Jointly Non-Sampling Learning for Knowledge Graph Enhanced Recommendation

Chong Chen, Min Zhang*, Weizhi Ma, Yiqun Liu, and Shaoping Ma
Department of Computer Science and Technology, Institute for Artificial Intelligence,
Beijing National Research Center for Information Science and Technology,
Tsinghua University, Beijing 100084, China
cc17@mails.tsinghua.edu.cn,z-m@tsinghua.edu.cn

ABSTRACT

Knowledge graph (KG) contains well-structured external information and has shown to be effective for high-quality recommendation. However, existing KG enhanced recommendation methods have largely focused on exploring advanced neural network architectures to better investigate the structural information of KG. While for model learning, these methods mainly rely on Negative Sampling (NS) to optimize the models for both KG embedding task and recommendation task. Since NS is not robust (e.g., sampling a small fraction of negative instances may lose lots of useful information), it is reasonable to argue that these methods are insufficient to capture collaborative information among users, items, and entities.

In this paper, we propose a novel Jointly Non-Sampling learning model for Knowledge graph enhanced Recommendation (JNSKR). Specifically, we first design a new efficient NS optimization algorithm for knowledge graph embedding learning. The subgraphs are then encoded by the proposed attentive neural network to better characterize user preference over items. Through novel designs of memorization strategies and joint learning framework, JNSKR not only models the fine-grained connections among users, items, and entities, but also efficiently learns model parameters from the whole training data (including all non-observed data) with a rather low time complexity. Experimental results on two public benchmarks show that JNSKR significantly outperforms the state-of-the-art methods like RippleNet and KGAT. Remarkably, JNSKR also shows significant advantages in training efficiency (about 20 times faster than KGAT), which makes it more applicable to real-world large-scale systems.

CCS CONCEPTS

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

KEYWORDS

Recommender Systems, Non-sampling Learning, Knowledge Graph, Efficient, Implicit Feedback

SIGIR '20, July 25–30, 2020, Virtual Event, China

ACM ISBN 978-1-4503-8016-4/20/07…$15.00
https://doi.org/10.1145/3397271.3401040

ACM Reference Format:
Chong Chen, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Jointly Non-Sampling Learning for Knowledge Graph Enhanced Recommendation. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20), July 25–30, 2020, Virtual Event, China.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3397271.3401040

1 INTRODUCTION

With the vigorous development of the Internet, recommender systems have been widely deployed in Web applications to address the information overload issue [4, 26]. Among the various recommendation methods, Collaborative Filtering (CF) [16, 18, 19, 25] gains significant attentions from researchers due to its elegant theory and good performance. However, conventional CF methods suffer from the inability of modeling side information [5, 38] such as user demographics, item attributes, and contexts, thus perform poorly in sparse situations where users and items have few interactions.

To provide more accurate recommendations, it is a trending topic to go beyond modeling user-item interactions and take side information into account [5, 13, 22, 38, 42, 48]. As shown in Figure 1, in real-world applications, there typically exist multiple relations (e.g., Categorization) between items and information values (e.g., Fast-food), and they are also particularly helpful to infer user preference. To consider both the relation type and information value, several recent efforts have attempted to leverage the graph of item side information, aka. Knowledge Graph (KG) [37] to construct recommendation models [3, 17, 34, 38, 40, 47]. The general assumption is that the item from recommender system can be linked to an entity in a knowledge graph, and the knowledge graph can provide extra information to generate more accurate item embeddings for recommendation.

However, it is challenging to effectively integrate KG into recommender systems. For both KG and implicit recommendation data (e.g., browsing histories, click logs), the true facts are rather limited, and non-observed instances, which are taken as negative examples in model learning, are of a much larger scale [16, 37]. To increase computational efficiency, existing methods mainly rely on negative sampling [25] for optimization. However, sampling a fraction of non-observed data as negative for training may ignore other useful examples, or lead to insufficient training of them [7, 9, 15, 43]. Essentially, sampling is biased, making it difficult to converge to the optimal ranking performance regardless of how many update steps have been taken. Moreover, KG enhanced methods usually need to optimize the loss function for both KG embedding task and

*Corresponding author

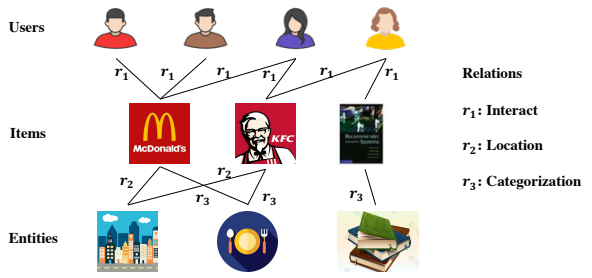

Figure 1: Illustration of the knowledge graph enhanced recommendation task, where the relation type and information value are both considered to construct the recommendation model. Item acts as the bridge to link user-item interactions and knowledge graph.

recommendation task. This produces a much larger randomness in total. As such, it is more difficult for sampling-based methods to achieve optimal performance.

Although several recent works have studied KG enhanced recommendation, they either focus on exploring advanced neural network architectures (e.g., Attention [42], Recurrent Neural Network (RNN) [30, 39], and Graph Neural Network (GNN) [38]) to regularize the model learning, or infer user preference by utilizing handcrafted meta-paths over KG [45]. Despite their success, these methods can not sufficiently express the complex relations among users, items, and entities due to the inherent weakness of negative sampling learning strategy.

Motivated by the above observations, we propose to apply non-sampling learning strategy for KG enhanced recommendation. In contrast to sampling, non-sampling strategy computes the gradient over the whole data (including all non-observed data). As such, it can easily converge to a better optimum in a more stable way [6–8, 15, 16, 43]. The difficulty in applying non-sampling strategy lies in the expensive computational cost. Although some studies have been made to explore efficient non-sampling CF methods for recommendation task [7, 15, 19, 46], they only focus on optimizing user-item relationships. Extending existing method to learn knowledge graph embedding, which consists of entity-relation-entity triplets, is not a trivial task.

In this paper, we design a novel Jointly Non-Sampling learning model for Knowledge graph enhanced Recommendation (JNSKR). To cope with the efficiency challenges caused by non-sampling strategy, we first design a new efficient optimization method to learn entity embeddings from KG. Motivated by the recent progress of representation learning [31, 38, 42], we then aggregate an item's surrounding entities with attention mechanism to learn more accurate item representation. Lastly, the two tasks (KG embedding and recommendation) are associated with a joint learning framework to simultaneously model the fine-grained connections among users, items, and entities. Our JNSKR is conceptually advantageous to existing KG enhanced recommendation methods in: 1) effective and stable non-sampling learning due to the consideration of all samples in each parameter update, and 2) much faster training process with the new proposed efficient optimization algorithm. To evaluate the recommendation performance and training efficiency of our model, we apply JNSKR on two public benchmarks with extensive experiments. The results indicate that our model significantly outperforms the state-of-the-art methods like RippleNet and KGAT, while maintaining a much simpler structure and fewer model parameters. Moreover, JNSKR shows significant advantages in training efficiency, which makes it more practical in real E-commerce scenarios.

The contributions of this work are summarized as follows:

(1) We highlight the importance of building KG enhanced recommendation models without negative sampling, and derive an efficient optimization method to learn from the whole knowledge graph with a controllable time complexity.
(2) We propose a novel end-to-end model JNSKR, which creatively addresses the KG enhanced recommendation task from the basic but important perspective of model learning. To the best of our knowledge, this is the first non-sampling learning method for KG enhanced recommendation.
(3) Extensive experiments on two public benchmarks show that JNSKR consistently and significantly outperforms the state-of-the-art models in terms of both recommendation performance and training efficiency. The source code of JNSKR and datasets used in the paper have been made available[1].

2 RELATED WORK

2.1 Knowledge Graph enhanced Recommendation

Incorporating a knowledge graph as side information has proven to be helpful for improving the performance of recommender systems. Some studies leverage the connections of entities in KG for embedding learning. For instance, Zhang et al. [47] adopt TransR [21] to learn item embeddings with the involvement of KG. Cao et al. [3] and Ai et al. [1] propose to jointly learn the models of recommendation and KG to achieve better recommendation accuracy. Wang et al. [35] propose a multi-task feature learning approach for knowledge graph enhanced recommendation. Another line of research proposes to perform propagation over the whole KG to assist in recommendation. Specifically, RippleNet [34] extends the user's interests along KG links to discover her potential interests. KPRN [39] automatically extracts paths connecting user-item pairs, and then models these paths via Recurrent Neural Network (RNN) for user preference modeling. KGCN [36] studies the utilization of Graph Convolutional Networks (GCN) for computing embeddings of items via propagation among their neighbors in KG. More recently, KGAT [38] recursively performs propagation over KG via Graph Neural Networks (GNN) and attention mechanism to refine entity embeddings.

Through the literature review above, it can be found that existing methods have largely focused on leveraging advanced neural network architectures to incorporate KG information. While for model learning, these works typically rely on the non-robust negative sampling strategy. Although significant improvements have been achieved, the performance of these methods can still be limited by the inherent weakness of negative sampling. In existing methods,

[1] https://github.com/chenchongthu/JNSKR

there is a lack of in-depth exploration of the basic but very important learning strategy, which is the main concern of our JNSKR model.

2.2 Non-sampling Learning for Top-K Recommendation

For implicit data, the observed interactions are rather limited, and non-observed examples are of a much larger scale. To learn from such a sparse data, there are generally two optimization strategies: 1) negative sampling strategy [5, 14, 25] and 2) non-sampling (whole-data based) strategy [7, 15, 16]. The first strategy samples a fraction of negative instances from non-observed entries, while the second one sees all the non-observed data as negative. In previous work (especially neural recommendation studies), negative sampling is widely adopted for efficient training. However, some recent studies have shown that sampling would inevitably limit the recommendation performance as it can ignore some important examples, or lead to insufficient training of them [7, 15, 43, 46]. In contrast, non-sampling strategy leverages the whole data with a potentially better coverage, but inefficiency can be an issue [7]. Some efforts have been devoted to resolving the inefficiency issue of non-sampling learning. For instance, Pilaszy et al. [24] describe an approximate solution of Alternating Least Squares (ALS). He et al. [15] propose an efficient ALS with non-uniform missing data. Some researchers [43, 46] study fast Batch Gradient Descent (BGD) methods. Recently, Chen et al. [7, 8] derive a flexible non-sampling loss for neural recommendation models, which achieves both effective and efficient performance.

Despite the success of existing non-sampling studies, they mainly focus on CF methods that only consider the two-element relationship between users and items. It is non-trivial to directly apply these methods for learning KG enhanced recommendation which consists of entity-relation-entity triplets. To the best of our knowledge, this is the first work to study efficient non-sampling method for KG enhanced recommendation.

3 PRELIMINARIES

We first introduce the key notations and problem formulation, and then provide an introduction to the efficient non-sampling collaborative filtering methods.

3.1 Notations and Problem Formulation

Table 1 depicts the notations and key concepts. We denote the user and item sets as $\mathbf{U}$ and $\mathbf{V}$, respectively. The user-item interaction matrix is denoted as $\mathbf{Y} = [y_{uv}] \in \{0, 1\}$, indicating whether u has an interaction with item v. In addition to the user-item matrix, we have knowledge graph information for items (e.g., item attributes and external knowledge), which is defined as an undirected graph $\mathbf{G} = (\mathbf{E}, \mathbf{R})$. Formally, it is presented as $\{(h, r, t)|h, t \in \mathbf{E}, r \in \mathbf{R}\}$, where each triplet describes that there is a relationship r between entity h and entity t. Given a target user u, the KG enhanced recommendation task is to recommend a list of items that u may be interested in, which is formally defined as:

Input: Users $\mathbf{U}$, items $\mathbf{V}$, user-item interactions $\mathbf{Y}$, and knowledge graph $\mathbf{G}$.

Table 1: Summary of symbols and notations.

Symbol	Description
$\mathbf{U}, \mathbf{V}$	Set of users and items, respectively
$\mathbf{E}, \mathbf{R}$	Set of entities and relations, respectively
$\mathbf{B}$	Batch of items
$\mathbf{Y}$	User-item interactions
$\mathcal{Y}$	Set of user-item pairs whose values are non-zero
$\mathbf{G}$	Knowledge graph
$\mathcal{G}$	Set of entity-relation-entity triplets whose values are non-zero
c_{uv}	Weight of entry y_{uv}
w_{hrt}	Weight of entry g_{hrt}
$\mathbf{p}_u, \mathbf{q}_v$	Latent vectors of user u and item v, respectively
$\mathbf{e}_h, \mathbf{e}_t$	Latent vectors of entities h and t, respectively
$\mathbf{r}_k$	Latent vectors of relation k
d	Latent factor number

Output: A ranked item list based on the probability $\hat{y}_{uv}$ that user u would interact with item v (from high to low).

3.2 Efficient Non-sampling Collaborative Filtering

Recently, some studies have realized that the non-sampling strategy is much helpful for achieving optimal recommendation performance [7, 15, 16, 46]. We make a brief introduction to efficient non-sampling collaborative filtering methods, which is designed for learning user preferences over items. For implicit data, a commonly used non-sampling loss is to minimize the differences between user feedback y_{uv} and predicted result $\hat{y}_{uv}$ [16]:

$$\mathcal{L}(\Theta) = \sum_{u \in \mathbf{U}} \sum_{v \in \mathbf{V}} c_{uv} (y_{uv} - \hat{y}_{uv})^2 \tag{1}$$

where c_{uv} denotes the weight of entry y_{uv}. In implicit feedback learning, missing entries are usually assigned a zero y_{uv} value but non-zero c_{uv} weight.

The time complexity of computing this loss is $O(|\mathbf{U}||\mathbf{V}|d)$, which is generally computationally prohibitive as $|\mathbf{U}||\mathbf{V}|$ can easily reach billion level or even higher in real life. To address the inefficiency issue of non-sampling learning, several methods have been proposed [7, 15, 43, 46]. Specifically, Chen et al. [7, 8] derive an efficient loss for generalized Matrix Factorization (MF), and prove that for a generalized matrix factorization framework whose prediction function is Eq.(2), the gradient of loss Eq.(1) is exactly equal to that of Eq.(3) if the instance weight c_{uv} is simplified to c_v.

$$\hat{y}_{uv} = \mathbf{h}^T (\mathbf{p}_u \odot \mathbf{q}_v) \tag{2}$$

$$\begin{aligned} \tilde{\mathcal{L}}(\Theta) = & \sum_{u \in \mathbf{U}} \sum_{v \in \mathbf{V}^+} \left((c_v^+ - c_v^-) \hat{y}_{uv}^2 - 2 c_v^+ \hat{y}_{uv} \right) \\ & + \sum_{i=1}^{d} \sum_{j=1}^{d} \left((h_i h_j) \left(\sum_{u \in \mathbf{U}} p_{u,i} p_{u,j} \right) \left(\sum_{v \in \mathbf{V}} c_v^- q_{v,i} q_{v,j} \right) \right) \end{aligned} \tag{3}$$

where $\mathbf{p}_u \in \mathbb{R}^d$ and $\mathbf{q}_v \in \mathbb{R}^d$ are latent vectors of user u and item v, $\mathbf{h} \in \mathbb{R}^d$ is the prediction vector, $\odot$ denotes the element-wise product of vectors.

The complexity of Eq.(3) is $O((|\mathbf{U}| + |\mathbf{V}|)d^2 + |\mathcal{Y}|d)$ while that of Eq.(1) is $O(|\mathbf{U}||\mathbf{V}|d)$. Since $|\mathcal{Y}|$ is the number of positive user-item interactions and $|\mathcal{Y}| \ll |\mathbf{U}||\mathbf{V}|$ in practice, the complexity is reduced by several magnitudes. The proof can be made by reformulating the expensive loss over all negative instances using a partition and a decouple operation, which largely follows from that in [7, 8] with little variations. To avoid repetition, it is omitted here.

Considering that existing KG enhanced recommendation works have largely ignored the study on model learning, we believe it is of critical importance to develop a method that can learn fine-grained connectivities among users, items and entities in an efficient and effective manner. To this end, we take inspiration from the recent developments of efficient CF methods, and propose a novel model JNSKR, which is, to the best of our knowledge, the first non-sampling learning method for KG enhanced recommendation.

4 METHODOLOGY

This section presents our proposed JNSKR model. The overall model architecture is described in Figure 2. From the figure, we first make a simple high-level overview of our model:

(1) The goal of JNSKR is to make accurate recommendations with the help of item knowledge graph. In particular, the recommendation and KG parts are jointly optimized through a non-sampling learning strategy, which is more effective and stable due to the consideration of all entries in each parameter update.
(2) The input of JNSKR contains user behaviors and item knowledge, which are firstly converted to dense vector representations through embeddings. Item acts as a bridge to connect the joint learning process. The output $\hat{y}_{uv}$ is a predicted score indicating user u's preference for item v.
(3) The structure of JNSKR consists of three main components: 1) KG embedding part, which learns structural KG information through the proposed efficient non-sampling method; 2) attentive user-item preference modeling part, which infers the user-item preference score with an attention mechanism; and 3) joint learning part that integrates the above two parts in an end-to-end fashion.

4.1 Efficient Non-sampling Knowledge Graph Embedding

Knowledge graph embedding is an effective way to convert entities and relations as vector representations while preserving the graph structure. It has been widely used in knowledge enhanced recommendation algorithms [1, 34, 38]. Existing knowledge graph embedding methods [2, 21, 44] mainly leverage negative sampling for model optimization, which however, has been shown not robust in recent studies [7, 43]. In this paper, we propose to apply non-sampling strategy for knowledge graph embedding learning. Specifically, for a batch of entities $\mathbf{B}$, the squared loss in graph embedding learning is defined as:

$$\begin{aligned}\mathcal{L}_{KG}(\Theta) &= \sum_{h\in\mathbf{B}}\sum_{t\in\mathbf{E}}\sum_{r\in\mathbf{R}} w_{hrt}(g_{hrt} - \hat{g}_{hrt})^2 \\ &= \sum_{h\in\mathbf{B}}\sum_{t\in\mathbf{E}}\sum_{r\in\mathbf{R}} w_{hrt}(g^2_{hrt} - 2g_{hrt}\hat{g}_{hrt} + \hat{g}^2_{hrt})\end{aligned} \tag{4}$$

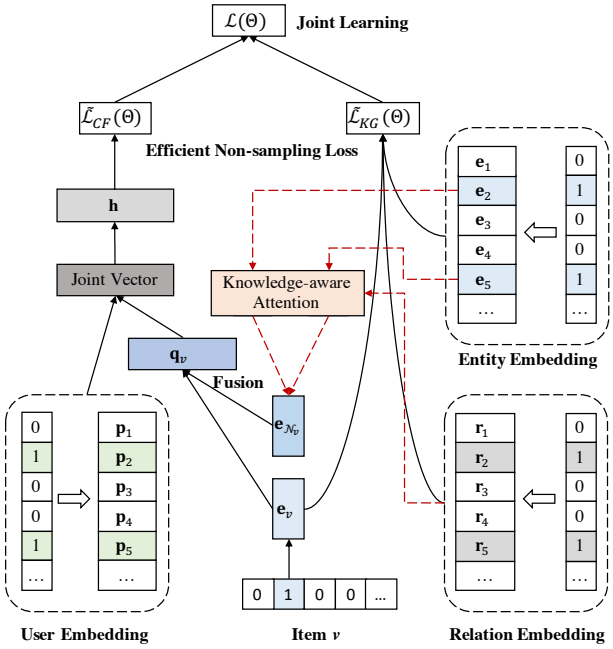

Figure 2: Illustration of our Jointly Non-Sampling model for Knowledge graph enhanced Recommendation (JNSKR).

where w_{hrt} denotes the weight of entry g_{hrt}, $g_{hrt} = 1$ if there is a relation r between h and t, and $g_{hrt} = 0$ otherwise. Since $g_{hrt} \in \{0, 1\}$, it can be replaced by a constant to simplify the equation. Also, the loss of non-observed data can be expressed by the residual between the loss of all data and that of positive data. We have the following derivation:

$$\begin{aligned}\tilde{\mathcal{L}}_{KG}(\Theta) &= -2\sum_{h\in\mathbf{B}}\sum_{t\in\mathbf{E}^+}\sum_{r\in\mathbf{R}^+} w^+_{hrt}\hat{g}_{hrt} + \sum_{h\in\mathbf{B}}\sum_{t\in\mathbf{E}}\sum_{r\in\mathbf{R}} w_{hrt}\hat{g}^2_{hrt} \\ &= \overbrace{\sum_{h\in\mathbf{B}}\sum_{t\in\mathbf{E}^+}\sum_{r\in\mathbf{R}^+}\left((w^+_{hrt} - w^-_{hrt})\hat{g}^2_{hrt} - 2w^+_{hrt}\hat{g}_{hrt}\right)}^{\mathcal{L}^P_{KG}(\Theta)} \\ &\quad + \overbrace{\sum_{h\in\mathbf{B}}\sum_{t\in\mathbf{E}}\sum_{r\in\mathbf{R}} w^-_{hrt}\hat{g}^2_{hrt}}^{\mathcal{L}^A_{KG}(\Theta)}\end{aligned} \tag{5}$$

where the Θ-invariant constant value has been eliminated, $\mathcal{L}^P_{KG}(\Theta)$ denotes the loss for positive data, and $\mathcal{L}^A_{KG}(\Theta)$ denotes the loss for all data. The computational bottleneck lies in $\mathcal{L}^A_{KG}(\Theta)$.

As can be seen from Eq.(5), to address the inefficiency issue of $\mathcal{L}^A_{KG}(\Theta)$, $\hat{g}^2_{hrt}$ need to be a score function that can be properly expanded. Translational distance models like TransR [21] do not meet this requirement since the expansion of $\hat{g}^2_{hrt}$ will introduce new terms. Thus we employ DistMult [44], another state-of-the-art factorization-based KG embedding method here. It defines the scoring function as:

$$\hat{g}_{hrt} = \mathbf{e}_h^T \cdot diag(\mathbf{r}) \cdot \mathbf{e}_t = \sum_i^d e_{h,i} r_i e_{t,i} \tag{6}$$

where $diag(\mathbf{r})$ denotes a diagonal matrix whose diagonal elements equal to $\mathbf{r}$ correspondingly.

Based on a decouple manipulation for the inner product operation, the summation operator and elements in $\mathbf{e}_h$, $\mathbf{e}_t$ and $\mathbf{r}$ can be rearranged:

$$\begin{aligned} \hat{g}_{hrt}^2 &= \sum_i^d e_{h,i} r_i e_{t,i} \sum_j^d e_{h,j} r_j e_{t,j} \\ &= \sum_i^d \sum_j^d (e_{h,i} e_{h,j})(r_i, r_j)(e_{t,i} e_{t,j}) \end{aligned} \tag{7}$$

By substituting Eq.(7) in $\mathcal{L}_{KG}^A(\Theta)$, we can see that if simplify w_{hrt} to a uniform [16] or entity-dependent [15, 20] parameter, the interaction among $e_{h,i}$, $e_{t,i}$, and r_i can be properly separated. Then, the optimization of $\sum_{h\in\mathbf{B}} w_h^- e_{h,i} e_{h,j}$, $\sum_{t\in\mathbf{E}} e_{t,i} e_{t,j}$, and $\sum_{r\in\mathbf{R}} r_i r_j$ are independent from each other, and we can achieve a significant speed-up by precomputing the three terms. The final efficient non-sampling loss for KG embedding learning is as follows:

$$\begin{aligned} \tilde{\mathcal{L}}_{KG}(\Theta) &= \mathcal{L}_{KG}^P(\Theta) \\ &+ \sum_{i=1}^d \sum_{j=1}^d \left(\left(\sum_{r\in\mathbf{R}} r_i r_j \right) \left(\sum_{h\in\mathbf{B}} w_h^- e_{h,i} e_{h,j} \right) \left(\sum_{t\in\mathbf{E}} e_{t,i} e_{t,j} \right) \right) \end{aligned} \tag{8}$$

The rearrangement of nested sums in $\mathcal{L}_{KG}^A(\Theta)$ is the key transformation that allows the fast optimization. The computing complexity of $\mathcal{L}_{KG}^A(\Theta)$ is reduced from $O(|\mathbf{B}||\mathbf{E}||\mathbf{R}|d)$ to $O((|\mathbf{B}| + |\mathbf{E}| + |\mathbf{R}|)d^2)$.

4.2 User-Item Preference Modeling

Next, we build upon the architecture of graph attention network [32, 38] to learn user preference over items. The preference prediction framework we adopt here is the neural form of MF [14], which is:

$$\hat{y}_{uv} = \mathbf{h}^T \left(\mathbf{p}_u \odot \mathbf{q}_v \right) \tag{9}$$

where $\mathbf{h} \in \mathbb{R}^d$ is the prediction vector, $\odot$ denotes the element-wise product of vectors. $\mathbf{p}_u$ and $\mathbf{q}_v$ are the representations of user u and item v, respectively. $\mathbf{p}_u$ is randomly initialized through embedding and then learnt during model training. Our focus is $\mathbf{q}_v$ here, as item acts as a bridge between KG and users.

For an item v, its final representation $\mathbf{q}_v$ is not only determined by its own message, but also influenced by the neighbored entities and relations. Obviously, the relation types and entity values are both important to characterize an item. For example, a user may pay more attention to genres when selecting a movie, and among all the genres, she is more interested in action than romantic. Since attention mechanism [4, 5, 10, 41] has a superior ability to assign non-uniform weights according to input instances, it is adopted in our model to learn fine-grained item embeddings.

We use $\mathcal{N}_v = \{(v, r, t) | g_{vrt} = 1\}$ to denote the neighbored knowledge triplets of v. To characterize item v, we define:

$$\begin{aligned} \mathbf{q}_v &= \mathbf{e}_v + \mathbf{e}_{\mathcal{N}_v} \\ &= \mathbf{e}_v + \sum_{(v,r,t)\in\mathcal{N}_v} \alpha_{(r,t)} \mathbf{e}_t \end{aligned} \tag{10}$$

where $\alpha_{(r,t)}$ is the attention weight, indicating how much information being propagated from t to v conditioned to relation r. $\mathbf{e}_v$ is item's basic feature vector and $\mathbf{e}_{\mathcal{N}_v}$ represents the information of v's knowledge triplets. More precisely, we define $\alpha_{(r,t)}$ as:

$$\begin{aligned} \alpha^*_{(r,t)} &= \mathbf{h}_\alpha^T \sigma(\mathbf{W}_1 \mathbf{e}_t + \mathbf{W}_2 \mathbf{r} + \mathbf{b}) \\ \alpha_{(r,t)} &= \frac{\exp(\alpha^*_{(r,t)})}{\sum_{(v,r',t')\in\mathcal{N}_v} \exp(\alpha^*_{(r',t')})} \end{aligned} \tag{11}$$

where $\mathbf{W}_1 \in \mathbb{R}^{k\times d}$, $\mathbf{W}_2 \in \mathbb{R}^{k\times d}$, $\mathbf{b} \in \mathbb{R}^k$, and $\mathbf{h}_\alpha \in \mathbb{R}^k$ are parameters of the attention network. k is the dimension of attention size, and σ is the nonlinear activation function *ReLU* [23]. Attention weights across all triplets are normalized by the softmax function.

Now we have completed the modeling of item v. Based on the learnt representation vectors, the prediction part aims to generate a score that indicates a user's preferences for an item. Note that our prediction part (Eq. (9)) satisfies the requirements of Eq.(3) [7], thus for a batch of items $\mathbf{B}$, we have the following efficient non-sampling loss function:

$$\begin{aligned} \tilde{\mathcal{L}}_{CF}(\Theta) = &\sum_{u\in\mathbf{U}^+} \sum_{v\in\mathbf{B}} \left((c_v^+ - c_v^-) \hat{y}_{uv}^2 - 2 c_v^+ \hat{y}_{uv} \right) \\ &+ \sum_{i=1}^d \sum_{j=1}^d \left((h_i h_j) \left(\sum_{u\in\mathbf{U}} p_{u,i} p_{u,j} \right) \left(\sum_{v\in\mathbf{B}} c_v^- q_{v,i} q_{v,j} \right) \right) \end{aligned} \tag{12}$$

4.3 Jointly Multi-task Learning

To effectively learn parameters for recommendation, as well as preserve the well-structured information of knowledge graph, we integrate the recommendation part (i.e., $\tilde{\mathcal{L}}_{CF}(\Theta)$) and the knowledge graph embedding part (i.e., $\tilde{\mathcal{L}}_{KG}(\Theta)$) in an end-to-end fashion through a jointly multi-task learning framework:

$$\mathcal{L}(\Theta) = \tilde{\mathcal{L}}_{CF}(\Theta) + \mu \tilde{\mathcal{L}}_{KG}(\Theta) + \lambda \|\Theta\|_2^2 \tag{13}$$

where $\tilde{\mathcal{L}}_{CF}(\Theta)$ is the recommendation loss from Eq.(12), $\tilde{\mathcal{L}}_{KG}(\Theta)$ is the KG embedding loss from Eq.(8), and μ is the parameter to adjust the weight proportion of each term. L_2 regularization parameterized by λ on Θ is conducted to prevent overfitting.

Note that previous KG enhanced recommendation methods [35, 36, 38, 39, 42] typically adopt negative sampling for model learning. To generate a training batch, these methods need to sample negative instances for both recommendation task and knowledge graph embedding task. This produces a much larger randomness than single-task learning, and would inevitably lead to information loss. Different from previous work, the parameters in our model are jointly optimized without negative sampling. The training procedure of JNSKR is illustrated in Algorithm 1.

To optimize the objective function, we use mini-batch Adagrad [11] as the optimizer. Its main advantage is that the learning rate can be self-adaptive during the training phase, which eases the pain of choosing a proper learning rate. Dropout is an effective

Algorithm 1 JNSKR Learning algorithm

Require: Training data $\{\mathbf{Y}, \mathbf{U}, \mathbf{V}, \mathbf{G}, \mathbf{E}, \mathbf{R}\}$ learning rate η; embedding size d
Ensure: Neural parameters Θ

```
Randomly initialize neural parameters Θ
while Stopping criteria is not met do
    while An epoch is not end do
        Randomly draw a mini-batch items {B}, training instances {Y_B, G_B}
        Compute the loss L̃_KG(Θ) (Eq.(8))
        Compute the loss L̃_CF(Θ) (Eq.(12))
        L(Θ) ← L̃_CF(Θ) + μL̃_KG(Θ)
        Update model parameters
    end while
end while
return Θ
```

solution to prevent neural networks from overfitting [29], which randomly drops part of neurons during training. In this work, we employ the node dropout technique to randomly drop ρ percent of $\mathbf{q}_v$, where ρ is the dropout ratio.

4.4 Discussion

We first discuss the time complexity of our model. The complexity of our JNSKR can be divided into two parts. For knowledge graph embedding (Eq.(8)), updating a batch of items takes $O((|\mathbf{B}| + |\mathbf{E}| + |\mathbf{R}|)d^2 + |\mathcal{G}_\mathbf{B}|d)$, where $\mathcal{G}_\mathbf{B}$ denotes positive knowledge triples of this batch. For recommendation task (Eq.(12)), updating a batch of items takes $O((|\mathbf{B}| + |\mathbf{U}|)d^2 + |\mathcal{Y}_\mathbf{B}|d)$ (the time overhead of attention network is rather small and can be ignored), where $\mathcal{Y}_\mathbf{B}$ denotes positive user-item interactions of this batch. Therefore, the total cost of Algorithm 1 for one batch over all parameters is $O((|\mathbf{B}|+|\mathbf{E}|+|\mathbf{R}|)d^2+|\mathcal{G}_\mathbf{B}|d+(|\mathbf{B}|+|\mathbf{U}|)d^2+|\mathcal{Y}_\mathbf{B}|d)$. For the original regression loss, it takes $O((|\mathbf{B}||\mathbf{E}||\mathbf{R}| + |\mathbf{B}||\mathbf{U}|)d)$. Since $|\mathcal{G}_\mathbf{B}| \ll |\mathbf{B}||\mathbf{E}||\mathbf{R}|$, $|\mathcal{Y}_\mathbf{B}| \ll |\mathbf{B}||\mathbf{U}|$, and $d \ll |\mathbf{B}|$ in practice, the computational complexity of our model is reduced by several magnitudes.

The proposed efficient learning algorithm of our JNSKR is based on Eq.(3) [7], which is not applicable for models with non-linear prediction layers. Thus our current JNSKR framework has a linear prediction layer on the top. We leave the extensions as future work. Nevertheless, it is worth mention that compared to the state-of-the-art deep learning methods — the 1-layer NFM [13], RNN based RippleNet [34], and GNN based KGAT [38], our JNSKR achieves significant improvements on Top-K recommendation performance, while maintaining a much simpler structure, fewer model parameters, and much fast training process. We show the details in Section 5.

5 EXPERIMENTS

5.1 Experimental Setup

5.1.1 ***Data Description.*** We experiment with two benchmark datasets: ***Amazon-book***[2] and ***Yelp2018***[3]. The two datasets have been recently extended for KG enhanced recommendation by the authors of [38][4]. We briefly introduce the two datasets:

- **Amazon-book**: Amazon datasets have been widely used for item recommendation [4, 12, 38]. In our experiments, we use Amazon-book of this collection. The item knowledge of Amazon-book is constructed by mapping the items into Freebase entities via title matching if there is a mapping available.
- **Yelp2018**: This dataset is adopted from the 2018 edition of the Yelp challenge, which recodes users' ratings on local businesses like restaurants and bars. The item knowledge is extracted from the local business information network (e.g., category, location, and attribute).

To ensure the KG quality, the two datasets are preprocessed to filter out infrequent entities (i.e., lower than 10 in both datasets) and retain the relations appearing in at least 50 triplets. Note that for objective comparison, in our experiments the two datasets are exactly the same as those used in [38]. The statistical details of these datasets are summarized in Table 2.

Table 2: Statistical details of the evaluation datasets.

		Amazon-book	*Yelp2018*
User-Item Interaction	**#Users**	70, 679	45, 919
	#Items	24, 915	45, 538
	#Interactions	847, 733	1, 185, 068
Knowledge Graph	**#Entities**	88, 572	90, 961
	#Relations	39	42
	#Triplets	2, 557, 746	1, 853, 704

5.1.2 ***Baselines.*** To evaluate the effectiveness, we compare our proposed JNSKR with plain CF methods (NCF and ENMF), featured-based method (NFM), and various KG enhanced methods (regularization-based CKE and CFKG, path-based RippleNet, and graph neural network-based KGAT), as follows:

- **NCF** [14]: This is a deep learning method which uses users' historical feedback for item ranking. It combines MF with a multilayer perceptron (MLP).
- **ENMF** [7, 8]: Efficient Neural Matrix Factorization is a newly proposed non-sampling recommendation method. It is a state-of-the-art method for Top-K recommendation which only based on the historical feedback information.
- **NFM** [13]: Neural factorization machine is one of the state-of-the-art feature-based methods which uses MLP to learn nonlinear and high-order interaction signals.
- **CKE** [47]: This is a representative regularization-based method, which exploits semantic embeddings derived from TransR [21] to enhance matrix factorization [25].
- **CFKG** [1]: The model applies TransE [2] on the unified graph including users, items, entities, and relations, casting the recommendation task as the prediction of (u, r, v) triplets.
- **RippleNet** [34]: This is one of the state-of-the-art path-based models, which enriches user representations by adding that of items within paths rooted at each user.

[2] http://jmcauley.ucsd.edu/data/amazon
[3] https://www.yelp.com/dataset/challenge
[4] https://github.com/xiangwang1223/knowledge_graph_attention_network

Table 3: Performance of different models on three datasets. ** denotes the statistical significance for $p < 0.01$, compared to the best baseline. "RI" indicate the average relative improvements of our JNSKR over the corresponding baseline.

Models	*Amazon-book*						
	Recall@10	**Recall@20**	**Recall@40**	**NDCG@10**	**NDCG@20**	**NDCG@40**	**RI**
NCF	0.0874	0.1319	0.1924	0.0724	0.0895	0.1111	+17.03%
ENMF	0.1002	0.1472	0.2085	0.0797	0.0998	0.1215	+5.49%
NFM	0.0891	0.1366	0.1975	0.0723	0.0913	0.1152	+14.44%
CKE	0.0875	0.1343	0.1946	0.0705	0.0885	0.1114	+17.14%
CFKG	0.0769	0.1142	0.1901	0.0603	0.077	0.0985	+32.62%
RippleNet	0.0883	0.1336	0.2008	0.0747	0.0910	0.1164	+13.99%
KGAT	0.1017	0.1489	0.2094	0.0814	0.1006	0.1225	+4.31%
JNSKR	**0.1056****	**0.1558****	**0.2178****	**0.0842****	**0.1068****	**0.1271****	–
Models	*Yelp2018*						
	Recall@10	**Recall@20**	**Recall@40**	**NDCG@10**	**NDCG@20**	**NDCG@40**	**RI**
NCF	0.0389	0.0653	0.1060	0.0603	0.0802	0.1087	+14.28%
ENMF	0.0403	0.0711	0.1109	0.0611	0.0877	0.1097	+9.15%
NFM	0.0396	0.0660	0.1082	0.0603	0.0810	0.1094	+13.03%
CKE	0.0399	0.0657	0.1074	0.0608	0.0805	0.1091	+13.13%
CFKG	0.0288	0.0522	0.0904	0.0450	0.0644	0.0897	+44.27%
RippleNet	0.0402	0.0664	0.1088	0.0613	0.0822	0.1097	+11.90%
KGAT	0.0418	0.0712	0.1128	0.0630	0.0867	0.1129	+7.26%
JNSKR	**0.0456****	**0.0749****	**0.1209****	**0.0687****	**0.0917****	**0.1211****	–

The results of KGAT are the same as those reported in [38] since we share exactly the same data splits and experimental settings.

- **KGAT** [38]: A state-of-the-art KG enhanced model, which employs graph neural network and attention mechanism to learn high-order graph-structured data for recommendation.

5.1.3 ***Evaluation Metrics.*** For each dataset, we randomly select 80% of interaction history of each user to construct the training set, and treat the remaining as the test set. From the training set, we randomly select 10% of interactions as validation set to tune hyper-parameters. For each user, our evaluation protocol ranks all the items except the positive ones in the training set. To evaluate the effectiveness of top-K recommendation, we apply two widely-used evaluation protocols [5, 7, 38]: *Recall@K* and *NDCG@K*. *Recall@K* measures whether the ground truth is ranked among the top K items, while *NDCG@K* is a position-aware ranking metric.

5.1.4 ***Parameter Settings.*** The parameters for all baseline methods are initialized as in the corresponding papers, and are then carefully tuned to achieve optimal performances. The learning rate is tuned amongst [0.005, 0.01, 0.02, 0.05], the coefficient of L_2 normalization is searched in [10^{-5}, 10^{-4} ,..., 10^{-1}, 1]. To prevent overfitting, the dropout ratio is tuned in [0.0, 0.1, ..., 0.9]. The dimension of attention network k and the latent factor number d are tested in [16, 32, 64]. After the tuning process, the batch size is set to 512, the learning rate is set to 0.05, the embedding size d is set to 64, and the attention size k is set to 32. Regarding NFM, the number of MLP layers is set as 1 with 64 neurons according to the original paper [13]. For RippleNet, we set the number of hops and the memory size as 2 and 8, respectively, according to [34]. For KGAT, we set the depth as 3 with hidden dimension 64, 32, and 16, respectively, as suggested in [38]. For non-sampling methods ENMF and our JNSKR, the negative weights are calculated based on the frequency of items [7, 15]. For the optimization objective of JNSKR, we set the weight parameter $\mu = 0.01$.

5.2 Performance Comparison

The performance comparison results are presented in Table 3. To evaluate on different recommendation lengths, we set the length K = 10, 20, and 40 in our experiments. From the results, the following observations can be made:

First and foremost, our proposed JNSKR achieves the best performance on the two datasets, significantly outperforming all the state-of-the-art baseline methods with p-values smaller than 0.01. In particular, compared to KGAT — a recently proposed and very expressive graph neural network-based model, our JNSKR exhibits average improvements of 4.31% and 7.26% on Amazon-book and Yelp2018, respectively. This is very remarkable, since JNSKR is a shallow framework that has much fewer parameters. The substantial improvement could be attributed to the proposed non-sampling learning algorithm. The parameters in JNSKR is optimized on the whole data, while sample-based methods (NCF, NFM, CKE, CFKG, RippleNet, KGAT) only use a fraction of sampled data which would ignore important negative examples. Moreover, compared with the conventional CF methods NCF and ENMF which only consider user-item interactions, JNSKR shows the effectiveness of knowledge graph information for user preference modelling.

Second, non-sampling methods (ENMF and our JNSKR) generally perform better than sampling-based methods. For example, in Table 3, the performance of ENMF is better than NCF; and our JNSKR outperforms all the baselines. This is consistent with previous work [7, 15, 43, 46]. It indicates that negative sampling is a biased learning strategy and would inevitably lead to information loss.

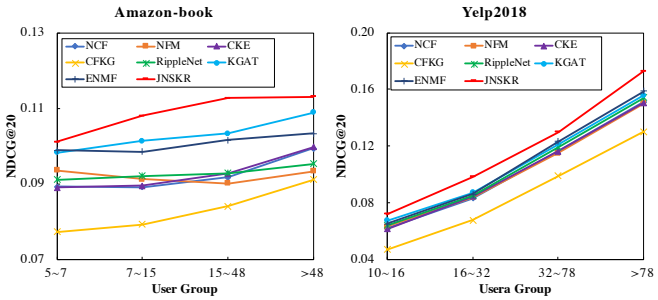

Figure 3: Performance comparison over the sparsity distribution of user groups on Amazon-book and Yelp2018 datasets.

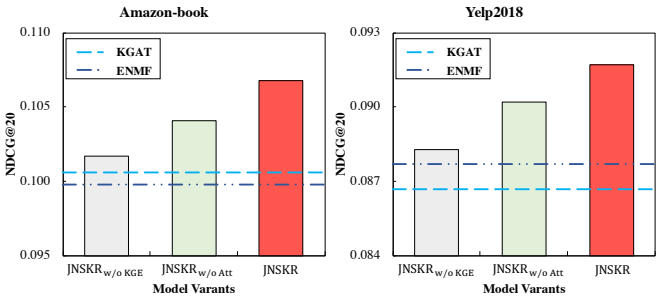

Figure 4: Performance of variants of JNSKR on Amazon-book and Yelp2018 datasets. The two dotted lines represent the results of KGAT (KG enhanced method) and ENMF (plain CF method) respectively, which are added as baselines.

Lastly, we observe that recent studies on KG enhanced recommendation have largely focused on advanced neural network structures. Although they do achieve better performance than conventional CF methods when adopting the same sampling-based learning strategy (e.g., KG baselines vs NCF), they are still limited by the inherent weakness of negative sampling. For example, in Table 3, even ENMF which utilizes no KG information performs better than NFM, CKE, CFKG, and RippleNet, and only slightly underperforms KGAT. It reveals that for recommendation task, a better learning strategy is even more important than advanced neural network structures. The performance gap between baselines and our JNSKR also reflects the value of learning KG enhanced recommendation without sampling.

5.3 Handling Data Sparsity Issue

Data sparsity is a big challenge in recommendation [33] because it is hard to establish optimal representations for inactive users with few interactions. KG enhanced recommendation provides a solution to alleviate the data sparsity issue. Thus we further investigate how our JNSKR model performs for the users with few records. Specifically, we perform experiments over user groups of different sparsity levels. We divide the test set into four groups based on interaction number per user, while trying to keep different groups have the same total interactions. Figure 3 illustrates the results w.r.t. NDCG@20 on different user groups in Amazon-book and Yelp2018. From the results, we have the following observations:

First, generally KG enhanced recommendation methods show better performance than methods using only user-item interactions. Considering that KG and user-item interactions are correlated, the information learned from KG can compensate for the shortage of user feedback on items. As a result, the use of KG produces great improvement when the training data are scarce. It is also worthwhile pointing out that on Yelp2018 dataset, the state-of-the-art KG enhanced baselines like RippleNet and KGAT only slightly outperform plain CF methods, One possible reason is that the preferences of users with too many interactions are too general to capture. High-order connectivity could introduce more noise into the user preferences, thus leading to the negative effect [38].

Second, our JNSKR consistently outperforms the other models including the state-of-the-art KG enhanced methods like RippleNet and KGAT. It verifies the effectiveness of JNSKR in addressing the data sparsity issue by applying non-sampling learning to leverage KG information.

Table 4: Performance comparison of joint learning and alternative learning on Amazon-book and Yelp2018 datasets. * denotes the statistical significance for $p < 0.05$

	Amazon-book		*Yelp2018*	
	Recall@20	**NDCG@20**	**Recall@20**	**NDCG@20**
$\text{JNSKR}_{\text{Alt}}$	0.1532	0.1041	0.0731	0.0896
JSNKR	**0.1558***	**0.1068***	**0.0749***	**0.0917***

5.4 Ablation Study

5.4.1 ***Effect of Knowledge Graph Embedding and Attention Mechanism.*** Our JNSKR utilizes a non-sampling strategy to learn knowledge graph embedding, and an attention mechanism to model user preference. In this section, we first conduct ablation study to understand their effect. Specifically, we build two variants of JNSKR:

- $\text{JNSKR}_{\text{w/o KGE}}$: The variant model of JNSKR without the knowledge graph embedding part (cf. Eq(8)).
- $\text{JNSKR}_{\text{w/o Att}}$: The variant model of JNSKR without using attention mechanism. A constant weight is assigned to item's knowledge triplets (cf. Eq(10)).

Figure 4 shows the performance of different variants. The results of the state-of-art methods ENMF (plain CF method) and KGAT (KG enhanced method) are also shown as baselines. From Figure 4, two observations are made:

First, when incorporating the KG embedding part, JNSKR performs better than $\text{JNSKR}_{\text{w/o KGE}}$ (p<0.01). And when the attention component is applied, the performances are also improved compared with the constant weight method $\text{JNSKR}_{\text{w/o Att}}$ (p<0.05). It indicates that both the two parts are helpful for modelling the fine-grained connectivity among users, items, and entities.

Second, even without the KG embedding part or attention component, our variants $\text{JNSKR}_{\text{w/o KGE}}$ and $\text{JNSKR}_{\text{w/o Att}}$ still perform better than the best baseline KGAT, indicating the effectiveness of our JNSKR by adopting non-sampling learning for user preference modelling.

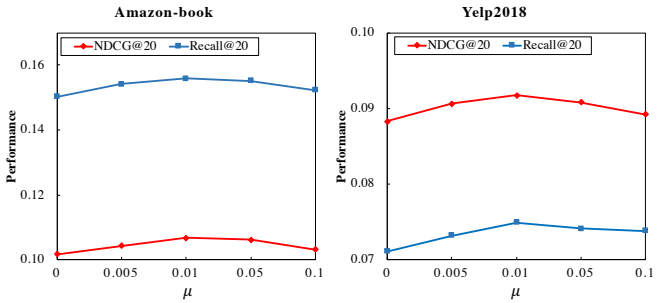

Figure 5: Effect of μ on Amazon-book and Yelp2018 datasets

5.4.2 ***Effect of Joint Learning.*** Previous KG enhanced methods [35, 38] like KGAT generally optimize $\mathcal{L}_{KG}$ and $\mathcal{L}_{CF}$ alternatively to increase computational efficiency. Specifically, in each epoch, these methods first fix the parameters of $\mathcal{L}_{CF}$ to train $\mathcal{L}_{KG}$, and then fix the parameters of $\mathcal{L}_{KG}$ to train $\mathcal{L}_{CF}$. We argue that this is actually a compromise method to alleviate the expensive computational cost, which would inevitably lead to insufficient training. Different from them, the parameters of our JNSKR are jointly optimized through the proposed efficient non-sampling framework, which achieves both effective and efficient performance. To further verify the effect of joint learning, we conduct experiments to test the performance of joint learning and alternative learning. The results are shown in Table 4, where $\text{JNSKR}_{\text{Alt}}$ is a variant that using alternative strategy for training. As shown in the table, the alternative learning $\text{JNSKR}_{\text{Alt}}$ performs worth than joint learning JNSKR. It makes sense since alternative learning fails to sufficiently and collaboratively model the representation relatedness on the granularity of users, items, and entities.

The coefficient μ in the joint loss function (cf. Eq(13)) explicitly guides the learning process of both KG embedding and recommendation, and thus helps to improve the model performance. To test the impact of μ, we also conduct experiments and the results are shown in Figure 5. We can see that with the increase of μ, the performance improves first and then starts to decrease. Since the primary target of JNSKR is recommendation other than learning KG embedding, it is necessary to ensure that $\mathcal{L}_{CF}$ is the key part of the total loss.

5.5 Efficiency Study

Many deep learning studies only focused on obtaining better results but ignored the computational efficiency of reaching the reported accuracy [27]. However, expensive training cost can limit the applicability of a model to real-world large-scale systems. In this section, we conduct experiments to explore the training efficiencies of our JNSKR and four state-of-the-art KG enhanced methods: CKE, CFKG, RippleNet, and KGAT. All experiments in this section are run on the same machine (Intel Xeon 8-Core CPU of 2.4 GHz and single NVIDIA GeForce GTX TITAN X GPU) for fair comparison on the efficiency. The comparison results among the overall training time of the above methods are shown in Table 5.

From the table, we can obviously observe that the overall training time of our JNSKR is **several magnitudes** faster than the baseline models. For example, on Amazon-book dataset, our JNSKR only needs 47 minutes to achieve the optimal performance, while the state-of-the-art models RippleNet and KGAT take about 50 hours and 45 hours, respectively. This acceleration is over 20 times, which is highly valuable in practice and is difficult to achieve with simple engineering efforts. In real E-commerce scenarios, the cost of training time is also an important factor to be considered [7]. Our JNSKR shows significant advantages in training efficiency, which makes it more practical in real life.

Table 5: Comparisons of runtime (second/minute/hour [s/m/h]). "S", "I", and "T" represent the training time for a single iteration, the number of iterations to converge, and the total training time, respectively.

Model	*Amazon-book*			*Yelp2018*		
	S	**I**	**T**	**S**	**I**	**T**
CKE	66s	200	220m	75s	200	250m
CFKG	27s	200	90m	45s	200	150m
RippleNet	15m	200	50h	11m	200	37h
KGAT	9m	300	45h	7m	300	35h
JNSKR	**14s**	200	**47m**	**16s**	200	**54m**

6 CONCLUSION AND FUTURE WORK

In this paper, we propose a novel Jointly Non-Sampling learning model for Knowledge graph enhanced Recommendation (JNSKR). Different from previous studies which mainly focus on exploring novel neural networks, we try to address the problem from the basic but important perspective of model learning. Specifically, we first design a new efficient non-sampling loss for knowledge graph embedding learning, whose complexity is reduced significantly. We then aggregate an item's surrounding entities with attention mechanisms to help learn accurate user preference over items. Our JNSKR is conceptually advantageous to existing methods in: 1) effective non-sampling learning and 2) efficient model training. Extensive experiments have been made on two real-life datasets. The proposed JNSKR consistently and significantly outperforms the state-of-the-art recommendation models in terms of both recommendation performance and training efficiency.

This work complements the mainstream sampling-based KG enhanced recommendation methods, and empirically shows that a proper learning method is even more important than advanced neural network structures. In the future, we will explore JNSKR on other related tasks like knowledge graph representation [21] and network embedding [28]. Also, we are interested in integrating more structural information such as social networks [7] and heterogeneous information networks [45, 49], to further improve our model.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments and suggestions. This work is supported by Natural Science Foundation of China (Grant No. 61672311, 61532011) and the National Key Research and Development Program of China (2018YFC0831900). Dr Weizhi Ma has been supported by Shuimu Tsinghua Scholar Program.

REFERENCES

[1] Qingyao Ai, Vahid Azizi, Xu Chen, and Yongfeng Zhang. 2018. Learning heterogeneous knowledge base embeddings for explainable recommendation. *Algorithms* 11, 9 (2018), 137.

[2] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Proceedings of NeuIPS*. 2787–2795.

[3] 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 *Proceedings of WWW*. 151–161.

[4] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-level Explanations. In *Proceedings of WWW*.

[5] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2019. Social Attentional Memory Network: Modeling Aspect-and Friend-level Differences in Recommendation. In *Proceedings of WSDM*.

[6] Chong Chen, Min Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Efficient Non-Sampling Factorization Machines for Optimal Context-Aware Recommendation. In *Proceedings of WWW*. 2400–2410.

[7] Chong Chen, Min Zhang, Chenyang Wang, Weizhi Ma, Minming Li, Yiqun Liu, and Shaoping Ma. 2019. An Efficient Adaptive Transfer Neural Network for Social-aware Recommendation. In *Proceedings of SIGIR*. ACM, 225–234.

[8] Chong Chen, Min Zhang, Yongfeng Zhang, Yiqun Liu, and Shaoping Ma. 2020. Efficient Neural Matrix Factorization without Sampling for Recommendation. *ACM Trans. Inf. Syst.* 38, 2, Article Article 14 (Jan. 2020).

[9] Chong Chen, Min Zhang, Yongfeng Zhang, Weizhi Ma, Yiqun Liu, and Shaoping Ma. 2020. Efficient Heterogeneous Collaborative Filtering without Negative Sampling for Recommendation. In *Proceedings of AAAI*.

[10] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Liqiang Nie, Wei Liu, and Tat-Seng Chua. 2017. Attentive collaborative filtering: Multimedia recommendation with item-and component-level attention. In *Proceedings of SIGIR*. 335–344.

[11] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, Jul (2011), 2121–2159.

[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 WWW*. 507–517.

[13] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of SIGIR*. 355–364.

[14] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of WWW*. 173–182.

[15] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of SIGIR*. 549–558.

[16] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of ICDM*. 263–272.

[17] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y Chang. 2018. Improving sequential recommendation with knowledge-enhanced memory networks. In *Proceedings of SIGIR*. ACM, 505–514.

[18] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of KDD*. 426–434.

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

[20] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of WWW*. 951–961.

[21] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of AAAI*.

[22] Weizhi Ma, Min Zhang, Yue Cao, Woojeong Jin, Chenyang Wang, Yiqun Liu, Shaoping Ma, and Xiang Ren. 2019. Jointly Learning Explainable Rules for Recommendation with Knowledge Graph. In *Proceedings of WWW*.

[23] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of ICML*. 807–814.

[24] István Pilászy, Dávid Zibriczky, and Domonkos Tikk. 2010. Fast als-based matrix factorization for explicit and implicit feedback datasets. In *Proceedings of Recsys*.

[25] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of UAI*. 452–461.

[26] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to recommender systems handbook. In *Recommender systems handbook*. Springer.

[27] Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2019. Green AI. *arXiv preprint arXiv:1907.10597* (2019).

[28] Yu Shi, Qi Zhu, Fang Guo, Chao Zhang, and Jiawei Han. 2018. Easing Embedding Learning by Comprehensive Transcription of Heterogeneous Information Networks. In *Proceedings of KDD*. 2190–2199.

[29] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research* 15, 1 (2014), 1929–1958.

[30] Zhu Sun, Jie Yang, Jie Zhang, Alessandro Bozzon, Long-Kai Huang, and Chi Xu. 2018. Recurrent knowledge graph embedding for effective recommendation. In *Proceedings of RecSys*. ACM, 297–305.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of NeuIPS*. 5998–6008.

[32] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph attention networks. In *ICLR*.

[33] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. 2017. Dropoutnet: Addressing cold start in recommender systems. In *Proceedings of NeuIPS*. 4957–4966.

[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 CIKM*. 417–426.

[35] Hongwei Wang, Fuzheng Zhang, Miao Zhao, Wenjie Li, Xing Xie, and Minyi Guo. 2019. Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. In *Proceedings of WWW*. ACM, 2000–2010.

[36] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. 2019. Knowledge graph convolutional networks for recommender systems. In *Proceedings of WWW*. ACM, 3307–3313.

[37] Quan Wang, Zhendong Mao, Bin Wang, and Li Guo. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering* 29, 12 (2017), 2724–2743.

[38] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. KGAT: Knowledge Graph Attention Network for Recommendation. In *Proceedings of KDD*.

[39] 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 AAAI*, Vol. 33. 5329–5336.

[40] 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.

[41] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* (2017).

[42] Xin Xin, Xiangnan He, Yongfeng Zhang, Yongdong Zhang, and Joemon Jose. 2019. Relational Collaborative Filtering: Modeling Multiple Item Relations for Recommendation. In *Proceedings of SIGIR*.

[43] Xin Xin, Fajie Yuan, Xiangnan He, and Joemon M Jose. 2018. Batch IS NOT Heavy: LearningWord Representations From All Samples. (2018).

[44] Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2014. Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575* (2014).

[45] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *Proceedings of WSDM*.

[46] Fajie Yuan, Xin Xin, Xiangnan He, Guibing Guo, Weinan Zhang, Chua Tat-Seng, and Joemon M Jose. 2018. fbgd: Learning embeddings from positive unlabeled data with bgd. (2018).

[47] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of KDD*. 353–362.

[48] Yongfeng Zhang, Xu Chen, et al. 2020. Explainable Recommendation: A Survey and New Perspectives. *Foundations and Trends® in Information Retrieval* 14, 1 (2020), 1–101.

[49] Huan Zhao, Quanming Yao, Jianda Li, Yangqiu Song, and Dik Lun Lee. 2017. Meta-graph based recommendation fusion over heterogeneous information networks. In *Proceedings of KDD*. ACM, 635–644.