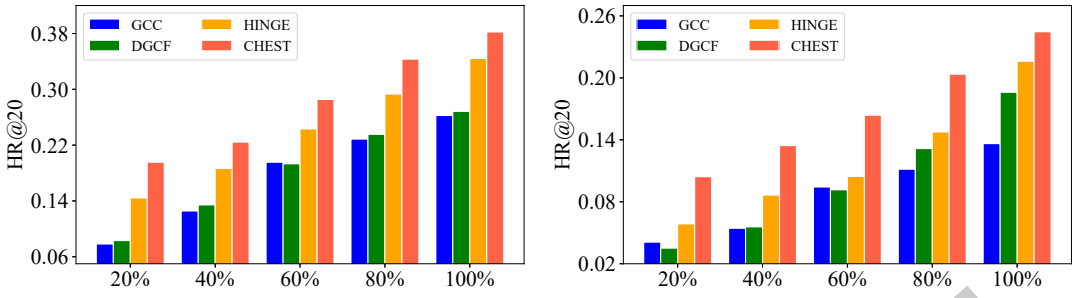
(a) Douban (b) Yelp

Fig. 7. Performance comparison *w.r.t.* different edge dropping rates on Douban and Yelp datasets.

We also investigate the influence of different meta-paths on the recommendation performance by gradually incorporating meta-paths into the subgraph. As shown in Figure 5, the performance of CHEST consistently improves with the incorporation of more meta-paths. The reason is that different meta-paths can introduce different aspects of information for modeling user-item interaction.

5.3.3 Hyperparameter Tuning. Our model consists of a composite embeddings layer and several Transformer layers. Here, we report the tuning results (HR@20) of different numbers of Transformer layers on Douban and Yelp datasets. The cases on other datasets or metrics are similar and omitted.

As shown in Figure 6, stacking Transformer layers can boost recommendation performance which verifies that deep self-attention architecture could help learn more complex node interactions within the subgraph. CHEST achieves the best performance when the layer number is set to 2, which indicates that CHEST can efficiently learn effective information from HIN for recommendation with two Transformer layers. The decline in performance when stacking more layers is largely due to the overfitting problem.



(a) Douban (b) Yelp

Fig. 8. Performance comparison *w.r.t.* different sparsity levels of training data on Douban and Yelp dataset.

5.3.4 Model Robustness on Incomplete HIN. Most of existing HIN-based recommenders usually assume that the original heterogeneous graph is reliable and complete. However, in real-world datasets, the constructed heterogeneous graphs are usually noisy or incomplete. To evaluate the robustness of our methods, we randomly drop different proportions of edges to construct incomplete heterogeneous graphs.

As shown in Figure 7, CHEST is consistently better than the best baseline method, especially at an extremely incomplete level (80%).It is because that we utilize three strategies to augment interaction-specific subgraph in the advanced course for pre-training, which enables CHEST to achieve good performance when dealing with incomplete heterogeneous graphs.

5.3.5 Model Robustness with Sparse Training Data. Recommender systems usually require a considerable amount of training data, thus they are likely to suffer from data sparsity problem in practice. This issue can be alleviated by our method because the proposed curriculum pre-training strategy can leverage intrinsic data correlations from input as auxiliary supervision signals. We simulate the data sparsity scenarios by using different proportions of the full training dataset, *i.e.,* 20%, 40%, 60%, 80%, and 100%.

Figure 8 shows the results of data sparsity analysis on Douban and Yelp datasets. As we can see, the performance substantially drops when less training data is used. While, CHEST achieves the best performance CHEST among all methods in different data sparsity scenarios. It is because CHEST utilizes an elementary-to-advanced training process to extract effective representations from HIN tailored to user-item interactions.

5.3.6 The Trade-off between Efficiency and Effectiveness. Besides recommendation performance, efficiency is also an important factor to consider in practical systems. Here, we analyze the trade-off between effectiveness and efficiency for different comparison methods. In particular, we conduct the analysis of the recommendation performance and inference time on Yelp dataset. For recommendation performance, we select HR@20 as the evaluation metric. For inference time, we report the total time cost of all users in the test set, where we perform the experiment for five times and report the average time. The experiments are executed on a Ubuntu 20.04 machine with Intel (R) Xeon (R) Platinum 8160 CPU and up to eight NVIDIA GeForce RTX 3090 GPUs. The rest experimental settings are similar to those in Section 5.1.

The results are shown in Figure 9. First, we can see that path-based baselines (*e.g.,* MCRec and MTRec) mostly perform better than other methods (*e.g.,* BPR, DGCF and HGT), while requiring longer inference time. The reason is that the path-based methods rely on capturing path-level fine-grained characteristics for recommendation, which is effective but time-consuming to compute. Besides, among the path-based baselines, HINGE captures user-specific paths for recommendation. Due to the rich contextual information, HINGE achieves the best performance

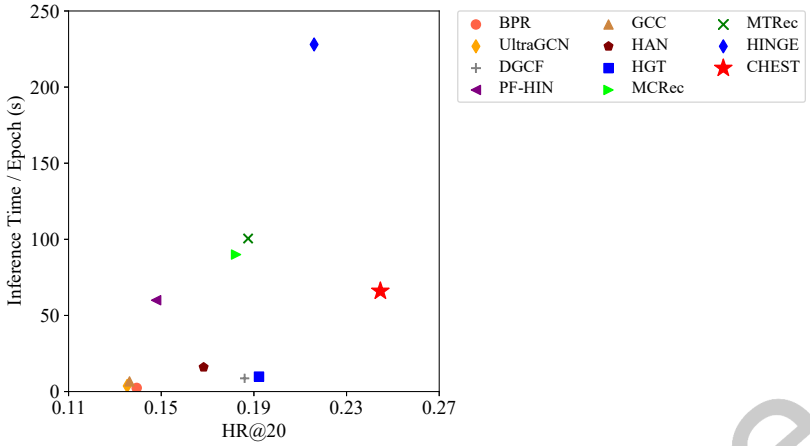


Fig. 9. Analysis of the trade-off between effectiveness and efficiency for different models in terms of HR@20 and inference time on Yelp dataset.

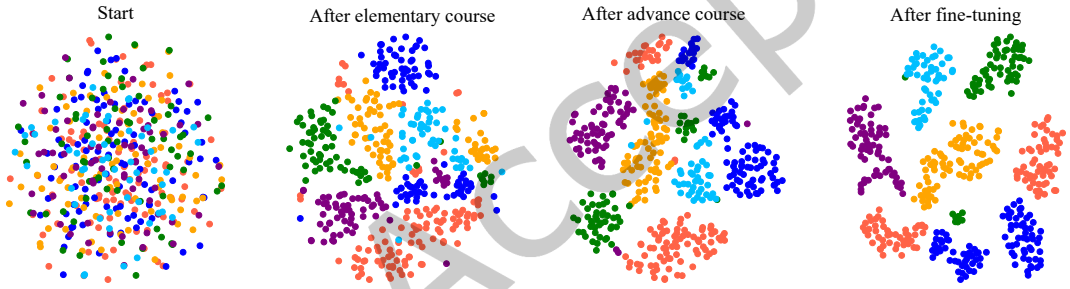


Fig. 10. Visualization of the learned user embeddings *w.r.t.* different phrases on Movielens dataset. Different colors correspond to the different occupations of the users.

but is the most time-consuming model. Finally, our CHEST method locates at the right and bottom part of the Figure 9. Compared with other methods, CHEST is able to achieve the best performance and meanwhile does not cause a higher cost of inference time. Since our CHEST only needs to model the interaction-specific subgraph consisting of several high-quality paths, it can effectively balance the two factors of effectiveness and efficiency.

5.4 Qualitative Analysis

The above results have shown the effectiveness of our curriculum pre-training strategy for the recommendation task. In this section, we present some qualitative analyses to understand why our approach works. Specially, we present two examples to qualitatively illustrate how the elementary-to-advanced training process improves the learning of data representations. We visualize the two-dimensional projections of learned user embeddings and subgraph representations on Movielens dataset using t-SNE algorithm [34].

As shown in Figure 10, various colors represent different *occupations* of users in Movielens dataset. Before pre-training, the representations of users with the same occupations are distributed randomly. However, after pre-training on the elementary course, our approach derives more coherent clusters corresponding to different

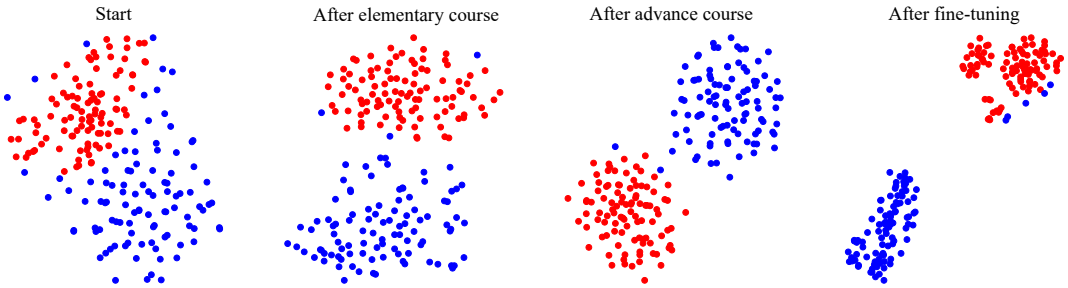


Fig. 11. Visualization of the learned subgraph representations *w.r.t.* different phrases on Movielens dataset. Positive samples and negative samples of the same interaction-specific subgraph are in red and blue respectively.

occupations. After the advanced course, we can see that the produced clusters of user representations are still separated clearly.

In the meantime, Figure 11 presents the distribution of positive samples (*i.e.*, augmented subgraphs) and negative samples of the original interaction-specific subgraph. As we can see, after training on the elementary course, the subgraph representations have not been aggregated into coherent clusters. One possible reason is that the elementary course only focuses on the local context information (*e.g.*, node, edge and path) by which our model is still unaware of the global information of the whole subgraph. While these subgraph representations are clearly separated into two clusters (1.e positive samples and negative samples) after the advanced course. This phenomenon verifies that the advanced course captures global context information of the subgraph.

The above findings indicate that our curriculum pre-training strategy is able to learn local and global semantics underlying HIN, which can enhance the modeling for user-item interaction.

6 CONCLUSION

In this paper, we proposed a curriculum pre-training based heterogeneous subgraph Transformer (CHEST) for HIN-based recommendation task. First, we proposed to use the interaction-specific heterogeneous subgraph to extract sufficient and relevant context information from HIN for each user-item pair. Then we designed the heterogeneous subgraph Transformer to model the subgraph, in which we incorporated a special composite embedding layer to capture graph structure and path-level semantics and a self-attentive layer to aggregate the representation for the user-item interaction subgraph. Furthermore, we designed a curriculum pre-training strategy to gradually learn from both local and global contexts in the subgraph tailored to the recommendation task, in which we devised an elementary-to-advanced learning process to learn effective representations with increasing difficulty levels. Extensive experiments conducted on three real-world datasets demonstrated the effectiveness of our proposed approach against a number of competitive baselines, especially when only limited training data is available.

Currently, we have shown that it is promising to utilize curriculum pre-training techniques for HIN-based recommendation. In future work, we plan to design a more general and effective pre-training strategy for improving more complex recommendation tasks, such as multimedia recommendation and conversational recommendation.

ACKNOWLEDGEMENT

The authors gratefully appreciate the anonymous reviewers for their valuable and detailed comments that greatly helped to improve the quality of this article.

REFERENCES

[1] Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum Learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09)*. ACM, 41–48.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A Simple Framework for Contrastive Learning of Visual Representations. In *ICML 2020, 13-18 July 2020, Virtual Event*, Vol. 119. PMLR, 1597–1607.

[3] Guanyi Chu, Xiao Wang, Chuan Shi, and Xunqiang Jiang. 2021. CuCo: Graph Representation with Curriculum Contrastive Learning. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conferences on Artificial Intelligence Organization, 2300–2306.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT 2019*. Association for Computational Linguistics, 4171–4186.

[5] Yuxiao Dong, Nitesh V. Chawla, and Ananthram Swami. 2017. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In *SIGKDD 2017*. ACM, 135–144.

[6] Yuxiao Dong, Ziniu Hu, Kuansan Wang, Yizhou Sun, and Jie Tang. 2020. Heterogeneous Network Representation Learning. In *IJCAI 2020*. 4861–4867.

[7] Yang Fang, Xiang Zhao, and Weidong Xiao. 2020. Exploring Heterogeneous Information Networks via Pre-Training. *CoRR* abs/2007.03184 (2020).

[8] Junliang Guo, Xu Tan, Linli Xu, Tao Qin, Enhong Chen, and Tie-Yan Liu. 2020. Fine-Tuning by Curriculum Learning for Non-Autoregressive Neural Machine Translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, New York, NY, USA, February 7-12, 2020*. AAAI Press, 7839–7846.

[9] Sheng Guo, Weilin Huang, Haozhi Zhang, Chenfan Zhuang, Dengke Dong, Matthew R. Scott, and Dinglong Huang. 2018. CurriculumNet: Weakly Supervised Learning from Large-Scale Web Images. In *ECCV 2018, Munich, Germany, September 8-14, 2018*, Vol. 11214. Springer, 139–154.

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. 2020. Momentum Contrast for Unsupervised Visual Representation Learning. In *CVPR 2020, Seattle, WA, USA, June 13-19, 2020*. IEEE, 9726–9735.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *WWW 2017, Perth, Australia, April 3-7, 2017*. ACM, 173–182.

[12] Dan Hendrycks, Kimin Lee, and Mantas Mazeika. 2019. Using Pre-Training Can Improve Model Robustness and Uncertainty. In *ICML 2019, 9-15 June 2019, Long Beach, California, USA*. 2712–2721.

[13] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Tianchi Yang. 2018. Local and Global Information Fusion for Top-N Recommendation in Heterogeneous Information Network. In *CIKM 2018*. 1683–1686.

[14] Binbin Hu, Chuan Shi, Wayne Xin Zhao, and Philip S. Yu. 2018. Leveraging Meta-path based Context for Top- N Recommendation with A Neural Co-Attention Model. In *SIGKDD 2018*. 1531–1540.

[15] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay S. Pande, and Jure Leskovec. 2020. Strategies for Pre-training Graph Neural Networks. In *ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*.

[16] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. 2020. GPT-GNN: Generative Pre-Training of Graph Neural Networks. In *KDD 2020*. 1857–1867.

[17] Ziniu Hu, Yuxiao Dong, Kuansan Wang, and Yizhou Sun. 2020. Heterogeneous Graph Transformer. In *WWW 2020*. ACM / IW3C2, 2704–2710.

[18] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. 2018. Improving Sequential Recommendation with Knowledge-Enhanced Memory Networks. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, SIGIR 2018, Ann Arbor, MI, USA, July 08-12, 2018*. ACM, 505–514.

[19] Folasade Olubusola Isinkaye, YO Folajimi, and Bolande Adefowoke Ojokoh. 2015. Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal* 16, 3 (2015), 261–273.

[20] Jiarui Jin, Jiarui Qin, Yuchen Fang, Kounianhua Du, Weinan Zhang, Yong Yu, Zheng Zhang, and Alexander J. Smola. 2020. An Efficient Neighborhood-based Interaction Model for Recommendation on Heterogeneous Graph. In *SIGKDD 2020*. 75–84.

[21] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

[22] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

[23] Lingpeng Kong, Cyprien de Masson d'Autume, Lei Yu, Wang Ling, Zihang Dai, and Dani Yogatama. 2020. A Mutual Information Maximization Perspective of Language Representation Learning. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

[24] Yehuda Koren and Robert M. Bell. 2015. Advances in Collaborative Filtering. In *Recommender Systems Handbook*. Springer, 77–118.

[25] Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix Factorization Techniques for Recommender Systems. *Computer* 42, 8 (2009), 30–37.
[26] Aaron Q. Li, Amr Ahmed, Sujith Ravi, and Alexander J. Smola. 2014. Reducing the sampling complexity of topic models. In *KDD '14, New York, NY, USA - August 24 - 27, 2014*. 891–900.
[27] Hui Li, Yanlin Wang, Ziyu Lyu, and Jieming Shi. 2020. Multi-task Learning for Recommendation over Heterogeneous Information Network. *IEEE Transactions on Knowledge and Data Engineering* (2020).
[28] Guang Ling, Michael R. Lyu, and Irwin King. 2014. Ratings meet reviews, a combined approach to recommend. In *Eighth ACM Conference on Recommender Systems, RecSys '14, Foster City, Silicon Valley, CA, USA - October 06 - 10, 2014*. ACM, 105–112.
[29] Cao Liu, Shizhu He, Kang Liu, and Jun Zhao. 2018. Curriculum Learning for Natural Answer Generation. In *IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*. ijcai.org, 4223–4229.
[30] Jinglin Liu, Yi Ren, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. Task-Level Curriculum Learning for Non-Autoregressive Neural Machine Translation. In *IJCAI 2020*. 3861–3867.
[31] Yuanfu Lu, Xunqiang Jiang, Yuan Fang, and Chuan Shi. 2021. Learning to pre-train graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4276–4284.
[32] Chen Luo, Wei Pang, Zhe Wang, and Chenghua Lin. 2014. Hete-CF: Social-Based Collaborative Filtering Recommendation Using Heterogeneous Relations. In *2014 IEEE International Conference on Data Mining, ICDM 2014, Shenzhen, China, December 14-17, 2014*. IEEE Computer Society, 917–922.
[33] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*. ACM, 287–296.
[34] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9, Nov (2008), 2579–2605.
[35] Kelong Mao, Jieming Zhu, Xi Xiao, Biao Lu, Zhaowei Wang, and Xiuqiang He. 2021. UltraGCN: Ultra Simplification of Graph Convolutional Networks for Recommendation. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. ACM, 1253–1262.
[36] Shanlei Mu, Yaliang Li, Wayne Xin Zhao, Siqing Li, and Ji-Rong Wen. 2022. Knowledge-Guided Disentangled Representation Learning for Recommender Systems. *ACM Trans. Inf. Syst.* 40, 1 (2022), 6:1–6:26.
[37] Tuan-Anh Nguyen Pham, Xutao Li, Gao Cong, and Zhenjie Zhang. 2016. A General Recommendation Model for Heterogeneous Networks. *IEEE Trans. Knowl. Data Eng.* 28, 12 (2016), 3140–3153.
[38] Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. 2020. GCC: Graph Contrastive Coding for Graph Neural Network Pre-Training. In *KDD 2020*. 1150–1160.
[39] Yuxiang Ren, Bo Liu, Chao Huang, Peng Dai, Liefeng Bo, and Jiawei Zhang. 2019. Heterogeneous Deep Graph Infomax. *CoRR* abs/1911.08538 (2019).
[40] Steffen Rendle. 2010. Factorization Machines. In *ICDM 2010, Sydney, Australia, 14-17 December 2010*. IEEE Computer Society, 995–1000.
[41] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. In *UAI 2009, Montreal, QC, Canada, June 18-21, 2009*. 452–461.
[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. 2015. ImageNet Large Scale Visual Recognition Challenge. *Int. J. Comput. Vis.* 115, 3 (2015), 211–252.
[43] Nikolaos Sarafianos, Theodore Giannakopoulos, Christophoros Nikou, and Ioannis A. Kakadiaris. 2017. Curriculum Learning for Multi-task Classification of Visual Attributes. In *ICCV Workshops 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2608–2615.
[44] Chuan Shi, Binbin Hu, Wayne Xin Zhao, and Philip S. Yu. 2019. Heterogeneous Information Network Embedding for Recommendation. *IEEE Trans. Knowl. Data Eng.* 31, 2 (2019), 357–370.
[45] Yizhou Sun, Jiawei Han, Xifeng Yan, Philip S. Yu, and Tianyi Wu. 2011. PathSim: Meta Path-Based Top-K Similarity Search in Heterogeneous Information Networks. *Proc. VLDB Endow.* 4, 11 (2011), 992–1003.
[46] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. 5998–6008.
[47] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph Attention Networks. *CoRR* abs/1710.10903 (2017).
[48] Petar Velickovic, William Fedus, William L. Hamilton, Pietro Liò, Yoshua Bengio, and R. Devon Hjelm. 2019. Deep Graph Infomax. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.
[49] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge Graph Convolutional Networks for Recommender Systems with Label Smoothness Regularization. *CoRR* abs/1905.04413 (2019).

[50] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S. Yu. 2019. Heterogeneous Graph Attention Network. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*. 2022–2032.

[51] Xiang Wang, Hongye Jin, An Zhang, Xiangnan He, Tong Xu, and Tat-Seng Chua. 2020. Disentangled Graph Collaborative Filtering. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*. ACM, 1001–1010.

[52] Yifan Wang, Suyao Tang, Yuntong Lei, Weiping Song, Sheng Wang, and Ming Zhang. 2020. DisenHAN: Disentangled Heterogeneous Graph Attention Network for Recommendation. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*. ACM, 1605–1614.

[53] Carl Yang, Yuxin Xiao, Yu Zhang, Yizhou Sun, and Jiawei Han. 2020. Heterogeneous Network Representation Learning: Survey, Benchmark, Evaluation, and Beyond. *CoRR* abs/2004.00216 (2020). https://arxiv.org/abs/2004.00216

[54] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. 2018. Graph Convolutional Neural Networks for Web-Scale Recommender Systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*. ACM, 974–983.

[55] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: a heterogeneous information network approach. In *WSDM 2014, New York, NY, USA, February 24-28, 2014*. 283–292.

[56] Xiao Yu, Xiang Ren, Yizhou Sun, Bradley Sturt, Urvashi Khandelwal, Quanquan Gu, Brandon Norick, and Jiawei Han. 2013. Recommendation in heterogeneous information networks with implicit user feedback. In *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*. 347–350.

[57] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V. Chawla. 2019. Heterogeneous Graph Neural Network. In *KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. 793–803.

[58] Huan Zhao, Quanming Yao, Yangqiu Song, James T. Kwok, and Dik Lun Lee. 2021. Side Information Fusion for Recommender Systems over Heterogeneous Information Network. *ACM Trans. Knowl. Discov. Data* 15, 4 (2021), 60:1–60:32.

[59] Jun Zhao, Zhou Zhou, Ziyu Guan, Wei Zhao, Wei Ning, Guang Qiu, and Xiaofei He. 2019. IntentGC: A Scalable Graph Convolution Framework Fusing Heterogeneous Information for Recommendation. In *KDD 2019, Anchorage, AK, USA, August 4-8, 2019*. 2347–2357.

[60] Wayne Xin Zhao, Junhua Chen, Pengfei Wang, Qi Gu, and Ji-Rong Wen. 2020. Revisiting Alternative Experimental Settings for Evaluating Top-N Item Recommendation Algorithms. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*. ACM, 2329–2332.

[61] Wayne Xin Zhao, Shanlei Mu, Yupeng Hou, Zihan Lin, Yushuo Chen, Xingyu Pan, Kaiyuan Li, Yujie Lu, Hui Wang, Changxin Tian, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2021. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. In *CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 - 5, 2021*. ACM, 4653–4664.

[62] Qiwei Zhong, Yang Liu, Xiang Ao, Binbin Hu, Jinghua Feng, Jiayu Tang, and Qing He. 2020. Financial Defaulter Detection on Online Credit Payment via Multi-view Attributed Heterogeneous Information Network. In *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*. ACM / IW3C2, 785–795.

[63] Kun Zhou, Hui Wang, Wayne Xin Zhao, Yutao Zhu, Sirui Wang, Fuzheng Zhang, Zhongyuan Wang, and Ji-Rong Wen. 2020. S3-Rec: Self-Supervised Learning for Sequential Recommendation with Mutual Information Maximization. In *CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020*. ACM, 1893–1902.

[64] Kun Zhou, Wayne Xin Zhao, Shuqing Bian, Yuanhang Zhou, Ji-Rong Wen, and Jingsong Yu. 2020. Improving Conversational Recommender Systems via Knowledge Graph based Semantic Fusion. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*. ACM, 1006–1014.

[65] Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. 2020. Deep Graph Contrastive Representation Learning. *CoRR* abs/2006.04131 (2020).