Zero-shot Item-based Recommendation via Multi-task Product Knowledge Graph Pre-Training

Ziwei Fan*
zfan20@uic.edu
University of Illinois Chicago
Chicago, IL, USA

Zhiwei Liu†
zhiweiliu@salesforce.com
Salesforce AI Research
Palo Alto, CA, USA

Shelby Heinecke
shelby.heinecke@salesforce.com
Salesforce AI Research
Palo Alto, CA, USA

Jianguo Zhang
jianguozhang@salesforce.com
Salesforce AI Research
Palo Alto, CA, USA

Huan Wang
huan.wang@salesforce.com
Salesforce AI Research
Palo Alto, CA, USA

Caiming Xiong
cxiong@salesforce.com
Salesforce AI Research
Palo Alto, CA, USA

Philip S. Yu
psyu@uic.edu
University of Illinois Chicago
Chicago, IL, USA

ABSTRACT

Existing recommender systems face difficulties with zero-shot items, *i.e.* items that have no historical interactions with users during the training stage. Though recent works extract universal item representation via pre-trained language models (PLMs), they ignore the crucial item relationships. This paper presents a novel paradigm for the Zero-Shot Item-based Recommendation (ZSIR) task, which pre-trains a model on product knowledge graph (PKG) to refine the item features from PLMs. We identify three challenges for pre-training PKG, which are multi-type relations in PKG, semantic divergence between item generic information and relations and domain discrepancy from PKG to downstream ZSIR task. We address the challenges by proposing four pre-training tasks and novel task-oriented adaptation (ToA) layers. Moreover, this paper discusses how to fine-tune the model on new recommendation task such that the ToA layers are adapted to ZSIR task. Comprehensive experiments on 18 markets dataset are conducted to verify the effectiveness of the proposed MPKG model .

CCS CONCEPTS

• **Information systems** → **Information systems applications**.

KEYWORDS

Multi-task Pre-Training, Product Knowledge Graph, Zero-shot Recommendation

*Work done during the internship at Salesforce AI Research.
†Corresponding Author

CIKM '23, October 21–25, 2023, Birmingham, United Kingdom

ACM ISBN 979-8-4007-0124-5/23/10…$15.00
https://doi.org/10.1145/3583780.3615110

ACM Reference Format:
Ziwei Fan, Zhiwei Liu, Shelby Heinecke, Jianguo Zhang, Huan Wang, Caiming Xiong, and Philip S. Yu. 2023. Zero-shot Item-based Recommendation via Multi-task Product Knowledge Graph Pre-Training. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM '23), October 21–25, 2023, Birmingham, United Kingdom.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3583780.3615110

1 INTRODUCTION

Recommender systems (RS) provide personalized information retrieval services to users, and have increasingly become an irreplaceable component in several web applications, such as fashion [7] and movies [9]. Most existing collaborative filtering methods [12, 15, 35] leverage collaborative signals from the historical interactions of users and items. However, collaborative filtering methods are unable to resolve the item cold-start problem. In the item cold-start problem, some items have few to no historical interactions [5, 14, 21, 23]. Without historical interactions, the representations of cold-start items are not optimized during collaborative filtering training. In this paper, we tackle the Zero-Shot Item-based Recommendation task (ZSIR). We present a toy example of ZSIR in Figure 1(a). Compared with other items, i_4 has no interactions with users, and hence i_4 is a zero-shot item.

Item-based recommendation methods [16, 26, 44] represent users as weighted sum of interacted items. The key to resolve the ZSIR task becomes learning representations for zero-shot items [5]. Because of the unavailability of interaction data, we believe the foundation is to infer embeddings of zero-shot items from generic side information, such as *titles* and *descriptions*. With the booming of large pre-trained language models (PLMs) [8, 28, 29, 31], powerful tools are available to infer universal item representations from textual data. PLMs have been explored in recent recommender systems [13, 17, 46]. In these works, PLMs are used to infer item representations, and these item representations are then used as input for downstream recommendation tasks. Nevertheless, we argue that direct inference of item representations from PLMs is far from aligning the semantics of items for recommendation, thus

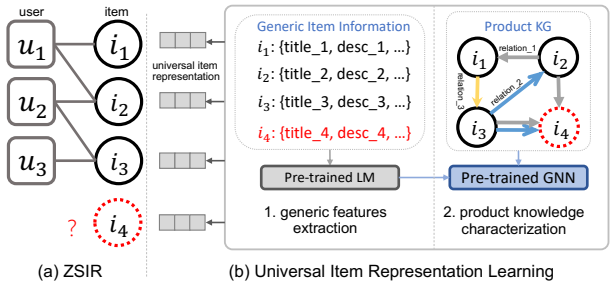


Figure 1: (a) A toy example of Zero-shot Item-based Recommendation (ZSIR), where the zero-shot item i_4 has no interaction with users. (b) A general framework for universal item representation learning, which contains universal feature extraction and product knowledge characterization.

impairing the ZSIR performance. High-quality universal item representations should not only capture the semantics of generic information, but should also incorporate recommendation-oriented knowledge [41, 42], such as complementary and substitution relationships among items.

To this end, we propose a novel universal item representation learning framework, which comprises two components, *i.e.* generic features extraction and product[1] knowledge characterization. The illustration of our universal item representation learning process is presented in Figure 1(b). The generic features extraction module employs a PLM to extract features from generic item side information, such as *titles*, *descriptions*, etc.

However, features directly extracted from generic information are hard to adapt to recommendation task. Therefore, we propose the product knowledge characterization module to enhance the universal representation of items for recommendation. To be more specific, we construct the product knowledge graph (PKG) to represent recommendation-oriented knowledge, where nodes are items and edges are different relations between items, such as complementary, substitution, and etc. Since those relations in PKG are usually retrieved from user-item interactions [43], leveraging PKG for refinement adapts the universal representation for recommendation. We pre-train a graph neural network (GNN) model [1, 2, 20, 27] to refine the features extracted from PLM such that the final universal item representations capture semantics relevant to recommendation tasks. It is noteworthy that we ensure the pre-trained GNN has the inductive ability for zero-shot items, since those items may not present in the pre-training stage of the GNN model. We demonstrate a toy example of the PKG over items in Figure 1(b), which contains three item-item relationships.

The challenges of pre-training PKG are from the following three perspectives: 1) Multi-type Relations in PKG; 2) Semantic Divergence between generic information and relations; 3) Domain Discrepancy from PKG to downstream ZSIR task.

Firstly, the multi-type relations intrinsically exist in PKG due to various item-item relationships. Pre-training a PKG encoder demands comprehensive characterization of those multiple relations, which is still under-explored. Secondly, the universal features are extracted from generic information, while the relations in PKG may reflect different semantics. For example, two items has similar *titles*, but they have no complementary relations. Ignoring the semantic divergence disables the unifying ability of graph encoder to incorporate both generic information and relation semantics. Thirdly, we use a pre-trained encoder for inferring item embeddings for ZSIR. However, because ZSIR task has distinct domain from PKG domain, though PKG yield similar embeddings for two items, users may have different preferences towards them in ZSIR task. Moreover, multiple relations may have different contributions to ZSIR task. We refer this as the domain discrepancy issue.

To resolve aforementioned problems, we propose a novel **M**ulti-task **P**roduct **K**nowledge **G**raph model for pre-training, and devise a novel paradigm to fine-tune the pre-trained model on recommendation task. MPKG is able to adapt to the downstream recommendation task and infer the universal representation of zero-shot items. To be more specific, we extract multiple single-relation PKG from the original PKG and adopt the SGCN [40] as the encoder for each single-relation PKG. Then, we endow this multi-relation graph encoder with adaptation ability to different tasks via novel Task-oriented Adaptation (ToA) layers. We also devise four pre-training tasks to optimize the graph encoder and ToA layers, which are Knowledge Reconstruction (KR), High-order Neighbor Reconstruction (HNR), universal Feature Reconstruction (FR), and Meta Relation Adaptation (MRA) tasks. Each task is associated with one type of ToA layer. Experiments are conducted on the cross-market dataset [3], which consists of 18 different markets data from Amazon. We summarize our contributions as follows:

- We propose a novel PKG pre-training and fine-tuning framework to tackle the ZSIR problem, which enhances the MPKG with inductive ability.
- We identify three type of challenges in PKG pre-training and devise four pre-training tasks. The MRA pre-training task is firstly proposed for adapting model to new downstream tasks.
- We propose a novel task-oriented adaption layer for each task, which adapts the embeddings from multi-relation graph encoder to different tasks.

2 RELATED WORK

2.1 Product Knowledge Graph

Product knowledge graphs (PKGs) are graphs whose nodes represent products (items) and whose edges represent various item-item relationships, such as complementary and substitute relationships. The item-item relationships are extracted from meta data of items or interaction data on items. Existing literature of PKG for recommendation mainly focuses on two directions, including PKG construction [11, 25, 43] and PKG utilization for recommendation [6, 34, 41, 45, 48, 49].

Regarding the PKG construction, the earliest work is Sceptre [25], which focuses on modeling product complementary and substitution relationships. PKG [43] is further introduced with three more knowledge relationships between products, including co-view and IsA. Moreover, Autoknow [11] builds the PKG from item textual knowledge and user-item interaction data. Those works suggest that item relationships described in PKG are more closely related to item semantics in recommendation [11, 43].

[1]The terms "product" and "item" are used interchangeably.

Utilizing PKG for recommendation also attracts increasing attention. RSC [48] proposed complement and substitution networks to improve rating prediction accuracy, demonstrating the effectiveness of additional item relationships. [41] developed Bayesian dual embedding framework to encode complementary item relationships for recommendation. Chorus [34] encoded item relationships into sequential recommendation with temporal kernel functions. In summary, constructing and leveraging PKGs is a promising direction for improving recommendation performance. In this work, we firstly propose to pre-train a PKG model and then fine-tune this model in recommendation task.

2.2 Graph Pre-Training

Graph neural networks (GNNs) encode rich relationships between nodes and formulates these connections as a graph. As a great amount of data can be represented as a graph, the GNN representation learning and its pre-training become an important but challenging research problem. Several classical GNN pre-training methods were developed for general graph tasks[19, 20, 24, 27, 33]. GPT-GNN [20] proposed two pre-training generation tasks, including node attributes generation and edge generation. The demonstration of GPT-GNN is conducted in downstream tasks with time and data shifts. Another representative work GCC [27] assumes that the graph structural property is transferable and universal across different networks. The authors defined the r-ego network as the positive subgraph and proposed the negative subgraph sampling into the contrastive learning. Lu,[24] proposed to bridge the gap between graph pre-training and fine-tuning with model-agnostic meta-learning strategy. It focused on subgraph learning node-level embedding learning for predicting the node connections, subgraph graph-level learning for predicting how close between the subgraph and the whole graph. Each subgraph is used as the support set for node and graph levels adapation meta-learning.

3 PRELIMINARIES

Our work is focused on strategically pre-training a graph neural network for PKG. We begin with the definition of PKG.

Definition 1. **Product Knowledge Graph (PKG).** A product knowledge graph (PKG) is denoted as $\mathcal{G} = \{\mathcal{I}, \mathcal{E}, \mathcal{R}, \mathbf{X}, \theta\}$, where $\mathcal{I}$ and $\mathcal{E}$ denote the sets of item nodes and edges, respectively. $\mathcal{R}$ is the relation type of edges, which is associated with $\mathcal{E}$ via a edge-type mapping function $\theta : \mathcal{E} \rightarrow \mathcal{R}$. $\mathbf{X} \in \mathbb{R}^{|\mathcal{I}| \times d}$ denotes the feature vector for nodes, which is extracted from item generic side information via PLMs. For each edge type $r \in \mathcal{R}$, we define its r-**PKG** as $\mathcal{G}^r = \{\mathcal{I}, \mathcal{E}^r, \mathbf{X}\}$, where $\mathcal{E}^r$ only has edges in relation r.

Note that PKG only has one node type, *i.e.* items, while having multiple edge types between items, *e.g. co-purchasing*, *co-view*, etc. To achieve knowledge-enhanced universal item representations, we pre-train a model that encodes nodes to embeddings in PKG.

Definition 2. **PKG Pre-training.** Given a PKG $\mathcal{G}$, the PKG pre-training task is to learn an encoder $\text{Enc}(\mathcal{G}) \rightarrow \mathbf{E} \in \mathbb{R}^{|\mathcal{I}| \times d}$, where each node $i \in \mathcal{I}$ is represented as an embedding $\mathbf{e}_i \in \mathbb{R}^d$.

In this work, we pre-train a graph encoder for PKG. The graph encoder preserves heterogeneous semantics of items, including both the features of items and their associated relations. The ultimate goal for pre-training a graph encoder is to tackle the ZSIR problem. Though the encoder $\text{Enc}(\cdot)$ is able to generate embeddings for all nodes in PKG $\mathcal{G}$, zero-shot items may not be present in the pre-training stage of the graph encoder. Thus, we require that the graph encoder can perform inductive on new items, defined below.

Definition 3. **Inductive Inference.** Given a PKG $\mathcal{G}$ and a pre-trained PKG encoder $\text{Enc}(\cdot)$, suppose there is a zero-shot item i^* with edges $\mathcal{E}_i^*$. The inductive inference is to encode an updated graph $\mathcal{G}^*$ which is constructed from $\mathcal{G}$ by including i^* and $\mathcal{E}_i^*$.

After inductive inference, we have embeddings for all items, including both warm items and zero-shot items. Thus, we can resolve the ZSIR task.

4 PROPOSED METHOD

In this section, we introduce the proposed framework MPKG for pre-training PKG encoder and fine-tuning the pre-trained model to the ZSIR task. The overall framework is shown in Figure (2).

4.1 Product Knowledge Graph Construction

Constructing a PKG for pre-training requires two crucial factors: (1) the universal features from item generic information; (2) item-item connections derived from either meta-data or user-item interactions. To be specific, the universal item features are task-invariant item generic features. For example, in this paper, we extract items feature embeddings, $\mathbf{X}$, from the pre-trained BERT [8] by using the concatenated description and title texts of items as input. We also analyze the effects of other PLMs [29, 31]. Item-item connections are derived from the collected feedback. Inspired by previous works [41, 43], our PKG consists of multiple item relationships, including complement, co-view, substitute, etc, which are extracted from user interaction data.

4.2 Multi-Relation Graph Encoder

In the pre-training stage, we first encode the PKG to obtain item embeddings over various relationships, which is shown in the upper left component in Fig. (2). The semantics of PKG contains multiple item-item relations. Therefore, during pre-training of the graph encoder, we simultaneously ingest both relations and node features for encoding the graph. We ensure the graph encoder has the inductive inference ability such that zero-shot item embeddings can be inferred during the evaluation stage.

Motivated by the effectiveness of existing works [15, 36, 38], we adopt a graph encoder based on the message-passing framework for each edge type. Specifically, for each edge type $r \in \mathcal{R}$, we extract the r-PKG as $\mathcal{G}^r = \{\mathcal{I}, \mathcal{E}^r, \mathbf{X}\}$. Following SGCN [40], we adopt the encoder to obtain the item embeddings with M layers of message aggregation as follows:

$$\text{Enc}(\mathcal{G}^r) \rightarrow \mathbf{E}^r = \left(\mathbf{D}^{-\frac{1}{2}}\tilde{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}}\right)^M \mathbf{X}\mathbf{W}^r, \tag{1}$$

where $\mathbf{E}^r \in \mathbb{R}^{|\mathcal{I}| \times d}$ is the embeddings of items w.r.t. relation r, $\mathbf{D}$ denotes the degree matrix of $\tilde{\mathbf{A}}$, $\tilde{\mathbf{A}}$ is the adjacency matrix with self-loop for r-PKG, and $\mathbf{W}^r \in \mathbb{R}^{d \times d}$ is the weight matrix. The advantage of this simple form and the removal of activation in each aggregation layer allows the pre-computation of high-order

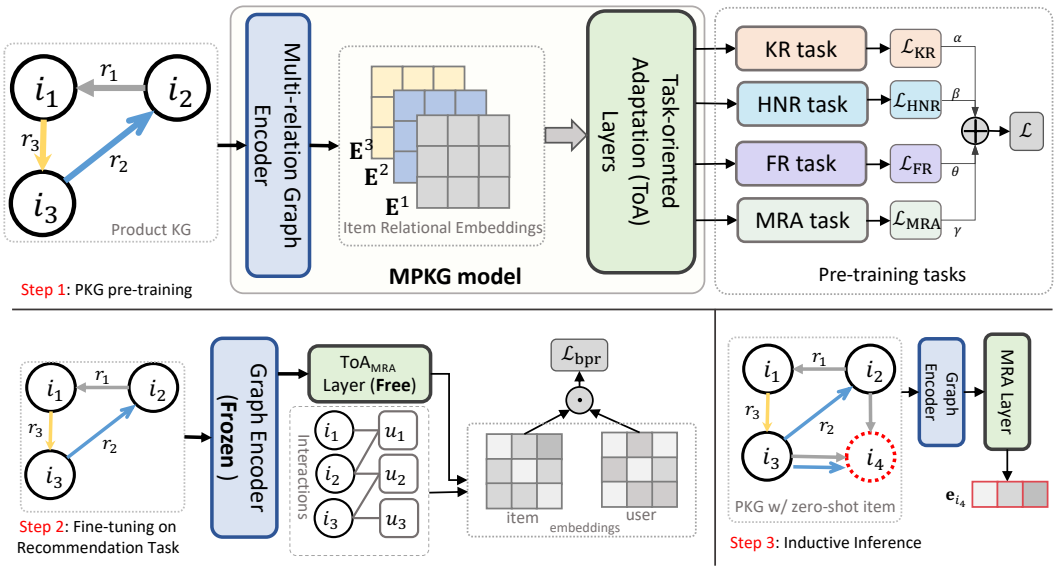


Figure 2: The framework of our proposed method. In step 1, we pre-train the MPKG model by using multiple pre-training tasks upon the PKG. Each task is associated with one type of ToA layer. Next in step 2, we fine-tune the model on the recommendation task with user-item interactions, which has frozen graph encoder and free ToA_{MRA} layer parameters. Finally, in step 3 we conduct inductive inference of the zero-shot item i_4. Best viewed in colors.

neighborhood connectivity matrix, which significantly increases the efficiency. Also, since the multi-layer message aggregation process, *i.e.*, the term $\left(\mathbf{D}^{-\frac{1}{2}}\tilde{\mathbf{A}}\mathbf{D}^{-\frac{1}{2}}\right)^M$ can be decoupled from the feature transformation step, *i.e.* the term $\mathbf{X}\mathbf{W}^r$, we can update the PKG with zero-shot items and conduct the message-passing directly on updated PKG, thus ensuring the inductive inference ability.

4.3 Task-oriented Adaptation Layer

Given $|\mathcal{R}|$ relationships, we obtain $|\mathcal{R}|$ item embedding matrices. However, our final goal is to resolve the ZSIR task, which requires the fusion of embeddings from all relations. Due to the domain discrepancy from our PKG pre-training tasks to the ZSIR task, we adapt the embeddings to different tasks such that semantics from multiple relations can be properly fused. Let $\{\mathbf{E}^r\}|_{r\in\mathcal{R}}$ denote the item embeddings for $|\mathcal{R}|$ relations. We define the fused embedding for a specific task t as:

$$\mathbf{E}_t = \text{ToA}_t(\{\mathbf{E}^r\}|_{r\in\mathcal{R}}), \tag{2}$$

where ToA_t can be arbitrary read-out functions, such as concat, mean-pooling, weighted-sum, etc. In the next sections, we discuss defining ToA layers for various pre-training tasks and fine-tuning on down-stream tasks.

4.4 Multi-task Pre-training

Our approach pre-trains a multi-relation graph encoder on four tasks for the PKG, which are presented as the step 1 in Figure (2). In this section, we introduce these pre-training tasks and define the corresponding ToA layers.

4.4.1 Knowledge Reconstruction (KR). Let (i, r, j) denote a knowledge triplet, where items i and j are connected by relation r. To preserve the original semantics from each relation, we adopt a knowledge reconstruction task with respect to each relation. To be concrete, we propose a link prediction task for each relation. In our link prediction task, the encoded item embeddings $\mathbf{E}^r$ must effectively reconstruct the item-item knowledge triplets under relation r. Therefore, in this knowledge reconstruction task for relation r, the ToA layer uses only the embedding $\mathbf{E}^r$. We calculate the knowledge reconstruction score s^r_{ij} as follows:

$$s^r_{ij} = \sigma(\mathbf{E}^r_i \cdot \mathbf{E}^r_j), \tag{3}$$

where $\sigma(\cdot)$ denotes the sigmoid activation function and $\mathbf{E}^r_i$ and $\mathbf{E}^r_j$ represent the embeddings under relation r for item i and j, respectively. Hereafter, we develop the item knowledge link reconstruction loss, $\mathcal{L}_{\text{KR}}$, as the BCE loss between positive triplet and negative triplet and sum over all relations:

$$\mathcal{L}_{\text{KR}} = \sum_{r\in\mathcal{R}} -\frac{1}{|\mathcal{E}^r|}\sum_{(i,j)\in\mathcal{E}^r}\left(\log s^r_{ij} + \log(1 - s^r_{ij_-})\right), \tag{4}$$

where $\mathcal{E}^r$ denotes all links under relation r and $(i, j_-) \notin \mathcal{E}^r$ is a negative sample to pair with the positive link.

4.4.2 High-order Neighbor Reconstruction (HNR). While the knowledge reconstruction task encourages the graph encoder to be relation-aware, due to sparsity of PKGs, it is insufficient to only consider direct neighbors. Thus, we leverage the higher-order neighbors in the PKG to fully reconstruct the semantics. Specifically, we enhance the embeddings by reconstructing the K-order neighbors, regardless of relationships, which is defined as the High-order Neighbor Reconstruction (HNR) task. This task simultaneously incorporates

semantics from all relations. Hence, we define the ToA layer for this task as the concatenation for all embeddings,

$$\mathbf{E}_{\text{HNR}} = \text{ToA}_{\text{HNR}}(\{\mathbf{E}^r\}|_{r\in\mathcal{R}}) = \text{Concat}(\{\mathbf{E}^r\}|_{r\in\mathcal{R}}), \tag{5}$$

where $\mathbf{E}_{\text{HNR}}$ and ToA_{HNR} denotes the item embeddings and ToA layer for this HNR task respectively. We first collect the K-order neighbors of each item, denoted as $\mathcal{N}_K(i)$. Then the neighbor reconstruction score a_{ij} between item i and j is defined as the soft dot-product:

$$a_{ij} = \sigma(\mathbf{E}_{\text{HNR}}(i) \cdot \mathbf{E}_{\text{HNR}}(j)), \tag{6}$$

where $\mathbf{E}_{\text{HNR}}(i)$ and $\mathbf{E}_{\text{HNR}}(j)$ represent the HNR embedding for item i and j, respectively. Finally, we optimize the task with BCE loss as follows:

$$\mathcal{L}_{HNR} = -\sum_{i\in\mathcal{I}}\sum_{j\in\mathcal{N}_K(i)} \left(\log a_{ij} + \log(1 - a_{ij\_})\right), \tag{7}$$

where j denote a K-hop neighbor of item i and j_- denotes a negative sample to pair with j such that $j_- \in \mathcal{I} \setminus \mathcal{N}_K(i)$

4.4.3 Feature Reconstruction (FR). The universal item features encode the basic item generic information and benefit the inductive inference for zero-shot items. However, since universal item features are extracted from PLMs, there is a large semantic divergence between the universal item features and output from multi-relation graph encoder. Therefore, we propose to use the feature reconstruction (FR) task to optimize the graph encoder such that semantic divergence is mitigated. Concretely, we propose to use the item embeddings from graph encoder to reconstruct the universal item features via a decoder. For this task, semantics from all relations are also harnessed. Hence, we define the ToA layer to be the same as in HNR task, *i.e.* the concatenation, as follows:

$$\mathbf{E}_{\text{FR}} = \text{ToA}_{\text{FR}}(\{\mathbf{E}^r\}|_{r\in\mathcal{R}}) = \text{Concat}(\{\mathbf{E}^r\}|_{r\in\mathcal{R}}), \tag{8}$$

where $\mathbf{E}_{\text{FR}}$ denotes the item embeddings for this FR task. Then, we input this $\mathbf{E}_{\text{FR}}$ to a decoder $\text{Dec}(\cdot)$ such that the universal feature from PLMs can be reconstructed, formulated as follows:

$$\tilde{\mathbf{X}} = \text{Dec}(\mathbf{E}_{\text{FR}}), \tag{9}$$

where $\tilde{\mathbf{X}}$ is the feature decoded from the concatenated relational embeddings. Though a wide range of decoders can tackle this FR task, we adopt one fully-connected layer as the decoder here in this paper. The reason is that a light-weight decoder is less complex to optimize and the output embeddings from graph encoder can be linearly aligned with universal features. We leave the investigation of other types of decoders as future work.

Finally, we optimize this task under the measurement of L_2 losses between orignal features and reconstructed features as follows:

$$\mathcal{L}_{\text{FR}} = \sum_{i\in\mathcal{I}} \|\mathbf{X}_i - \tilde{\mathbf{X}}_i\|_2^2, \tag{10}$$

where $\mathbf{X}_i$ and $\tilde{\mathbf{X}}_i$ are the universal and reconstructed features for item i, respectively.

4.4.4 Meta Relation Adaptation (MRA). Recall that the objective of pre-training a graph encoder is to yield embeddings for the items in the downstream ZSIR task. Nevertheless, due to the domain discrepency between PKG semantics and the ZSIR task, different item-item relations have unequal contributions. Therefore, we should devise a proper strategy to adapt relational embeddings to various tasks. Since during the pre-training stage, we have no access to downstream data, we propose a novel Meta Relation Adaptation (MRA) task. To be concrete, we treat one relation r as the target relation, and use embeddings from other relations to reconstruct the edges in r-PKG $\mathcal{G}^r$. We define this as the r-MRA task. Firstly, the ToA layer for r-MRA task is a weighted sum of all relational embeddings except the relation r embeddings, which is formulated as:

$$\mathbf{E}_{r\text{-MRA}} = \text{ToA}_{r\text{-MRA}}(\{\mathbf{E}^r\}|_{r\in\mathcal{R}}) = \sum_{r\in\mathcal{R}_{-r}} w_r\mathbf{E}_{\mathbf{r}}, \tag{11}$$

where $\mathcal{R}_{-r}$ denotes all relations but relation r, and $w_r \in \mathbb{R}$ is a scalar weights, denoting the contrition of each relation embeddings in $\mathcal{R}_{-r}$. In this paper, we use a self-excitation layer [18] to compute the weight w_r, which ingests the associated relation embeddings into two fully-connected layers and normalizes those weights w.r.t. each relation with a softmax function. The reason is self-excitation layer is easy to implement and fine-tune for new downstream tasks. We leave other methods for calculating the weights in future works. Next, we predict edges in r-PKG by a soft dot-product upon the r-MRA embedding. The prediction score b_{ij} between item i and j is formulated as follows:

$$b_{ij} = \sigma(\mathbf{E}_{r\text{-MRA}}(i) \cdot \mathbf{E}_{r\text{-MRA}}(j)), \tag{12}$$

where $\mathbf{E}_{r\text{-MRA}}(i)$ and $\mathbf{E}_{r\text{-MRA}}(j)$ represent the r-MRA embeddings for items i and i, respectively. The intuition for this meta relation adaption task is to simulate the process of adapting relation embeddings to new tasks. The r-MRA task views the edge prediction task on r relation as a new task and train the encoder to adapt the embeddings from other relation sematics to relation r. In this way, the encoder would have more generalizatio ability and endows the $\text{ToA}_{r\text{-MRA}}(\cdot)$ layer more flexibility for downstream task adaptation, thus resolving the domain discrepency problem betwen PKG semantics and ZSIR task. We will introduce how to fine-tune this layer in ZSIR task in the next section.

Next, we optimize the MRA tasks for all relations via MSE loss as follows:

$$\mathcal{L}_{\text{MRA}} = \sum_{r\in\mathcal{R}} -\frac{1}{|\mathcal{E}^r|}\sum_{(i,j)\in\mathcal{E}^r} \left(\log b_{ij} + \log(1 - b_{ij\_})\right), \tag{13}$$

where $\mathcal{E}^r$ denotes all edges under relation r and $(i, j_-) \notin \mathcal{E}^r$ is a negative sample to pair with the positive edge.

4.4.5 Final Pre-Training Loss. We present the entire training framework as a multi-task training framework. The final loss is calculated as the weighted sum of four proposed losses:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{KR}} + \beta\mathcal{L}_{\text{FR}} + \theta\mathcal{L}_{\text{HNR}} + \gamma\mathcal{L}_{\text{MRA}}, \tag{14}$$

where α, β, θ, and γ are hyper-parameters, and we choose them based on the best performance on the validation set.

4.5 Model Fine-tuning

In general, we could fine-tune the proposed model on any new tasks. We could update parameters in the graph encoder $\text{Enc}(\cdot)$ and ToA layers by defining new objective functions for new tasks. This work mainly fine-tunes the ToA_{MRA} layers for all relations as it is most relevant to the ZSIR task and more efficient to adapt without loading the entire PKG again in ZSIR. Hence, we only discuss how

to fine-tune ToA_{MRA} layers on ZSIR task in this paper. An example process is given in the step 2 of Figure (2). Any other tasks can be investigated in analogy.

Concretely, we update ToA_{MRA} layers with a recommendation objective function. Given the user-item interaction data, denoted as $\mathcal{D} = \{(u, i)|u \in \mathcal{U}, i \in \mathcal{I}\}$ where $\mathcal{U}$ is the user set, we optimize the pre-trained MPKG with only ToA_{MRA} layers as free parameters and all other parameters are fixed. During the fine-tuning stage, the item embeddings are produced similar to Eq. (11), but involving all relations in PKG, denoted as $\mathbf{E}_{\text{MRA}} = \sum_{r \in \mathcal{R}_r} w_r \mathbf{E}_\mathbf{r}$. Recall that w_r represents the contribution of each relation r and computed via self-excitation over the relational embedding $\mathbf{E}_r$. Hence, we optimize the self-excitation layers such that the contribution of each relation towards recommendation task can be characterized.

The recommendation task is to predict ranking scores between items and users. For each pair (u, i), we represent the user representation $\mathbf{e}_u$ as the mean aggregation for all interacted items, formulated as $\mathbf{e}_u = \frac{1}{|\mathcal{D}_u|} \sum_{i \in \mathcal{D}_u} \mathbf{E}_{\text{MRA}}(i)$, where $\mathcal{D}_u$ is the interacted items for user u and $\mathbf{E}_{\text{MRA}}(i)$ denotes the output embedding for item i. Then, we calculate the ranking score p_{ui} between user u and item i via the dot-product similarity as follows:

$$p_{ui} = \mathbf{e}_u \cdot \mathbf{E}_{\text{MRA}}(i). \tag{15}$$

Finally, we fine-tune the model via the BPR loss [30] as follows:

$$\mathcal{L}_{bpr} = \sum_{(u,i) \in \mathcal{D}} -\log \sigma \left(p_{ui} - p_{ui_-}\right), \tag{16}$$

where i_- is a negative item such that $(u, i_-) \notin \mathcal{D}$ for user u. After optimization, ToA_{MRA} layers are adapted to the recommendation task. Note that we could use any other functions to produce the final representation of users and items for recommendation task. This paper investigates the above methods as it is the minimal way to verify the effectiveness of the MPKG framework, no additional parameters being introduced during fine-tuning stage.

Hereafter, we utilize the fine-tuned model to conduct inductive inference for the zero-shot items, which is demonstrated in the step 3 of Figure 2. Finally, the prediction scores between users and all items are calculated as in Eq. (15.)

5 EXPERIMENTS

In this section, we demonstrate the effectiveness of our proposed universal pre-training PKG framework in several perspectives. We answer the following Research Questions (RQs) to validate the superiority:

- **RQ1:** Does MPKG generalize to downstream ZSIR tasks, especially for zero-shot items?
- **RQ2:** Does MPKG yield better universal item embeddings than other state-of-the-art models?
- **RQ3:** What are the contributions of multiple pre-training tasks?
- **RQ4:** What are the effects of MPKG variants?

5.1 Data Preparation

We conduct the experiments on the largest category *Home and Kitchen* category in Xmarket dataset[2]. The dataset consists of 18 markets, of which each has user-item reviews and item-item relationships as meta-data. We utilize the item-item relationships in meta-data as the PKG pre-training item relationships, including *alsoViewed, alsoBought, boughtTogether* as these are widely used item relationships for recommendation [45, 48, 49]. We aggregate item-item relationships pairs from all markets and construct the PKG. We list statistics of user-item interaction data of all markets in Table 1. The data statistics of the product knowledge graph are in Table 2. We concatenate the description and title texts as the universal textual information, and we extract the item universal features $\mathbf{X}$ using a pre-trained language model [10].

We rank the user-item interactions in chronological order. We use data in the earliest 80% time for training, the following 10% time for validation, and the last 10% period for testing. The items appearing in the training data are the train item set. For validation and testing items appearing in the train item set, we denote them as warm items, otherwise, we denote them as zero-shot (zs) items. To avoid the data leakage problem we delete all the cold items from PKG during training.

5.2 Evaluation Tasks

We present the effectiveness of our proposed pre-training PKG framework via two evaluation tasks, *i.e.* the *knowledge prediction* task and *zero-shot item-based recommendation* (ZSIR) task. The knowledge prediction task assesses the ability of our pre-trained GNN in recovering the semantics between items in the PKG. Specifically, the knowledge prediction task predicts the knowledge triplet links associated with items as head entities. The ZSIR task assesses the inference ability of MPKG on a downstream task.

The performance of both tasks is evaluated on all items and zero-shot items settings. For all downstream tasks, we predict top-N ranking lists from either the all item candidates, or only the test zero-shot items. We report the overall performance on both settings to demonstrate the recommendation ability of our model. The inductive inference experiments introduced in Section 3 infer the embeddings of zero-shot items with a updated PKG in test time. We report the testing performance based on the grid-searched best validation performance.

5.3 Baselines and Implementation

To validate the effectiveness of the proposed framework, we compare the model with the following two groups of related baselines: (1) Triplet-based heterogeneous graph methods, including TransE [4], TransD [22], DistMult [47], and TransH [37]; (2) Heterogeneous graph models, including GPT-GNN [20] with a generative graph model framework and HeCo [36] with the self-supervised graph learning architecture.

We implement MPKG in PyTorch and conduct the experiments with 4 V100 GPUs. We grid search important hyper-parameters in baselines and the proposed MPKG. During the pre-training stage, we can only access the knowledge triplets and we select the best pre-training MPKG based on the validation performance on validation set of knowledge triplets predictions. For all methods, we search the hidden dimension from $\{64, 128\}$, the L2 regularization weight from $\{1e-3, 1e-2, 1e-1, 5e-1\}$, the learning rate from $\{1e-3, 1e-4, 5e-3\}$, the batch size is set to be 256, the base GNN is SGC [40],

[2] https://xmrec.github.io/

Table 1: Home and Kitchen Dataset User-Item Interactions Statistics. For each market, statistics are reported in the following format: #of users/#of items/#of user-item interactions. We filter out markets with less than 1,000 users.

Brazil(br)	Japan(jp)	Mexico(mx)	Italian(it)	France(fr)	Spain(es)
1.7K/60K/10.5K	1.9K/5.6K/14K	3.5K/7K/16K	3.7K/11K/20K	5.7K/16K/38K	5.9K/10K/29K
Australia(au)	**Germany(de)**	**India(in)**	**Canada(ca)**	**United States(us)**	**United Kingdom(uk)**
13K/27.6K/121K	18K/31K/122K	22K/20K/114K	30K/38K/208K	1.7K/8K/8K	251K/66K/1.8M

Table 2: Product Knowledge Graph Statistics.

Edge Type	alsoBought	alsoViewed	boughtTogether	Total
#of items/#of edges	93,659/2,044,418	86,481/1,048,779	64,574/115,386	97,626/3,208,583

and the number of GNN layers is default at 3. For all triplet-based heterogeneous graph baselines, we search the hidden dimension and L2 regularization weight. For heterogeneous graph model GPT-GNN [20], we additionally search its attribute generation loss ratio from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$ and the queue size from $\{128, 256, 512\}$. For HeCo [36], we further search its dropout rate for features and attentions from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$.

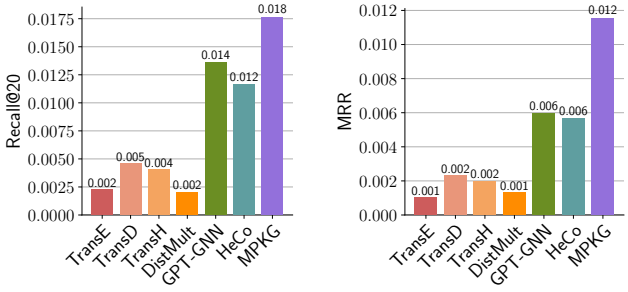
5.4 ZSIR Task Performance (RQ1)

We conduct the ZSIR evaluation in multiple markets. We report the performance on all items recommendation in Table 3 and the performance on only zero-shot items recommendation in Table 4. We only list 7 markets due to the space limitation. The first three from the left are the smallest 3 markets while the remaining 4 markets are the largest 4 markets. From both tables, we have the following observations:

- In both all items and zero-shot items recommendations, our proposed MPKG consistently achieves the best performance in all markets and all metrics. The relative improvements range from 23.08% to 83.33% in all items recommendation. For zero-shot items recommendation, the improvements are from 4.68% to 56.33%. These improvements demonstrate that the proposed MPKG framework successfully addresses the domain discrepancy between the PKG and the downstream ZSIR task in the zero-shot setting. We argue that the improvements result from the superior pre-training capability on handling multi-type item relationships and the adaptation layer to improve the generalization capability.
- The pre-training heterogeneous GNN baselines outperform the triplet-based methods. However, there is not a consistent winner among heterogeneous GNN baselines. This again demonstrates the importance of multi-type relations modeling in GNN.
- The improvements on low-resource markets are larger than the rich markets. For example, in all items recommendation, the low-resource markets have at least 36.53% relative improvements in NDCG@20 while the larger markets have at most 33.52%. This demonstrates that MPKG can benefit low-resource markets more than rich markets, indicating better generalization capability.

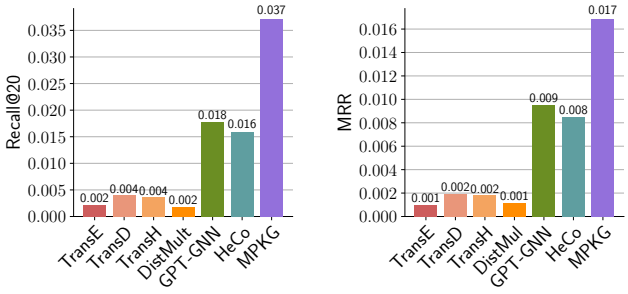
5.5 Knowledge Prediction Comparison (RQ2)

In this section, we validate the pre-training effectiveness of the proposed MPKG in learning item-item relationships predictions, in both warm items (seen items in training portion) and zero-shot items (unseen items). The knowledge prediction task validates the capability of pre-training with product knowledge graph information over existing methods.



(a) Comparison on Recall@20 **(b) Comparison on MRR**

Figure 3: Knowledge Prediction on Warm (Seen) Items.



(a) Comparison on Recall@20 **(b) Comparison on MRR**

Figure 4: Knowledge Prediction on Zero-Shot Items.

5.5.1 Warm Items Comparison. The knowledge prediction performance of product knowledge graph triplets on warm items are shown in Figure 3. We report the Recall@20 and MRR in Figure 3a and Figure 3b, respectively. We obtain the following observations from these comparisons:

- The proposed MPKG achieves the best warm item knowledge prediction performance in both metrics, with relative improvements from 28% to 100% in all metrics. We attribute this superior

Table 3: ZSIR Task Results on All Items Comparison. The best models are bolded and the second-best are underlined.

	NDCG@20							MRR						
Model	br	mx	es	in	ca	uk	us	br	mx	es	in	ca	uk	us
TransE	0.0357	0.0250	0.0196	0.0153	0.0149	0.0126	0.0166	0.0298	0.0186	0.0163	0.0121	0.0100	0.0080	0.0102
TransD	0.0345	0.0251	0.0189	0.0157	0.0148	0.0127	0.0159	0.0275	0.0199	0.0153	0.0117	0.0103	0.0080	0.0103
TransH	0.0393	0.0289	0.0208	0.0164	0.0160	0.0129	0.0197	0.0317	0.0229	0.0187	0.0143	0.0107	0.0083	0.0126
DistMult	0.0344	0.0233	0.0202	0.0159	0.0168	0.0125	0.0155	0.0253	0.0207	0.0195	0.0129	0.0111	0.0075	0.0088
GPT-GNN	0.0397	0.0304	0.0222	0.0182	0.0174	0.0140	0.0210	0.0329	0.0242	0.0251	0.0169	0.0113	0.0096	0.0158
HeCo	0.0384	0.0285	0.0222	0.0175	0.0175	0.0132	0.0217	0.0323	0.0246	0.0234	0.0156	0.0125	0.0090	0.0157
MPKG	**0.0542**	**0.0433**	**0.0407**	**0.0243**	**0.0204**	**0.0162**	**0.0257**	**0.0413**	**0.0332**	**0.0346**	**0.0208**	**0.0166**	**0.0139**	**0.0208**
Impro.	36.52%	42.43%	83.33%	33.52%	16.57%	15.71%	18.43%	25.53%	34.96%	37.85%	23.08%	32.8%	44.79%	31.65%

Table 4: ZSIR Task Results on Zero-Shot Items Comparison. The second-best and the best models are underlined and bolded.

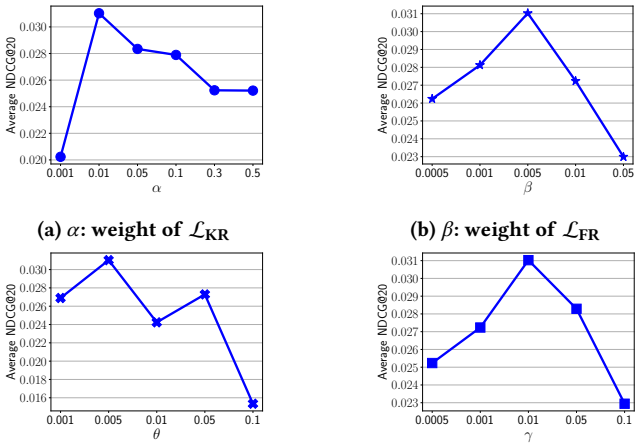
	NDCG@20							MRR						
Model	br	mx	es	in	ca	uk	us	br	mx	es	in	ca	uk	us
TransE	0.0467	0.0365	0.0312	0.0206	0.0162	0.0130	0.0215	0.0333	0.0245	0.0216	0.0150	0.0115	0.0096	0.0143
TransD	0.0454	0.0332	0.0304	0.0213	0.0175	0.0132	0.0205	0.0313	0.0227	0.0228	0.0130	0.0121	0.0088	0.0144
TransH	0.0501	0.0404	0.0312	0.0240	0.0197	0.0152	0.0252	0.0364	0.0280	0.0202	0.0167	0.0114	0.0095	0.0181
DistMult	0.0452	0.0332	0.0309	0.0202	0.0174	0.0129	0.0198	0.0290	0.0241	0.0227	0.0151	0.0125	0.0086	0.0126
GPT-GNN	0.0528	0.0426	0.0319	0.0236	0.0194	0.0147	0.0342	0.0404	0.0302	0.0229	0.0178	0.0124	0.0109	0.0240
HeCo	0.0508	0.0403	0.0326	0.0240	0.0210	0.0160	0.0324	0.0401	0.0288	0.0219	0.0180	0.0117	0.0101	0.0245
MPKG	**0.0601**	**0.0481**	**0.0452**	**0.0270**	**0.0227**	**0.0181**	**0.0358**	**0.0431**	**0.0348**	**0.0358**	**0.0213**	**0.0168**	**0.0129**	**0.0262**
Impro.	13.83%	12.91%	41.69%	12.5%	17.01%	23.13%	4.68%	6.68%	15.23%	56.33%	18.33%	35.48%	18.35%	6.94%

capability to the design of several proposed pre-training tasks as it mitigates the semantic divergence between generic information and item multi-relations.

- Among compared baselines, we observe that pre-training methods based on heterogeneous GNN (GPT-GNN, HeCo, and our MPKG) achieve better performances than triple-based methods. The heterogeneous GNN methods outperform triplet-based methods due to the stronger modeling capability of multi-relations in PKG while triplet-based methods only model direct connections and item features.

5.5.2 Zero-Shot Items Comparison. We further conduct the knowledge prediction task on zero-shot items. The zero-shot item embedding inference is corresponding to the inductive inference as in the step 3 in Figure 2 but without the fine-tuning step. The performance is shown in Figure 4. We also report the Recall@20 and MRR in Fig. (4a) and Fig. (4b), respectively. Zero-shot items evaluation verifies the induction capability of models and demonstrates the extent to which item embeddings generation can extend to zero-shot items. From the comparison, we have several observations:

- MPKG still achieves the best zero-shot item knowledge prediction performances in all metrics, with improvements from 88.9% to 105.6% over the best baseline model. The superiority in knowledge prediction performances demonstrates the effectiveness of MPKG in generalizing to zero-shot items.
- Among the two categories of baselines approaches, pre-training methods based on heterogeneous GNN still achieve more satisfactory item embeddings learning than triplet-based methods. It further demonstrates the necessity of GNN in generalizing item embeddings learning.



(a) α: weight of $\mathcal{L}_{\text{KR}}$ (b) β: weight of $\mathcal{L}_{\text{FR}}$

(c) θ: weight of $\mathcal{L}_{\text{HNR}}$ (d) γ: weight of $\mathcal{L}_{\text{MRA}}$

Figure 5: Item-based recommendation performance sensitivity of each component in Eq. (14).

Table 5: Effects of Pre-training Tasks.

	Knowledge Pred.		ZSIR	
Variant	MRR	Recall@20	MRR	NDCG@20
MPKG	0.0142	0.0255	0.0253	0.0310
w/o KR	0.0041	0.0097	0.0124	0.0152
w/o FR	0.0122	0.0224	0.0216	0.0242
w/o HNR	0.0135	0.0238	0.0234	0.0269
w/o MRA	0.0120	0.0217	0.0200	0.0245

Table 6: Effects of Graph Encoder Variants.

	Knowledge Pred.		ZSIR	
GNNs	MRR	Recall@20	MRR	NDCG@20
GCN-base	0.0108	0.0231	0.0199	0.0257
GAT-base	0.0117	0.0235	0.0214	0.0286
SGC-base	0.0122	0.0241	0.0223	0.0295

Table 7: Effects of PLM Variants.

	Knowledge Pred.		ZSIR	
PLMs	MRR	Recall@20	MRR	NDCG@20
Distill-Bert	0.0138	0.0238	0.0249	0.0298
Bert	0.0142	0.0255	0.0253	0.0310
Sentence-Bert	0.0145	0.0249	0.0250	0.0304

5.6 Pre-training Tasks Study (RQ3)

5.6.1 Impacts of Individual Task. We first show the effectiveness of each pre-training tasks as in Eq. (14) by removing one from the final loss. Recall that we have four pre-training tasks, Knowledeg Reconstruction (KR), High-order Neighbor Reconstruction (HNR), Feature Reconstruction (FR) and Meta Relation Adaptation (MRA). Our ablation study includes the performance on both knowledge prediction and ZSIR tasks after we remove each pre-training tasks, shown in Table 5. The ZSIR performance is reported as in the average of all markets. We observe that performance degrades when we remove each task individually. This demonstrates that all pre-training taskes are necessary for the satisfactory universal item embeddings learning.

We also visualize the sensitivity of hyper-parameters for each task loss as in the final loss Eq. (14), shown in Figure 5. The best weights are not the same for all components because the loss scales are varying for each loss component. We also observe that the performance drops more significantly for the k-hop neighbors reconstruction loss weight θ.

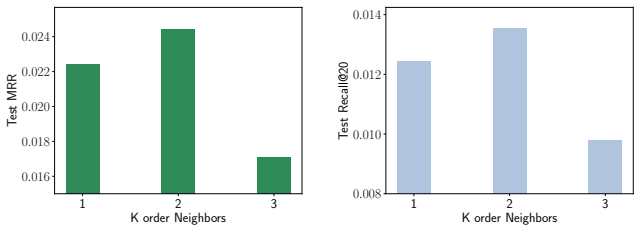
5.6.2 Impacts of K in HNR task. In this section, we investigate the effect of choosing different Ks in the k-hop neighbors reconstruction loss $\mathcal{L}_{HNR}$. We choose K from $\{1, 2, 3\}$ and test it on both tasks. The knowledge prediction task results are shown in Figure 6, and the item-based recommendation task results are shown in Figure 7. From both Figure 6 and Figure 7, we can see that when $K = 2$, the best performance is achieved. When we include high-order neighbors if $K = 3$, the performance drops significantly. The reason is that high-order neighbors might introduce more irrelevant noises.

5.7 Effects of Base Model Variants (RQ4)

We study the sensitivity of choosing different base models, including the PLM and graph encoder. Our proposed framework can adopt arbitrary different GNN encoder and PLM. The performance sensitivity of different graph encoder variants is shown in Table 6. For different textual language models that generate universal textual features, we show the results in Table 7.

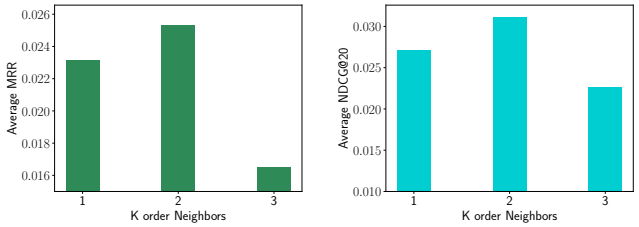
Our proposed MPKG adopts the efficient SGC as the base model. From Table 6, we observe that the GCN [39] achieves the worst performance. The second best GNN encoder is GAT [32]. The reason is that SGC is easier to learn and generalize to new data.

We also investigated the effects of using different textual language models to generate universal textual features, including Distill-Bert [31], Bert (adopted in this work) [8], and Sentence-Bert [29]. In Table 7, we can see that Sentence-Bert achieves marginally better performances than BERT. Moreover, Distill-Bert cannot achieve on-par performance, even though its efficiency is significantly better than other two.



(a) MRR **(b) Recall@20**

Figure 6: Performance Sensitivity of k-hop Neighbors on the Knowledge Prediction Task.



(a) MRR **(b) NDCG@20**

Figure 7: Performance Sensitivity of k-hop Neighbors on the ZSIR task.

6 CONCLUSION

In this work, we investigate the challenging problem of pre-training product knowledge graph to infer universal item representations for zero-shot item-based recommendation. We propose four pre-training tasks that comprehensively characterize PKG semantics and improve the adaptation ability of the model to new tasks, including knowledge reconstruction, feature reconstruction, high-order neighbors reconstruction, and the meta relation adaptation tasks. We also discuss how to leverage the recommendation task to fine-tune the novel task-oriented adaptation layers such that semantics in PKG can be adapted to new tasks. Though in this paper, we only discuss how to fine-tune the model for ZSIR task, our framework is a general pre-training paradiagm for PKG and adaptable to any other new tasks.

ACKNOWLEDGMENTS

This work is supported in part by NSF under grants III-1763325, III-1909323, III-2106758, and SaTC-1930941.

REFERENCES

[1] Yunsheng Bai, Hao Ding, Song Bian, Ting Chen, Yizhou Sun, and Wei Wang. 2019. Simgnn: A neural network approach to fast graph similarity computation. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 384–392.

[2] Yunsheng Bai, Hao Ding, Ken Gu, Yizhou Sun, and Wei Wang. 2020. Learning-based efficient graph similarity computation via multi-scale convolutional set matching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 3219–3226.

[3] Hamed Bonab, Mohammad Aliannejadi, Ali Vardasbi, Evangelos Kanoulas, and James Allan. 2021. Cross-Market Product Recommendation. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. ACM.

[4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).

[5] Yuwei Cao, Liangwei Yang, Chen Wang, Zhiwei Liu, Hao Peng, Chenyu You, and Philip S. Yu. 2023. Multi-task Item-attribute Graph Pre-training for Strict Cold-start Item Recommendation. arXiv:2306.14462 [cs.IR]

[6] Tong Chen, Hongzhi Yin, Guanhua Ye, Zi Huang, Yang Wang, and Meng Wang. 2020. Try this instead: Personalized and interpretable substitute recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 891–900.

[7] Xu Chen, Hanxiong Chen, Hongteng Xu, Yongfeng Zhang, Yixin Cao, Zheng Qin, and Hongyuan Zha. 2019. Personalized fashion recommendation with visual explanations based on multimodal attention network: Towards visually explainable recommendation. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 765–774.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[9] Qiming Diao, Minghui Qiu, Chao-Yuan Wu, Alexander J Smola, Jing Jiang, and Chong Wang. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 193–202.

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

[11] Xin Luna Dong, Xiang He, Andrey Kan, Xian Li, Yan Liang, Jun Ma, Yifan Ethan Xu, Chenwei Zhang, Tong Zhao, Gabriel Blanco Saldana, et al. 2020. Autoknow: Self-driving knowledge collection for products of thousands of types. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2724–2734.

[12] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM conference on recommender systems*. 101–109.

[13] Shijie Geng, Shuchang Liu, Zuohui Fu, Yingqiang Ge, and Yongfeng Zhang. 2022. Recommendation as Language Processing (RLP): A Unified Pretrain, Personalized Prompt & Predict Paradigm (P5). *arXiv preprint arXiv:2203.13366* (2022).

[14] Bowen Hao, Jing Zhang, Hongzhi Yin, Cuiping Li, and Hong Chen. 2021. Pre-training graph neural networks for cold-start users and items representation. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*. 265–273.

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

[16] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. 2018. Nais: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering* 30, 12, 2354–2366.

[17] Yupeng Hou, Shanlei Mu, Wayne Xin Zhao, Yaliang Li, Bolin Ding, and Ji-Rong Wen. 2022. Towards Universal Sequence Representation Learning for Recommender Systems. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 585–593.

[18] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.

[19] Weihua Hu, Bowen Liu, Joseph Gomes, Marinka Zitnik, Percy Liang, Vijay Pande, and Jure Leskovec. 2019. Strategies for pre-training graph neural networks. *arXiv preprint arXiv:1905.12265* (2019).

[20] Ziniu Hu, Yuxiao Dong, Kuansan Wang, Kai-Wei Chang, and Yizhou Sun. 2020. Gpt-gnn: Generative pre-training of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1857–1867.

[21] Zhaoxin Huan, Gongduo Zhang, Xiaolu Zhang, Jun Zhou, Qintong Wu, Lihong Gu, Jinjie Gu, Yong He, Yue Zhu, and Linjian Mo. 2022. An Industrial Framework for Cold-Start Recommendation in Zero-Shot Scenarios. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 3403–3407.

[22] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*. 687–696.

[23] Jingjing Li, Mengmeng Jing, Ke Lu, Lei Zhu, Yang Yang, and Zi Huang. 2019. From zero-shot learning to cold-start recommendation. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 4189–4196.

[24] 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.

[25] Julian McAuley, Rahul Pandey, and Jure Leskovec. 2015. Inferring networks of substitutable and complementary products. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 785–794.

[26] Xia Ning and George Karypis. 2012. Sparse linear methods with side information for top-n recommendations. In *Proceedings of the sixth ACM conference on Recommender systems*. 155–162.

[27] 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 *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1150–1160.

[28] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research* 21, 1 (2020), 5485–5551.

[29] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

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

[31] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108* (2019).

[32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*. https://openreview.net/forum?id=rJXMpikCZ

[33] Chen Wang, Yueqing Liang, Zhiwei Liu, Tao Zhang, and S Yu Philip. 2021. Pre-training graph neural network for cross domain recommendation. In *2021 IEEE Third International Conference on Cognitive Machine Intelligence (CogMI)*. IEEE, 140–145.

[34] Chenyang Wang, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Make it a chorus: knowledge-and time-aware item modeling for sequential recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 109–118.

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

[36] Xiao Wang, Nian Liu, Hui Han, and Chuan Shi. 2021. Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 1726–1736.

[37] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 28.

[38] Wei Wei, Chao Huang, Lianghao Xia, Yong Xu, Jiashu Zhao, and Dawei Yin. 2022. Contrastive meta learning with behavior multiplicity for recommendation. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1120–1128.

[39] Max Welling and Thomas N Kipf. 2016. Semi-supervised classification with graph convolutional networks. In *J. International Conference on Learning Representations (ICLR 2017)*.

[40] Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019. Simplifying graph convolutional networks. In *International conference on machine learning*. PMLR, 6861–6871.

[41] Da Xu, Chuanwei Ruan, Jason Cho, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Knowledge-aware complementary product representation learning. In *Proceedings of the 13th International Conference on Web Search and Data Mining*. 681–689.

[42] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962* (2020).

[43] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Product knowledge graph embedding for e-commerce. In *Proceedings of the 13th international conference on web search and data mining*. 672–680.

[44] Feng Xue, Xiangnan He, Xiang Wang, Jiandong Xu, Kai Liu, and Richang Hong. 2019. Deep item-based collaborative filtering for top-n recommendation. *ACM Transactions on Information Systems (TOIS)* 37, 3, 1–25.

[45] An Yan, Chaosheng Dong, Yan Gao, Jinmiao Fu, Tong Zhao, Yi Sun, and Julian McAuley. 2022. Personalized complementary product recommendation. In

Companion Proceedings of the Web Conference 2022. 146–151.

[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. (2021).

[47] Zhao Zhang, Fuzhen Zhuang, Meng Qu, Fen Lin, and Qing He. 2018. Knowledge graph embedding with hierarchical relation structure. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 3198–3207.

[48] Tong Zhao, Julian McAuley, Mengya Li, and Irwin King. 2017. Improving recommendation accuracy using networks of substitutable and complementary products. In *2017 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 3649–3655.

[49] Zhiheng Zhou, Tao Wang, Linfang Hou, Xinyuan Zhou, Mian Ma, and Zhuoye Ding. 2022. Decoupled Hyperbolic Graph Attention Network for Modeling Substitutable and Complementary Item Relationships. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2763–2772.