DiffKG: Knowledge Graph Diffusion Model for Recommendation

Yangqin Jiang
University of Hong Kong
Hong Kong, China
mrjiangyq99@gmail.com

Yuhao Yang
University of Hong Kong
Hong Kong, China
yuhao-yang@outlook.com

Lianghao Xia
University of Hong Kong
Hong Kong, China
aka_xia@foxmail.com

Chao Huang*
University of Hong Kong
Hong Kong, China
chaohuang75@gmail.com

ABSTRACT

Knowledge Graphs (KGs) have emerged as invaluable resources for enriching recommendation systems by providing a wealth of factual information and capturing semantic relationships among items. Leveraging KGs can significantly enhance recommendation performance. However, not all relations within a KG are equally relevant or beneficial for the target recommendation task. In fact, certain item-entity connections may introduce noise or lack informative value, thus potentially misleading our understanding of user preferences. To bridge this research gap, we propose a novel knowledge graph diffusion model for recommendation, referred to as DiffKG. Our framework integrates a generative diffusion model with a data augmentation paradigm, enabling robust knowledge graph representation learning. This integration facilitates a better alignment between knowledge-aware item semantics and collaborative relation modeling. Moreover, we introduce a collaborative knowledge graph convolution mechanism that incorporates collaborative signals reflecting user-item interaction patterns, guiding the knowledge graph diffusion process. We conduct extensive experiments on three publicly available datasets, consistently demonstrating the superiority of our DiffKG compared to various competitive baselines. We provide the source code repository of our proposed DiffKG model at the following link: https://github.com/HKUDS/DiffKG.

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Recommendation, Diffusion Model, Knowledge Graph Learning

ACM Reference Format:

Yangqin Jiang, Yuhao Yang, Lianghao Xia, and Chao Huang. 2024. DiffKG: Knowledge Graph Diffusion Model for Recommendation. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining (WSDM '24), March 4–8, 2024, Merida, Mexico.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3616855.3635850

*Chao Huang is the corresponding author.

WSDM '24, March 4–8, 2024, Merida, Mexico

ACM ISBN 979-8-4007-0371-3/24/03…$15.00
https://doi.org/10.1145/3616855.3635850

1 INTRODUCTION

In the context of the information overload problem, recommendation systems have gained substantial influence in the modern web landscape. These systems have become an integral part of the online experience by effectively connecting users with items that align with their individual interests. Collaborative filtering (CF), one of the leading paradigms for recommendation systems, postulates that users who engage in similar interaction modes also share similar interests towards items. This approach has garnered considerable attention and proven to be highly effective in delivering personalized recommendations to users [11, 15, 19, 21].

The recommendation performance in practical scenarios is significantly hindered by the inherent sparsity of user-item interactions [36, 40]. To mitigate this issue, the integration of a knowledge graph (KG) as a comprehensive information network for items has emerged as a new trend in collaborative filtering, known as knowledge-aware recommendation. Researchers have explored knowledge-aware recommendation through two primary approaches: embedding-based methods and path-based methods. Embedding-based methods [2, 28, 42] have been employed to enhance the modeling of users and items by incorporating transition-based knowledge graph embeddings into item representations. On the other hand, path-based methods [34, 41] focus on extracting semantically meaningful meta-paths from the knowledge graph and leveraging them to perform complex modeling of users and items. To combine the strengths of embedding-based and path-based methods, recent research has turned to GNNs as a powerful tool. GNN methods leverage the capabilities of propagation and aggregation over the knowledge graph to capture high-order information [30, 32, 33].

Despite the demonstrated effectiveness of existing knowledge graph (KG)-aware recommendation methods, their performance heavily relies on high-quality input knowledge graphs and can be adversely affected by the presence of noise. In practical scenarios, knowledge graphs often suffer from sparsity and noise, characterized by long-tail entity distributions and topic-irrelevant connections between items and entities [17, 27]. To address these challenges, recent research has proposed the utilization of contrastive learning (CL) techniques to enhance knowledge-aware recommendation. For instance, the KGCL approach [39] leverages stochastic graph augmentation on the knowledge graph and employs CL to address the long-tail issues within the KG. Likewise, the MCCLK [44] and KGIC [45] methods introduce cross-view CL paradigms between the knowledge graph and user-item graph, aiming to integrate external item knowledge into the modeling of user-item interactions. However, it is worth noting that these methods predominantly rely on simplistic random augmentation or intuitive cross-view information, overlooking the substantial amount of irrelevant information present in the knowledge graph for the specific recommendation task. Thus, it is of paramount importance to effectively filter out noisy knowledge graph information, leading to a more resilient encoding of user preferences.

This research introduces an innovative model known as DiffKG for knowledge-aware recommender systems. Drawing inspiration from recent advancements in diffusion models, we propose a unique knowledge graph diffusion paradigm that effectively balances corruption and reconstruction. Our approach involves a progressive forward process where the initial knowledge graph undergoes step-by-step corruption through the introduction of random noises. This incremental corruption process accumulates noises over multiple iterations, which are then iteratively recovered to restore the original knowledge graph structures. By employing this tractable forward process, we establish a feasible posterior and enable reverse generation using flexible neural networks to model complex distributions iteratively. To address the challenge of noisy information within the knowledge graph, we introduce a KG filter that eliminates irrelevant and erroneous data, aligning seamlessly with the learning of user preferences. Additionally, we devise a collaborative knowledge graph convolution mechanism, which enhances our diffusion model by integrating collaborative signals into the KG diffusion process. It ensures the retention of relevant knowledge during the diffusion process. Furthermore, we propose a KG diffusion-enhanced data augmentation paradigm to benefit the model with the enriched information and improved learning capabilities.

In summary, this paper makes the following contributions:

- We present a novel recommendation model called DiffKG, which leverages task-relevant item knowledge to enhance the collaborative filtering paradigm. Our approach introduces a new framework that allows for the distillation of high-quality signals from the aggregated representation of noisy knowledge graphs.
- We propose an integration of the generative diffusion model with the knowledge graph learning framework, designed for knowledge-aware recommendation. This integration allows us to effectively align the semantics of knowledge-aware items with collaborative relation modeling for recommendation purposes.
- Our extensive experimental evaluations substantiate the substantial performance gains achieved by our DiffKG framework when compared to various baseline models across diverse benchmark datasets. Notably, our approach effectively tackles the challenges stemming from data noise and data scarcity, which are known to exert a negative impact on the accuracy of recommendation.

2 PRELIMINARIES

We introduce the key concepts that form the paper foundation and provide a formal definition of the KG-enhanced recommendation.

User-Item Interaction Graph. Consider a typical recommendation scenario with a set of users denoted as $\mathcal{U}$ and a set of items denoted as $\mathcal{I}$. Each individual user u belongs to the set $\mathcal{U}$, and each item i belongs to the set $\mathcal{I}$. To represent the collaborative signals between users and items, we construct a binary graph denoted as $\mathcal{G}_u = (u, y_{u,i}, i)$. Here, $y_{ui} = 1$ indicates that user u has interacted with item i, while $y_{u,i} = 0$ signifies the absence of such interaction.

Knowledge Graph. The knowledge graph is denoted as $\mathcal{G}_k = (h, r, t)$ and serves to organize external item attributes by incorporating various types of entities and their corresponding relationships. Each triplet (h, r, t) within the knowledge graph characterizes the semantic relatedness between the head entity h and the tail entity t, connected by the relation r. The entities h and t encompass items and their associated concepts, such as directors for movies. By utilizing this supplementary knowledge graph, we can effectively model and analyze the intricate relationships that exist between items and entities. This, in turn, empowers us to gain a more comprehensive and nuanced understanding of the item attributes.

We define the KG-enhanced recommendation task as follows: given the user-item interaction graph $\mathcal{G}_u$ and the associated knowledge graph $\mathcal{G}_k$, our objective is to train a recommender model $\mathcal{F}(u, i|\mathcal{G}_u, \mathcal{G}_k, \Theta)$ with learnable parameters Θ. This model aims to predict the likelihood of user u interacting with item i.

3 THE PROPOSED DIFFKG FRAMEWORK

In this section, we present the technical design of our proposed DiffKG, accompanied by the overall model architecture depicted in Fig. 1. Our model includes a heterogeneous knowledge aggregation module, a knowledge graph diffusion model, and a KG diffusion-enhanced data augmentation paradigm. These components effectively capture diverse relationships in the KG and ensure high-quality KG information for enhancing recommendation.

3.1 Heterogeneous Knowledge Aggregation

To handle the heterogeneity of knowledge relations in real-world knowledge graphs, we employ a relation-aware knowledge embedding layer inspired by graph attention mechanisms utilized in previous works such as [25, 32, 39]. This layer enables effective capturing of diverse relationships inherent in the connection structure of the knowledge graph. By incorporating a parameterized attention matrix, it projects entity-dependent context and relation-dependent context into specific representations, overcoming the limitations of manually designing path generation on knowledge graphs. The message aggregation mechanism between an item and its connected entities can be described as follows:

$$\begin{aligned} \mathbf{x}_i =& Drop(Norm(\mathbf{x}_i + \sum_{e\in\mathcal{N}_i} \alpha(e, r_{e,i}, i)\mathbf{x}_e)), \\ \alpha(e, r_{e,i}) =& \frac{\exp(LeakyReLU(r_{e,i}^T W[\mathbf{x}_e||\mathbf{x}_i]))}{\sum_{e\in\mathcal{N}_i}\exp(LeakyReLU(r_{e,i}^T W[\mathbf{x}_e||\mathbf{x}_i])))} \end{aligned} \tag{1}$$

In the knowledge aggregation process, $\mathcal{N}_i$ represents the neighboring entities of item i based on different types of relations $r_{e,i}$ in the knowledge graph $\mathcal{G}_k$. The embeddings of the item and entity are denoted as $\mathbf{x}_i \in \mathbb{R}^d$ and $\mathbf{x}_e \in \mathbb{R}^d$, respectively. To prevent overfitting, we apply the dropout function denoted as $Drop$, and for normalization, we use the function $Norm$. The term $\alpha(e, r_{e,i}, i)$ represents the estimated entity-specific and relation-specific attentive relevance during the knowledge aggregation process, capturing the

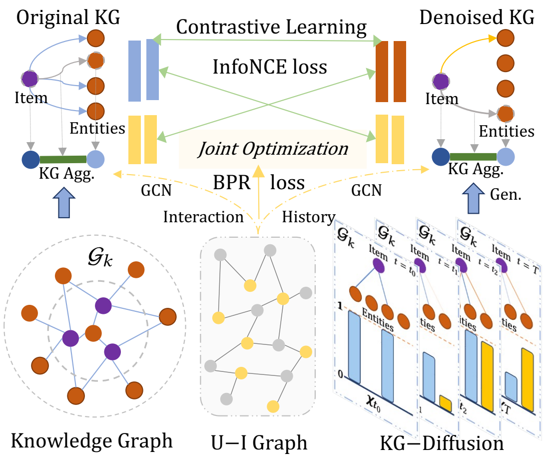


Figure 1: Overall framework of the proposed DiffKG model.

distinct semantics of relationships between item i and entity e. A parametric weight matrix $W \in \mathbb{R}^{d\times 2d}$ is customized to the input item and entity representations, and a non-linear transformation is induced using the *LeakyReLU* activation function. Notably, we incorporate random dropout operations on the knowledge graphs before heterogeneous knowledge aggregation. This is because a sparse knowledge graph inherently has the potential to significantly enhance the performance of the recommender system.

3.2 KG-enhanced Data Augmentation

Contrastive learning has recently gained remarkable success in the realm of recommendation systems. In the context of knowledge graph-enhanced recommendation, methods like KGCL [39], MCCLK [44], and KGIC [45] have introduced contrastive learning techniques. However, these approaches often rely on simplistic random augmentation methods or simplistic cross-view contrasts between the raw knowledge graph view and collaborative filtering view. Unfortunately, the random augmentation can introduce unwanted noise, and the supplementary knowledge graph view may contain irrelevant information. It is crucial to acknowledge that within the wealth of semantic relationships present in a knowledge graph, only a subset is truly relevant to the downstream recommendation task. Failing to address these irrelevant knowledge relationships can have a detrimental impact on the recommendation performance.

To tackle these challenges, we propose the use of a generative model to reconstruct a subgraph $\mathcal{G}_k^{'}$ of the knowledge graph $\mathcal{G}_k$ that specifically contains the relationships relevant to the downstream recommendation task. In Section 3.3, we will provide a detailed explanation of this generative model. Once we have constructed the task-related knowledge graph, we encode the representations of users and items using a combination of the graph-based collaborative filtering framework and heterogeneous knowledge aggregation. Taking inspiration from the effectiveness of the simplified graph convolutional network in LightGCN [6], we design our own local graph embedding propagation layer, which can be described as:

$$\mathbf{x}_u^{(l+1)} = \sum_{i\in\mathcal{N}_u} \frac{\mathbf{x}_i^{(l)}}{\sqrt{|\mathcal{N}_u|\cdot|\mathcal{N}_i|}}; \mathbf{x}_i^{(l+1)} = \sum_{u\in\mathcal{N}_i} \frac{\mathbf{x}_u^{(l)}}{\sqrt{|\mathcal{N}_i|\cdot|\mathcal{N}_u|}}, \quad (2)$$

We utilize $\mathbf{x}_u^{(l)}$ and $\mathbf{x}_i^{(l)}$ to represent the encoded representations of user u and item i at the l-th graph propagation layer. The neighboring items/users of user u/item i are denoted as $\mathcal{N}_u$ and $\mathcal{N}_i$ respectively. By employing multiple graph propagation layers, the graph-based collaborative filtering (CF) framework captures collaborative signals of higher order. In our encoding pipeline, both $\mathcal{G}_k$ and $\mathcal{G}_k^{'}$ are employed for heterogeneous knowledge aggregation, allowing us to generate input item feature vectors while preserving the semantic information of the knowledge graph. These item embeddings are subsequently fed into the graph-based CF framework to refine their representations further.

Once we have established two knowledge-enhanced graph views, we consider the view-specific embeddings of the same node as positive pairs (e.g., $(\mathbf{x}_u^{'}, \mathbf{x}_u^{''})|u \in \mathcal{U}$). On the other hand, we regard the embeddings of different nodes in the two views as negative pairs (e.g., $(\mathbf{x}_u^{'}, \mathbf{x}_v^{''})|u, v \in \mathcal{U}, u \neq v$). To formalize this, we define a contrastive loss function that aims to maximize the agreement among positive pairs and minimize the agreement among negative pairs. The contrastive loss can be expressed as follows:

$$\mathcal{L}_{cl}^{user} = \sum_{u\in\mathcal{U}} -\log \frac{\exp(s(\mathbf{x}_u^{'}, \mathbf{x}_u^{''})/\tau)}{\sum_{v\in\mathcal{U}} \exp(s(\mathbf{x}_u^{'}, \mathbf{x}_v^{''})/\tau)}, \quad (3)$$

The similarity between two vectors is measured using the cosine similarity function, denoted as $s(\cdot)$. The hyper-parameter τ, referred to as the temperature, is used in the softmax operation. We obtain the contrastive loss of the user side as $\mathcal{L}_{cl}^{user}$, and similarly, we compute the contrastive loss of the item side as $\mathcal{L}_{cl}^{item}$. By combining these two losses, we obtain the objective function for the self-supervised task, which can be represented as $\mathcal{L}cl = \mathcal{L}_{cl}^{user} + \mathcal{L}_{cl}^{item}$.

3.3 Diffusion with Knowledge Graph

Drawing inspiration from the effectiveness of diffusion models in data generation from noisy inputs, such as diffusion models presented in works like [8, 22, 31], we propose a knowledge graph diffusion model. Our purpose is to generate a recommendation-relevant subgraph $\mathcal{G}_k^{'}$ from the original knowledge graph $\mathcal{G}_k$. To achieve this, the model is trained to identify true relationships between items and entities in a knowledge graph that has been corrupted by a noise diffusion process. Our method employs a forward process that gradually introduces noise to the relations in the knowledge graph, simulating the corruption of relations. Then, through iterative learning, we aim to recover the original relations in the knowledge graph. This iterative denoising training enables DiffKG to model complex relation generation procedures and reduce the impact of noisy relations. Ultimately, the restored relation probabilities are utilized to reconstruct the subgraph $\mathcal{G}_k^{'}$ from the original knowledge graph $\mathcal{G}_k$.

3.3.1 **Noise Diffusion Process.** In Fig. 2, we can observe that our knowledge graph (KG) diffusion, similar to other diffusion models, consists of two essential processes: the forward process and the reverse process. In order to apply these processes to the KG, we represent the KG using an adjacency matrix. Specifically, let's consider an item i that has relations with entities in the entity set $\mathcal{E}$. We denote these relations as $\mathbf{z}_i = [\mathbf{z}_i^0, \mathbf{z}_i^1, \cdots, \mathbf{z}_i^{|\mathcal{E}|-1}]$, where $\mathbf{z}_i^e = 1$ or 0. This binary value indicates whether item i has a relation

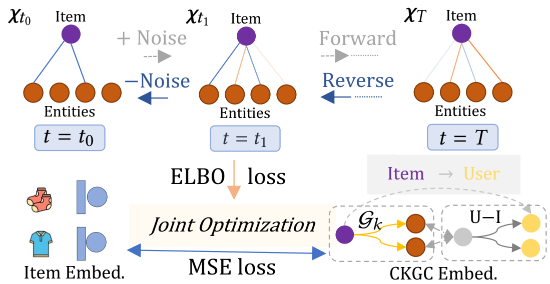


Figure 2: Diffusion Model with Knowledge Graph.

with entity e or not. In the forward process, the original structure of the knowledge graph (KG) is corrupted by adding Gaussian noises step by step. We initialize the initial state χ_0 as the original adjacency matrix $\mathbf{z}_i$ of the item. This means that $\chi_0 = \mathbf{z}_i$. The forward process then constructs $\chi_{1:T}$ in a Markov chain by gradually adding Gaussian noise in T steps. We parameterize the transition from χ_{t-1} to χ_t as:

$$q(\chi_t|\chi_{t-1}) = \mathcal{N}(\chi_t; \sqrt{1-\beta_t}\chi_{t-1}, \beta_t \boldsymbol{I}), \tag{4}$$

$t \in 1, \cdots, T$ represents the diffusion step. $\mathcal{N}$ denotes the Gaussian distribution, and $\beta_t \in (0,1)$ controls the scale of the Gaussian noise added at each step t. As $T \rightarrow \infty$, the state χ_T converges towards a standard Gaussian distribution. By utilizing the reparameterization trick and taking advantage of the additivity property of two independent Gaussian noises, we can directly derive the state χ_t from the initial state χ_0. Formally, we describe this process as follows:

$$q(\chi_t|\chi_0) = \mathcal{N}(\chi_t; \sqrt{\bar{\alpha}_t}\chi_0, (1-\bar{\alpha}_t)\boldsymbol{I}), \bar{\alpha}_t = \prod_{t'=1}^{t}(1-\beta_{t'}). \tag{5}$$

χ_t can be reparameterized as follows:

$$\chi_t = \sqrt{\bar{\alpha}_t}\chi_0 + \sqrt{1-\bar{\alpha}_t}\boldsymbol{\epsilon}, \boldsymbol{\epsilon} \sim \mathcal{N}(0, \boldsymbol{I}). \tag{6}$$

To regulate the addition of noises in $\chi_{1:T}$, we incorporate a linear noise scheduler that implements $1-\bar{\alpha}t$ using three hyperparameters: s, αlow, and αup. The linear noise scheduler is defined as follows:

$$1-\bar{\alpha}_t = s \cdot \left[\alpha_{low} + \frac{t-1}{T-1}(\alpha_{up} - \alpha_{low})\right], t \in \{1, \cdots, T\}. \tag{7}$$

The linear noise scheduler uses three hyperparameters: $s \in [0,1]$ controls the noise scales, while $\alpha_{low} < \alpha_{up} \in (0,1)$ set the upper and lower bounds for the added noises.

Next, the diffusion model learns to remove the added noises from χ_t in order to recover χ_{t-1} using neural networks. Starting from χ_T, the reverse process gradually reconstructs the relations within the knowledge graph (KG) through the denoising transition step. The denoising transition step is outlined as follows:

$$p_\theta(\chi_{t-1}|\chi_t) = \mathcal{N}(\chi_{t-1}; \boldsymbol{\mu}_\theta(\chi_t, t), \Sigma_\theta(\chi_t, t)). \tag{8}$$

We utilize neural networks parameterized by θ to generate the mean $\boldsymbol{\mu}\theta(\chi_t, t)$ and covariance $\Sigma\theta(\chi_t, t)$ of a Gaussian distribution.

3.3.2 **Optimization of KG Diffusion Process.** To optimize our model, we maximize the Evidence Lower Bound (ELBO) of the likelihood of the original knowledge graph relations χ_0. Following the approach described in [31], we can summarize the optimization objective of our probabilistic diffusion process as follows:

$$\begin{aligned}\log p(\chi_0) \geq\ & \mathbb{E}_{q(\chi_1|\chi_0)}[\log p_\theta(\chi_0|\chi_1)] \\ & - \sum_{t=2}^{T}\mathbb{E}_{q(\chi_t|\chi_0)}[D_{KL}(q(\chi_{t-1}|\chi_t,\chi_0)||p_\theta(\chi_{t-1}|\chi_t))].\end{aligned} \tag{9}$$

The optimization objective of diffusion model consists of two terms. The first term measures the recovery probability of χ_0, representing the ability of the model to reconstruct the original knowledge graph. The second term regulates the recovery of χ_{t-1} for t ranging from 2 to T in the reverse process.

The second term in the optimization objective aims to make the distribution $p_\theta(\chi_{t-1}|\chi_t)$ approximate the tractable distribution $q(\chi_{t-1}|\chi_t, \chi_0)$ through the KL divergence $D_{KL}(\cdot)$. Following [31], the second term $\mathcal{L}_t$ at step t is as follows:

$$\mathcal{L}_t = \mathbb{E}_{q(\chi_t|\chi_0)}\left[\frac{1}{2}\left(\frac{\bar{\alpha}_{t-1}}{1-\bar{\alpha}_{t-1}} - \frac{\bar{\alpha}_t}{1-\bar{\alpha}_t}\right)||\hat{\chi}_\theta(\chi_t, t) - \chi_0||_2^2\right], \tag{10}$$

where $\hat{\chi}_\theta(\chi_t, t)$ is the predicted χ_0 based on χ_t and t. To calculate Eq. 10, we implement $\hat{\chi}_\theta(\chi_t, t)$ by neural networks. Specifically, we instantiate $\hat{\chi}_\theta(\cdot)$ via a Multi-Layer Perceptron (MLP) that takes χ_t and the step embedding of t as inputs to predict χ_0.

For the first term, we use $\mathcal{L}_{first}$ to denote the negative of the first term in Eq. 9 and it can be calculate as follows:

$$\begin{aligned}\mathcal{L}_{first} &\triangleq -\mathbb{E}_{q(\chi_1|\chi_0)}[\log p_\theta(\chi_0|\chi_1)] \\ &= \mathbb{E}_{q(\chi_1|\chi_0)}\left[||\hat{\chi}_\theta(\chi_1, 1) - \chi_0||_2^2\right],\end{aligned} \tag{11}$$

where we estimate the Gaussian log-likelihood log $p(\chi_0|\chi_1)$ by unweighted $-||\hat{\chi}_\theta(\chi_1, 1) - \chi_0||_2^2$. It is easy to find that $\mathcal{L}_{first}$ is equal to $\mathcal{L}_1$ based on Eq. 10. Therefore, the first term in Eq. 9 can be considered as $-\mathcal{L}_1$.

According to Eq. 10, ELBO in Eq. 9 can be formulated as $-\mathcal{L}_1 - \sum_{t=2}^{T}\mathcal{L}_t$. Hence, to maximize the ELBO, we can optimize θ in $\hat{\chi}_\theta(\chi_t, t)$ by minimizing $\sum_{t=1}^{T}\mathcal{L}_t$. Specifically, we uniformly sample step t to optimize $\mathcal{L}_{elbo}$ over $t \sim \mathcal{U}(1, T)$. Formally, the ELBO loss $\mathcal{L}_{elbo}$ is shown below:

$$\mathcal{L}_{elbo} = \mathbb{E}_{t\sim\mathcal{U}(1,T)}\mathcal{L}_t. \tag{12}$$

3.3.3 **Knowledge Graph Generation with Diffusion Model.** In contrast to other diffusion models that randomly draw Gaussian noises for reverse generation, we have designed a simple inference strategy that aligns with the training of DiffKG for relation prediction in knowledge graphs (KGs). This strategy avoids corrupting the KG with pure noises, as doing so would severely compromise the informative structure of the KG.

In our inference strategy, we begin by corrupting the original KG relations χ_0 in a step-by-step manner during the forward process, resulting in $\chi_{T'}$. We then set $\hat{\chi}_T = \chi_{T'}$ and perform reverse denoising, where we ignore the variance and use $\hat{\chi}_{t-1} = \mu\theta(\hat{\chi}_t, t)$ for deterministic inference. Next, we reconstruct the structure of the modified KG $\mathcal{G}_k'$ using $\hat{\chi}_0$. For each item i, we select the top k $\hat{z}_i^j$ ($j \in [0, |\mathcal{E}|-1]$, $j \in \mathcal{J}$, and $|\mathcal{J}| = k$) and add k relations between item i and entities $j \in \mathcal{J}$. It aims to preserve the informative structure of the KG while incorporating noise during the forward process and deterministic inference during the reverse process.

3.3.4 Collaborative Knowledge Graph Convolution.

To mitigate the potential limitations of the diffusion model in generating a denoised knowledge graph that encompasses pertinent relationships for downstream recommendation tasks, we propose a collaborative knowledge graph convolution (CKGC) mechanism. This novel approach capitalizes on the user-item interaction data to assimilate supervisory signals from recommendation tasks into the optimization of KG diffusion. Through the aggregation of user-item interaction data, our method enhances the model's capacity to capture user preferences and seamlessly incorporates them into the denoised knowledge graph, thereby enhancing its relevance to recommendation tasks. This amalgamation of user preferences introduces a valuable dimension to the optimization process of KG diffusion, effectively bridging the divide between knowledge graph denoising and recommendation tasks.

The loss of collaborative knowledge graph convolution, denoted as $\mathcal{L}_{ckgc}$, is computed by incorporating user-item interaction information and knowledge graph predictions into the item embedding generation process. Specifically, we begin by aggregating the user-item interaction information $\mathcal{A}$ with the predicted relation probabilities from the knowledge graph, represented as $\hat{\chi}_0$. This aggregation updates the user-item interaction matrix, effectively integrating the knowledge graph information. Next, we combine this updated user-item matrix with the user embeddings $\mathbf{E}_u$ to obtain an item embedding $\mathbf{E}_i^{'}$ that jointly incorporates both the knowledge graph and user information. Finally, we calculate the mean squared error (MSE) loss between the aggregated item embedding $\mathbf{E}_i^{'}$ and the original item embedding $\mathbf{E}_i$, and optimize it alongside the ELBO loss ($\mathcal{L}_{elbo}$). The formal expression for the loss $\mathcal{L}_{ckgc}$ is as follows:

$$\mathcal{L}_{ckgc} = \left\| \left[\mathcal{A} \cdot \hat{\chi}_0^{\top} \right]^{\top} \cdot \mathbf{E}_u - \mathbf{E}_i \right\|_2^2 \tag{13}$$

3.4 The Learning Process of DiffKG

The training of our DiffKG consists of two primary components: training for the recommendation task and training for KG diffusion. The joint training of KG diffusion encompasses two loss components: the ELBO loss and the CKGC loss, which are optimized simultaneously. As a result, the loss function for KG diffusion can be expressed as follows:

$$\mathcal{L}_{kgdm} = (1 - \lambda_0)\mathcal{L}_{elbo} + \lambda_0 \mathcal{L}_{ckgc}, \tag{14}$$

To balance the contributions of the ELBO loss and the CKGC loss, we introduce a hyperparameter λ_0 that controls their respective strengths. For the recommendation task, we incorporate the original Bayesian personalized ranking (BPR) recommendation loss along with the contrastive loss $\mathcal{L}_{cl}$ mentioned earlier. The BPR loss, denoted as $\mathcal{L}_{bpr}$, is defined as follows:

$$\mathcal{L}_{bpr} = \sum_{(u,i,j) \in O} -\log\sigma(\hat{y}_{ui} - \hat{y}_{uj}), \tag{15}$$

The training data is represented as $O = (u, i, j)|(u, i) \in O^+, (u, j) \in O^-$, where O^+ denotes the observed interactions and O^- represents the unobserved interactions obtained from the Cartesian product of user set $\mathcal{U}$ and item set $\mathcal{I}$ excluding O^+. With these definitions, the integrative optimization loss for the recommendation task is:

$$\mathcal{L}_{rec} = \mathcal{L}_{bpr} + \lambda_1 \mathcal{L}_{cl} + \lambda_2 ||\theta||_2^2, \tag{16}$$

Table 1: Statistics of the experimental datasets.

Statistics	Last-FM	MIND	Alibaba-iFashion
# Users	23,566	100,000	114,737
# Items	48,123	30,577	30,040
# Interactions	3,034,796	2,975,319	1,781,093
# Density	2.7e-3	9.7e-4	5.2e-4
Knowledge Graph			
# Entities	58,266	24,733	59,156
# Relations	9	512	51
# Triplets	464,567	148,568	279,155

The learnable model parameters are denoted as Θ, which encompasses the trainable variables within the model. Additionally, λ_1 and λ_2 are hyperparameters that determine the respective strengths of the CL-based loss and the L_2 regularization term.

4 EXPERIMENTS

To evaluate the effectiveness of our DiffKG, we have designed a series of experiments to address the following research questions:

- **RQ1**: How does the performance of our DiffKG compare to a diverse range of state-of-the-art recommendation systems?
- **RQ2**: What distinct contributions do the key components of our DiffKG offer to the overall performance? Additionally, how does the model's performance adapt and respond to variations in hyperparameter settings?
- **RQ3**: How does our proposed DiffKG demonstrate its effectiveness in overcoming the obstacles of data sparsity and noise?
- **RQ4**: To what degree does our proposed DiffKG model provide a high level of interpretability for recommendation, facilitating a thorough comprehension of its decision-making process?

4.1 Experimental Settings

4.1.1 Dataset.

To ensure a comprehensive and diverse evaluation, we have incorporated three distinct public datasets that represent different real-life scenarios: Last-FM (music), MIND (news), and Alibaba-iFashion (e-commerce). To preprocess the data, we have applied the 10-core technique, filtering out users and items with occurrence counts below 10. For the Last-FM dataset, we have employed a mapping approach to associate the items with Freebase entities and extract knowledge triplets, following methodologies inspired by [32] and [43]. In the case of the MIND dataset, we have followed the practices outlined in [24] to collect the knowledge graph (KG) from Wikidata, focusing on representative entities within the original data. As for the Alibaba-iFashion dataset, we have manually constructed the KG, utilizing the category information as valuable knowledge [33]. Detailed statistics for the three datasets and their corresponding KGs can be found in Table 1.

4.1.2 Evaluation Protocols.

To avoid bias from negative sampling in evaluation [12], we report performance metrics under the full-rank setting, as done in the research works [18, 32, 33]. We utilize Recall@N and NDCG@N as top-N recommendation metrics, with N=20, a commonly used value [7, 32].

4.1.3 Compared Baseline Methods.

For a comprehensive evaluation, we thoroughly compare our DiffKG with a diverse set of baselines derived from different research streams.

Collaborative Filtering Methods.

- **BPR** [21]: This method effectively utilizes pairwise ranking loss derived from implicit feedback for matrix factorization.
- **NeuMF** [7]: It incorporates MLP into matrix factorization and learns enriched user and item representations while capturing the feature interactions between them.
- **GC-MC** [1]: By proposing a graph auto-encoder, GC-MC predicts unknown ratings by exploiting the underlying graph structure.
- **LightGCN** [6]: Conducting an in-depth analysis of modules within standard GCN for collaborative data, LightGCN proposes a simplified GCN model tailored specifically for graph CF task.
- **SGL** [37]: SGL introduces data augmentation techniques such as random walk and feature dropout to generate multiple views.

Embedding-based Knowledge-aware Recommenders.

- **CKE** [42]: By integrating collaborative filtering and KG embeddings, CKE empowers the recommendation system with a deeper understanding of item relationships.
- **KTUP** [2]: This approach enables mutual complementation between collaborative filtering and knowledge graph signals, allowing for a more comprehensive recommendation process.

GNN-based KG-enhanced Recommenders.

- **KGNN-LS** [29]: KGNN-LS considers user preferences towards different knowledge triplets in graph convolution. It introduces label-smoothing as regularization to encourage similar user preference weights between closely related items in the KG.
- **KGCN** [30]: It aggregates knowledge for item representations by incorporating high-order information using GNNs.
- **KGAT** [32]: It introduces the concept of collaborative KG to apply attentive aggregation on the joint user-item-entity graph.
- **KGIN** [33]: It models user intents for relations and employs relational path-aware aggregation to capture rich information from the composite knowledge graph.

Self-Supervised Knowledge-aware Recommenders.

- **MCCLK** [44]: It employs contrastive learning in a hierarchical manner. It aims to mine useful structural information from the user-item-entity graph and its subgraphs.
- **KGCL** [39]: By leveraging self-supervised learning, KGCL effectively incorporates KG information while addressing noise and improving recommendation accuracy.

4.2 RQ1: Overall Performance Comparison

We have evaluated the overall performance of all the methods, and the results are summarized in Table 2. Based on the findings, we have made the following observations:

- The performance evaluation of all methods consistently demonstrates that our proposed DiffKG outperforms all baseline approaches. This highlights the effectiveness of our DiffKG in enhancing recommendations with task-relevant KG signals. Specifically, our carefully designed graph diffusion model serves as a powerful graph generator, producing knowledge graphs that incorporate task-specific entity relationships. This enriched knowledge graph enhances the effectiveness of data augmentation, resulting in improved recommendation accuracy.

Table 2: Performance comparison on Last-FM, MIND, Alibaba-iFashion datasets in terms of *Recall@20* and *NDCG@20*.

Model	Last-FM		MIND		Alibaba-iFashion	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
BPR	0.0690	0.0585	0.0384	0.0253	0.0822	0.0501
NeuMF	0.0699	0.0615	0.0308	0.0237	0.0506	0.0276
GC-MC	0.0709	0.0631	0.0386	0.0261	0.0845	0.0502
LightGCN	0.0738	0.0647	0.0419	0.0253	0.1058	0.0652
SGL	0.0879	0.0775	<u>0.0429</u>	0.0275	0.1141	0.0713
CKE	0.0845	0.0718	0.0387	0.0247	0.0835	0.0512
KTUP	0.0865	0.0671	0.0362	0.0302	0.0976	0.0634
KGNN-LS	0.0881	0.0690	0.0395	0.0302	0.0983	0.0633
KGCN	0.0879	0.0694	0.0396	<u>0.0302</u>	0.0983	0.0633
KGAT	0.0870	0.0743	0.0340	0.0287	0.0957	0.0577
KGIN	0.0900	<u>0.0779</u>	0.0357	0.0225	0.1144	<u>0.0723</u>
MCCLK	0.0671	0.0603	0.0327	0.0194	0.1089	0.0707
KGCL	<u>0.0905</u>	0.0769	0.0399	0.0247	<u>0.1146</u>	0.0719
DiffKG	**0.0980**	**0.0911**	**0.0615**	**0.0389**	**0.1234**	**0.0773**

- The performance evaluation clearly demonstrates the superiority of knowledge-aware recommenders that incorporate knowledge graph information compared to traditional approaches like BPR and NeuMF. This highlights the valuable role of knowledge graphs in addressing the sparsity issue inherent in collaborative filtering. The noticeable performance gap between our DiffKG and other knowledge-aware models, such as KGAT, KGIN, and KGCL, suggests that knowledge graphs often contain irrelevant relations that can negatively impact recommendation quality.
- The comparative performance of KGCL highlights the effectiveness of incorporating KG-based item semantic relatedness and leveraging self-supervised signals to explicitly address the interaction sparsity issue. KGCL focuses on augmenting the user-item interaction matrix with KG guidance, while our DiffKG takes a different approach by utilizing a task-related knowledge graph generated through our designed KG diffusion model.

4.3 RQ2: Ablation Study

4.3.1 **Key Module Ablation.** This study aims to evaluate the effectiveness of the key modules incorporated in our proposed DiffKG. To establish a comparative analysis with the original method, we have developed three distinct model variants, which are outlined:

- "w/o CL": This variant involves the removal of the KG-enhanced data augmentation module in recommendation.
- "w/o DM": We replace our diffusion model with variational graph autoencoder, which is a widely-used generative model.
- "w/o CKGC": This variant excludes the collaborative knowledge graph convolution from the KG diffusion model optimization.

The ablation study results, as presented in Table 3, yield important insights, leading to the following key conclusions: i) Removal of KG-enhanced contrastive learning results in significant performance degradation across all cases. This finding validates the effectiveness of incorporating additional self-supervised signals using the knowledge graph. ii) Ablation of the knowledge graph diffusion model component demonstrates its crucial role in improving the performance of our DiffKG. In all cases, the inclusion of our designed diffusion model contributes to better results, affirming

Table 3: Ablation study on key components of DiffKG.

Ablation Settings	Last-FM		MIND		Alibaba-iFashion	
	Recall	NDCG	Recall	NDCG	Recall	NDCG
DiffKG	**0.0980**	**0.0911**	**0.0615**	**0.0389**	**0.1234**	**0.0773**
DiffKG w/o CL	0.0737	0.0638	0.0398	0.0246	0.1126	0.0700
DiffKG w/o DM	0.0961	0.0891	0.0573	0.0358	0.1218	0.0767
DiffKG w/o CKGC	0.0976	0.0905	0.0605	0.0378	0.1228	0.0770

the effectiveness of capturing task-relevant relations through the diffusion process. Notably, the larger performance drop observed in Last-FM and MIND datasets suggests a higher level of noise present in their respective knowledge graphs. iii) The absence of the collaborative knowledge graph convolution module leads to performance degradation across all cases. This underscores the significance of collaborative knowledge graph convolution in our DiffKG, as it facilitates the integration of user collaborative knowledge into the training of the diffusion model for recommendation.

4.3.2 **Sensitivity to Key Hyperparameters.** In this study, we focus on examining the effects of different hyperparameters on our method. Specifically, we conduct a thorough analysis of hyperparameters in both the data augmentation and knowledge graph diffusion modules. To present our findings, we report the corresponding results on the MIND dataset, as demonstrated in Figure 3.

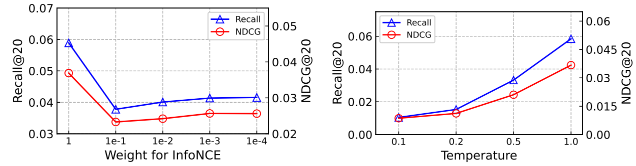
We thoroughly analyzed hyperparameters for our DiffKG, specifically focusing on λ_1 (InfoNCE loss weight) and τ (softmax temperature). Figure 3(a) showcased the best performance with $\lambda_1 = 1$ and $\tau = 1$, emphasizing the significance of CL. Additionally, in the knowledge graph diffusion model, Figure 3(b) demonstrated minimal accuracy impact when increasing diffusion steps due to low noise levels. We selected $T = 5$ to balance performance and computation. Notably, the best performance was achieved with $T^{'} = 0$ to avoid excessive corruption of the original KG.

4.4 RQ3: Further Investigation on DiffKG

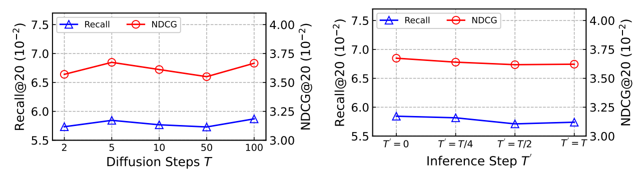
Sparse User Interaction Data. In order to assess the performance of our DiffKG in handling sparse data, we conducted an evaluation on both users and items. For users, we divided them into five groups, each containing an equal number of users. The interaction density within these groups gradually increased from Group 1 to Group 5, representing varying levels of sparsity. A similar approach was employed for processing the items. The test results from this evaluation are presented in Figure 4.

Knowledge Graph Noise. To assess our DiffKG's ability to filter out irrelevant relations from the KG, we injected noisy triplets into the data and compared its performance with other knowledge-aware recommender systems. Specifically, we randomly added 10% noisy triplets to the existing KG while keeping the test set unchanged, simulating a scenario with a large number of topic-irrelevant relations. The test results can be found in Fig. 5.

- The evaluation of sparse data recommendation clearly demonstrates the superior performance of our DiffKG compared to KGCL. This notable improvement serves as strong evidence of the effectiveness of our KG-enhanced data augmentation. It effectively tackles the challenge posed by task-irrelevant relations within the KG, which have the potential to mislead the encoding of user preferences in the recommendation process.

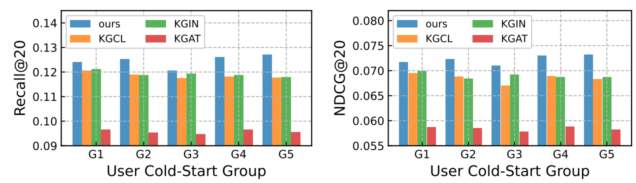


(a) Hyperparameter Analysis on Data Augmentation

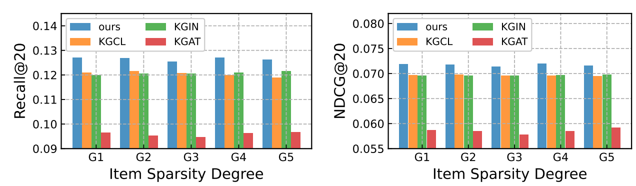


(b) Hyperparameter Analysis on Diffusion Model

Figure 3: Hyperparameter Analysis on MIND Dataset.



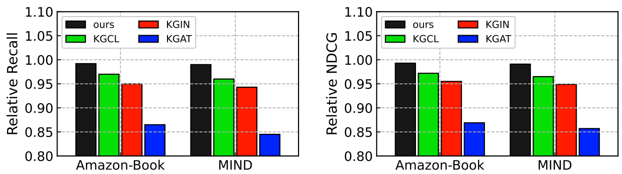
(a) Performance *w.r.t.* cold-start user groups



(b) Performance *w.r.t.* sparse item groups

Figure 4: Performance *w.r.t* different data sparsity degrees.

- In the recommendation scenario with long-tail item distributions, our DiffKG significantly improves recommendation performance for such items. This highlights its effectiveness in mitigating popularity bias, as other baseline methods tend to neglect less popular items. Additionally, our DiffKG outperforms competitive KG-aware recommendation systems like KGAT and KGIN. This suggests that blindly incorporating all KG information into collaborative filtering may introduce noise from irrelevant item relations and fail to alleviate popularity bias effectively.
- Among the various knowledge-aware recommendation models, our DiffKG consistently achieves the highest performance. This can be attributed to the task-specific knowledge graph generated by the diffusion model. Notably, DiffKG demonstrates the most effective noise alleviation, as evidenced by the lowest average performance decrease in the presence of KG noise, as depicted in Figure 5. This serves as compelling evidence of the remarkable ability of our DiffKG to discover relevant information from a noisy KG, effectively supporting user preference modeling.



(a) Relative Recall (b) Relative NDCG

Figure 5: Performance in alleviating KG noise.

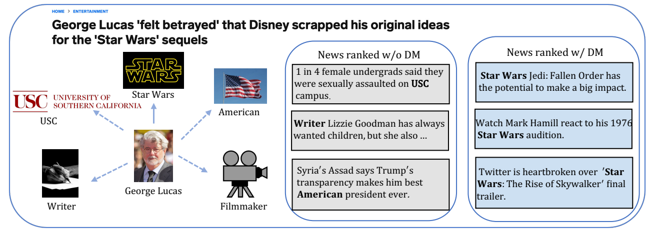


Figure 6: Relevant News w/ and w/o KG diffusion.

4.5 RQ4: Case Study

We performed a case study on news recommendation, comparing the results with and without our knowledge graph diffusion model. The findings are shown in Figure 6, highlighting the impact of our KG diffusion on recommendation accuracy. We examine the assessment of the Star Wars sequel by renowned filmmaker George Lucas and its relevance to the provided KG information. The KG includes entities such as "American," "USC," "writer," and "filmmaker," which are unrelated to the news at hand. This noise in the KG can introduce bias and misguide user representation. Without the knowledge graph diffusion model, the model ranks unrelated news articles covering topics such as "USC," "Lizzie Goodman" (a writer), and "Syria." However, with the integration of KG diffusion paradigm, our DiffKG effectively filters out irrelevant KG information, resulting in more pertinent news articles. These articles include a discussion on a Star Wars video game, an actor's involvement in the Star Wars film, and social media commentary on the Star Wars movie. By accurately leveraging and filtering KG information, our model demonstrates improved performance in recommendation tasks, illustrating its effectiveness in enhancing relevance and mitigating the impact of irrelevant information in the KG.

5 RELATED WORK

5.1 Knowledge-aware Recommender Systems

Existing knowledge-aware recommendation methods can be categorized into embedding-based, path-based, and GNN-based approaches. GNN-based methods, such as KGCN [30], KGAT [32], and KGIN [33], combine the strengths of both paradigms and effectively extract valuable information from the knowledge graph. KGCN utilizes a fixed number of neighbors for item representation aggregation, while KGAT employs Graph Attention Networks (GATs) to assign weights based on the importance of knowledge neighbors. KGIN incorporates user preferences and relational embeddings in the aggregation layer. These GNN-based methods enhance recommendation systems by leveraging the power of GNNs and the rich information in the knowledge graph [29, 30, 32, 33].

5.2 Data Augmentation for Recommendation

Data augmentation techniques, combined with self-supervised learning (SSL), have emerged as a promising approach to enhance recommendation systems. By leveraging additional supervision signals extracted from raw data, SSL-based data augmentation methods can address data sparsity and improve recommendation performance [3, 37]. Contrastive learning-based data augmentation methods, such as those proposed in [35, 37], generate augmented views of user or item representations. By training models to differentiate between positive and negative pairs [13, 38], these methods effectively addresses data sparsity and enhances recommendation performance through self-supervised learning. Additionally, inspired by natural language processing tasks like BERT, masking and reconstruction augmentation techniques involve masking or hiding certain items or parts of user-item interactions and training the model to predict the missing elements. This process forces the model to learn contextual relationships in the recommendation process [20, 23]. By incorporating SSL-based data augmentation techniques into recommendation systems, models can address data sparsity, capture complex patterns, and improve the generalization ability of recommender systems.

5.3 Diffusion Probabilistic Models

Diffusion probabilistic models have gained considerable attention and showcased great potential in a range of fields, spanning computer vision and natural language processing. In the context of vision, diffusion models have been particularly effective in tasks such as image generation [5, 9] and inpainting [16]. In the context of text generation, a generative model is trained to recover the original text from the perturbed data [4, 14]. In addition, diffusion models have also found application in diverse domains, including graph learning for the purpose of graph generation. For example, GraphGDP [10] proposes a continuous-time generative diffusion process for permutation invariant graph generation. Digress [26] employs a discrete denoising diffusion model that utilizes a graph transformer network to iteratively modify graphs with noise, resulting in the generation of graphs. Recently, diffusion probabilistic models have also been explored in the realm of recommendation [31].

6 CONCLUSION

This research introduces DiffKG, a novel recommendation model that leverages task-specific item knowledge to enhance the collaborative filtering paradigm. The framework proposes a unique methodology for extracting high-quality signals from noisy knowledge graphs. By seamlessly integrating a generative diffusion model with a knowledge graph learning framework tailored for knowledge-aware recommender systems, the model effectively aligns the semantic aspects of knowledge-enhanced items with collaborative relation modeling, resulting in precise recommendations. Through extensive evaluations on diverse benchmark datasets, our proposed DiffKG framework demonstrates significant performance improvements compared to various baseline models. Furthermore, our approach effectively addresses the challenge of noisy data, which is known to impede the accuracy of recommender systems.

REFERENCES

[1] Rianne van den Berg, Thomas N Kipf, and Max Welling. 2017. Graph convolutional matrix completion. *arXiv preprint arXiv:1706.02263* (2017).

[2] 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 *WWW*. 151–161.

[3] Mengru Chen, Chao Huang, Lianghao Xia, Wei Wei, Yong Xu, and Ronghua Luo. 2023. Heterogeneous graph contrastive learning for recommendation. In *WSDM*. 544–552.

[4] Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2022. DiffuSeq: Sequence to Sequence Text Generation with Diffusion Models. In *ICLR*.

[5] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. 2022. Vector quantized diffusion model for text-to-image synthesis. In *CVPR*. 10696–10706.

[6] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *SIGIR*. 639–648.

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

[8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. 2020. Denoising diffusion probabilistic models. *NeurIPS* 33 (2020), 6840–6851.

[9] Jonathan Ho, Chitwan Saharia, William Chan, David J Fleet, Mohammad Norouzi, and Tim Salimans. 2022. Cascaded diffusion models for high fidelity image generation. *JMLR* 23, 1 (2022), 2249–2281.

[10] Han Huang, Leilei Sun, Bowen Du, Yanjie Fu, and Weifeng Lv. 2022. Graphgdp: Generative diffusion processes for permutation invariant graph generation. In *ICDM*. IEEE, 201–210.

[11] Yehuda Koren, Steffen Rendle, and Robert Bell. 2021. Advances in collaborative filtering. *Recommender systems handbook* (2021), 91–142.

[12] Walid Krichene and Steffen Rendle. 2020. On sampled metrics for item recommendation. In *KDD*. 1748–1757.

[13] Xuewei Li, Aitong Sun, Mankun Zhao, Jian Yu, Kun Zhu, Di Jin, Mei Yu, and Ruiguo Yu. 2023. Multi-Intention Oriented Contrastive Learning for Sequential Recommendation. In *WSDM*. 411–419.

[14] Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *NeurIPS* 35 (2022), 4328–4343.

[15] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. 2018. Variational autoencoders for collaborative filtering. In *WWW*. 689–698.

[16] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. 2022. Repaint: Inpainting using denoising diffusion probabilistic models. In *CVPR*. 11461–11471.

[17] Jay Pujara, Eriq Augustine, and Lise Getoor. 2017. Sparsity and noise: Where knowledge graph embeddings fall short. In *EMNLP*. 1751–1756.

[18] Xubin Ren, Wei Wei, Lianghao Xia, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. Representation Learning with Large Language Models for Recommendation. *arXiv preprint arXiv:2310.15950* (2023).

[19] Xubin Ren, Lianghao Xia, Jiashu Zhao, Dawei Yin, and Chao Huang. 2023. Disentangled Contrastive Collaborative Filtering. *arXiv preprint arXiv:2305.02759* (2023).

[20] Yuyang Ren, Zhang Haonan, Luoyi Fu, Xinbing Wang, and Chenghu Zhou. 2023. Distillation-Enhanced Graph Masked Autoencoders for Bundle Recommendation. In *SIGIR*. 1660–1669.

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

[22] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. 2015. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*. PMLR, 2256–2265.

[23] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *CIKM*. 1441–1450.

[24] Yu Tian, Yuhao Yang, Xudong Ren, Pengfei Wang, Fangzhao Wu, Qian Wang, and Chenliang Li. 2021. Joint knowledge pruning and recurrent graph convolution for news recommendation. In *SIGIR*. 51–60.

[25] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[26] Clément Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. 2023. DiGress: Discrete Denoising diffusion for graph generation. In *ICLR*.

[27] Guanying Wang, Wen Zhang, Ruoxu Wang, Yalin Zhou, Xi Chen, Wei Zhang, Hai Zhu, and Huajun Chen. 2018. Label-free distant supervision for relation extraction via knowledge graph embedding. In *EMNLP*. 2246–2255.

[28] Hongwei Wang, Fuzheng Zhang, Xing Xie, and Minyi Guo. 2018. DKN: Deep knowledge-aware network for news recommendation. In *WWW*. 1835–1844.

[29] Hongwei Wang, Fuzheng Zhang, Mengdi Zhang, Jure Leskovec, Miao Zhao, Wenjie Li, and Zhongyuan Wang. 2019. Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *KDD*. 968–977.

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

[31] Wenjie Wang, Yiyan Xu, Fuli Feng, Xinyu Lin, Xiangnan He, and Tat-Seng Chua. 2023. Diffusion Recommender Model. In *SIGIR*.

[32] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. 2019. Kgat: Knowledge graph attention network for recommendation. In *KDD*. 950–958.

[33] Xiang Wang, Tinglin Huang, Dingxian Wang, Yancheng Yuan, Zhenguang Liu, Xiangnan He, and Tat-Seng Chua. 2021. Learning intents behind interactions with knowledge graph for recommendation. In *WWW*. 878–887.

[34] Xiang Wang, Dingxian Wang, Canran Xu, Xiangnan He, Yixin Cao, and Tat-Seng Chua. 2019. Explainable reasoning over knowledge graphs for recommendation. In *AAAI*, Vol. 33. 5329–5336.

[35] Wei Wei, Chao Huang, Lianghao Xia, and Chuxu Zhang. 2023. Multi-Modal Self-Supervised Learning for Recommendation. In *WWW*. 790–800.

[36] Wei Wei, Xubin Ren, Jiabin Tang, Qinyong Wang, Lixin Su, Suqi Cheng, Junfeng Wang, Dawei Yin, and Chao Huang. 2023. LLMRec: Large Language Models with Graph Augmentation for Recommendation. *arXiv preprint arXiv:2311.00423* (2023).

[37] Jiancan Wu, Xiang Wang, Fuli Feng, Xiangnan He, Liang Chen, Jianxun Lian, and Xing Xie. 2021. Self-supervised graph learning for recommendation. In *SIGIR*. 726–735.

[38] Yuhao Yang, Chao Huang, Lianghao Xia, Chunzhen Huang, Da Luo, and Kangyi Lin. 2023. Debiased Contrastive Learning for Sequential Recommendation. In *WWW*. 1063–1073.

[39] Yuhao Yang, Chao Huang, Lianghao Xia, and Chenliang Li. 2022. Knowledge graph contrastive learning for recommendation. In *SIGIR*. 1434–1443.

[40] Tiansheng Yao, Xinyang Yi, Derek Zhiyuan Cheng, Felix Yu, Ting Chen, Aditya Menon, Lichan Hong, Ed H Chi, Steve Tjoa, Jieqi Kang, et al. 2021. Self-supervised learning for large-scale item recommendations. In *CIKM*. 4321–4330.

[41] Xiao Yu, Xiang Ren, Yizhou Sun, Quanquan Gu, Bradley Sturt, Urvashi Khandelwal, Brandon Norick, and Jiawei Han. 2014. Personalized entity recommendation: A heterogeneous information network approach. In *WSDM*. 283–292.

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

[43] Wayne Xin Zhao, Gaole He, Kunlin Yang, Hongjian Dou, Jin Huang, Siqi Ouyang, and Ji-Rong Wen. 2019. Kb4rec: A data set for linking knowledge bases with recommender systems. *Data Intelligence* 1, 2 (2019), 121–136.

[44] Ding Zou, Wei Wei, Xian-Ling Mao, Ziyang Wang, Minghui Qiu, Feida Zhu, and Xin Cao. 2022. Multi-level cross-view contrastive learning for knowledge-aware recommender system. In *SIGIR*. 1358–1368.

[45] Ding Zou, Wei Wei, Ziyang Wang, Xian-Ling Mao, Feida Zhu, Rui Fang, and Dangyang Chen. 2022. Improving knowledge-aware recommendation with multi-level interactive contrastive learning. In *CIKM*. 2817–2826.