Tiger: Transferable Interest Graph Embedding for Domain-Level Zero-Shot Recommendation

Jianhuan Zhuo*
zhuojianhuan@iie.ac.cn
Institute of Information Engineering, Chinese Academy of Sciences & School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China

Jianxun Lian
jianxun.lian@outlook.com
Microsoft Research Asia
Beijing, China

Lanling Xu
xulanling_sherry@163.com
Gaoling School of Artificial Intelligence, Renmin University of China
Beijing, China

Ming Gong
Linjun Shou
Daxin Jiang
migon@microsoft.com
lisho@microsoft.com
djiang@microsoft.com
Microsoft STC Asia
Beijing, China

Xing Xie
xingx@microsoft.com
Microsoft Research Asia
Beijing, China

Yinliang Yue†
yueyinliang@iie.ac.cn
Institute of Information Engineering, Chinese Academy of Sciences & School of Cyber Security, University of Chinese Academy of Sciences
Beijing, China

ABSTRACT

Recommender systems play a significant role in online services and have attracted wide attention from both academia and industry. In this paper, we focus on an important, practical, but often overlooked task: domain-level zero-shot recommendation (DZSR). The challenge of DZSR mainly lies in the absence of collaborative behaviors in the target domain, which may be caused by various reasons, such as the domain being newly launched without existing user-item interactions, or users' behaviors being too sensitive to collect for training. To address this challenge, we propose a **T**ransferable **I**nterest **G**raph **E**mbedding technique for **R**ecommendations (Tiger). The key idea is to connect isolated collaborative filtering datasets with a knowledge graph tailored to recommendations, then propagate collaborative signals from public domains to the zero-shot target domain. The backbone of Tiger is the transferable interest extractor, which is a simple yet effective graph convolutional network (GCN) aggregating multiple hops of neighbors on a shared interest graph. We find that the bottom layers of GCN preserve more domain-specific information while the upper layers represent universal interest better. Thus, in Tiger, we discard the bottom layers of GCN to reconstruct user interest so that collaborative signals can be successfully propagated to other domains, and retain the bottom layers of GCN to include domain-specific information for items. Extensive experiments with four public datasets demonstrate that Tiger can effectively make recommendations for a zero-shot domain and outperform several alternative baselines.

CCS CONCEPTS

• **Information systems** → **Personalization**; **Collaborative filtering**; **Content ranking**.

KEYWORDS

Recommender System, Knowledge Graph, Zero-shot Learning

ACM Reference Format:
Jianhuan Zhuo, Jianxun Lian, Lanling Xu, Ming Gong, Linjun Shou, Daxin Jiang, Xing Xie, and Yinliang Yue. 2022. Tiger: Transferable Interest Graph Embedding for Domain-Level Zero-Shot Recommendation. In *Proceedings of the 31st ACM Int'l Conference on Information and Knowledge Management (CIKM '22), Oct. 17–21, 2022, Atlanta, GA, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3511808.3557472

1 INTRODUCTION

Recommender systems play a critical role for online platforms in connecting users with their interested items. In the past decades, numerous methods have been proposed around how to leverage users' historical behaviors for better recommendations, such as collaborative filtering [13, 20], feature interactions [10, 29], sequential recommendations [15, 19], and multi-interest user modeling [21, 24]. However, the line of research on how to cold start a recommender system is almost blank. Different from user/item cold-start problems [1, 33, 50] which make recommendations for new users or items in a mature domain, recommender system cold-starting means the target domain is brand new without any user-item interactions. In this paper, we call the task of recommender system cold-starting as domain-level zero-shot recommendation.

The domain-level zero-shot recommendation (DZSR) is an important and practical task, typical applications of which include: (1) when an online service provider intends to launch a new domain

*Work is done during the internship at Microsoft.
†Corresponding author

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

ACM ISBN 978-1-4503-9236-5/22/10.
https://doi.org/10.1145/3511808.3557472

that needs recommendations, e.g., a news provider such as MSN wants to insert a few personalized advertisements into the content feed. It only has user-news interactions but no user-ads interactions just yet; a video-focused content provider such as TikTok wants to expand its business to e-commerce recommendations, but currently it does not have user-commodity interaction logs. (2) when the target domain's user behavior data is too sensitive to collect for model training since the strict privacy protection policy. Hence, users' data, such as email history or passwords from their input methods, can only be used on their own devices for inferring but not be uploaded to the server for model training.

In order to provide high-quality recommendations for a zero-shot domain, a promising and intuitive way is to represent users' preference by their behaviors from some available source domains, and represent target domain's items with attributes. The key point lies in finding an intermediary, so that users' preference can be propagated to the target domain in a zero-shot manner. Some earlier works [7, 44, 46] use text as the intermediary, since language is a kind of universal knowledge that can be shared across domains. However, the drawbacks of text as intermediary are mainly two-fold: (1) recommendations based on language models demonstrate strong linguistics bias [46]. In many cases, items' true meaning cannot be reflected by their literal name. For example, *12 monkeys* is actually a thriller and science fiction film; *Fantastic Beasts and Where to Find Them* turns out to be a prequel to the *Harry Potter* series. (2) learning a text encoder for universal item representation still requires specific domains as proxy supervised learning tasks [7, 44], which is prone to overfitting the training signals and makes it hard to propagate collaborative signals to zero-shot domains.

In this paper, we advocate to utilize knowledge graph (KG) as the intermediary for user preference propagation. The basic assumption is that, from the perspective of common sense, users' behaviors are universally connected and associated by high-level interest, so that users' preference can be propagated even though the item sets of original domains do not have direct overlap, which is intuitively reasonable. A KG, which contains massive structural and semantic triples to connect entities, can serve as a type of common knowledge to bridge items. For example, for two domains, movie and book, their items are closely connected by high-level concepts such as Science Fiction, Romance, and Historical. However, traditional KG embedding methods such as TransE [3] and TransR [41] are trained towards reconstructing the structural information of a KG. As a result, users' collaborative signals are neither incorporated nor propagated on the graph. A proper KG embedding model for our task should possess these properties: (*P1: recommendation-oriented*) the model can reflect preference similarity, e.g., like-minded users' interacted items should be similar in the embedding space; (*P2: preference propagation*) the generated embeddings are universal and transferable, so that collaborative preference can be propagated to different domains which are connected on the KG; (*P3: domain adaptation*) the model can perform domain adaptation, i.e., when making recommendations for a target domain, distinctive patterns of the target domain can be preserved.

To this end, we propose **T**ransferable **I**nterest **G**raph **E**mbedding for **R**ecommendations (Tiger) to address the domain-level zero-shot problem. The backbone of Tiger is a simple yet effective graph convolutional network (GCN) over a KG, which aggregates multiple hops of neighbors on the KG to represent an entity. The GCN is trained with datasets of collaborative user-item interactions, which, on the one hand, satisfies the *recommendation-oriented* property; on the other hand, has the flexibility to absorb any public dataset derived from real-world recommender systems for facilitating training, as long as items in the dataset can be linked to the KG. The most challenging part for Tiger is how to ensure the *preference propagation* property. We empirically find that a naive GCN cannot perform zero-shot inference, which tends to fit in-domain relationship well but fails to learn transferable information. To overcome this problem, we propose to discard bottom layers of GCN and only aggregate high-level ones to reconstruct user interest. As for items, we retain the bottom layers of GCN so that item embeddings are a combination of universal patterns (which is carried by the high-level layers of GCN) and domain-specific patterns (which is carried by the bottom layers of GCN), which eventually fulfills the *domain adaptation* property. To verify the effectiveness of Tiger, we conduct extensive experiments with four public datasets, with several interesting observations highlighted as follows:

(1) Bottom-layer discarding is the key to propagate collaborative signals to a boarder range of the shared interest graph, resulting in better transferring capability. Within a reasonable threshold (such as 4 in Amazon Movie dataset), the performance improves as the number of discarded layers increase.
(2) Unlike traditional GCNs which quickly fall into over-smoothing problems, in Tiger, the item GCN equipped with the layer discarding mechanism can go much deeper, e.g., in the Amazon Movie dataset, the number of GCN layers can be as many as 10.
(3) Out-domain dataset, in which both users and items have no overlap with the source and target domain, can be easily adsorbed by Tiger to further improve zero-shot performance.
(4) Tiger's zero-shot performance can approach about 50% accuracy of the oracle model in Hit@100, which is far better than a random guess (4% of oracle model) and significantly outperforms several content-based baselines.
(5) Besides the zero-shot scenario, Tiger can also improve warmed-up recommendation, thanks to its ability in encoding collaborative signals from different domains into the interest graph.

The main contributions of this paper are summarized as follows:

- We highlight the importance as well as challenges of domain-level zero-shot recommender systems. To the best of our knowledge, our proposed Tiger is the first work to pre-train an interest graph for zero-shot recommendation.
- We design simple yet effective modules in Tiger, including transferable interest extractor, user interest reconstructor and domain adaptation, to facilitate information propagation to new domains.
- We conduct extensive experiments with four real-world datasets. Experimental results demonstrate that Tiger can achieve reasonable zero-shot recommendations and significantly outperform several competitive baselines [1].

2 PRELIMINARY

In this paper, we formulate the domain-level zero-shot recommendation (DZSR) task as follows: In DZSR, we consider two kinds of

[1]https://github.com/JianhuanZhuo/Tiger-Code-and-Dataset-for-CIKM2022

domains, denoted as target domain $\mathcal{T}$ and source domain $\mathcal{S}$. The target domain $\mathcal{T}$ is the cold-start goal of the DZSR task, while the source domain $\mathcal{S}$ is used to characterize the potential interest of users. For simplicity, $\mathcal{D} \in \{\mathcal{T}, \mathcal{S}\}$ is used to indicate one of domains involved throughout this paper. User sets and item sets involved are denoted as $\mathcal{U}_\mathcal{D}$ and $\mathcal{V}_\mathcal{D}$ respectively, where the user set of target domain is the subset of source domain, i.e., $\mathcal{U}_\mathcal{T} \subset \mathcal{U}_\mathcal{S}$, but item sets on two domains are totally different, i.e., $\mathcal{V}_\mathcal{T} \cap \mathcal{V}_\mathcal{S} = \emptyset$.

All possible interaction set between users and items is $\mathcal{I}_\mathcal{D} = \mathcal{U}_\mathcal{D} \times \mathcal{V}_\mathcal{D}$, where $\times$ denotes Cartesian product. Among them, the observed interaction history of users in $\mathcal{D}$ domain are collected as $\mathcal{P}^+_\mathcal{D} = \{(u, v)|y(u, v) = 1, u \in \mathcal{U}, v \in \mathcal{V}_\mathcal{D}\} \subset \mathcal{I}_\mathcal{D}$, where $y : \mathcal{I}_\mathcal{D} \to \{0, 1\}$ is the labeling function to indicate if the user has interacted with the item (e.g. view, click, purchase). Oppositely, the unobserved interactions are collected as $\mathcal{P}^-_\mathcal{D} = \{(u, v)|y(u, v) = 0, u \in \mathcal{U}, v \in \mathcal{V}_\mathcal{D}\} \subset \mathcal{I}_\mathcal{D}$. For convenience, $\mathcal{H}^\mathcal{D}_u = \{v|y(u, v) = 1, v \in \mathcal{V}_\mathcal{D}\}$ is used to indicate the item set that user u has interacted in domain $\mathcal{D}$. Interactions on the target domain are also collected to evaluate the performance of models under the DZSR setting.

The aim of DZSR is to make cold-start recommendations on the target domain by absorbing universal and transferable knowledge from the source domain. To distinguish the DZSR task from other related tasks introduced in Section 5, a comparison is conducted in Table 1, from which we can see that the unique characteristic of DZSR is that user-item interactions are totally unavailable in the target domain in the training stage.

Table 1: A comparison of DZSR with related tasks.

Task	interactions used to train	
	source domain	target domain
In-domain Rec. [30, 34, 39]	×	✓
Cold-start Rec. [1, 26, 33]	✓	partial
Cross-domain Rec. [25, 51]	✓	✓
Domain-level ZSR [this paper]	✓	×

3 METHODOLOGY

In this section, we introduce our solution to DZSR based on training transferable representation with the help of knowledge graphs. We start by introducing the overall framework of interest graph embedding, and then we will describe the key components in detail.

3.1 Interest Graph Framework

As items from different domains have no overlaps, the critical challenge of DZSR is how to guide the model to learn users' transferable interest embeddings beyond the concrete item-relevant preference limited to the source domain. The "transferable" means the generated embedding from the source domain should be generalized enough to model items in the target domain. For example, if a user has interacted with the book *Harry Potter and the Philosopher's Stone*, models under the DZSR task should capture the user's interest like "things about Wizarding World". The high-level interest is

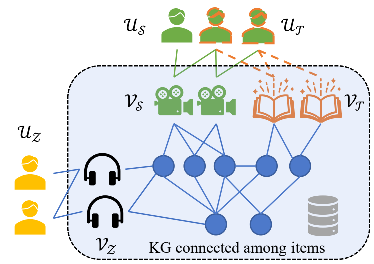


Figure 1: Interest graph: connecting isolated user-item datasets with an knowledge graph

suitable for cold-start domains, which realizes the zero-shot recommendation of *Fantastic Beasts and Where to Find* in the movie domain or *Hedwig's Theme* in the music domain.

Hence, we advocate the interest graph framework, a KG-based universal framework for the domain-level zero-shot recommendation. The core of our framework is to connect items of all datasets by a tailored knowledge graph (KG) and build the transferable embeddings of users' interest from the common-sense perspective. Therefore, the interest graph can be used to propagate users' preferences even though the item sets of original domains do not have overlap. To achieve this objective, we introduce a KG as the intermediary for user preference propagation as the dashed box in Figure 1. Concretely, knowledge graph $\mathcal{G}$ is provided in the form of entity-relation-entity triplet set $\mathcal{G} = \{(h, r, t)|h, r \in \mathcal{E}, r \in \mathcal{R}\}$, where (h, r, t) denotes the head entity h and the tail entity t are connected by a relation r. To describe the attributes of items with rich side information of KG, we introduce an alignment set $\mathcal{A} = \{(v, e)|v \in \mathcal{V}, e \in \mathcal{E}\}$, where (v, e) denotes that item v in recommendations and entity e in KG is the same object conceptually.

The aim of our interest graph framework becomes to establish the prediction function $\hat{y} = f(\mathcal{T}|\mathcal{G}, \mathcal{S}, \Theta^*)$, where parameter Θ^* is estimated only from the source domain $\mathcal{S}$ as Eq. (1). In other words, the collaborative signals can be effectively propagated from source domains to the zero-shot target domain.

$$\Theta^* = \mathop{\mathbf{arg\,min}}_{\Theta} \mathcal{L}(y(\mathcal{S}), \mathcal{S}; \mathcal{G}) \tag{1}$$

To train the parameter Θ^*, the interest graph framework improves from two fundamental subjects: item and user. For item, we extract the transferable representation of items by decomposing the semantic representation into different levels and aggregating high-order GCN layers. For user, we reconstruct the interest representation of users by transferring interests from interaction histories in the source domain. Corresponding to the above two steps, our framework contains two well-designed modules: the transferable interest extractor and the user interest reconstructor. An overview of our model is offered in Figure 2.

3.2 Transferable Interest Extractor

The transferable interest extractor aims to discover the deeper hidden interest carried by items, which are transferable via knowledge graph to items in new domains. Specifically, a multi-layer GCN is applied on items' knowledge graph to decompose items' semantic

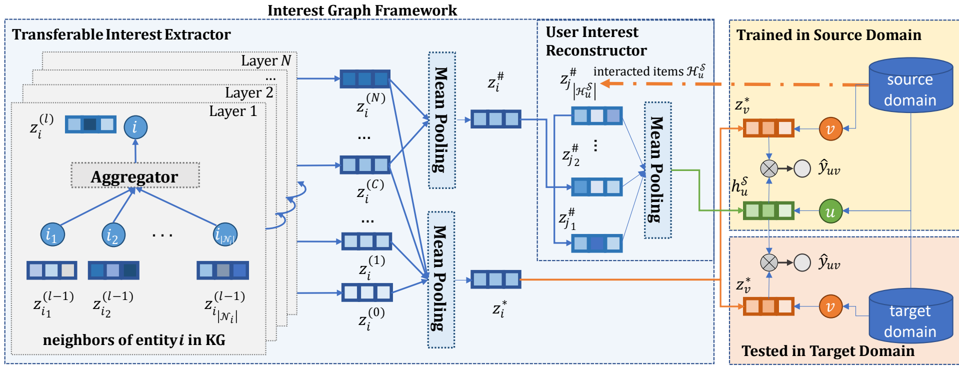


Figure 2: An overview of the proposed Tiger model

features into different levels, in which the embeddings of high-order GCN layers that fuse higher hop neighbor entities are expected to be more transferable.

To obtain embeddings fused with L hop neighborhood information, the GCN needs to perform graph convolutional operation L times on the knowledge graph to generate $L+1$ refined embeddings $[\mathbf{z}_i^{(0)}, \mathbf{z}_i^{(1)}, \ldots, \mathbf{z}_i^{(L)}]$, where $\mathbf{z}_i^{(l)} \in \mathbb{R}^d$ is the GCN output of item i on layer l and d is the dimensional size of embedding. At the input layer, we directly use the learnable embedding of an entity as the input of the graph convolutional network:

$$\mathbf{z}_i^{(0)} = \mathbf{e}_i \tag{2}$$

where $\mathbf{e}_i \in \mathbb{R}^d$ is the original embedding vector of entity i. From 1-st to L-th layers, following the traditional GCN pattern, we use the message passing mechanism to aggregate neighborhood information of the given node to obtain the enhanced representation.

$$\mathbf{z}_i^{(l)} = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \mathbf{z}_j^{(l-1)} \tag{3}$$

where $\mathbf{z}_i^{(l)} \in \mathbb{R}^d$ denotes the output vector of entity i at layer l.

With $L+1$ refined embeddings generated by GCN, we guide the model to learn transferable interest beyond the concrete item-relevant preference limited to the source domain. Empirically, about 98% of items in the source domain take at least two hops in KG to connect to items in the target domain, which means the embeddings of the low-hop GCN are too specific to be shared with the target domain. Furthermore, the lower layers are prone to overfitting the training signals and make it hard to effectively propagate collaborative signals to zero-shot domains. Hence, besides pooling all layer output embeddings of entities as Eq. (4) to obtain the complete semantic representation, we discard the lower output embeddings by a hyper-parameter C to obtain the transferable interest graph embedding from high-order GCN layers as Eq. (5).

$$\mathbf{z}_i^{*} = \frac{1}{N+1} \sum_{j=0}^{N} \mathbf{z}_i^{j} \tag{4}$$

$$\mathbf{z}_i^{\#} = \frac{1}{N-C+1} \sum_{j=C}^{N} \mathbf{z}_i^{j} \tag{5}$$

Since all items are linked well to the KG in the DZSR task, the representation of item v is directly assigned by its corresponding entity embedding of i in the KG, i.e., $\mathbf{z}_v^{*} = \mathbf{z}_i^{*}$ and $\mathbf{z}_v^{\#} = \mathbf{z}_i^{\#}$.

3.3 User Interest Reconstructor

Via mining interaction histories in the source domain, the user interest reconstructor aims to reconstruct users' transferable interest representation, which also works in the target domain for predicting interaction behaviors. Typically, an intuitive approach is characterizing each user as an embedding (UserAsEmb), that is, users are represented by a learnable parameter vector:

$$\mathbf{h}_u = \mathbf{e}_u \tag{6}$$

where $\mathbf{e}_u \in \mathbb{R}^d$. Since the target domain is unavailable during the training stage, the single user embedding only trained in the source domain leads to a sub-optimal result under the domain-level cold-start setting. To narrow the semantic gap between the source and target domain, we dynamically reconstruct the representation of users from their historical behaviors $\mathcal{H}_u^{\mathcal{S}}$ in the source domain $\mathcal{S}$, which can be formulated as follows:

$$\mathbf{h}_u^{\mathcal{S}} = \frac{1}{|\mathcal{H}_u^{\mathcal{S}}|} \sum_{v \in \mathcal{H}_u^{\mathcal{S}}} \mathbf{z}_v^{\#} \tag{7}$$

In Eq. (7), $\mathbf{z}_v^{\#}$ is used to reconstruct the user's interest rather than $\mathbf{z}_v^{*}$ since bottom layers of GCN encode more item-relevant information while high-layer ones are more general interest-relevant. For the user representation under the domain-level cold-start setting, Tiger is encouraged to learn the potential general interest rather than the preference for specific items, improving the generalization ability of the learned user representation across different domains.

3.4 Domain Adaptation

Based on the layered design of the interest graph framework, we can achieve domain adaptation naturally and intuitively. When predicting a user's preference score for an item, we adopt the operation of inner product between the full semantic representation $\mathbf{h}_v^{*}$ of item v and the transferable interest graph embedding $\mathbf{z}_u^{\mathcal{S}}$ of user u reconstructed from domain S. The full semantic representation $\mathbf{h}_v^{*}$ of items are used as a combination of universal patterns (which is carried by the upper layers of GCN) and domain-specific patterns (which is carried by the bottom layers of GCN). Given the candidate pair (u, v), the predicted score is calculated as follows:

$$\hat{y}_{uv} = \mathbf{z}_v^{*\top} \mathbf{h}_u^{\mathcal{S}} \tag{8}$$

During model evaluation in the target domain, Eq. (8) is also used to infer users' preferences on items. Finally, we use the popular BPR [30] loss function to optimize the whole model.

$$\mathcal{L} = -\sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{H}_u^S} \sum_{v' \notin \mathcal{H}_u^S} \ln \sigma(\hat{y}_{uv} - \hat{y}_{uv'}) \tag{9}$$

4 EXPERIMENT

We conduct comprehensive experiments to answer the following research questions:

- **RQ1** How powerful is Tiger for the DZSR task?
- **RQ2** What are the most difficult challenges of DZSR and how Tiger addresses them?
- **RQ3** How can Tiger benefit from the out-domain dataset?
- **RQ4** Can Tiger continue to assist recommender systems after their cold-start stage?

4.1 Dataset

To fully evaluate the performance of Tiger in a real-world setting, we conduct experiments on four public representative datasets including different overlap levels of users and items:

- Amazon Movies & TV (AM) and Amazon Books (AB) are two subsets of the Amazon datasets [2], which contain product reviews and metadata from Amazon [12] and nowadays have become popular benchmark datasets for recommender systems. We use "reviewerID" to bridge users across the two datasets.
- Movielens(ML) [3] dataset [11] contains anonymous movie ratings to describe users' preferences on movies, which is widely used in the evaluation of recommender systems.
- LastFM(LFM) [4] dataset contains music listening information from the world's largest online music service Last.fm [4]. The ML and LFM datasets are used to extend the source domain and verify if Tiger can benefit from the public datasets without overlapping entities in the target domain.

For AB and AM datasets, we only keep users with more than three historical interactions on the both datasets. The domain-level cold-start performance is evaluated on the target domain with users represented by their behaviors on the source domain. User behaviors in the target domain are only used for evaluation, not for training or user modeling. For ML and LFM datasets, their users have no overlap with two Amazon datasets' users. Both datasets are filtered and reduced from giant versions (ML20M and LFM1B) into about a million records. All datasets are split by the popular One-Leave-Out strategy [13, 30]. Specifically, for each user, we randomly select two historical interactions from the dataset, one for validation, the other for testing. The rest of the dataset is regarded as the training set. Importantly, dataset of the target domain is also split into training, validation and test sets, in which the training set is only used for training the oracle models and not available for models under the zero-shot setting. The validation and test sets are shared for all models to ensure fair evaluations.

All items involved are linked to Freebase[5], an online knowledge base containing massive structured triples. To link items in the recommendation dataset to entities in the knowledge graph, we follow the suggestion of KB4Rec [48] to obtain the corresponding entity of items in a retrieval fashion: for each item, we collect the product title from the metadata and use the title as the query to receive the top-1 entity by the Google Knowledge Graph Search API[6], in which only the first 64 characters of the title are used to form the keyword due to the limitation of API. The KG linkage for AmazonBook is taken from the public data of KB4Rec directly. To facilitate research, items that get empty results from API or do not exist in Freebase are filtered in our experiments. We extract a sub-graph from the full Freebase version with the entities linked by items mentioned above as seeds to reduce computing costs. Specifically, the inter-linkages between the linked entity seeds and their neighbors are collected to form a sub-graph with 3,599,000 entities and 32,372,637 edges. All experiments in this paper are conducted on the same sub-KG dataset. The detailed statistics of datasets are shown in Table 2. Both the dataset and source code will be released upon acceptance of this paper.

Table 2: Some basic dataset statistics

Dataset	**AB**	**AM**	**ML**	**LFM**
#User	11,240	11,240	6,040	18,029
#Item	47,377	16,100	3,655	311,994
#Interaction	202,223	142,395	997,580	1,006,639
#Entity	3,599,000			
#Relation	2,089			
#Triple	32,372,637			

4.2 Baselines

In terms of baselines, we compare Tiger with a series of models, including existing methods and their alternates. Specifically, the random and oracle baselines are used to indicate the lower and upper bounds of performance. We carefully reproduce these baselines according to their original papers and open-source codes, and try our best to ensure fair comparisons in our experiment. The baseline models involved in our experiment are as follows:

- **Random**. As the lower bound, the random baseline directly recommends a random item for the given user. Any valuable model is expected to perform better than random results.
- **Oracle**. As the upper bound, the oracle baseline is trained with the full interactions of the target domain. BPR [30] is a classical and robust baseline in the in-domain recommendation field. Since it performs well in many situations, we use it as the oracle.
- **NLP-based**. Textual content is another alternative side information to represent items in a universal way. For instance, [7, 44, 46] utilize pre-trained language models to encode items with descriptions for zero-shot predictions. Thus, we collect the review text

[2]https://jmcauley.ucsd.edu/data/amazon/
[3]https://grouplens.org/datasets/movielens/
[4]https://grouplens.org/datasets/hetrec-2011/
[5]https://developers.google.com/freebase
[6]https://developers.google.com/knowledge-graph

of items and employ SBert [28] to generate items' embeddings. SBert is pre-trained with sentence representation tasks. We do not fine-tune SBert on the source domain's user-item interactions data because in our experiments it leads to a worse performance.

• **KGE-based**. Since traditional knowledge graph embedding (KGE) models learn entity embeddings in a fully self-supervised way based on the KG structure, they can be naturally regarded as competitive baselines for zero-shot recommendations. Thus, we use TransE [3] to generate representations of items, which are used to replace the extractor module in Section 3.2. Fundamentally, KGE belongs to content-based recommendation models, with content being the KG embedding rather than raw attributes.

• **GCN-based**. The GCN-based approaches are widely used to integrate knowledge graph information into the collaborative signals to improve the generalization ability. KGCN [37] is a classical knowledge graph enhanced GCN-based recommender system. To compare fairly, we use the mean-aggregator version of KGCN to evaluate its performance under the DZSR setting.

• **Tiger (UserAsEmb)**. The UserAsEmb alternative of Tiger directly uses the ID-embedding to represent users as Eq. (6) instead of our reconstructed interest graph embeddings from historical interactions in the source domain as Eq. (7).

4.3 Evaluation and Other Settings

We use two popular measures to evaluate all models: Hit Ratio (H@K) and Normalized Discounted Cumulative Gain (N@K), where K is selected from classical settings {10, 100} in consideration of both the precision and recall property. The higher value of all measures means the better performance. In the test phase, all models are asked to rank all items that each user has not interacted with. In order to reduce the impact of random noise, each experiment is independently repeated three times on the same condition, and the average performance is reported in this paper.

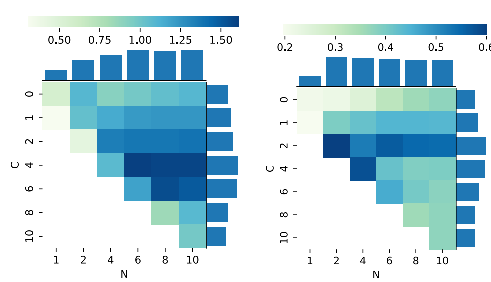
We use PyTorch [27] and Adagrad [8] optimizer to implement all models. For reproducibility and scalability, we use the popular framework DGL [38] to construct the graph convolutional network and perform the message passing mechanism. To ensure a fair comparison, the dimension d is assigned as 32 and batch size is set to 8192 for all experiments. In the training stage, we select the learning rate $\mu \in \{0.001, 0.003, 0.01, 0.03, 0.3, 1.0\}$, the number of GCN layers $N \in \{1, 2, 4, 6, 8, 10\}$, and the number of discarded layers $C \in \{0, 1, 2, 4, 6, 8, 10\}$. Each experiment runs 100 epochs for all datasets. Grid search with early stop strategy on NDCG@100 of the validation set is adopted to determine the best hyper-parameter configuration as follows: the learning rate μ is 0.3, the number of GCN layers N is 6 for AmazonMovie, 6 for AmazonBook, the number of discarded C is 4 for AmazonMovie and 2 for AmazonBook. All experiments are trained on the source domain and evaluated on the validation set of the target domain, then performances on the test set of the target domain are reported.

4.4 Overall Performance (RQ1)

The performance of different models is summarized in Table 3. Since we have two datasets with overlapping users, i.e., the AB and AM, we switch their roles acting as the zero-shot target domain (abbreviated as target-AB and target-AM), so that we have two groups of results in Table 3, from which we have the following observations: (1) In general, our proposed Tiger and its variants, significantly outperform the other baselines (*Random, NLP-based, TransE*, and *KGCN*) in the zero-shot setting, on both target-AB and target-AM scenarios. Compared with the oracle BPR model, the best setting of Tiger can perform as well as about 50% performance of the oracle results in terms of N@100, which indicates how powerful Tiger is in making recommendations for a completely new domain. (2) Both *NLP-based* and *TransE* are content-based zero-shot methods. On both target-AB and target-AM, *TransE* consistently outperforms *NLP-based*, which indicates that knowledge graph is a better intermediary for cross-domain information transferring in the recommendation scenario. (3) *KGCN* is better than *TransE*, which demonstrates that training entity embeddings with the source domain's user behaviors can help to distill the useful recommendation-oriented knowledge from the task-agnostic structure-oriented knowledge graph. (4) Compared with *TransE* and *KGCN*, the Tiger methods demonstrate absolute advantages in performance. Note that all these methods consume the same type of data, which is the interest graph. This phenomenon proves the effectiveness of Tiger, as well as its key components including GCN layers discarding and domain item adaptation. (5) By comparing *Tiger(UserAsEmb)* with *Tiger(normal)*, we can observe that reconstructing users' interest from their history behaviors is better than directly learning user latent embeddings as their interest. (6) The out-domain datasets, such as ML and LFM, can be easily absorbed into Tiger and boost the final performance significantly. This phenomenon positively supports our claim – Tiger is an effective approach to facilitate collaborative signals distributed in various domains to propagate on the interest graph. The detailed analysis of the out-domain will be carried out in section 4.6.

4.5 Ablation study (RQ2)

In this section, we provide some in-depth analysis of how each key component in Tiger contributes to solving the challenges of DZSR.



(a) AmazonMovie (b) AmazonBook

Figure 3: Performance comparison between different settings. The darker means the better.

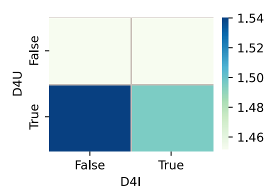
4.5.1 What is the best setting of GCN layers? As mentioned in Section 3.2, one of the biggest challenges for DZSR is to facilitate collaborative signals to be propagated on the interest graph, and the key solution is bottom-layer discarding. Thus, N and C become critical

Table 3: Performance comparison. The higher value of all measures means the better performance. The best zero-shot result is highlighted in bold and the runner-up is underlined, the same below. * indicates the oracle result.

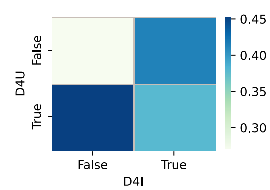
Model	Source	Target	H@10	N@10	H@100	N@100
Random	-	AM	0.0620	0.0282	0.6211	0.1300
BPR (Oracle)	-	AM	2.8440*	1.4066*	14.1040*	3.5393*
NLP-based	AB	AM	0.1307	0.0488	1.2900	0.2586
TransE	AB	AM	0.3203	0.1580	1.4858	0.3719
KGCN	AB	AM	0.5368	0.2491	3.6032	0.8155
Tiger (UserAsEmb)	AB	AM	0.7711	<u>0.4198</u>	4.7242	1.1510
Tiger (normal)	AB	AM	<u>0.9312</u>	0.3751	**7.3072**	<u>1.5401</u>
Tiger (+ out domain)	ML+LFM+AB	AM	**1.0854**	**0.7484**	<u>7.1886</u>	**1.8811**
Random	-	AB	0.0211	0.0096	0.2111	0.0442
BPR (Oracle)	-	AB	0.7859*	0.4051*	3.9472*	1.0014*
NLP-based	AM	AB	0.0505	0.0202	0.4580	0.0948
TransE	AM	AB	0.0623	0.0293	0.3915	0.0913
KGCN	AM	AB	0.0860	0.0487	0.7117	0.1657
Tiger (UserAsEmb)	AM	AB	0.2343	0.1185	1.2604	0.3100
Tiger (normal)	AM	AB	<u>0.3055</u>	<u>0.1370</u>	<u>1.9692</u>	<u>0.4519</u>
Tiger (+ out domain)	ML+LFM+AM	AB	**0.5872**	**0.3392**	**2.5178**	**0.5659**

hyper-parameters. We vary their values, with $N \in \{1, 2, 4, 6, 8, 10\}$, $C \in \{0, 1, 2, 4, 6, 8, 10\}$ and satisfy the condition $S \leq N$ to see how Tiger will be impacted. We plot the NDCG@100 score of Tiger in the form of heat map in Figure 3, in which a joint cell indicates the model score under the specific joint configuration of C and N, and a marginal bar on the row (or column) indicates the best score (we called it *margin score* hereinafter) for a specific C (or N). From Figure 3 we can observe that: (1) From N horizontal margin scores, a larger N brings better performance in both datasets because a bigger size of GCN layers can carry more universal interest. However, with the N growing up more than 6, the gain becomes not statistically significant by increasing N. Hence, N should be a trade-off between computing cost and performance. (2) The C vertical margin scores demonstrate a trend that, within a certain threshold, e.g., $C = 4$ for AmazonMovie and $C = 2$ for AmazonBook, the performance increases significantly with the increase of C, which means that bottom layers of item GCN capture more local domain's information, so discarding them can force information to be propagated to a broader range on the interest graph. However, after the proper threshold, further increasing C will lead to a performance drop. This is because the aggregated graph embedding is too coarse to carry the precise interest of users when C is too large.

4.5.2 Are users' preferences really decomposed? The ablation study in Section 4.5.1 has shown that the upper GCN layers capture general preferences, while the lower GCN layers preserve local representations. To further verify such assumption, analytical experiments are conducted to determine if the user' preference is really decomposed into different levels. Based under the optimal setting provided in Section 3, we alter GCN levels of users and items with two model settings: (1) **D4U**: whether to use $\mathbf{z}_v^{\#}$ to discard the lower GCN embedding of historical items when reconstructing the user interest in Eq. (7) or $\mathbf{z}_v^{*}$; (2) **D4I**: whether to use $\mathbf{z}_v^{\#}$ when presenting the item itself in Eq. (8) or $\mathbf{z}_v^{*}$; We evaluate all four settings of D4U and D4I combinations on the same experiment condition and report their performances on NDCG@100. For intuitive analysis, we visualize the comparison in Figure 4.



(a) AmazonMovie

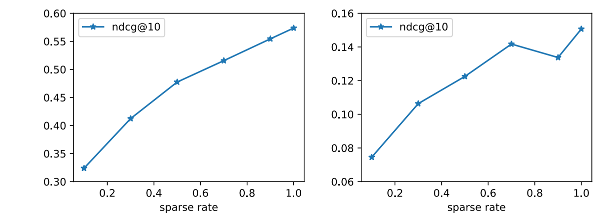


(b) AmazonBook

Figure 4: Result comparison of decomposed representations

As shown in Figure 4, we observe that (1) The result is terrible when both D4U and D4I settings are false in all datasets, confirming the necessity of discarding operation on GCN output embeddings of items to extract more transferable information. (2) As the setting of discarding for user reconstruction and keeping for the item itself performs best, we learn that lower GCN layer embeddings encode the item-relevant information and facilitate accurate predictions.

4.5.3 Does a larger size of Knowledge Graph help? How does the size of knowledge graph affect the performance and parameter behavior of our model? First, several "sparse" versions of knowledge graph dataset with a size of p are conducted by random dropping triplets of original KG, where $p \in \{0.1, 0.3, 0.5, 0.7, 0.9, 1.0\}$. Then, models with the same training strategy described in Section 4 are trained on different sparse sizes of KG. Finally, performances are reported and reorganized in Figure 5. From Figure 5, we observed: our model does generate a transferable representation with the help of knowledge graph. With a larger size of knowledge graph trained on, the model performs better, which suggests researchers collecting more KG data to better serve in the industrial scenario.



(a) AmazonMovie (b) AmazonBook

Figure 5: Performance under different sparse levels of KG

4.6 Tiger with out-domain datasets (RQ3)

Besides the datasets on the same platform, there are numerous public datasets collected in different ways available for training recommender systems, which we call the out-domain datasets. As the training of Tiger does not include any domain-specific parameters but the shared graph embedding $\mathbf{E} = \{\mathbf{e}_v | v \in \mathcal{G}\}$, another interesting question raises: does Tiger benefits from public datasets without overlapped users or items to propagate collaborative signals to the target domain? To answer this question, we set up two domain-level cold-start settings with out-domain datasets. Then models are trained on the two settings respectively and tested on the source and target domain datasets as before.

4.6.1 History available setting (HAS). In this setting, out-domain datasets are jointly trained with the source domain. Concretely, we extend source domain $\mathcal{S}$ of Eq. (1) with out-domain datasets $\mathcal{Z}$ as:

$$\Theta^* = \mathop{\mathbf{arg\,min}}_{\Theta} \mathcal{L}(y(\mathcal{S} + \mathcal{Z}), \mathcal{S} + \mathcal{Z}; \mathcal{G}) \tag{10}$$

4.6.2 History protected setting (HPS). For some purposes like the access limitation, privacy protection policy, or real-time response requirement at the mobile endpoint, it is a practical consideration that the historical interactions of users are available in the stage of testing but not in training, which means the model has to be trained in out-domain datasets without any overlapping users or items. Concretely, we replace source domain $\mathcal{S}$ of Eq. (1) with out-domain datasets $\mathcal{Z}$ as:

$$\Theta^* = \mathop{\mathbf{arg\,min}}_{\Theta} \mathcal{L}(y(\mathcal{Z}), \mathcal{Z}; \mathcal{G}) \tag{11}$$

From the results in Table 4, we observe that: (1) With out-domain datasets in HAS, the models on extended training datasets achieve a better performance, which confirms Tiger to absorb the universal knowledge from public datasets to improve prediction accuracy of the DZSR task. (2) The setting trained on the HPS without any user overlapped, still achieves promising performance, which makes it possible for pre-trained Tiger to cold-start unrelated recommender systems as long as items in the dataset can be linked to the KG.

4.7 Tiger for normal recommendation (RQ4)

With discussion and verification above, the Tiger model has been proved competent for the DZSR task. In need of industrial practice, a follow-up question is whether Tiger still plays a role after the cold-start stage. Thus, experiments are conducted on the oracle setting, where models are trained on the training set of the target domain to represent a recommender system that has already warmed up. From Table 5, we can see that Tiger can not only work on domain-level cold-start recommendation but also achieve promising performance in the subsequent optimization. From Figure 6, we get consistent observations with related literature on knowledge-aware recommendations: when $C = 0$, using more GCN layers can fuse more semantic KG information and therefore lead to a better performance. Different from the behavior analysed in Section 4.5.3, C is discouraged because the in-domain scenario asks for more item-relevant clues, which confirms that the performance of Tiger is exactly based on the transferable interest discussed in Section 4.5.2.

Table 4: Performance comparison of Tiger trained on different datasets

Training	Target	H@10	N@10	H@100	N@100
AB	AM	0.9312	0.3751	7.3072	1.5401
ML	AM	0.6139	0.2391	**7.3577**	1.4408
LFM	AM	0.9253	0.4784	6.3968	1.4474
ML+AB	AM	**1.1210**	0.5418	6.7438	1.6184
LFM+AB	AM	1.1121	0.3882	7.2598	1.4984
ML+LFM+AB	AM	1.0854	**0.7484**	7.1886	**1.8811**
AM	AB	0.3055	0.1370	1.9692	0.4519
ML	AB	0.3203	0.1456	1.7527	0.4121
LFM	AB	0.1275	0.0397	2.1501	0.4275
ML+AM	AB	0.5397	0.3013	2.0848	**0.6007**
LFM+AM	AB	0.4938	0.3211	1.6770	0.5434
ML+LFM+AM	AB	**0.5872**	**0.3392**	**2.5178**	0.5659

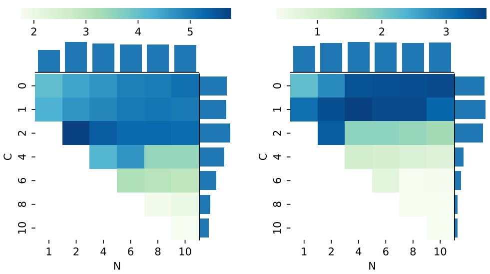
Table 5: Performance comparison in the subsequent optimization

Model	Target	H@10	N@10	H@100	N@100
Random	AM	0.0620	0.0282	0.6211	0.1300
BPR	AM	2.8440	1.4066	14.1040	3.5393
KGCN	AM	4.2912	2.1425	20.1097	5.1583
Tiger	AM	**5.0445**	**2.5381**	**21.9781**	**5.7812**
Random	AB	0.0211	0.0096	0.2111	0.0442
BPR	AB	0.7859	0.4051	3.9472	1.0014
KGCN	AB	3.3007	1.8283	12.0196	3.5146
Tiger	AB	**4.8221**	**2.5774**	**15.3203**	**4.6344**

5 RELATED WORK

5.1 Knowledge-enhanced Recommendations

Knowledge graphs (KGs) contain massive structural information of entities. As an external data source, KGs have great potentials to improve the accuracy, interpretability and diversity of recommender systems. DKN [35] introduces knowledge-aware convolutional neural networks (KCNN), which inject external information from KGs into text in news articles to get better news representations. KRED [23] is a more flexible and efficient framework, which can refine an arbitrary base article representation with knowledge, and the refined representation can benefit multiple downstream tasks related



(a) AmazonMovie (b) AmazonBook

Figure 6: Performance with different settings of GCN layer for the normal recommendation

to news recommendations. Considering that a KG naturally provides closer connections between entities, [34] tries to propagate a user's preference over a KG, so that the user-item interaction sparsity problem can be alleviated. The proposed method, RippleNet, is basically a memory network which captures the user's multi-hop preference on the KG. In contrast to RippleNet, KGCN [37] and KGAT [39] leverages knowledge gragh convolution networks to integrate neighborhood representation for items. [36] and [5] use a multi-task training framework to train the recommendation task and the knowledge graph embedding task. Another line of research is KG reasoning, in which the goal is to find high-quality paths on the KG to connect two nodes. KPRN [40] enumerates all the possible connecting paths, then uses a path encoder to select best paths among all the path candidates. Next, PGPR [45] and ADAC [47] formulate the KG reasoning as a path finding task rather than a path enumeration task, and use reinforcement learning techniques to learn navigation policies. [22] learns a subgraph generator to extract the most important subgraph on the KG for a given item, the the relations of two items can be inferred from their corresponding generated subgraph. Different from all the aforementioned works, in this paper, we discuss how to connect isolated users' collaborative behaviors from different domains with the help of KGs, and perform zero-shot recommendations in a new domain.

5.2 Cross-domain Recommendations

To alleviate the prevalent data sparsity problem in recommender systems, cross-domain recommendation (CDR) [2] has emerged to utilize information across domains. Based on different approaches of knowledge transfer, previous works can be roughly divided into content-based methods [9, 14, 32], embedding-based methods [6, 18, 25, 51] and model-based methods [16, 17]. Content-based approaches mainly leverage attribute-level relevance across domains by linking features of users and items, such as user reviews [32], item tags [9] and knowledge graphs [14]. The second category focuses on embedding sharing [6, 51] or embedding mapping [18, 25] of overlapping users/items. DTCDR [51] shares the embedding of common users in the combination layer to integrate multi-domain knowledge, while HeroGRAPH [6] combines in-domain embedding and heterogeneous graph embedding connecting multiple domains to obtain the enhanced representation of entities. Another research direction of embedding-based methods is to learn a mapping function from the source to the target domain such as EMCDR [25], SSCDR [18]. As for model-based CDR approaches, like XPTRANS [17], the core idea is collaborative training through cross-connections between models. Conventional CDR still requires sparse interactive data in the target domain. In other words, existing CDR methods can not solve the proposed domain-level cold-start problem without any interactions, which is the main contribution of our work.

5.3 Cold-start Recommendations

How to make predictions for newly join users/items that have no or very few interaction logs is a challenging task. In this line of research, the key is to effectively leverage side information to make up the missing signals of collaborative behaviors. Existing approaches mainly include transforming from content-based (CB) embeddings to collaborative filtering (CF) embeddings [1, 43, 49, 52], adaptive fusion of CB and CF embeddings [31, 33], and warm-up CF embedding with meta learning [26, 42, 50]. For example, [1] learns a deep neural network (DNN) to transform items' content, including tags, numeric features, and textual content, into their CF representations. [43] uses contrastive learning to better align items' content signals with collaborative signals. [33] uses a dropout mechanism to train user/item DNNs, so that content information and collaborative signals can be fused adaptively according to each specific input data. [31] learns attention components to adaptively fuse CB and CF signals in an explicit manner. In addition to inferring CF embeddings from items' side information, [26] introduces a meta-learning-based method that can generate good initial CF embeddings and speed up CF embeddings' refinement process. However, all these methods assume that there exist some collaborative behaviors in the target recommendation domain. The goal is to build a desirable bridge or fusion method between content information and collaborative behaviors, which is not a domain-level zero-shot problem.

6 CONCLUSION

This paper discusses the task of domain-level zero-shot recommendation (DZSR). Unlike user/item cold-start or cross-domain recommendation tasks, in DZSR, we cannot access user-item interaction logs in the target domain. We propose a solution named *Tiger*, which aims to project and fuse users' universal preferences into a common interest graph bridging different domains' collaborative behaviors. The embeddings learned by Tiger are transferable to the target domain in a zero-shot prediction manner. Through extensive experiments, we verify that some fundamental mechanisms in Tiger, such as bottom-layer discarding operation, domain adaptation, and connection of out-domain datasets, play an essential role in learning meaningful transferable embeddings. Tiger is essentially a type of knowledge graph pre-training model for recommender systems. Pre-training with large-scale universal datasets has been proven to be an effective approach for language models. Tiger acts as a prior work to leverage knowledge graph for bridging isolated recommender system datasets. In future works, we will develop better entity linking tools so that more recommendation datasets can be linked to the interest graph, and perform truly large-scale interest graph pre-training for recommender systems to further shrink the gap between Tiger's and the oracle model's performance.

REFERENCES

[1] Oren Barkan, Noam Koenigstein, Eylon Yogev, and Ori Katz. 2019. CB2CF: A Neural Multiview Content-to-Collaborative Filtering Model for Completely Cold Item Recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems* (Copenhagen, Denmark) *(RecSys '19)*. Association for Computing Machinery, New York, NY, USA, 228–236. https://doi.org/10.1145/3298689.3347038

[2] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2007. Cross-Domain Mediation in Collaborative Filtering. In *User Modeling 2007, 11th International Conference, UM 2007, Corfu, Greece, June 25-29, 2007, Proceedings (Lecture Notes in Computer Science, Vol. 4511)*, Cristina Conati, Kathleen F. McCoy, and Georgios Paliouras (Eds.). Springer, 355–359. https://doi.org/10.1007/978-3-540-73078-1_44

[3] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-Relational Data. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (Lake Tahoe, Nevada) *(NIPS'13)*. Curran Associates Inc., Red Hook, NY, USA, 2787–2795.

[4] Iván Cantador, Peter Brusilovsky, and Tsvi Kuflik. 2011. 2nd Workshop on Information Heterogeneity and Fusion in Recommender Systems (HetRec 2011). In *Proceedings of the 5th ACM conference on Recommender systems* (Chicago, IL, USA) *(RecSys 2011)*. ACM, New York, NY, USA.

[5] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. 2019. Unifying Knowledge Graph Learning and Recommendation: Towards a Better Understanding of User Preferences. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia (Eds.). ACM, 151–161. https://doi.org/10.1145/3308558.3313705

[6] Qiang Cui, Tao Wei, Yafeng Zhang, and Qing Zhang. 2020. HeroGRAPH: A Heterogeneous Graph Framework for Multi-Target Cross-Domain Recommendation. In *Proceedings of the 3rd Workshop on Online Recommender Systems and User Modeling co-located with the 14th ACM Conference on Recommender Systems (RecSys 2020), Virtual Event, September 25, 2020 (CEUR Workshop Proceedings, Vol. 2715)*, João Vinagre, Alípio Mário Jorge, Marie Al-Ghossein, and Albert Bifet (Eds.). CEUR-WS.org. http://ceur-ws.org/Vol-2715/paper6.pdf

[7] Hao Ding, Yifei Ma, Anoop Deoras, Yuyang Wang, and Hao Wang. 2021. Zero-shot recommender systems. *arXiv preprint arXiv:2105.08318* (2021).

[8] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research* 12, 61 (2011), 2121–2159.

[9] Ignacio Fernández-Tobías and Iván Cantador. 2014. Exploiting Social Tags in Matrix Factorization Models for Cross-domain Collaborative Filtering. In *Proceedings of the 1st Workshop on New Trends in Content-based Recommender Systems co-located with the 8th ACM Conference on Recommender Systems, CBRecSys@RecSys 2014, Foster City, Silicon Valley, California, USA, October 6, 2014 (CEUR Workshop Proceedings, Vol. 1245)*, Toine Bogers, Marijn Koolen, and Iván Cantador (Eds.). CEUR-WS.org, 34–41. http://ceur-ws.org/Vol-1245/cbrecsys2014-paper06.pdf

[10] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. *arXiv preprint arXiv:1703.04247* (2017).

[11] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[12] Ruining He and Julian McAuley. 2016. Ups and Downs: Modeling the Visual Evolution of Fashion Trends with One-Class Collaborative Filtering. In *Proceedings of the 25th International Conference on World Wide Web (WWW '16)*. International World Wide Web Conferences Steering Committee, 507–517. https://doi.org/10.1145/2872427.2883037

[13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.

[14] Benjamin Heitmann and Conor Hayes. 2016. SemStim: Exploiting Knowledge Graphs for Cross-Domain Recommendation. In *IEEE International Conference on Data Mining Workshops, ICDM Workshops 2016, December 12-15, 2016, Barcelona, Spain*, Carlotta Domeniconi, Francesco Gullo, Francesco Bonchi, Josep Domingo-Ferrer, Ricardo Baeza-Yates, Zhi-Hua Zhou, and Xindong Wu (Eds.). IEEE Computer Society, 999–1006. https://doi.org/10.1109/ICDMW.2016.0145

[15] Balázs Hidasi and Alexandros Karatzoglou. 2018. Recurrent neural networks with top-k gains for session-based recommendations. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 843–852.

[16] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, 667–676. https://doi.org/10.1145/3269206.3271684

[17] Meng Jiang, Peng Cui, Nicholas Jing Yuan, Xing Xie, and Shiqiang Yang. 2016. Little Is Much: Bridging Cross-Platform Behaviors through Overlapped Crowds. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA*, Dale Schuurmans and Michael P. Wellman (Eds.). AAAI Press, 13–19. http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12009

[18] SeongKu Kang, Junyoung Hwang, Dongha Lee, and Hwanjo Yu. 2019. Semi-Supervised Learning for Cross-Domain Recommendation to Cold-Start Users. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 1563–1572. https://doi.org/10.1145/3357384.3357914

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

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

[21] Chao Li, Zhiyuan Liu, Mengmeng Wu, Yuchi Xu, Huan Zhao, Pipei Huang, Guoliang Kang, Qiwei Chen, Wei Li, and Dik Lun Lee. 2019. Multi-interest network with dynamic routing for recommendation at Tmall. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 2615–2623.

[22] Danyang Liu, Jianxun Lian, Zheng Liu, Xiting Wang, Guangzhong Sun, and Xing Xie. 2021. Reinforced Anchor Knowledge Graph Generation for News Recommendation Reasoning. In *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, Feida Zhu, Beng Chin Ooi, and Chunyan Miao (Eds.). ACM, 1055–1065. https://doi.org/10.1145/3447548.3467315

[23] Danyang Liu, Jianxun Lian, Shiyin Wang, Ying Qiao, Jiun-Hung Chen, Guangzhong Sun, and Xing Xie. 2020. KRED: Knowledge-Aware Document Representation for News Recommendations. In *Fourteenth ACM Conference on Recommender Systems* (Virtual Event, Brazil) *(RecSys '20)*. Association for Computing Machinery, New York, NY, USA, 200–209. https://doi.org/10.1145/3383313.3412237

[24] Zheng Liu, Jianxun Lian, Junhan Yang, Defu Lian, and Xing Xie. 2020. Octopus: Comprehensive and elastic user representation for the generation of recommendation candidates. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 289–298.

[25] Tong Man, Huawei Shen, Xiaolong Jin, and Xueqi Cheng. 2017. Cross-Domain Recommendation: An Embedding and Mapping Approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, Carles Sierra (Ed.). ijcai.org, 2464–2470. https://doi.org/10.24963/ijcai.2017/343

[26] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. 2019. Warm Up Cold-Start Advertisements: Improving CTR Predictions via Learning to Learn ID Embeddings. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Paris, France) *(SIGIR'19)*. Association for Computing Machinery, New York, NY, USA, 695–704. https://doi.org/10.1145/3331184.3331268

[27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, and et al. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2019/hash/bdbca288fee7f92f2bfa9f7012727740-Abstract.html

[28] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084

[29] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000.

[30] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence (UAI'09)*. AUAI Press, 452–461.

[31] Shaoyun Shi, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Attention-Based Adaptive Model to Unify Warm and Cold Starts Recommendation. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management* (Torino, Italy) *(CIKM '18)*. Association for Computing Machinery, New York, NY, USA, 127–136. https://doi.org/10.1145/3269206.3271710

[32] Shulong Tan, Jiajun Bu, Xuzhen Qin, Chun Chen, and Deng Cai. 2014. Cross domain recommendation based on multi-type media fusion. *Neurocomputing* 127 (2014), 124–134. https://doi.org/10.1016/j.neucom.2013.08.034

[33] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. DropoutNet: Addressing Cold Start in Recommender Systems. In *Proceedings of the 31st International Conference on Neural Information Processing Systems* (Long Beach, California, USA) *(NIPS'17)*. Curran Associates Inc., Red Hook, NY, USA, 4964–4973.

[34] Hongwei Wang, Fuzheng Zhang, Jialin Wang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2018. RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018*, Alfredo Cuzzocrea, James Allan, Norman W. Paton, Divesh Srivastava, Rakesh Agrawal, Andrei Z. Broder, Mohammed J. Zaki, K. Selçuk Candan, Alexandros Labrinidis, Assaf Schuster, and Haixun Wang (Eds.). ACM, 417–426. https://doi.org/10.1145/3269206.3271739

[35] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep Knowledge-Aware Network for News Recommendation. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web, WWW 2018, Lyon, France, April 23-27, 2018*, Pierre-Antoine Champin, Fabien L. Gandon, Mounia Lalmas, and Panagiotis G. Ipeirotis (Eds.). ACM, 1835–1844. https://doi.org/10.1145/3178876.3186175

[36] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia (Eds.). ACM, 2000–2010. https://doi.org/10.1145/3308558.3313411

[37] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge Graph Convolutional Networks for Recommender Systems. In *The World Wide Web Conference, WWW 2019, San Francisco, CA, USA, May 13-17, 2019*, Ling Liu, Ryen W. White, Amin Mantrach, Fabrizio Silvestri, Julian J. McAuley, Ricardo Baeza-Yates, and Leila Zia (Eds.). ACM, 3307–3313. https://doi.org/10.1145/3308558.3313417

[38] Minjie Wang, Da Zheng, Zihao Ye, Quan Gan, Mufei Li, Xiang Song, Jinjing Zhou, Chao Ma, Lingfan Yu, Yu Gai, Tianjun Xiao, Tong He, George Karypis, Jinyang Li, and Zheng Zhang. 2019. Deep Graph Library: A Graph-Centric, Highly-Performant Package for Graph Neural Networks. *arXiv preprint arXiv:1909.01315* (2019).

[39] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (Eds.). ACM, 950–958. https://doi.org/10.1145/3292500.3330989

[40] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 5329–5336.

[41] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge Graph Embedding by Translating on Hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence* (Québec City, Québec, Canada) *(AAAI'14)*. AAAI Press, 1112–1119.

[42] Tianxin Wei, Ziwei Wu, Ruirui Li, Ziniu Hu, Fuli Feng, Xiangnan He, Yizhou Sun, and Wei Wang. 2020. Fast adaptation for cold-start collaborative filtering with meta-learning. In *2020 IEEE International Conference on Data Mining (ICDM)*. IEEE, 661–670.

[43] Yinwei Wei, Xiang Wang, Qi Li, Liqiang Nie, Yan Li, Xuanping Li, and Tat-Seng Chua. 2021. Contrastive Learning for Cold-Start Recommendation. In *Proceedings of the 29th ACM International Conference on Multimedia* (Virtual Event, China) *(MM '21)*. Association for Computing Machinery, New York, NY, USA, 5382–5390. https://doi.org/10.1145/3474085.3475665

[44] Tao Wu, Ellie Ka-In Chio, Heng-Tze Cheng, Yu Du, Steffen Rendle, Dima Kuzmin, Ritesh Agarwal, Li Zhang, John Anderson, Sarvjeet Singh, Tushar Chandra, Ed H. Chi, Wen Li, Ankit Kumar, Xiang Ma, Alex Soares, Nitin Jindal, and Pei Cao. 2020. Zero-Shot Heterogeneous Transfer Learning from Recommender Systems to Cold-Start Search Retrieval. In *Proceedings of the 29th ACM International Conference on Information and Knowledge Management* (Virtual Event, Ireland) *(CIKM '20)*. Association for Computing Machinery, New York, NY, USA, 2821–2828. https://doi.org/10.1145/3340531.3412752

[45] Yikun Xian, Zuohui Fu, S Muthukrishnan, Gerard De Melo, and Yongfeng Zhang. 2019. Reinforcement knowledge graph reasoning for explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 285–294.

[46] Yuhui Zhang, Hao Ding, Zeren Shui, Yifei Ma, James Zou, Anoop Deoras, and Hao Wang. 2021. Language Models as Recommender Systems: Evaluations and Limitations. In *I (Still) Can't Believe It's Not Better! NeurIPS 2021 Workshop*.

[47] Kangzhi Zhao, Xiting Wang, Yuren Zhang, Li Zhao, Zheng Liu, Chunxiao Xing, and Xing Xie. 2020. Leveraging Demonstrations for Reinforcement Recommendation Reasoning over Knowledge Graphs. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 239–248.

[48] Wayne Xin Zhao, Gaole He, Kunlin Yang, Hong-Jian Dou, Jin Huang, Siqi Ouyang, and Ji-Rong Wen. 2019. KB4Rec: A Data Set for Linking Knowledge Bases with Recommender Systems. *Data Intelligence* 1, 2 (2019), 121–136. https://doi.org/10.1162/dint_a_00008

[49] Jiawei Zheng, Qianli Ma, Hao Gu, and Zhenjing Zheng. 2021. Multi-View Denoising Graph Auto-Encoders on Heterogeneous Information Networks for Cold-Start Recommendation. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining* (Virtual Event, Singapore) *(KDD '21)*. Association for Computing Machinery, New York, NY, USA, 2338–2348. https://doi.org/10.1145/3447548.3467427

[50] Yujia Zheng, Siyi Liu, Zekun Li, and Shu Wu. 2021. Cold-start Sequential Recommendation via Meta Learner. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*. 4706–4713. https://ojs.aaai.org/index.php/AAAI/article/view/16601

[51] Feng Zhu, Chaochao Chen, Yan Wang, Guanfeng Liu, and Xiaolin Zheng. 2019. DTCDR: A Framework for Dual-Target Cross-Domain Recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019*, Wenwu Zhu, Dacheng Tao, Xueqi Cheng, Peng Cui, Elke A. Rundensteiner, David Carmel, Qi He, and Jeffrey Xu Yu (Eds.). ACM, 1533–1542. https://doi.org/10.1145/3357384.3357992

[52] Ziwei Zhu, Shahin Sefati, Parsa Saadatpanah, and James Caverlee. 2020. *Recommendation for New Users and New Items via Randomized Training and Mixture-of-Experts Transformation.* Association for Computing Machinery, New York, NY, USA, 1121–1130. https://doi.org/10.1145/3397271.3401178