# A Survey on Knowledge Graph-Based Click-Through Rate Prediction

Ying Jin[a], Yanwu Yang[a]
[a]School of Management, Huazhong University of Science and Technology, Wuhan 43004, China,
{jinying.isec, yangyanwu.isec}@gmail.com

**Abstract:** With the rapid development of the Internet era, accurate click-through rates (CTR) prediction is crucial for optimizing recommender systems. Existing CTR prediction models often encounter challenges related to data sparsity and cold start problems. Knowledge graph (KG), with rich semantic and structural relationships, has shown great potential in addressing these issues. In this paper, we provide a comprehensive literature review of recent advances in CTR prediction based on KG. Specifically, we categorize various approaches into embedding-based, path-based and propagation-based models and discuss their advantages and limitations. We further summarize evaluation metrics and datasets related to this research domain and discuss the performance of each model on these datasets. Moreover, we identify prevailing research trends, primary challenges and promising future research directions. This literature review aims to offer insights into the advancements made in KG-based CTR prediction and provide a foundation for future research in this area.
**Keywords:** click-through rate, knowledge graph, CTR Prediction, prediction models

## 1. Introduction

In recent years, users have been inundated with vast amounts of information, making it increasingly arduous for them to locate content that is both pertinent and tailored to their specific preferences. Recommender systems have become instrumental in alleviating information overload by filtering and suggesting personalized content based on user preferences. Among various techniques employed in recommender systems, Click-Through Rate (CTR) aims to estimate the probability of a user clicking on a recommended item. Accurate CTR prediction not only delivers personalized content that resonates with users, significantly shortening the user's information acquisition time but also has a positive impact on users' purchase intentions, enhancing revenue for businesses (Li et al., 2023c; Wang et al., 2019c).

CTR prediction heavily relies on historical user-item interactions. Nevertheless, both users and items undergo frequent updates in practical scenarios. Consequently, data sparsity arises when there are limited historical interactions for certain users or items. The more extreme cold-start problem arises under the circumstances of rare data available for new users or items un-interacted. It is worth noting that data sparsity and cold-start problems are still stubborn obstacles, hindering the model from effectively learning from the limited information. Additionally, existing CTR prediction models tend to neglect the semantics and intricate relationships between various entities, leading to an inadequate exploration of underlying factors that motivate a user to click on a spacific item. To solve these issues, the integration of knowledge graph (KG) has emerged as a promising approach to enhance CTR prediction. KG provides a powerful framework to organize and store information in a semantic and structural manner. KG has been widely utilized in various domains, such as recommender systems and question answering. Microsoft Satori is a commonly used KG in our collected articles. By leveraging the comprehensive and interconnected nature of KG, CTR prediction models can capture fine-grained correlations between entities, model long-range dependencies, and discover latent user preferences.

Extensive studies have provided fruitful approaches for KG-based CTR prediction. However, there is a lack of methodical classification and synthesized analysis of their respective strengths and weaknesses, making it challenging for researchers to gain a comprehensive understanding of different models. We aim to provide a comprehensive overview of the state-of-the-art models in KG-based CTR prediction. The objectives of this review can be outlined as follows. First, we aim to categorize the various approaches employed in KG-based CTR prediction and compare the advantages and disadvantages of each category. Second, we are intended to discuss how various models perform on various datasets. Additionally, this survey is expected to shed light on research trends, challenges, and future directions in this area.

There are a limited number of recently published literature reviews on CTR prediction (Yang and Zhai, 2022) and KG in recommender systems (Gao et al., 2020; Guo et al., 2020a; Khan et al., 2022a). Yang and Zhai (2022) conducted a literature review on CTR prediction models with respect to modeling frameworks, and discussed their advantages; Gao et al. (2020) provided a survey on deep learning models on KG to facilitate recommendations; Guo et al. (2020a) and Khan et al. (2022a) reviewed KG-based recommender systems.

In contrast to previous works, our survey delves into various models proposed in this domain and classifies them into three categories: embedding-based, path-based and propagation-based models, from the perspective of how to integrate KG into CTR prediction. Within each subcategory, we systematically dissect the important modules involved and summarize their principles and contributions. We also consolidate commonly used datasets and evaluation metrics, and discuss the model performance under different settings. Furthermore, we analyze the strengths and drawbacks of these models and identify prospective research directions in KG-based CTR prediction. By understanding the advancements and challenges in this area, researchers and practitioners can gain insights into the effective utilization of KG in CTR prediction models.

The structure of this survey is outlined as follows. Section 2 provides how to search and identify articles included in this review. Section 3 introduces the related concepts and definitions of CTR prediction and KG, along with notations used throughout the paper. Section 4 reviews the existing literature on KG-based CTR prediction. Section 5 summarizes datasets and evaluation metrics commonly used in this field and discusses the model performance on various application scenarios. Section 6 presents the analysis of current trends and challenges and future research directions. Section 7 concludes the survey by summarizing the key findings and suggestions for the field.

## 2. Literature Search and Study Identification

This survey searches for articles on KG-based CTR prediction in six prominent acamdeic databases: Web of Science, ACM Digital Library, IEEE Xplore, EBSCOhost, ScienceDirect and ABI/Inform Global. For identifying relevant articles, the scope of our survey is based on the keywords ("knowledge graph" OR "knowledge aware") AND ("CTR prediction" OR "click-through rate prediction" OR "click prediction") through a full-text search under the restriction of English. The search across the Web of Science yields 15 pertinent publications. The ACM Digital Library encompasses 127 relevant records. IEEE Xplore covers 30 relevant papers. EBSCOhost contains 3 matching entries. ScienceDirect returns 62 relevant literature sources. Additionally, the ABI/Inform Global business database contributes 40 aligned research works. Furthermore, we broaden our search by exploring citations and related articles of those previously identified on Google Scholar. The additional research yields 15 articles. After conducting the initial search, to identify whether the retrieved articles address KG-based CTR prediction problems, each article undergoes a manual screening process involving reviewing the title, abstract, full-text and experiments, which exclude 112 articles. Finally, the literature search yields a selection of 86 research publications, consisting of 42 journal articles, 41 conference articles and 3 pre-prints, spanning the period from 2018 to 2024. The literature search process is illustrated in Figure 1. The collected literature for this survey is presented in Table A1.

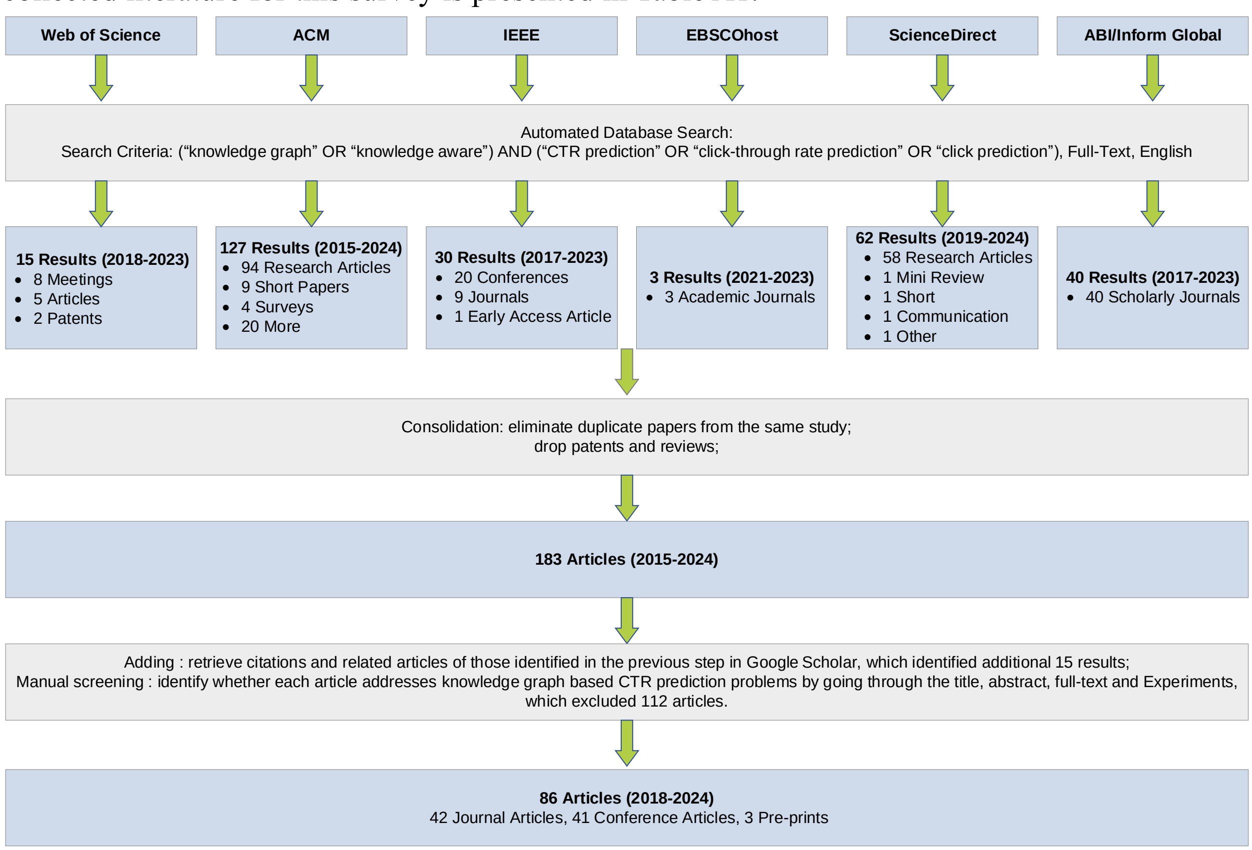


Figure 1. Literature search and study identification.

As shown in Figure 2, we visualize the number of publications in this review. The visualization demonstrates a steady growth in the number of publications year over year since 2018. The upward trend indicates increasing interests and research efforts in the field of KG-based CTR prediction.

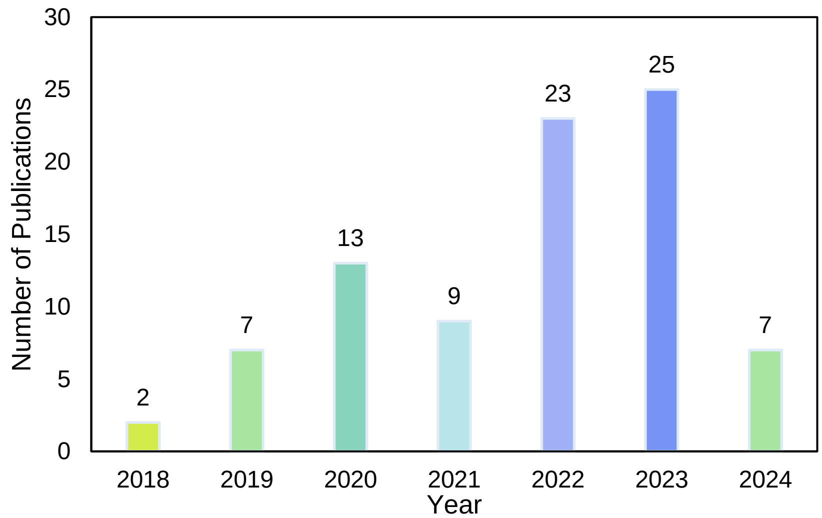


Figure 2. The number of publications from 2018 to 2024.

## 3. Preliminaries

In this section, we illustrate basic knowledge in KG-based CTR prediction, including KG and CTR prediction. For convenience, notations and their corresponding descriptions are detailed in Table 1. Throughout this paper, we adhere to the convention that matrices are symbolized by uppercase characters in bold, while vectors are denoted by bold lowercase characters.

Table 1. Notations and descriptions.

| Notations | Descriptions |
|---|---|
| $u$ | User |
| $v$ | Item |
| $\mathbf{u}$ | Representation of user $u$ |
| $\mathbf{v}$ | Representation of user $v$ |
| $\mathcal{U} = \{u_1, u_2, \dots, u_{\lvert\mathcal{U}\rvert}\}$ | A set of users with the number of $\lvert\mathcal{U}\rvert$ |
| $\mathcal{V} = \{v_1, v_2, \dots, v_{\lvert\mathcal{V}\rvert}\}$ | A set of items with the number of $\lvert\mathcal{V}\rvert$ |
| $\mathcal{F}$ | Click-through rate prediction function |
| $\Theta$ | Parameters of prediction function $\mathcal{F}$ |
| $\hat{y}_{uv}$ | Predicted user $u$'s click probability to item $v$ |
| $\mathcal{G}$ | Knowledge graph (KG) |
| $Y$ | User-item interaction matrix |
| $e_h, e_t$ | Head and tail entity in KG |
| $r$ | Relation in KG |
| $\mathbf{e}_h, \mathbf{e}_t$ | Representation of head and tail entity in KG |
| $\mathbf{r}$ | Representation of relation in KG |
| $V$ | The set of entities in the KG |

| | |
|---|---|
| $E$ | The set of relations in the KG |
| $\mathcal{A}$ | The set of entity types of the KG |
| $\mathcal{R}$ | The set of relation types of the KG |
| $G_T$ | Network schema |
| $\mathcal{N}_e^h$ | $h$-hop neighbors of entity $e$ |
| $\mathcal{S}_e^h$ | The entity triplet set with $h$-hop neighbors. |
| $\mathcal{L}$ | The loss function |
| $d$ | The embedding dimension |
| $\mathbf{W}$, $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3$, $\mathbf{W}_4$ $\mathbf{W}$, $\mathbf{b}$, $\mathbf{b}_1$, $\mathbf{b}_2$ | Trainable parameters |
| $\lambda$, $\lambda_1$, $\lambda_2$ | The balancing parameter |
| $\sigma(\cdot)$ | The normalized function |
| $\oplus$ | Concatenation |
| $\odot$ | Element-wise product |

## 3.1 Knowledge Graph

KG is a type of directed heterogeneous graph with the structural representation of knowledge, in which nodes represent entities and edges represent relationships between these entities. The KG $\mathcal{G} = (V, E)$ is composed of massive $(e_h, r, e_t)$ triplets, where $e_h \in V$ is the head entity, $r \in E$ is the relation entity, and $e_t \in V$ is the tail entity. For example, the triplet (Titanic, film.director, James Cameron) states the fact that James Cameron is the director of the film Titanic. There are type mapping functions: $\phi: V \to \mathcal{A}$ and $\psi: E \to \mathcal{R}$, where each entity $v \in V$ belongs to an entity type: $\phi(V) \in \mathcal{A}$ and each relation $r \in E$ belongs to an entity type: $\psi(r) \in \mathcal{R}$.

### 3.1.1 Meta-path

Meta-path refers to the specific path with a sequence of node types and edge types in the network schema $G_T = (\mathcal{A}, \mathcal{R})$, which can be noted as $\mathcal{P} = A_0 \xrightarrow{R_1} A_1 \xrightarrow{R_2} \cdots \xrightarrow{R_k} A_k$. $R_1 \circ R_2 \cdots \circ R_k$ is the combination of relation types from the entity type $A_0$ to $A_k$, where $A_i \in \mathcal{A}$ and $R_i \in \mathcal{R}$ for $i = 0, 1, \cdots, k$. Meta-path can be used to represent contextual information and extract relation patterns within the KG. For example, in a movie recommendation scenario, the meta-path "User-Movie-Genre-Movie-User" corresponds to the relationships between users and movies through the same movie genre, which provides a flexible and interpretable way to infer user preferences.

### 3.1.2 $H$-hop Neighbor

$H$-hop neighbor refers to the neighbor node with distance $H$ from the given node. In the path $e_0 \xrightarrow{r_1} e_1 \xrightarrow{r_2} \cdots \xrightarrow{r_H} e_H$, $e_H$ is the $H$-hop neighbor of $e_0$, which can be noted as $e_H \in \mathcal{N}_{e_0}^H$. Exploring the multi-order neighbors allows for a more comprehensive understanding of related entities and relations.

### 3.1.3 Entity Triplet Set

The entity triplet set is a collection of triplets that represent relationships between entities. For the entity $e \in \mathcal{G}$, the entity triplet set with $h$-hop neighbors can be represented as $\mathcal{S}_e^h = \{(e_h, r, e_t) | (e_h, r, e_t) \in \mathcal{G} \text{ and } e_h \in \mathcal{N}_e^{h-1}\}, h = 1, 2, \cdots, H$.

## 3.2 CTR Prediction

Click-Through Rate (CTR) prediction is an essential task in recommender systems. It refers to predicting the probability that a user will click on a specific item, which can be formulated as

$$\hat{y}_{uv} = \mathscr{F}(\mathbf{u}, \mathbf{v}; \Theta, Y), \quad (1)$$

where $\hat{y}_{uv}$ denotes the predicted probability that the user $u$ will interact with the item $v$, $\Theta$ denotes the parameters of the prediction function $\mathscr{F}$, and $Y$ denotes the user-item interaction matrix. $y_{uv} = 1$ denotes the user $u$ clicks the item $v$, otherwise $y_{uv} = 0$.

KG-based CTR prediction leverages KG to extract relevant features and learn patterns that can help predict the probability of a user clicking on a particular item. Each user or item can be associated with entities in the KG. These connections enable the model to leverage the rich information in the KG leverage the rich information present in the KG to improve the accuracy of CTR prediction models. Given the user-item interaction matrix $Y$ and KG $\mathcal{G}$, the task aims to learn a prediction function

$$\hat{y}_{uv} = \mathscr{F}(\mathbf{u}, \mathbf{v}; \Theta, Y, \mathcal{G}). \quad (2)$$

# 4. Categorization for KG-based CTR Prediction Models

According to how to utilize KG in the CTR prediction, KG-based CTR prediction models in the state-of-the-art literature can be classified into three categories, including embedding-based, path-based and propagation-based models as shown in Table 2. Specifically, we introduce modeling frameworks in each category and discuss their strengths and limitations. In the table, "IKG" denotes KG for items, "UIKG" denotes a united graph of item KG and user-item interactions, and "U&IKG" denotes KG for both users and items.

Table 2. Categorization for KG-based CTR prediction models in this survey.

| Category | Sub-Category | Model | Framework | KGE | KG Type | Reference |
|---|---|---|---|---|---|---|
| Embedding | One-by-one learning | DKN | CNN, DNN, Attention | TransD | IKG | Wang et al. (2018b) |
| | | DMCM | CNN, DNN, Attention | TransD | IKG | Wang et al. (2019d) |
| | | KG-RWSNM | CNN, Attention | TransD | IKG | Yang et al. (2020) |
| | | K-DCN | CNN, DNN, Attention | GNN, TransE | U&IKG | Wong et al. (2021) |
| | | CareGraph | Attention | TransE | IKG | Tangruamsub et al. (2022) |
| | | KANN | Attention | TransE | IKG | Liu and Miyazaki (2022) |
| | Joint learning | KGFER | CNN, MLP | TransR | IKG | Fan et al. (2022) |
| | | KGBPR | Factorization Machines | TransR | IKG | Ma et al. (2023) |
| | Alternate learning | MKR | MLP | End-to-end | IKG | Wang et al. (2019b) |
| | | MUKG | MLP | End-to-end | IKG | Sun and Shagar (2020) |
| | | CAKR | MLP, Attention | End-to-end | IKG | Huang et al. (2022) |

| | | | | | | |
|---|---|---|---|---|---|---|
| Path | Meta-path | SAGE | GNN, Attention | TransR | UIKG | Khan et al. (2022) |
| | | H-SAGE | DNN, Autoencoder | TransD | UIKG | Khan et al. (2023) |
| | Path-embedding | KARN | CNN, RNN, Attention | End-to-end | UIKG | Zhu et al. (2020) |
| | | MRP2Rec | RNN | End-to-end | UIKG | Wang et al. (2020b) |
| | | MTBRN | RNN, Attention | End-to-end | IKG | Feng et al. (2020b) |
| | | DISL | SEP2Vec | End-to-end | IKG | Li et al. (2023c) |
| | | CKGE | Transformer | End-to-end | U&IKG | Yang et al. (2023b) |
| Propagation | Refinement of the user | RippleNet | Attention | End-to-end | IKG | Wang et al. (2018a) |
| | | AKUPM | Attention | TransR | IKG | Tang et al. (2019) |
| | | KGDAM | Attention | End-to-end | IKG | Li et al. (2019) |
| | | GFEN | GNN, Attention | End-to-end | IKG | Yu et al. (2020) |
| | | KCER | Attention | End-to-end | IKG | Wang et al. (2020a) |
| | | LAGI | DNN, Attention | End-to-end | IKG | Zhang and Yang (2020) |
| | | DKEN | CNN, DNN, Attention | End-to-end | IKG | Guo et al. (2020) |
| | | SYT_RippleNet | Attention | node2vec | IKG | Yang et al. (2021) |
| | | HRS | MLP, Attention | End-to-end | IKG | Dong et al. (2022) |
| | | Ripp-MKR | MLP, Attention | End-to-end | IKG | Wang et al. (2021) |
| | | ReBKC | RNN, Attention | End-to-end | IKG | Hui et al. (2022) |
| | | GRE | GNN, Attention | End-to-end | IKG | Wang et al. (2022) |
| | | KEMIM | Attention | End-to-end | IKG | Yang et al. (2023a) |
| | | KGAN | Attention | TransE, TransD, MLP, DistMult | IKG | Zhang et al. (2023) |
| | | Ripple-DF | Attention | End-to-end | IKG | Hai and Hongyan (2023) |
| | | KGDAE | Attention | End-to-end | IKG | Gao et al. (2023) |
| | Refinement of the item | KGCN | GNN, Attention | End-to-end | IKG | Wang et al. (2019c) |
| | | KGNN-LS | GNN, Attention | End-to-end | IKG | Wang et al. (2019a) |
| | | TEKGR | GNN, RNN, Attention | End-to-end | IKG | Lee et al. (2020) |
| | | ATBRG | GNN, Attention | End-to-end | IKG | Feng et al. (2020a) |
| | | KGCN-CF | GNN, Attention | End-to-end | IKG | Hou et al. (2021) |
| | | KCRec | GNN, Attention | End-to-end | IKG | Zhang et al. (2021) |
| | | RKG | GNN, MLP, Attention | End-to-end | IKG | Shu and Huang (2021) |
| | | KRNN | GNN, Attention | End-to-end | IKG | Ma et al. (2022a) |
| | | CG-KGR | GNN, Attention | End-to-end | IKG | Chen et al. (2022b) |
| | | COAT | GNN, Attention | End-to-end | IKG | Dai et al. (2022) |
| | | UBAR | GNN, Attention | End-to-end | IKG | Wu et al. (2022) |
| | | KE-GCN | GNN, Attention | TransR | IKG | Tang et al. (2022) |
| | | DFM-GCN | GNN, DeepFM, DNN | End-to-end | IKG | Xiao et al. (2022) |
| | | KGAFM | GNN, AFM, Attention, MLP | End-to-end | UIKG | Xie et al. (2022) |
| | | DeepFM_GCN | GNN, DeepFM, DNN | End-to-end | IKG | Chen et al. (2022a) |
| | | ERSIF-KR | GNN, Attention | RotatE | IKG | He et al. (2023) |
| | | KGG | GNN, GAN, Attention | End-to-end | IKG | Song et al. (2023) |
| | | CRMMAN | GNN, Attention | End-to-end | IKG | Hu et al. (2023) |
| | | HRR | GNN, Attention | End-to-end | IKG | Li et al. (2023a) |
| | | Multi-Rec | GNN, Attention | End-to-end | IKG | Shu and Huang (2023) |
| | | DCRAN | GNN, Attention | TransR | IKG | Duan et al. (2023a) |
| | | KNIFE | GNN | DistMult | IKG | Yao et al. (2023) |
| | | SEKGAT | GNN, Attention | End-to-end | IKG | Li et al. (2023b) |
| | | KGDIE | GNN, Attention, Transformer | RTransD | IKG | Wang et al. (2023c) |
| | | EPAN-SERec | GNN, Attention | End-to-end | IKG | Tang et al. (2024) |

| | | | | | |
|---|---|---|---|---|---|
| | CKLF | GNN, Attention | TransD | IKG | Chen et al. (2024) |
| Refinement of both user and item representation | KNI | GNN, Attention | End-to-end | UIKG | Qu et al. (2019) |
| | CKAN | GNN, Attention | End-to-end | UIKG | Wang et al. (2020c) |
| | MVIN | GNN, Attention | End-to-end | IKG | Tai et al. (2020) |
| | KCAN | GNN, Attention | TransH | UIKG | Tu et al. (2021) |
| | DEKR | GNN, Attention | End-to-end | U&IKG | Cao et al. (2021) |
| | MNI | GNN, MLP, Attention | End-to-end | UIKG | Ma et al. (2021) |
| | BKANE | RNN, GNN, Attention | End-to-end | UIKG | Lyu et al. (2022) |
| | GACF | GNN, Attention | End-to-end | UIKG | Elahi and Halim (2022) |
| | KGET | GNN, Attention | End-to-end | UIKG | Ma et al. (2022b) |
| | KGHR | GNN, CNN, Attention | End-to-end | UIKG | Sun and Li (2022) |
| | MCCLK | GNN, Attention | End-to-end | UIKG | Zou et al. (2022a) |
| | KGIC | GNN, Attention | End-to-end | UIKG | Zou et al. (2022b) |
| | RKAC | GNN, Attention | End-to-end | UIKG | Qian et al. (2022) |
| | RFAN | GNN, Attention | TransR | UIKG | Duan et al. (2023b) |
| | DIKGNN | GNN, Attention | TransH | U&IKG | Tu et al. (2023) |
| | UPIACM | GNN, RNN, Attention | End-to-end | IKG | Wang et al. (2023b) |
| | DS-KGAT | GNN, RNN, Attention | End-to-end | IKG | Peng et al. (2023) |
| | CurvRec | GNN, Attention | End-to-end | UIKG | Wang et al. (2023a) |
| | P-GCN | GNN, Attention | End-to-end | UIKG | Bai et al. (2023) |
| | QTEF-CRL | GNN, Attention | End-to-end | UIKG | Ong et al. (2023) |
| | SKGCR | GNN, Attention | End-to-end | UIKG | Liu et al. (2023) |
| | MKNBL | GNN, Attention | End-to-end | UIKG | Wang et al. (2024a) |
| | RKGCN | GNN, Attention | End-to-end | IKG | Li et al. (2024) |
| | KFGAN | GNN, Attention | End-to-end | UIKG | Wang et al. (2024b) |
| | RAKCR | GNN, Attention | End-to-end | UIKG | Cui et al. (2024) |
| | KGCAN | GNN, Attention | End-to-end | UIKG | Elahi et al. (2024) |

## 4.1 Embedding-based Models

Embedding-based models enable the incorporation of rich semantic knowledge into CTR prediction utilizing knowledge graph embedding (KGE) methods. KGE represents entities and relations in the KG as continuous and low-dimensional vectors and information and captures semantic information in the KG.

In general, embedding-based CTR prediction models primarily consist of two modules: the KGE module and the CTR prediction module (Guo et al., 2020a). The KGE module transforms entities (e.g., movies, actors and directors) and relations among entities into low-dimensional vectors by using TransX series embedding techniques (Bordes et al., 2013; Ji et al., 2015; Lin et al., 2015). The CTR prediction module takes knowledge embeddings as input together with inherent user features and item attributes to infer the clicking probability.

By considering how to integrate the two modules, embedding-based models can be further categorized into three types, including one-by-one learning, joint learning and alternate learning, as shown in Figure 3.

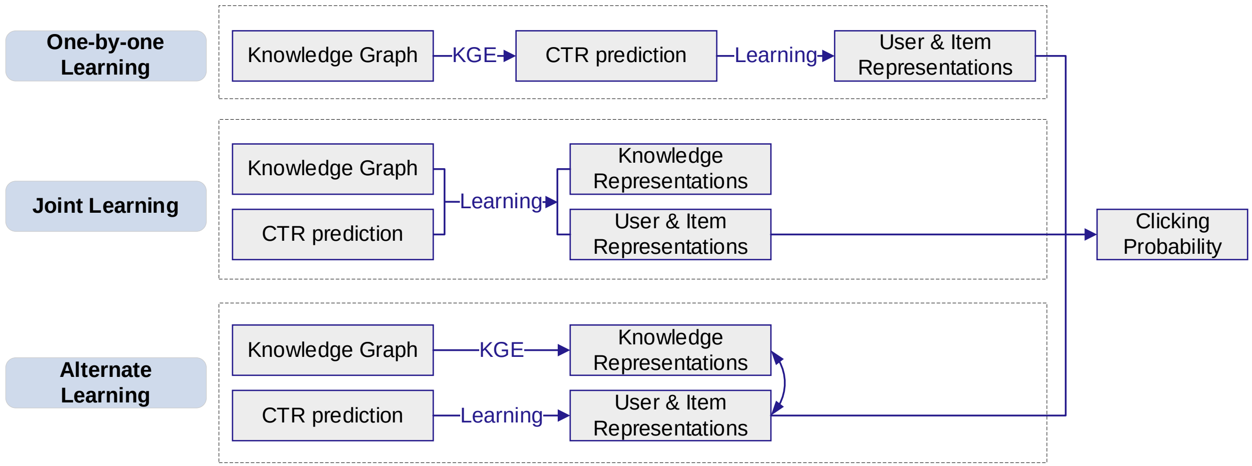


Figure 3. Learning processes of embedding-based models.

#### 4.1.1 One-by-one Learning Models

In one-by-one learning models, graph embedding and CTR prediction are trained separately in a sequential manner: KGE generates embeddings of entities and relations in a KG (the first stage), which are fed into CTR prediction (the second stage). In the first stage, three KGE models are developed in the literature.

**(1) TransD** (Ji et al., 2015). TransD projects entity embeddings into a different space for each relation, allowing the model to differentiate the semantics of different relations. The score function is given as

$$f_r(e_h, e_t) = \left\| (\mathbf{w}_r \mathbf{w}_h^\top + \mathbf{I})\mathbf{e}_h + \mathbf{r} - (\mathbf{w}_r \mathbf{w}_t^\top + \mathbf{I})\mathbf{e}_t \right\|_2^2, \tag{3}$$

where $f_r(\cdot)$ denotes the score function; $\mathbf{w}_r$, $\mathbf{w}_h$ and $\mathbf{w}_t$ are mapping vectors; $\mathbf{I}$ is an identity matrix.

In the literature, TransD has been used to generate knowledge-aware embeddings for news CTR prediction (Wang et al., 2018b; Yang et al., 2020) and movie CTR prediction (Wang et al., 2019d) where each word in the news (or movie) is associated with a relevant entity in a KG.

**(2) TransE** (Bordes et al., 2013). TransE models the relationship between entities as a translation operation in the embedding space to obtain $\mathbf{e}_h + \mathbf{r} \cong \mathbf{e}_t$ by minimizing the energy function,

$$f_r(e_h, e_t) = \|\mathbf{e}_h + \mathbf{r} - \mathbf{e}_t\|. \tag{4}$$

Wong et al. (2021) constructed a conversation KG from information about users, items and conversations for the scenario of conversational recommender systems. Both semantic and structural information in the KG is encoded via GCN and TransE jointly. The layer-wise GCN is represented as

$$\mathbf{X}^{(h+1)} = \sigma(\mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}\mathbf{X}^{(h)}\mathbf{W}^{(h)}), \tag{5}$$

where $\mathbf{X}^{(h)}$ is a layer-specific representation matrix of entities, $\mathbf{A}$ is the adjacency matrix, $\mathbf{D}$ is the diagonal degree matrix of $\mathbf{A}$ and $\mathbf{W}^{(h)}$ is a trainable weight matrix of layer $h$. During the cotraining procedure, the output embedding matrix of GCN is fed into TransE as the input, aiming to minimize the margin-based ranking loss:

$$\mathcal{L} = \sum_{(e_h,r,e_t)\in\mathcal{G}} \max\left(0, f_r(e_h, e_t) + \gamma - f_r(e_h{}', e_t{}')\right), \tag{6}$$

where $f_r(e_h{}', e_t{}')$ replaces the head or tail with the entity in $\mathcal{G}$ randomly to generate negative samples, and $\gamma$ is the margin.

Liu and Miyazaki (2022) introduced KG for movie CTR prediction, which models knowledge-level user-item interactions. The entities extracted from reviews are extended to KG and encoded by TransE. The embedding of the central entity concatenated with the average of context embeddings forms the final entity representation matrix.

**(3) TransR** (Lin et al., 2015). TransR represents entities and relations in separate embedding spaces, in which the loss function is defined as

$$f_r(e_h, e_t) = \|\mathbf{M}_r\mathbf{e}_h + \mathbf{r} - \mathbf{M}_r\mathbf{e}_t\|_2^2, \tag{7}$$

where $\mathbf{M}_r$ denotes the projection matrix to map the entity from its own space to the relation space of $r$.

Tangruamsub et al. (2022) encoded KG via TransR and introduced KGE to CTR prediction in health nudges with relevant attribute information.

During the second stage, Wang et al., (2018b, 2019d) and Yang et al. (2020) developed CNN-based models to perform prediction. Wang et al. (2018b) devised the component KCNN to fuse word-level and knowledge-level embeddings and generate the final item representation $\mathbf{v}_j$, i.e. news. To characterize the user's personal interests, an attention network is used to dynamically aggregate the user's clicked history with different weights:

$$\mathbf{u}_i = \sum_{k=1}^{N} s_{v_k,v_j}\mathbf{v}_k, \tag{8}$$

where $s_{v_k,v_j}$ denotes the normalized impact weight between the candidate item $v_j$ and the clicked item $v_k$. Finally, given the user representation $\mathbf{u}_i$ and item representation $\mathbf{v}_j$, the click-through rate is predicted by DNN: $\hat{y}_{uv} = DNN(\mathbf{u}_i, \mathbf{v}_j)$.

Wang et al. (2019d) fed entity embeddings into a CNN-based network along with item embeddings and genre embeddings to obtain the final item representations. The next procedure is to model dynamic user representations and output probability, which is the same as (Wang et al., 2018b).

Yang et al. (2020) obtained the context embedding by taking the average of embeddings of entities associated with the central entity. Next, a CNN module extracts item features fusing multiple perspectives, including topic embeddings, entity embeddings and context embeddings. As for the user side, a social network is constructed by reposting and commenting. To quantify the influence of active users, a random sampling mechanism is adopted to obtain neighbors. The sampling probability $s_{i,j}$ is defined with the weight of the adjacent edge between node $i$ and $j$:

$$s_{i,j} = \frac{\exp(n_{i,j})}{\sum_{i,k\in T}\exp(n_{i,k})}, \tag{9}$$

where $n_{i,j}$ is the number between node $i$ and $j$ in the social network $T$. The social network feature can be obtained by

$$\mathbf{u}_i = \sum_{k=1}^{N} s_{i,k} \mathbf{u}_k^{soical}, \tag{10}$$

where $\mathbf{u}_k^{soical}$ is the representation of the related user in the social network of the user $u_i$. Finally, the item representation and the user representation are multiplied to predict the probability of the user clicking on the candidate item.

Wong et al. (2021) predicted the clicking probability based on DCN. They pre-trained and fine-tuned entity embeddings and adopted the DCN model to fuse representations of the user's state, dialogue interaction and common features.

Tangruamsub et al. (2022) and Liu and Miyazaki (2022) employed attention mechanism as the primary technique. Tangruamsub et al. (2022) aggregated one-hop neighbor embeddings of clicked items with assigned weights to distinguish the user preference, which is concatenated with the user click history embedding and user profile embedding to generate the final user embedding.

Liu and Miyazaki (2022) designed two key layers: the inner-attention layer and outer-attention layer. The inner layer performs multi-head self-attention on the review entity matrix for the user $u_i$ and item $v_j$ and obtain respective representation matrix $\mathbf{U}_i$ and $\mathbf{V}_j$. The outer-attention models the interactions between users and movies. The final user representation $\mathbf{u}_i$ can be formulated as

$$\begin{cases} \mathbf{A}_{ij} = \text{softamx}(\frac{\mathbf{U}_i^Q \mathbf{V}_j^K}{\sqrt{d}}) \\ \mathbf{u}_i = ReLU(\mathbf{W}_u \mathbf{A}_{ij} \mathbf{V}_j^V + \mathbf{b}_u) \end{cases}, \tag{11}$$

where $\mathbf{A}_{ij}$ is the attention score; $\mathbf{U}_i^Q$ is the query matrix after linear transformation of $\mathbf{U}_i$; $\mathbf{V}_j^K$ and $\mathbf{V}_j^V$ are the key and value matrix after linear transformation of $\mathbf{V}_j$, respectively; $\mathbf{W}_u$ and $\mathbf{b}_u$ are trainable parameters. Likewise, the final item representation $\mathbf{v}_j$ can be obtained. At last, the prediction layer outputs the probability by the element-wise product of $\mathbf{u}_i$ and $\mathbf{v}_j$.

### 4.1.2 Joint Learning Models

One-by-one learning models are indeed more suitable for in-graph tasks, such as KG completion. The loose coupling between the KGE module and the CTR prediction module may result in obtained embeddings not being optimized specifically for CTR prediction tasks. Joint learning models aim to address the mismatch between the two modules, which are trained jointly in an end-to-end fashion. These models can benefit from the complementary information provided by each module.

For the KGE module, Fan et al. (2022) and Ma et al. (2023) adopted TransR to embed entities and relations into the entity space and relation space, respectively.

For the CTR prediction module, there are three ways to handle joint feature learning. Fan et al. (2022) employed a two-dimensional CNN to integrate neighboring entities and relations and a fully connected network to generate item representations. Instead of simply predicting with the inner product of the user's and item's representations, they mapped the item's representation from its own space to the user's latent space ahead of this step. The final objective function can be formulated as

$$\begin{cases} \mathcal{L} = \lambda_1 \mathcal{L}_{CF} + (1-\lambda_1)\mathcal{L}_{KG} + \lambda_2 \|\Theta\|_2^2 \\ \mathcal{L}_{CF} = \sum_{(u_i,v_j)\in R} \sum_{(u_i,v_j{}')\notin R} \log\left(1 + (\hat{y}_{u_i v_j} - \hat{y}_{u_i v_j{}'})\right) \\ \mathcal{L}_{KG} = \sum_{(e_h,r,e_t)\in \mathcal{G}} \sum_{(e_h,r,e_t{}')\in \mathcal{G}} \log\left(1 + (f_r(e_h, e_t) - f_r(e_h, e_t{}'))\right) \end{cases}, \tag{12}$$

where $\mathcal{L}_{CF}$ is the soft-margin loss to train the CTR prediction module, $\mathcal{L}_{KG}$ is the objective function in the KGE task, $(e_h, r, e_t{}')$ represents the negative sample of KG, $R$ is the set of observed user-item interactions.

Ma et al. (2023) combined structural knowledge with Bayesian personalized ranking Matrix Factorization (BPRFM), which aims to factorize the user-item interaction matrix into two lower-dimensional matrices for the user and item and uses a Bayesian approach to model the ranking of items for each user.

### 4.1.3 Alternate Learning Models

Alternate learning models further optimize knowledge transmission and utilize the KGE task to assist the CTR prediction task. The two tasks exhibit a strong correlation rather than being mutually independent, which take turns iteratively updating their parameters by the intermediate bridging module.

Wang et al. (2019b) proposed a multi-task feature learning framework where the KGE task serves as regularization for the CTR prediction task. The following works (Sun and Shagar, 2020; Huang et al., 2022) can be regarded as extensions of (Wang et al., 2019b). In the research (Wang et al., 2019b) and (Sun and Shagar, 2020), the two tasks are bridged by the designed cross&compress unit, which shares the feature representations between items and entities to complement each other. The cross&compress unit $C(\cdot)$ of layer $l$ can be expressed as $[\mathbf{v}_{l+1}, \mathbf{e}_{l+1}] = C(\mathbf{v}_l, \mathbf{e}_l)$. Considering employing outer product for the cross operation and one-dimensional vector parameters for the compression operation in (Wang et al., 2019b) is insufficient, Huang et al. (2022) proposed the cross-attention fusion module to optimize the relevance between the CTR prediction task and KGE task. The transfer weights of items and entities are extended to two-dimensional semantic space with Hadamar product. The following attention mechanism fuses intermediate features:

$$\begin{cases} \tilde{\mathbf{v}} = s_{1,1}\mathbf{v} + s_{1,2}\mathbf{e} \\ \tilde{\mathbf{e}} = s_{2,1}\mathbf{v} + s_{2,2}\mathbf{e}\text{'} \end{cases} \tag{13}$$

where $s_{i,j}$ denotes the attention score between the item and entity vector.

In the CTR prediction module, $L$ cross&compress units are adopted to learn the item representation,

$$\mathbf{v}^L = \mathbb{E}_{e\in S(v)}\left[C^L(\mathbf{v}, \mathbf{e})[\mathbf{v}]\right], \tag{14}$$

where $S(v)$ denotes the set of connected entities of item $v$. A $L$-layer MLP is utilized to extract the dense features for the user $u$:

$$\mathbf{u}^L = M^L(\mathbf{u}), \tag{15}$$

where $M(\cdot)$ denotes the fully-connected neural network layer. Then, given the user representation $\mathbf{u}^L$ and item representation $\mathbf{v}^L$, the clicking probability can be predicted by the inner product or MLP.

As for the KGE module, the head $e_h$ and relation $r$ are learned by cross&compress units, which are concatenated and fed into a $K$-layer MLP to predict the tail embedding $\hat{\mathbf{e}}_t$. The similarity score $s(e_h, r, e_t)$ is calculated by normalized inner product:

$$\begin{cases} \mathbf{e}_h^L = \mathbb{E}_{v \in S(e_h)}\left[C^L(\mathbf{v}, \mathbf{e}_h)[\mathbf{e}]\right] \\ \mathbf{r}^L = M^L(\mathbf{r}) \\ \hat{\mathbf{e}}_t = M^K(\mathbf{e}_h^L \oplus \mathbf{r}) \\ s(e_h, r, e_t) = \sigma(\mathbf{e}_t^T \hat{\mathbf{e}}_t) \end{cases}, \tag{16}$$

where $S(e_h)$ is the set of connected items of entity $e_h$. The complete objective function is defined as

$$\begin{aligned} \mathcal{L} &= \mathcal{L}_{CF} + \lambda_1 \mathcal{L}_{KG} + \lambda_2 \|\Theta\|_2^2 \\ &= \sum_{u \in \mathcal{U}, v \in \mathcal{V}} \mathcal{J}\left(\hat{y}_{uv}, y_{uv}\right) - \lambda_1 \left(\sum_{(e_h, r, e_t) \in \mathcal{G}} s(e_h, r, e_t) - s(e_h', r, e_t')\right) + \lambda_2 \|\Theta\|_2^2, \end{aligned} \tag{17}$$

where $\mathcal{J}(\cdot)$ the cross-entropy function.

### 4.1.4 Summary of Embedding-based Models

One-by-one learning models are straightforward and flexible to implement by decoupling the training process since each module can be optimized using different techniques or algorithms. However, this type of models are well-suited for in-graph applications and fail to characterize user-item interactions effectively. Joint learning enables the model to exploit interdependencies and correlations between graph embeddings and CTR prediction. They may suffer from increasing computational requirements for large-scale KG compared with one-by-one learning models. Alternate learning models can leverage sharing information and mutual feedback to refine the embeddings progressively. Despite the advantages, alternate learning models require sophisticated design to avoid information loss from the intermediate bridging module.

## 4.2 Path-based Models

Path-based models leverage the connectivity and semantic relationships within a KG to assist CTR prediction. These models aim to extract meaningful information about user preferences and item relevance. According to the way the paths are modeled, we classify these models into meta-path based models and path-embedding based models.

### 4.2.1 Meta-path Based Models

Meta-path based models focus on designing meta-paths and measuring node similarity to capture meaningful patterns of connectivity between entities. By incorporating meta-paths into the modeling process, CTR prediction models can better understand user behaviors.

The first issue is how to construct meta-paths to effectively capture structural connections. Khan et al. (2022b) extracted high-order relevant nodes based on semantic and structural similarity and integrated them into distinct meta-paths. The inter-node similarity $s_{i,j}$ is formulated by the objective function:

$$s_{i,j} = TNP(i,j) + LOC(i,j), \tag{18}$$

where $TNP(\cdot)$ determines the connection possibility between node $i$ and $j$, $LOC(\cdot)$ denotes appearance-based similarity calculated by cosine similarity and unified Adamic-Adar and Jaccard similarities. The follow-up work (Khan et al., 2023) also measured the semantic correlation between entities and transformed pertinent entities and relations into unique meta-paths.

The next procedure is to tackle the constructed subgraph. Khan et al. (2022b) designed Interaction-enhanced Knowledge Network (IKN) to model high-order implicit interactions over the similarity-attributed graph. Simultaneously, explicit features are integrated into user-item interactions to emphasize the prominent nodes in IKN by GNN. Finally, the prediction module receives entity representations and hidden information to make predictions regarding the desired preferences. In the other work (Khan et al., 2023), the unique meta-paths are converted to hash-codes and placed in semantic hash-buckets, leveraging Deep-Probabilistic (dProb) technique. The formal hash-codes $h_i = \{\pm 1\}^k$ of node $i$, where $k$ denotes the bit length, is fed to DNN to calculate the mutual likelihood of the corresponding entities:

$$\mathbf{h}_i^{(l+1)} = \sigma(\mathbf{W}^{(l)}\mathbf{h}_i^{(l)} + \mathbf{b}^{(l)}), \tag{19}$$

where $\mathbf{h}_i^{(l)}$ is the output of layer $l$. The total probability $P(n_{cj}|\mathbf{h}_i)$ of dProb can be formulated as

$$\begin{cases} P(n_{cj}|\mathbf{h}_i) = \frac{P(n_{cj}|\mathbf{h}_c,\mathbf{h}_j)+P(p_{cz}|\mathbf{h}_c,\mathbf{h}_z)}{2} \\ P(n_{cj}|\mathbf{h}_c,\mathbf{h}_j) = \Phi^{n_{cj}}\Psi^{1-n_{cj}} = \begin{cases} n_{cj} = 1, & if\ \Phi \\ n_{cj} = 0, & if\ \Psi \end{cases}, \\ P(p_{cz}|\mathbf{h}_c,\mathbf{h}_z) = \Phi^{p_{cz}}\Psi^{1-p_{cz}} = \begin{cases} p_{cz} = 1, & if\ \Phi \\ p_{cz} = 0, & if\ \Psi \end{cases} \end{cases} \tag{20}$$

where $n_{cj}$ is the interlinking probability of concerned hash-codes $\mathbf{h}_c$ and $\mathbf{h}_j$, $p_{cz}$ is the path selection probability from node $c$ to node $j$ of concerned hash-codes $\mathbf{h}_c$ and $\mathbf{h}_z$, respectively. $\Phi = x\sigma(x)$, $\Psi = 1 - x\sigma(x)$, and $x = \Delta(h_c, h_j)$, where $\Delta$ denotes the distance. To maintain semantic relevance among the information instances, a linear auto-encoder regression is employed to project the matrices of entities and relations in hamming space. Subsequently, dProb is again used to retrieve the required hash-codes and send back them to the presentation module to generate potential preferences.

### 4.2.2 Path-embedding Based Models

Meta-path based models are reliant on information similarity constraints, which often face difficulties in effectively capturing nuanced relationships and intricate patterns in paths. In contrast, path-embedding based models encode multiple paths between user-item pairs or item-item pairs within KG. By aggregating the embeddings of these paths, path-embedding based models aim to incorporate rich contextual information into the models and enhance the prediction accuracy.

As to path extraction, one implementation is to build paths within the global space of user-item interactions and KG. Zhu et al. (2020) mined path connectivity between users and items in the KG. Wang et al. (2020b) extracted high-order relational paths by the strategy of random walks. Some other implementations are based on item-related relationships. Feng et al. (2020b) incorporated KG and item-item similarity graph and constructed multiplex relation paths between user behaviors and items to infer the reason driving a user to click on a target item. To effectively extract paths, the breadth-first search (BFS) and greedy-selection principle are employed in the item-item similarity graph and the BFS principle is also used in the KG to keep the shortest paths. Li et al. (2023c) designed two kinds of paths based on historical interest sequence: the first one $\mathcal{P}_p$ is defined by the shortest path connecting two clicked items in the KG, which depicts the user behavior explicitly; the second one $\mathcal{P}_s$ is defined by the number of shared entities in the KG, which depicts the user interest fluctuations implicitly. Yang et al. (2023b) sampled paths connecting entity pairs by biased random walks. The probability distribution of the next entity can be computed as

$$P(\mathbf{e}_{i+1}|\mathbf{e}_i) = \begin{cases} \frac{w_{i,i+1}}{\sum_{e_j \in \mathcal{N}_{e_i}} w_{i,j}}, \exists\left(e_i, r_{i,i+1}, e_{i+1}\right) \in \mathcal{G} \\ 0, \ otherwise \end{cases}, \tag{21}$$

where $P(\mathbf{e}_{i+1}|\mathbf{e}_i)$ is the probability of choosing the next entity from $e_i$ to $e_{i+1}$, and $w_{i,i+1}$ is related to the amount of entity $e_{i+1}$ and relation $r_{i,i+1}$.

For explicitly encoding path information, there are three types of solutions as follows. Zhu et al. (2020), Wang et al. (2020b) and Feng et al. (2020b) utilized LSTM-based techniques to extract path features. Concretely, Zhu et al. (2020) exploited the user's click history sequence and devised the SRA component with stacked LSTM to capture the item sequence representation as $\{\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_t\}$:

$$\mathbf{h}_j = LSTM(\mathbf{h}_{j-1} + \mathbf{v}_j), \tag{22}$$

where $\mathbf{h}_j$ denotes the hidden states of LSTM. The following attention network differentiates the contributions of previous hidden states $\mathbf{H} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_t]$:

$$\begin{cases} \mathbf{A} = \text{softmax}(\mathbf{W}_1 \sigma(\mathbf{W}_2 \mathbf{H})) \\ \mathbf{a} = f_a(\mathbf{A}\mathbf{H}^T) \\ \bar{\mathbf{u}} = \mathbf{a} \oplus \mathbf{h}_t \end{cases}, \tag{23}$$

where $\mathbf{A}$ denotes the attention score matrix weighting items, $f_a(\cdot)$ is to calculate the average of the matrix row vectors, and $\bar{\mathbf{u}}$ is the user's history interest representation. Then the A-SRA component is designed to extract the user's potential intent representation $\tilde{\mathbf{u}}$, in which the SRA network is again used to encode each path between the user-item pair, and an attention mechanism is used to integrate multiple path feature representations. Finally, the representations of the user's history interest and potential intent are fused to calculate the clicking probability.

Wang et al. (2020b) encoded the path $\mathbf{p} = \{\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_L\}$ connecting the entity $e_h$ and $e_t$ by LSTM, which learns the relation representations for each path:

$$\begin{cases} \tilde{\mathbf{c}}_i = tanh(\mathbf{W}_c(\mathbf{e}_{i-1} \oplus \mathbf{r}_i) + \mathbf{b}_c) \\ \mathbf{f}_i = tanh(\mathbf{W}_f(\mathbf{e}_{i-1} \oplus \mathbf{r}_i) + \mathbf{b}_f) \\ \mathbf{g}_i = tanh\left(\mathbf{W}_g(\mathbf{e}_{i-1} \oplus \mathbf{r}_i) + \mathbf{b}_g\right) \\ \mathbf{o}_i = tanh(\mathbf{W}_o(\mathbf{e}_{i-1} \oplus \mathbf{r}_i) + \mathbf{b}_o) \\ \mathbf{c}_i = \mathbf{f}_i \odot \mathbf{c}_{i-1} + \mathbf{g}_i \odot \tilde{\mathbf{c}}_i \\ \mathbf{e}_i = \mathbf{o}_i \odot tanh\ (\mathbf{c}_{i-1}) \end{cases}, \tag{24}$$

where $\mathbf{c}_i$ and $\tilde{\mathbf{c}}_i$ are the current memory state and candidate memory state, respectively; $\mathbf{f}_i$, $\mathbf{g}_i$ and $\mathbf{o}_i$ represent the forget, input and output gate, respectively; $\mathbf{W}_c$, $\mathbf{W}_f$, $\mathbf{W}_g$ and $\mathbf{W}_o$ are mapping coefficient matrices; $\mathbf{b}_c$, $\mathbf{b}_f$, $\mathbf{b}_g$ and $\mathbf{b}_o$ are bias vectors. The latest state $\mathbf{e}_L$ is to establish the representation of path representation $\bar{\mathbf{p}}$. The following procedure involves predicting the existence of a relation path between the entity $e_h$ and $e_t$. Thus, the probability score is calculated as $\|\bar{\mathbf{p}} - \mathbf{r}\|$. A lower score indicates that the possibility of the path being true is higher.

Feng et al. (2020b) encoded each path by Bi-LSTM and obtained the representation $\mathbf{p}$. The path representation matrix $\mathbf{P}_{cf}$ of item-item similarity graph and $\mathbf{P}_{kg}$ of KG are fused to capture higher order interactions $\mathbf{P}_{fu}$,

$$\mathbf{P}_{fu} = \left\{\mathbf{p}_i \odot \mathbf{p}_j | 1 \le i \le k, i+1 \le j \le k\right\}, \tag{25}$$

where $\mathbf{p}_i \in \mathbf{P}_{cf} \cup \mathbf{P}_{kg}$ and $k$ is the number of all paths. The following attention mechanism is employed to aggregate paths from the path set,

$$\begin{cases} a_i = \frac{\exp(\mathbf{p}_i \mathbf{W} \mathbf{u})}{\sum_{j=1}^{n} \exp(\mathbf{p}_j \mathbf{W} \mathbf{u})} \\ \mathbf{a} = \sum_{i=1}^{n} a_i\, \mathbf{p}_i \end{cases}, \tag{26}$$

where $\mathbf{a}$ denotes the path representation, and $n$ denotes the path number of the path set. The path representation $\mathbf{a}_{cf}$, $\mathbf{a}_{kg}$ and $\mathbf{a}_{fu}$ are obtained for $\mathbf{P}_{cf}$, $\mathbf{P}_{kg}$ and $\mathbf{P}_{fu}$, respectively. The final clicking probability can be calculated as follows,

$$\hat{y}_{uv} = \sigma(M(M(M(\mathbf{u} \oplus \mathbf{v} \oplus \mathbf{a}_{cf} \oplus \mathbf{a}_{kg} \oplus \mathbf{a}_{fu})))). \tag{27}$$

Yang et al. (2023b) introduced Transformer to distinguish the path importance and made mask prediction to predict masked entities. To learn high-quality representations, one random path is retained and the remaining paths are masked given the entity pair.

Although previous works can learn the semantic information from the connection patterns automatically, they rarely explore the dynamic user interests via KG. In contrast, Li et al. (2023c) captured both the entity semantics over time and relations under different granularities to characterize user preference. The two kinds of paths $\mathcal{P}_M (M = p, s)$ are embedded by SEP2Vec with the objective of maximizing the average log probability

$$\frac{1}{L}\sum_{j=\omega}^{L-\omega} \log P(\mathcal{P}_{M_{j-1,j}} | \mathcal{P}_{M_{j-1-\omega,j-\omega}}, \cdots, \mathcal{P}_{M_{j-1+\omega,j+\omega}}), \tag{28}$$

where $L$ denotes the length of the path, $\omega$ is the size of the sliding window over the path. The two paths are merged by the entropy-aware pooling layer to generate user representations.

#### 4.2.3 Summary of Path-based Models

Meta-path based models provide an intuitive and interpretable way to reason about user selections. The performance of this type of models highly depends on the design of appropriate meta-paths, which requires specialized domain knowledge. The construction of meta-paths can be unscalable and challenging. For instance, in the news scenario where each news is associated with multiple entities, the identification of relevant meta-paths can be complex and time-consuming. Path-embedding based models can more explicitly model fine-grained interaction patterns. Compared to meta-path based models, they may have higher computational complexity. That means efficiency can be an issue, particularly in large-scale KGs.

### 4.3 Propagation-based Models

Embedding-based models lack the utilization of higher-order structural information, while path-based models may not be flexible enough in handling paths. Propagation-based models combine embedding-based and path-based models with the core idea of embedding propagation. They can capture fine-grained semantic information and structural connections simultaneously. The typical way is to utilize GNN-based techniques to aggregate high-order neighboring information and update representations iteratively. By leveraging the power of GNNs, these models can efficiently capture complex relationships and dependencies within a graph structure. Based on the specific type of entity being refined, these models are categorized into three categories: refinement of user representation, refinement of item representation, and refinement of both user and item representation.

#### 4.3.1 Refinement of User Representation

This type of models refine user representations based on historical clicked items. The key idea is performing preference propagation to extend user's potential interests in the KG. The iterative outward propagation relies on the multi-hop triplet sets $\mathcal{S}_u^h$ $(h = 1, 2, \cdots, H)$ in the KG. The general steps can be illustrated as (1) generating the user preference representation $\mathbf{o}_u^h$ by aggregating each order neighbors in $\mathcal{S}_u^h$; (2) combining $\mathbf{o}_u^h$ of all orders to calculate the final user representation. The process starts from the user's clicked items and ends with distant entities in the KG.

There are various solutions to generate user preference representations in each hop. Wang et al. (2018a) proposed a pioneering work that combines embedding-based models and path-based models in knowledge-enhanced CTR prediction. This innovative model drew inspiration from the physical phenomenon of water ripples, illustrating the propagation of user preferences. The propagation process of $h$ order calculates the similarity $p_i^h$ of the item $\mathbf{v}$ and head entity $\mathbf{e}_{h_i}^h$

measured in the relation space $\mathbf{R}_i^h$. The weighted summation of the relevance probabilities over the tail entities $\mathbf{e}_{t_i}^h$ yields the user preference representation $\mathbf{o}_u^h$,

$$\begin{cases} p_i^h = \frac{\exp(\mathbf{v}^T \mathbf{R}_i^h \mathbf{e}_{h_i}^h)}{\sum_{(e_{h_k}, r_k, e_{t_k}) \in \mathcal{S}_u^h} \exp(\mathbf{v}^T \mathbf{R}_k^h \mathbf{e}_{h_k}^h)} \\ \mathbf{o}_u^h = \sum_{(e_{h_i}, r_i, e_{t_i}) \in \mathcal{S}_u^h} p_i^h \, \mathbf{e}_{t_i}^h \end{cases}, \tag{29}$$

where for $h = 2, \cdots, H$, $\mathbf{v}$ is replaced with $\mathbf{o}_u^{h-1}$.

The following extensions (Hui et al., 2022; Wang et al., 2020a; Zhang and Yang, 2020) improved the process of historical user-item interactions in (Wang et al., 2018a). Zhang and Yang (2020) extracted local interests based on the user's historically clicked items via multi-layer fully connected neural networks and captured global interests based on KG. Wang et al. (2020a) learned the representation of knowledge context based on the specific user-item interaction and improved the calculation of the attention score as

$$\begin{cases} p_i^h = \frac{\exp(tanh\ (\mathbf{s} + \mathbf{s} \odot \mathbf{r}_i^h + \mathbf{b}) {\mathbf{e}_{t_i}^h}^T)}{\sum_{(e_{h_k}, r_k, e_{t_k}) \in \mathcal{S}_u^h} \exp(tanh\ (\mathbf{s} + \mathbf{s} \odot \mathbf{r}_k^h + \mathbf{b}) {\mathbf{e}_{t_k}^h}^T)} \\ \mathbf{s} = \frac{\mathbf{u} \odot \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|} \end{cases}. \tag{30}$$

Hui et al. (2022) learned historical preference by modeling the historical item sequence via LSTM and self-attention mechanism. Before embedding propagation in the KG, the entity embedding is generated by the weighted sum of associated triplets:

$$\begin{cases} \mathbf{c}_{ij} = \mathbf{W}_1 (\mathbf{e}_h \oplus \mathbf{r}_i \oplus \mathbf{e}_{t_j}) \\ a_{ij} = \text{softmax}(\mathbf{W}_2 \boldsymbol{c}_{ij}) \\ \bar{\mathbf{e}}_h = \sigma(\sum_{i=1}^{M} \sum_{j=1}^{N} a_{ij} \mathbf{c}_{ij}) \end{cases}, \tag{31}$$

where $\mathbf{c}_{ij}$ denotes the embedding of the triplet $(e_h, r_i, e_{t_j})$, $a_{ij}$ is the attention value for the triplet, and $\bar{\mathbf{e}}_h$ is the updated entity embedding.

Considering existing works are unaware of relationships between users and entities, Tang et al. (2019) explored the interactions among entities and categorized them into two types: inter-entity-interaction to distinguish the contributions of different entities to user representations and intra-entity-interaction to depict different characteristics of an entity involving in different relations. To depict the intra-entity-interaction, each entity in the KG is embedded into relation space by TransR for proper characteristics. To model the inter-entity-interaction, the entity propagation is similar to (Wang et al., 2018a), which refines the user representation by entities correlated with engaged items. The self-attention mechanism is utilized to differentiate the impacts of items on the user preference $\mathbf{o}_u^h$,

$$\begin{cases} \mathbf{V}_u^h = \mathbf{Q}_u^h = \mathbf{K}_u^h = \left[\mathbf{R}_1^h \mathbf{e}_{h_1}^h, \mathbf{R}_2^h \mathbf{e}_{h_2}^h, \cdots, \mathbf{R}_N^h \mathbf{e}_{h_N}^h\right] \\ \mathbf{o}_u^h = \mathbf{V}_u^h \text{softmax}(\frac{{\mathbf{Q}_u^h}^T \mathbf{K}_u^h}{\sqrt{d}}) \end{cases}. \tag{32}$$

Zhang et al. (2023) strengthened the utilization of relations in KG and designed intra-group and inter-group attention mechanisms to propagate user preferences. The intra-group aggregation generates user preference $\mathbf{o}_u^k$ grouped by different relation-links, where $k$ is to discriminate each distinct group. The inter-group aggregation utilizes the attention mechanism to perform weighted summation on $\mathbf{o}_u^k$ and generates the final user representation $\mathbf{u}$.

The next step is to combine multi-hop user representations. Wang et al. (2018a) and Hui et al. (2022) applied the sum operation, Zhang and Yang (2020) adopted the concatenation operation, and Wang et al. (2020a) and Tang et al. (2019) leveraged the attentive aggregation operation.

After propagating $H$ hops from the user's engaged items, Wang et al. (2018a) obtained the user representation by combining the user preference representations,

$$\mathbf{u} = \mathbf{o}_u^1 + \mathbf{o}_u^2 + \cdots + \mathbf{o}_u^H. \tag{33}$$

Finally, the sigmoid function outputs the predicted probability by the inner product of the user representation and item representation,

$$\hat{y}_{uv} = \sigma(\mathbf{u}^T\mathbf{v}). \tag{34}$$

The prediction module of (Zhang and Yang, 2020) takes the local user representation $\bar{\mathbf{u}}$, global user representation $\tilde{\mathbf{u}}$ and item representation $\mathbf{v}$ as input. The predicted probability is formulated by DNN: $\hat{y}_{uv} = DNN(\bar{\mathbf{u}}, \tilde{\mathbf{u}}, \mathbf{v})$.

In the research (Wang et al., 2020a), given the representation matrix $\mathbf{O}_u = [\mathbf{o}_u^1, \mathbf{o}_u^2, \cdots, \mathbf{o}_u^H]$, the knowledge fusion layer adopts the multi-dimensional attention to generate the enhanced context representation as follows,

$$\begin{cases}\mathbf{A} = \mathrm{softmax}(\mathbf{W}_1 tanh(\mathbf{W}_2\mathbf{u} + \mathbf{W}_3\mathbf{v} + \mathbf{W}_4\mathbf{O}_u^T + \mathbf{b}_1) + \mathbf{b}_2) \\ \mathbf{z}_c = f_a(\mathbf{A}\mathbf{O}_u)\end{cases}, \tag{35}$$

where $\mathbf{z}_c$ is the context representation, $f_a(\cdot)$ denotes the average pooling operation. The final user representation is calculated by the gating mechanism:

$$\begin{cases}\mathbf{g} = \sigma(\mathbf{W}_{g1}\mathbf{u} + \mathbf{W}_{g2}\mathbf{z}_c + \mathbf{b}_g) \\ \tilde{\mathbf{u}} = \mathbf{g}\odot\mathbf{u} + (\mathbf{1} - \mathbf{g})\odot\mathbf{z}_c\end{cases}, \tag{36}$$

where $\mathbf{g}$ denotes the learned gate signal to balance the contributions of two representations and $\mathbf{W}_{g1}$ $\mathbf{W}_{g2}$ and $\mathbf{b}_g$ are learnable model parameters.

To characterize user interest, Tang et al. (2019) obtained the final user representation by attentive aggregation of user preference in each order,

$$\begin{cases}\mathbf{Q}_u = \mathbf{v} \\ \mathbf{V}_u = \mathbf{K}_u = [\mathbf{o}_u^1, \mathbf{o}_u^2, \cdots, \mathbf{o}_u^H] \\ \mathbf{u} = \mathbf{V}_u \mathrm{softmax}(\frac{\mathbf{Q}_u^T\mathbf{K}_u}{\sqrt{d}})\end{cases}. \tag{37}$$

Given the user and item representation, the CTR can be predicted by Equation 34.

There are other extensions of (Wang et al., 2018a) to further improve knowledge-aware CTR prediction. Specifically, Yang et al. (2021) applied (Wang et al., 2018a) to music CTR prediction and the difference is using node2vec to embed the KG. Gao et al. (2023) explored driving tourist decision-making and applied (Wang et al., 2018a) to capture fine-grained preference patterns. Guo et al. (2020b) optimized the sampling strategy with max pooling and CNN to extract more effective nodes during propagation. Yu et al. (2020) adopted (Wang et al., 2018a) as the basic module and utilized GCN to model local high-order feature interactions, which are combined via a fully connected layer to learn global high-order feature interactions. The user representation fuses the output of (Wang et al., 2018a) and GCN in each layer. Yang et al. (2023a) modeled explicit user interest, which calculates the weighted sum of interacted items through the attention mechanism, and potential user interest, which propagates preferences through high-order connections in the KG.

Dong et al., (2022) and Wang et al., (2021) extended (Wang et al., 2018a) to a multi-task feature learning framework (Wang et al., 2019b), which incorporates the KGE module into the CTR prediction module by preference propagation:

$$\hat{y}_{uv} = \sigma(M^L(\sum_{h=1}^{H}\sum_{(e_{h_i},r_i,e_{t_i})\in\mathcal{S}_u^h}\frac{\exp(\mathbf{v}^T\mathbf{R}_i^h\mathbf{e}_{h_i}^h)}{\sum_{(e_{h_k},r_k,e_{t_k})\in\mathcal{S}_u^h}\exp(\mathbf{v}^T\mathbf{R}_k^h\mathbf{e}_{h_k}^h)}\mathbf{e}_{t_i}^h)^T\mathbf{v}^L). \tag{38}$$

Wang et al. (2022) utilized GAT to aggregate one-hop neighbors and enriched the representation of user preferences. This model addresses the limitation that GAT lacks full use of relations between head entities and tail entities. It divides associated triplets of each clicked item (head entity) into different triplet group $TG_i$ according to the different relation $r_i$. For each $TG_i$, the aggregation from tail entities to head entities can be expressed as

$$\begin{cases} s_k = FNN(\mathbf{W}\mathbf{e}_{t_k}\oplus\mathbf{W}\mathbf{e}_h) \\ \bar{\mathbf{e}}_h = \sum_{k=1}^{N+1}\mathrm{softmax}(s_k)\mathbf{W}\mathbf{e}_{t_k}, \\ \tilde{\mathbf{e}}_h = \mathbf{r}_i\bar{\mathbf{e}}_h \end{cases} \tag{39}$$

where $s_k$ denotes the similarity between the head entity $e_h$ and tail entity $e_{t_k}$, $FNN(\cdot)$ is the feedforward neural network, $\bar{\mathbf{e}}_h$ and $\tilde{\mathbf{e}}_h$ denote the enriched head entity representation, and $N+1$ is the number of tail entities and the head entity itself in $TG_i$. Then all $\tilde{\mathbf{e}}_h$ learned from different triplet groups are concatenated to obtain user preference.

Li et al. (2019) designed a dual attention mechanism to make use of external knowledge: the inter-item attention mechanism to weigh the relevance between items and entities and the inter-layer attention mechanism to adaptively aggregate the user preference in each hop. The main difference is that it quantifies user-item interaction behaviors with explicit ratings to measure the degree of user preference,

$$\mathbf{u} = \sigma(\frac{w+o}{z+o}\mathbf{v} + (1-\frac{w+o}{z+o})\tilde{\mathbf{u}}). \tag{40}$$

where $\tilde{\mathbf{u}}$ denotes the user representation learned from context entities, $w$ is the explicit rating, $z$ is the upper limit of the rating, and $o$ is a set threshold.

### 4.3.2 Refinement of Item Representation

The works mentioned above utilize outward propagation iteratively to refine user representations. Another trend is to capture item-related higher-order interactions in an inward aggregation manner. Since the user has different preferences for different items, each neighbor in $\mathcal{N}_v^h$ $(h = 1, 2, \cdots, H)$

is assigned different weights with the attention mechanism. The procedure in each order $h$ is expressed as: (1) aggregating neighbors of entity $e_i$ via the aggregator $AGG(\cdot)$ to generate a context representation, $\mathbf{e}_{\mathcal{N}_i}^{h-1} = AGG\left(\mathbf{e}_j^{h-1}\right)$, $e_j \in \mathcal{N}_i^{h-1}$; (2) updating the $h$-order entity representation with the combination of $\mathbf{e}_i^{h-1}$ and $\mathbf{e}_{\mathcal{N}_i}^{h-1}$ by the updater $g(\cdot)$, $\mathbf{e}_i^h = g\left(\boldsymbol{e}_i^{h-1}, \mathbf{e}_{\mathcal{N}_i}^{h-1}\right)$.

These attentive aggregators utilize the information from the knowledge triplets to determine the weights for each edge in the graph. Wang et al. (2019c) proposed KG convolutional networks to capture both high-order structure and semantic information in the KG automatically. The weight of each neighbor $\pi_j$ characterizes the importance of relation $r$ to specific user $u$,

$$\pi_j = \mathbf{u}^T \mathbf{r}. \tag{41}$$

The aggregation of neighbor information is as follows,

$$\begin{cases} \mathbf{e}_{\mathcal{N}_i}^{h-1} = \sum_{e_j \in \mathcal{N}_i^{h-1}} \tilde{\pi}_j \mathbf{e}_j^{h-1} \\ \tilde{\pi}_j = \frac{\exp(\pi_j)}{\sum_{e_j \in \mathcal{N}_i^{h-1}} \exp(\pi_j)} \end{cases}, \tag{42}$$

where $\tilde{\pi}_m$ is normalized attention score.

Tang et al. (2022) improved the attention mechanism when capturing inter-item relations, which considers users, relations and entities:

$$\pi = \alpha g(\mathbf{u}, \mathbf{r}) + (1 - \alpha) g(\mathbf{u}, \mathbf{e}), \tag{43}$$

where $\alpha$ is to balance the weights of two vectors.

He et al. (2023) pointed out it is important to consider that unrated items may not have been presented to the user, rather than assuming that they are disliked or unintended. To mitigate exposure bias, they exploited user similarity based on the items clicked by both users and only one user. Given the user similarity connections, layer-wise GCN generates user representations. Moreover, this model captures correlations between high-order and low-order neighbor information during propagation.

$$\mathbf{e}_{\mathcal{N}_i}^h = \sum_{e_j \in \mathcal{N}_i^h} \tilde{\pi}_j s_{h-1,h} \mathbf{e}_j^h, \tag{44}$$

where $s_{h-1,h}$ denotes the cosine similarity of relation $r$ of order $h-1$ and $h$.

Xie et al. (2022) applied embedding propagation to agricultural products and used AFM to aggregate neighbors:

$$\mathbf{e}_{\mathcal{N}_m}^{h-1} = \sum_{e_i \in \mathcal{N}_m^{h-1}} \sum_{e_j \in \mathcal{N}_m^{h-1}} \pi_{ij} \tilde{\mathbf{e}}_i^{h-1} \odot \tilde{\mathbf{e}}_j^{h-1}, \tag{45}$$

where $\tilde{\mathbf{e}}_i^{h-1}$ and $\tilde{\mathbf{e}}_j^{h-1}$ are weighted representations of neighbor $e_i$ and $e_j$, respectively; $\pi_{ij}$ denotes the attention weight between two neighbors.

Li et al. (2023a) utilized multi-head attention in the propagation process to enhance the model's expressive capability.

$$\mathbf{e}_i^h = ||_{k=1}^K \sigma\left(\sum_{j \in \mathcal{N}_i^h \cup \{i\}} \pi_{ij}^{h,k} \mathbf{W}^{h,k} \mathbf{e}_j^{h-1,k}\right), \tag{46}$$

where $k$ denotes the $k$-th head, and $\mathbf{W}^{h,k}$ is a learnable parameter.

Chen et al. (2024) argued that both an exact match and a sense of discovery (i.e., diversity and serendipity) created by the item can improve CTR accuracy. Therefore, they applied cognitive psychology-related theories to activate diversified and high-ordered entities in KG. For the neighbor $e_j$, the activation probability is

$$\pi_j = \frac{\beta(\mathbf{e}_i^h, \mathbf{r}_{ij}^h)\beta(\boldsymbol{e}_j^h, \boldsymbol{r}_{ij}^h)}{dis(\mathbf{e}_i^h, \mathbf{e}_i^h)}, \tag{47}$$

where $\beta$ denotes the probability of the entity along with the relation and can be calculated by element-wise summation, $dis(\cdot)$ denotes the distance between the two entities. Thus, the item representation is refined by aggregating activated entities.

Some other works (Chen et al., 2022b; Duan et al., 2023a; Feng et al., 2020a; Hou et al., 2021; Hu et al., 2023; Lee et al., 2020; Li et al., 2023b; Song et al., 2023; Tang et al., 2024; Wang et al., 2023c; Wu et al., 2022; Yao et al., 2023; Zhang et al., 2021) not only learn item features from KG, but also capture user behaviors and collaborative signals from user-item interactions. Specifically, Hou et al. (2021) considered (Wang et al., 2019c) ignored inter-user and user-item interactions and incorporated a user-based collaborative filtering algorithm. Song et al. (2023) used the model in (Wang et al., 2019c) to mine local interest features and GAN to learn global interest features. Zhang et al. (2021) not only modeled higher-order interactions in the KG to learn item representations but also constructed a user-adjacency graph and user-features graph to learn user representations by GCN. Lee et al. (2020) utilized GNN to model topic-enriched KG for news CTR prediction. The final representation of a news title is the concatenation of word-level and knowledge-level representations. The user's final embedding is represented by clicked news. Hu et al. (2023) utilized GAT to learn item representations and multi-head self-attention mechanism to learn user representations with items reflecting user's preference and dislike. Li et al. (2023b) adopted GAT to learn knowledge-aware item representations and implemented four types of operations to aggregate neighbors, including concat pooling, average pooling, maximum pooling and minimum pooling aggregators. Tang et al. (2024) enriched item representations by fusing user preference embeddings and knowledge embeddings learned from GCN and TransH, respectively.

Chen et al. (2022b) and Duan et al. (2023a) proposed attentive knowledge-aware graph convolutional networks with collaborative guidance to balance information aggregation from user-item interactions and KG. To extract user-item interactive information, collaboration attention given the target interaction pair is formulated as

$$\pi_{uv} = \mathbf{u}^T \mathbf{M}_{r_{uv}} \mathbf{v}, \tag{48}$$

where $\mathbf{M}_{r_{uv}}$ denotes the transformation matrix for the relation between the user and interacted items. User-centric and item-centric interactive information is aggregated and updated by information propagation. Then three types of guidance signal encoders are implemented to encode user-item pairs:

(1) Sum encoder utilizes summation of representations $\mathbf{u}$ and $\mathbf{v}$.

$$f_{sum} = \mathbf{u} + \mathbf{v} \tag{49}$$

(2) Pairwise-max encoder takes the element-wise maximum values of two representations.

$$f_{pmax} = \text{pmax}(\mathbf{u}, \mathbf{v}) \tag{50}$$

(3) Linear combination encoder receives a linear combination of two representations.

$$f_{comb} = \alpha\mathbf{u} + (1 - \alpha)\mathbf{v}, \alpha \in (0,1) \tag{51}$$

External knowledge is extracted with the guidance of collaborative signals. The attention score $\pi_{uve}$ between the user-item pair and the triplet in the KG is calculated as

$$\begin{cases} \pi_{uve} = \mathbf{v}^T \mathbf{M}_{r_{uve}} \mathbf{e} \\ \mathbf{M}_{r_{uve}} = f(\mathbf{u}, \mathbf{v}) \odot \mathbf{M}_r \end{cases}, \tag{52}$$

where $\mathbf{M}_r$ denotes relation-specific matrix for KG and $f(\mathbf{u}, \mathbf{v})$ denotes collaborative signal guidance. Item-related high-order knowledge is captured by GCN-based representation learning.

Dai et al. (2022) modeled the user-item graph and KG separately and simultaneously with two GNNs. The former leverages GCN to learn high-order collaborative signals as illustrated in Equation 5. The latter designs personalized GAT to capture fine-grained semantics, in which the attention score considers user features, target entity, relation and neighbor entity simultaneously:

$$\pi = f\left(\mathbf{u}, \mathbf{e}_i, \mathbf{r}_{ij}, \mathbf{e}_j\right) \tag{53}$$

There are four types of functions to implement $f(\cdot)$:

(1) Inner-Product attention utilizes inner-product to learn relation-aware and entity-aware attention scores.

$$\pi = \mathbf{u}\mathbf{r}_{ij} + \mathbf{e}_i\mathbf{e}_j \tag{54}$$

(2) Sum attention takes the summation of four representations, followed by a nonlinear transformation.

$$\pi = \sigma(\mathbf{W}(\mathbf{u} + \mathbf{e}_i + \mathbf{r}_{ij} + \mathbf{e}_j) + \mathbf{b}) \tag{55}$$

(3) Concat attention concatenates the four representations.

$$\pi = \sigma(\mathbf{W}(\mathbf{u} \oplus \mathbf{e}_i \oplus \mathbf{r}_{ij} \oplus \mathbf{e}_j) + \mathbf{b}) \tag{56}$$

(4) Bi-interaction combines two operators of summation and element-wise product.

$$\pi = \sigma\left(\mathbf{W}_1\left(\mathbf{u} + \mathbf{e}_i + \mathbf{r}_{ij} + \mathbf{e}_j\right) + \mathbf{b}_1\right) + \sigma\left(\mathbf{W}_2\left(\mathbf{u} \odot \mathbf{e}_i \odot \mathbf{r}_{ij} \odot \mathbf{e}_j\right) + \mathbf{b}_2\right) \tag{57}$$

The entity representations are learned by stacked aggregation layers. The final item representations are generated by combinations of the user-item graph and KG.

Wu et al. (2022) improved (Wang et al., 2019c) by introducing user behavior information such as browsing and adding-to-cart, which describes the user's motivation in user-item interactions. Each user behavior graph with the behavior type $t$ is defined as $\mathcal{BG}_t$, which includes multiple paths that start from a specific user and end with interacted items. The representation learning process for user behaviors can be expressed as

$$\mathbf{b}_t = g_t(\mathcal{BG}_t). \tag{58}$$

The comparative learning is utilized to learn the differences between the user behavior graphs. The negative cases are generated as

$$\hat{\mathbf{b}}_t = g_{t-1}(\mathcal{BG}_t). \tag{59}$$

The positive and negative samples are compared by distance function $d(\cdot)$:

$$\mathcal{L}_{t-1,t} = -\ln \frac{\exp\left(d(\mathbf{b}_t, \hat{\mathbf{b}}_t)\right)}{\exp\left(d(\mathbf{b}_t, \hat{\mathbf{b}}_t)\right) + \exp\left(d(\mathbf{b}_t, \mathbf{b}_{t-1})\right)}. \tag{60}$$

The final user representation is the aggregation of user embedding and behavior representations. What's more, the main difference of item representations is that a multi-headed self-attentive mechanism follows GCN to further extract association from the neighborhood information.

Feng et al. (2020a) constructed an adaptive target-behavior relational graph with graph connect and graph prune techniques to effectively distill structural information over KG. The paths connecting entities and items are restored and entities not connecting different items are pruned. The next step is to propagate user preference over the extracted sub-graph for target user-item pairs by relation-aware aggregation.

$$\begin{cases} \pi_{hrt} = \mathrm{softmax}(\mathbf{r}\mathbf{W}_1 f(\mathbf{e}_h^h \oplus \mathbf{e}_t^h)) \\ \mathbf{e}_{\mathcal{N}_h}^h = \sum_{(e_h, r, e_t) \in \mathcal{S}_{e_h}^h} \pi_{hrt}\, \mathbf{e}_t^h \end{cases}, \tag{61}$$

where $\mathcal{S}_{e_h}^h$ is the triplet set connecting with head entity $e_h$ in the $h$-th layer, and $f(\cdot)$ denotes the single-layer perceptron. The user representation is obtained with related items by an attention mechanism:

$$\begin{cases} \pi_j = \mathrm{softmax}(\tilde{\mathbf{v}}_j \mathbf{W}_2 \tilde{\mathbf{v}}) \\ \tilde{\mathbf{u}} = \sum_{v_j \in \mathcal{V}^u} \pi_j\, \tilde{\mathbf{v}}_j \end{cases}, \tag{62}$$

where $\tilde{\mathbf{v}}$ denotes the learned representation of target item, $\tilde{\mathbf{v}}_j$ denotes the learned representation of the user's interacted item in the set $\mathcal{V}^u$, and $\tilde{\mathbf{u}}$ denotes the final user representation.

Yao et al. (2023) constructed a user-item bipartite graph and KG to learn user and item representations, respectively. KG is embedded by DistMult (Yang et al., 2015) with the bilinear score function

$$f_r(e_h, e_t) = \mathbf{e}_h \mathbf{M}_r \mathbf{e}_t, \tag{63}$$

where $\mathbf{M}_r$ is a relation-specific diagonal matrix. They designed a hierarchical network structure to balance semantic representations generated from text information and graph representations learned by GCN. The process can be formulated as follows:

$$\mathcal{H}(\mathbf{e}_{gp}, \mathbf{e}_{se}) = \frac{{\mathbf{e}_{gp}}^\top \mathbf{g}}{(\mathbf{e}_{gp} + \mathbf{e}_{se})^\top \mathbf{g}} \mathbf{e}_{gp} + \frac{{\mathbf{e}_{se}}^\top \mathbf{g}}{(\mathbf{e}_{gp} + \mathbf{e}_{se})^\top \mathbf{g}} \mathbf{e}_{se}, \tag{64}$$

where $\mathbf{e}_{gp}$ and $\mathbf{e}_{se}$ denote graph representation and semantic representation, respectively; $\mathbf{g}$ is a gate vector to balance the fusion ratio.

Wang et al. (2023c) proposed relation-aware translating embeddings via dynamic mapping matrix (RTransD) to embed KG and the loss function is

$$f_r(e_h, e_t) = \left\| (\mathbf{w}_r^h {\mathbf{w}_e}^\top + \mathbf{I})\mathbf{e}_h + \mathbf{r} - (\mathbf{w}_r^t {\mathbf{w}_e}^\top + \mathbf{I})\mathbf{e}_t \right\|_2^2, \tag{65}$$

where $\mathbf{w}_r^h$ and $\mathbf{w}_r^t$ are two relation mapping vectors for the head and tail entity, respectively; $\mathbf{w}_e$ denotes the entity mapping vectors. The learned embeddings are aggregated to generate item representations, which are then fed into Transformer to model user behavior sequences and gain user interest. Finally, an MLP network combines the user interest and item representation and outputs the prediction.

The next issue is how to update the entity representation. Among the following three types of updaters, the sum updater and concat updater are the most commonly implemented.

(1) Sum updater utilizes the summation of two representations as the interaction operator.

$$\mathbf{e}_i^h = \sigma(\mathbf{W}(\boldsymbol{e}_i^{h-1} + \mathbf{e}_{\mathcal{N}_i}^{h-1}) + \mathbf{b}) \quad (66)$$

(2) Concat updater concatenates two representations with the concatenation operator.

$$\mathbf{e}_i^h = \sigma(\mathbf{W}(\boldsymbol{e}_i^{h-1} \oplus \mathbf{e}_{\mathcal{N}_i}^{h-1}) + \mathbf{b}) \quad (67)$$

(3) Neighbor updater only receives neighbor representation to update the node representation.

$$\mathbf{e}_i^h = \sigma(\mathbf{W}\mathbf{e}_{\mathcal{N}_i}^{h-1} + \mathbf{b}) \quad (68)$$

Finally, the $H$-order entity representation incorporates itself and context information up to $H$ hops away. Thus, the user's potential interests can be extended in the knowledge in a wider way.

Some works further improve the item propagation in KG. In general user-item interactions are sparse, thus GCN utilizing relation score as the only supervised signal tends to cause overfitting. As the extensions of (Wang et al., 2019c), two works (Wang et al., 2019a; Ma et al., 2022a) focus on solving this issue. Wang et al. (2019a) extended GNNs architecture to learn personalized item representations in KG and developed label smoothness regularization to assist the training of edge weights. The layer-wise GNN can be expressed as Equation 5, in which $\mathbf{A}[i,j]$ is calculated from the relation score between entity $e_i$ and $e_j$ in Equation 41. Motivated by the label smoothness assumption that adjacent entities in the KG are likely to have similar relevancy labels, the energy function is minimized as

$$E(l_u, \mathbf{A}) = \frac{1}{2}\sum_{e_i \in \mathcal{E}, e_j \in \mathcal{E}} \mathbf{A}[i,j](l_u(e_i) - l_u(e_j))^2, \quad (69)$$

where $l_u(v) = 1$ if the user $u$ has engaged with item $v$, otherwise $l_u(v) = 0$. The loss between the true label $y_{uv}$ and the predicted label $\hat{l}_u(v)$ serves as a supervised signal for updating the edge weights matrix $\mathbf{A}$:

$$\mathcal{L}_{LS} = \sum_{u \in \mathcal{U}, v \in \mathcal{V}} \mathcal{J}(y_{uv}, \hat{l}_u(v)). \quad (70)$$

Ma et al. (2022a) also solved the over-smoothing problem of GNNs and proposed KG random neural networks. Concretely, a random dropout strategy is designed to generate the perturbed entity feature matrices $\tilde{\mathbf{E}}^s$ ($s = 1, 2, \cdots, S$) for $S$ times by randomly dropping out some feature representations. Since the dropped information is complemented by neighbors, the structural and feature information is not damaged. Then high-order neighbor information is propagated via GCN over the perturbed feature matrices. The objective function is

$$\begin{cases} \mathcal{L} = \mathcal{L}_{base} + \lambda_1 \mathcal{L}_{con} + \lambda_2 \|\Theta\|_2^2 \\ \mathcal{L}_{base} = \frac{1}{S}\sum_{s=1}^{S}(\sum_{(u,v)\in R} \mathcal{J}(y_{uv}, \hat{y}_{uv}{}^s) - \sum_{(u,v)\notin R} \mathcal{J}(y_{uv}, \hat{y}_{uv}{}^s)) \\ \mathcal{L}_{con} = \frac{1}{S-1}\sum_{s=2}^{S}\|\bar{\mathbf{E}}^s - \bar{\mathbf{E}}^{s-1}\|_2^2 \end{cases}, \quad (71)$$

where $\bar{\mathbf{E}}^s$ denotes the entity feature matrix after propagation in the $s$-th time.

Another trend is to perform graph representation learning under the multi-task learning framework. Shu and Huang (2021) considered three modules: recommender system module, KG feature learning module and KG structure learning module. With a similar structure to (Wang et al., 2019b), these three modules share information through cross and exchange units. The recommender system module obtains user representations by MLP and item representations from cross units. KG feature learning module updates the representation of head entities with transferred information from cross and exchange units and learns relation representations by MLP. KG structure learning module generates entity representations from exchange units and utilizes GAT to aggregate neighbor information. With information transfer, multiple tasks can correct each other and assist each other. Likewise, Shu and Huang (2023) adopted GCN to perform KG structure learning. Chen et al. (2022a) and Xiao et al. (2022) proposed a multi-task feature learning framework based on alternate learning, which adopts DeepFM to capture collaborative signals and GCN to mine entity features in the KG.

### 4.3.3 Refinement of Both User and Item Representation

Another type of models is to integrate the user-item interaction graph and KG and form the user-item KG. The joint learning of collaborative signals and external knowledge refines both user and item representations. Their main difference is to design different attention mechanisms to aggregate neighbors.

Qu et al. (2019) observed that most previous models exhibit early summarization behavior in aggregating neighborhood information before learning user-item interactions and neglect neighbor-neighbor interactions. To solve the problem, they proposed a bi-attention network to enrich neighborhood information.

$$\begin{cases}\pi_{ij} = \mathrm{softmax}_{ij}(\mathbf{W}(\mathbf{u}^h \oplus \mathbf{e}_i^h \oplus \mathbf{v}^h \oplus \mathbf{e}_j^h) + \mathbf{b}) \\ \hat{y}_{uv} = \sum_{i \in \mathcal{N}_u} \sum_{j \in \mathcal{N}_v} \pi_{ij}\, \mathbf{e}_i^{h^T} \mathbf{e}_j^h \end{cases}, \tag{72}$$

where $\mathbf{u}^h$, $\mathbf{v}^h$, $\mathbf{e}_i^h$ and $\mathbf{e}_j^h$ denote $h$-order representations of the user $u$, item $v$, user's neighbor and item's neighbor, respectively. The high-order neighborhood information can be obtained by GCN or GAT.

Wang et al. (2020c), Qian et al. (2022), Ma et al. (2022b), Liu et al. (2023) and Wang et al. (2024a) employed a heterogeneous propagation strategy to encode both collaborative signals and knowledge associations. Given each triplet $(e_h, r, e_t)$ related to the head entity $e_h$, the attentive weight to aggregate tail entities is calculated as

$$\begin{cases}\mathbf{z} = ReLU(\mathbf{W}_1(\mathbf{e}_h \oplus \mathbf{r}) + \mathbf{b}_1) \\ \pi = \sigma(\boldsymbol{W}_3 ReLU(\boldsymbol{W}_2 \boldsymbol{z} + \boldsymbol{b}_2) + \boldsymbol{b}_3) \end{cases}. \tag{73}$$

The user representations $\mathbf{u}^1$, $\mathbf{u}^2$,…$\mathbf{u}^H$ or item representations $\mathbf{v}^1$, $\mathbf{v}^2$,…$\mathbf{v}^H$ obtained in each layer are aggregated through aggregators. Then inner product of two representations is conducted to output the clicking probability. After embedding propagation, Ma et al. (2022b) also explored transductive learning to model unknown items, in which learned item representations are fed into two individual knowledge propagation layers to further extrapolate the KG.

Duan et al. (2023b) observed that some existing works fail to capture relations in the neighborhood and proposed relation-fused attention to characterize each neighbor:

$$\begin{cases} \mathbf{z} = \mathbf{W}_1(\mathbf{e}_h \oplus \mathbf{r} \oplus \mathbf{e}_t) \\ \pi = LeakyReLU(\mathbf{W}_2\mathbf{z}) \end{cases}. \tag{74}$$

Bai et al. (2023) considered the difference of feature information within different relations and proposed a relation-enhanced GCN, which aggregates the high-order neighbor nodes under different relations.

$$\mathbf{e}_{\mathcal{N}_i}^h = \sum_{r\in\mathcal{R}} \sum_{e_j\in\mathcal{N}_i^r} \pi_{ij} \mathbf{W}_r^{h-1} \mathbf{e}_j^{h-1}, \tag{75}$$

where $\pi_{ij}$ denotes the edge weight, $\mathbf{W}_r^{h-1}$ is a transformation matrix under the relation $r$, and $\mathcal{N}_i^r$ is the set of neighbors of the $e_i$ under the relation $r$.

Elahi and Halim (2022) exploited the user-specific attention mechanism to assign different importance to neighbor entities:

$$\begin{cases} \mathbf{z} = (\mathbf{r} \oplus \mathbf{e}_t)\mathbf{W}_1 \\ \tilde{\mathbf{u}} = ReLU(\mathbf{u}\mathbf{W}_2 + \mathbf{b}_1) \\ \pi = tanh\,((\mathbf{e}_h \oplus \mathbf{z})\mathbf{W}_3 + \mathbf{b}_2)\tilde{\mathbf{u}} \end{cases}, \tag{76}$$

where $\tilde{\mathbf{u}}$ is to reflect the personalized preference calculated from user embedding.

Similarly, Elahi et al. (2024) explicitly encoded both relational and contextual information by high-order propagation and also designed a user-specific attention mechanism to aggregate neighbors personally, which can be calculated as

$$\pi = \mathbf{W}\mathbf{e}_t tanh(\mathbf{W}\mathbf{e}_h + \mathbf{r})\tilde{\mathbf{u}}. \tag{77}$$

Wang et al. (2024b) proposed a fine-grained attention network to integrate high-order connections along relation paths. The attention weight to aggregate neighbors is calculated as

$$\begin{cases} \mathbf{z} = ReLU(\mathbf{W}_1((\mathbf{e}_h + \mathbf{N}_{e_h}) \oplus (\mathbf{r}^h \odot \mathbf{p}^h)) + \mathbf{b}_1) \\ \pi = \sigma(\mathbf{W}_3 ReLU(\mathbf{W}_2\mathbf{z} + \mathbf{b}_2) + \mathbf{b}_3) \end{cases}, \tag{78}$$

where $\mathbf{p}^h$ is the relation representation of the path from the entity $\mathbf{e}_h^1$ to $\mathbf{e}_t^{h-1}$, $\mathbf{N}_{e_h}$ denotes the representation of entities connected on the path.

Lyu et al. (2022) proposed an attention mechanism based on recurrent neural network for long-distance propagation over KG. The attention weights of neighbor entities can control the attenuation of information propagation:

$$\begin{cases} \mathbf{x} = \mathbf{e}_h \odot \mathbf{r} \\ \mathbf{z} = tanh(\mathbf{W}_z\mathbf{x} + \mathbf{W}_h\mathbf{h}_0 + \mathbf{b}_z) \\ \mathbf{f} = \sigma\left(\mathbf{W}_f\mathbf{x} + \mathbf{W}_h\mathbf{h}_0 + \mathbf{b}_f\right) \\ \mathbf{i} = \sigma(\mathbf{W}_i\mathbf{x} + \mathbf{W}_h\mathbf{h}_0 + \mathbf{b}_i) \\ \mathbf{o} = \sigma(\mathbf{W}_o\mathbf{x} + \mathbf{W}_h\mathbf{h}_0 + \mathbf{b}_o) \\ \mathbf{c} = \mathbf{f} \odot \mathbf{c}_0 + \mathbf{i} \odot \mathbf{z} \\ \pi = \mathbf{o} \odot tanh\,(\mathbf{c}) \end{cases}, \tag{79}$$

where $\mathbf{x}$ is the input of attention; $\mathbf{z}$ is the update of cell state; $\mathbf{h}_0$ and $\mathbf{c}_0$ denote the hidden states with random initialization; $\mathbf{W}_z$, $\mathbf{W}_f$, $\mathbf{W}_i$, and $\mathbf{W}_o$ denote transformation matrices; $\mathbf{b}_z$, $\mathbf{b}_f$, $\mathbf{b}_i$, and $\mathbf{b}_o$ denote bias; $\mathbf{i}$, $\mathbf{o}$ and $\mathbf{f}$ represent input, output and forget gate, respectively.

Tai et al. (2020) leveraged preference propagation similar to (Wang et al., 2018a) to generate knowledge-enhanced user preferences and designed a mixing layer to refine item representations. For depth, high-order entity information $\mathbf{e}_w^d$ and neighbor information $\mathbf{n}_w^d$ are aggregated to generate the next-order representation $\mathbf{e}_w^{d+1}$. For width, mixing the layer-wise GCN information allows for comparisons of entity-entity interactions in different orders.

$$\begin{cases}\mathbf{e}_{w+1}^1 = \mathbf{M}_w\left(\mathbf{e}_w^1 \oplus \mathbf{e}_w^2 \cdots \oplus \mathbf{e}_w^{l_d}\right) \\ \mathbf{e}_w^{d+1} = \sigma(\mathbf{W}(\mathbf{e}_w^d + \mathbf{n}_w^d) + \mathbf{b})\end{cases}. \tag{80}$$

where $\mathbf{M}_w$ is the layer matrix to generate the next wide layer entity representation $\mathbf{e}_{w+1}^1$; $w$=1, ⋯, $l_w$-1, and $d$=1, ⋯, $l_d$-1; $l_w$ and $l_d$ are the number of wide and deep layers, respectively.

Tu et al. (2021) focused on effective KG distillation and refinement. They used TransH (Wang et al., 2014) first to learn latent embeddings of entities and relations in the KG:

$$f_r(e_h, e_t) = \left\|\left(\mathbf{e}_h - \mathbf{w}_r^\top \mathbf{e}_h \mathbf{w}_r\right) + \mathbf{r} - \left(\mathbf{e}_t - \mathbf{w}_r^\top \mathbf{e}_t \mathbf{w}_r\right)\right\|_2^2. \tag{81}$$

Knowledge-aware attention is defined to propagate embeddings on the KG based on GCN:

$$\pi = \cos\left(\mathbf{e}_h + \mathbf{r}, \mathbf{e}_t\right), \tag{82}$$

where $\pi$ is regarded as sampled probability to distill subgraphs rather than the whole graph. To refine the KG, conditional attention mechanism over propagation is adopted for layer $h$, which contains two aspects: above attention $\pi_j$ independent of target and $\pi_{j|\mathcal{T}}$ to measure the importance of the entity to each target user-item pair $\mathcal{T}$,

$$\begin{cases}\pi_{j|\mathcal{T}} = \mathbf{a}^T(\mathbf{W}_1 \mathbf{e}_{\mathcal{T}}^h \oplus \mathbf{W}_2 \mathbf{e}_j^h) \\ \alpha_{j|\mathcal{T}} = \mathrm{softmax}(LeakyReLU(\pi_j \pi_{j|\mathcal{T}}))\end{cases}, \tag{83}$$

where $\alpha_{j|\mathcal{T}}$ is the conditional attention, $\mathbf{a}$ is a weight vector, $\mathbf{e}_{\mathcal{T}}^h$ denotes the concatenation of entities in target $\mathcal{T}$. After conducting information propagation, the final user and item representations are obtained.

Wang et al. (2023b) proposed a GCN-based aggregation model, which replaces the binary adjacency matrix with a relation-aware attentional adjacency matrix. The convolution computation on a given graph under relation $r_i$ can be expressed as

$$\mathbf{X}_{r_i}^{(h+1)} = \sigma(\mathbf{A}_{r_i} \mathbf{X}_{r_i}^{(h)} \mathbf{W}^{(h+1)} + \mathbf{b}^{(h+1)}), \tag{84}$$

where $\mathbf{X}_{r_i}^{(h+1)}$ denotes the node representations matrix, and $\mathbf{A}_{r_i}$ denotes the attentional adjacency matrix. The representations of potentially interested items in the KG and the interacted item sequence learned by Bi-GRU are integrated to obtain the user representations. A similar process is employed for the item-side representations.

Wang et al. (2023a) aggregated mixed-curvature manifold vectors in the tangent space to capture local geometric structural properties.

$$\mathbf{e}_i^h = \exp(\sum_{j\in\mathcal{N}_i^h} \pi_{ij}^h \log \mathbf{e}_j^{h-1}), \tag{85}$$

where $\pi_{ij}^h$ is the weight measuring the importance of the neighbor node $e_j$ to the central node $e_i$.

Another trend is to investigate contrastive learning between different parts of the graph. Zou et al. (2022a) proposed multi-level cross-view contrastive learning for knowledge-aware CTR prediction. For local-level contrastive learning, Light-GCN is utilized to encode collaborative user-item graph and semantic item-entity graph.

$$\mathcal{L}^{local} = -\log \frac{e^{\cos(\mathbf{v}_i^s,\mathbf{v}_i^c)/\tau}}{e^{\cos(\mathbf{v}_i^s,\mathbf{v}_i^c)/\tau} + \sum_{k\neq i} e^{\cos(\mathbf{v}_i^s,\mathbf{v}_k^s)/\tau} + \sum_{k\neq i} e^{\cos(\mathbf{v}_i^s,\mathbf{v}_k^c)/\tau}}, \tag{86}$$

where $\mathbf{v}_i^s$ and $\mathbf{v}_i^c$ denote the representation of item $v_i$ learned from collaborative and semantic, respectively; $\tau$ is a temperature parameter. The user-item-entity graph is considered as a global structural view and encoded by path-aware GNN to aggregate multi-hop neighbors. Global-level contrastive learning is performed between the global-level view and the local-level view. The local-view contrastive loss $\mathcal{L}_v^g$ and global-view contrastive loss $\mathcal{L}_v^l$ in the item side is defined as

$$\begin{cases} \mathcal{L}_v^g = -\log \frac{e^{\cos(\mathbf{v}_i^g,\mathbf{v}_i^l)/\tau}}{e^{\cos(\mathbf{v}_i^g,\mathbf{v}_i^l)/\tau} + \sum_{k\neq i} e^{\cos(\mathbf{v}_i^g,\mathbf{v}_k^g)/\tau} + \sum_{k\neq i} e^{\cos(\mathbf{v}_i^g,\mathbf{v}_k^l)/\tau}} \\ \mathcal{L}_v^l = -\log \frac{e^{\cos(\mathbf{v}_i^l,\mathbf{v}_i^g)/\tau}}{e^{\cos(\mathbf{v}_i^l,\mathbf{v}_i^g)/\tau} + \sum_{k\neq i} e^{\cos(\mathbf{v}_i^l,\mathbf{v}_k^l)/\tau} + \sum_{k\neq i} e^{\cos(\mathbf{v}_i^l,\mathbf{v}_k^g)/\tau}} \end{cases}, \tag{87}$$

where $\mathbf{v}_i^g$ and $\mathbf{v}_i^l$ denote the representation of item $v_i$ learned from global and local views, respectively. The contrastive loss $\mathcal{L}_u^g$ and $\mathcal{L}_u^l$ in the user side can be generated in a similar way. Then the global contrastive loss is defined as

$$\mathcal{L}^{global} = \frac{1}{2|\mathcal{V}|}\sum_{v\in\mathcal{V}}(\mathcal{L}_v^g + \mathcal{L}_v^l) + \frac{1}{2|\mathcal{U}|}\sum_{u\in\mathcal{U}}(\mathcal{L}_u^g + \mathcal{L}_u^l). \tag{88}$$

Similarly, Ong et al. (2023) took user-item, item-entity and user-entity graphs as local views and user-item-entity collaborative KG as global views to perform local-level and global-level contrastive learning. Later, Zou et al. (2022b) proposed a follow-up contrastive learning model, which conducts layer-wise contrasting between different parts within graphs. First, local graphs consisting of user-item interactions and related knowledge triplets and non-local graphs with high-order co-occurrence items and external knowledge facts are constructed. To balance information utilization, intra-graph contrastive learning is facilitated between sparse collaborative signals and redundant knowledge connections in each local and non-local graph. Then inter-graph contrastive learning is conducted between local and non-local graphs to provide adequate knowledge extraction. Liu et al. (2023) explored self-supervised signals and employed contrastive learning to enhance representations from the user and item views. Wang et al. (2024b) performed contrastive learning to learn the potential semantic information within the KG, which utilizes the certain propagation layer and the other layers to form positive pairs and negative pairs, respectively.

Some other works also jointly learn the collaborative signal and KG to refine user and item representations simultaneously in various application scenarios. Ma et al. (2021) achieved latent feature sharing between collaborative signals and KG under the multi-task learning framework, in which GAT is used to propagate preference with both users' and items' neighbors. Cao et al. (2021) applied KG to match appropriate machine learning methods for given datasets. They constructed relevant KG incorporating description information for dataset and method entities, which are modeled by high-order information aggregation based on the framework of GNN. Cui et al. (2024) leveraged fine-grained features from user reviews, which are aligned with KG entities to characterize user preferences, then aggregated neighborhood information weighted by sentiment relationships to generate personalized representations of users and items. Li et al. (2024) utilized models proposed in (Wang et al., 2018a, 2019c) to learn user and item representations, respectively. Sun and Li (2022) fused multi-source auxiliary information for music recommendation and captured users' high-order personalized interests in the music KG. Peng et al. (2023) divided the user behavior sequence into multiple session graphs, where GAT is applied to represent the interest of the session. At the same time, the user's interest in each session is propagated on the KG. Then, to capture evolving user interests, Bi-LSTM is utilized to model the interactions between sessions.

#### 4.3.4 Summary of Propagation-based Models

Obviously, the first two categories of models refine only one type of entity and rarely model explicit collaborative signals, which leads to insufficient learning and suboptimal performance. However, the third category of models deserve an in-depth study of integration of entities and relations from different latent spaces.

### 4.4 Summary

Embedding-based models exhibit greater flexibility but are more suitable for graph completion or other within-graph applications. They overlook high-order interaction relationships among entities in the KG and may not capture the nuanced complexities of user-entity interactions. Path-based models leverage the KG more naturally and intuitively. However, designing effective meta-paths can be challenging and cumbersome. Decomposing the intricate user-item connectivity into individual linear paths may result in a loss of valuable information. These models struggle to sufficiently explore the fundamental factors driving user-item interactions. The deeper they delve into multi-hop paths, the further they may drift away from accurately capturing user preferences (Fan et al., 2022). Moreover, in sparse scenarios, the quality of path-based models may be compromised. Propagation-based models, which have emerged as a recent research trend, combine the advantages of embedding-based and path-based models by utilizing the idea of embedding propagation. These models autonomously mine high-order paths and interactions. However, the increasing number of neighbors in large datasets poses a challenge to computational complexity.

## 5. Datasets and Evaluations

In this section, we summarize the datasets, KG and evaluation metrics used in KG-based CTR prediction models and discuss experimental results reported in each paper.

## 5.1 Datasets and KG

### 5.1.1 Datasets Description

Table A2 shows the detailed information of each dataset, such as statistics of users, items and interactions. The most common public datasets are MovieLens-1M, MovieLens-20M, Book-Crossing and Last.FM. Most private datasets are scraped from different websites based on various application scenarios.

### 5.1.2 KG Construction

The CTR prediction dataset is linked to an external knowledge base to obtain rich information. Knowledge repositories exploited in our collected papers are summarized in Table A3. The commonly used knowledge repositories are Microsoft Satori, Freebase and Wikidata.

We summarize the most common construction of relevant KG for each dataset. For the dataset MovieLens-1M, MovieLens-20M, Book-Crossing and Last.FM, the KG is commonly constructed based on the external knowledge base Microsoft Satori. Specifically, the first step is to extract a subset of triples with relations to the certain scenario and confidence levels above 0.9 from the complete KG. Given the sub-KG, the following step is to identify the ID of the item matching with the tail of triples. To ensure simplicity, items with no matching entities or multiple matching entities are discarded. The final step is to match the IDs with the heads and tails of all triples and select the well-matched ones from the sub-KG. For the dataset Bing-News and Adressa News, the construction process is similar except that entities are extracted in news titles using entity linking tools and relations are not restricted in any domain. For the dataset IMDb and Amazon-movies&TV-A, entities in the reviews are linked to the KG Wikidata and extended to one-hop distance. For the dataset Amazon-Movies&TV-B, Amazon-Book-D, Amazon-Music, Amazon-Clothing, Amazon-Cross, Official-Account, Mini-Program, Nudge, MOOCCube, Agricultural-Products, Fund and ML, KG represents relations between items or item attributes without the support of external knowledge, such as purchased together by the same user, purchased after viewing the reviews of an item, etc. For the dataset Amazon-Book-B, Amazon-Book-C, Amazon-Book-E, Amazon-Book-F and Taobao-B, the items are mapped into knowledge base Freebase using the linkage method as described in (Zhao et al., 2019). For the dataset Yelp and E-commerce, KG construction follows the procedure described in (Luo et al., 2020). For the dataset Food.com and MyFitnessPal, the items are mapped into the KG FoodKG. The corresponding KG of the Weibo dataset is constructed based on Wikidata. For the dataset Dianping-Food, the KG is constructed by introducing Meituan Brain of Meituan-Dianping Group. For the dataset Alipay, relevant entities and relations are selected from the KG Antfin Digital Local Service KG. For the dataset StackOverflow, the KG is constructed by means of SWKG.

## 5.2 Evaluation Metrics

Evaluation metrics to assess the performance of CTR prediction models are presented in Table 3. These metrics include area under the ROC curve (AUC), Accuracy, F1-score (F1), Recall, Precision, cross-entropy loss function (Logloss), mean absolute error (MAE), mean squared error (MSE), root mean squared error (RMSE), mean average precision (mAP), Hit@K, normalized discounted cumulative gain at K (NDCG@K) and relative improvement (RI). In particular, AUC, Accuracy and F1 are commonly used evaluation metrics.

Table 3. Summary of evaluation metrics.

| Metrics | Definition | References |
|---|---|---|
| AUC | AUC calculates the area beneath the receiver operating characteristic (ROC) curve, reflecting the balance achieved between the true positive rate (TPR) and the false positive rate (FPR). $TPR = \frac{TP}{TP + FN}$ $FPR = \frac{FP}{FP + TN}$ | Tang et al. (2019); Feng et al. (2020a); Lyu et al. (2022); Huang et al. (2022); Tangruamsub et al. (2022); Chen et al. (2022b); Wang et al. (2020c); Hou et al. (2021); Chen et al. (2022a); Qu et al. (2019); Tu et al. (2021); Cao et al. (2021); Xiao et al. (2022); Guo et al. (2020); Wang et al. (2018b); Wang et al. (2019d); He et al. (2023); Yu et al. (2020); Elahi and Halim (2022); Wang et al. (2022); Khan et al. (2023); Dong et al. (2022); Wong et al. (2021); Zou et al. (2022b); Zhang et al. (2021); Wang et al. (2019c); Ma et al. (2022b); Ma et al. (2022a); Liu and Miyazaki (2022); Wang et al. (2019a); Tang et al. (2022); Tang et al. (2022); Wang et al. (2020a); Ma et al. (2021); Wang et al. (2020b); Feng et al. (2020b); Sun and Shagar (2020); Zou et al. (2022a); Wang et al. (2021); Wang et al. (2019b); Yang et al. (2021); Tai et al. (2020); Dai et al. (2022); Sun and Li (2022); Hui et al. (2022); Li et al. (2019); Xie et al. (2022); Qian et al. (2022); Duan et al. (2023b); Wang et al. (2018a); Khan et al. (2022); Yang et al. (2020); Wu et al. (2022); Zhang and Yang (2020); Shu and Huang (2021); Song et al. (2023); Hu et al. (2023); Tu et al. (2023); Ma et al. (2023); Wang et al. (2023b); Yang et al. (2023b); Peng et al. (2023); Yang et al. (2023a); Li et al. (2023a); Zhang et al. (2023); Wang et al. (2023a); Shu and Huang (2023); Bai et al. (2023); Ong et al. (2023); Duan et al. (2023a); Hai and Hongyan (2023); Li et al. (2023b); Gao et al. (2023); Yao et al. (2023); Elahi et al. (2024); Chen et al. (2024); Wang et al. (2024b); Cui et al. (2024); Tang et al. (2024); Liu et al. (2023); Wang et al. (2023c); Li et al. (2024) |
| Accuracy | $Accuracy = \frac{TP + TN}{N}$ | Tang et al. (2019); Huang et al. (2022); Qu et al. (2019); Cao et al. (2021); Guo et al. (2020); Yu et al. (2020); Wang et al. (2022); Khan et al. (2023); Dong et al. (2022); Fan et al. (2022); Zhang et al. (2021); Liu and Miyazaki (2022); Tang et al. (2022); Wang et al. (2020a); Ma et al. (2021); Wang et al. (2020b); Sun and Shagar (2020); Wang et al. (2021); Wang et al. (2019b); Yang et al. (2021); Tai et al. (2020); Lee et al. (2020); Sun and Li (2022); Hui et al. (2022); Xie et al. (2022); Duan et al. (2023b); Wang et al. (2018a); Khan et al. (2022); Zhang and Yang (2020); Song et al. (2023); Ma et al. (2023); Wang et al. (2023b); Wang et al. (2023a); Hai and Hongyan (2023); Gao et al. (2023); Wang et al. (2024a); Tang et al. (2024); Wang et al. (2023c); Li et al. (2024) |
| Recall | $Recall = \frac{TP}{TP + FN}$ | Chen et al. (2022a); Wang et al. (2019d); Yang et al. (2020); Yang et al. (2023b) |
| Precision | $Precision = \frac{TP}{TP + FP}$ | Chen et al. (2022a); Yang et al. (2020) |
| mAP | mAP is the mean of the AP scores, which are the average of the precision scores at different recall levels. | Chen et al. (2024) |
| F1 | $F1 = \frac{2 * Precision * Recall}{Precision + Recall}$ | Lyu et al. (2022); Chen et al. (2022b); Wang et al. (2020c); Hou et al. (2021); Chen et al. (2022a); Cao et al. (2021); Xiao et al. (2022); Wang et al. (2018b); He et al. (2023); Elahi and Halim (2022); Wang et al. (2022); Zou et al. (2022b); Fan et al. (2022); Zhang et al. (2021); Wang et al. (2019c); Ma et al. (2022b); Ma et |

| | | |
|---|---|---|
| | | al. (2022a); Tang et al. (2022); Zou et al. (2022a); Lee et al. (2020); Dai et al. (2022); Sun and Li (2022); Li et al. (2019); Qian et al. (2022); Yang et al. (2020); Wu et al. (2022); Shu and Huang (2021); Yang et al. (2023b); Wang et al. (2023b); Yang et al. (2023a); Wang et al. (2023a); Shu and Huang (2023); Bai et al. (2023); Ong et al. (2023); Duan et al. (2023a); Li et al. (2023b); Elahi et al. (2024); Chen et al. (2024); Wang et al. (2024b); Cui et al. (2024); Tang et al. (2024); Liu et al. (2023) |
| Logloss | $logloss = -\frac{1}{N}\sum_{i=1}^{N}(y_i log\hat{y}_i + (1 - y_i) log\,(1-\hat{y}_i))$ | Feng et al. (2020b); Peng et al. (2023); Yao et al. (2023) |
| MAE | $\frac{1}{N}\sum_{i=1}^{N}\lvert\hat{y}_i - y_i\rvert$ | Wang et al. (2024a) |
| MSE | $\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2$ | Chen et al. (2024) |
| RMSE | $RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}$ | Liu and Miyazaki (2022); Wang et al. (2024a) |
| Hit@K | Hit@K calculates the proportion of instances where at least one relevant click appears in the top-K positions. | Yang et al. (2023b; Zhu et al. (2020) |
| NDCG@K | NDCG@K measures the quality of the predicted ranking by considering the relevance of clicks up to position K. | Zhu et al. (2020); Hu et al. (2023); Yang et al. (2023b) |
| RI | $RI = \frac{\lvert AUC(model) - AUC(base)\rvert}{AUC(base)} * 100\%$ | Feng et al. (2020a); Peng et al. (2023) |

## 5.3 Performance Discussion

In this section, we discuss and summarize model performance on each dataset reported by each literature, which provides meaningful insights and serves several purposes. First, examining how well each model performs across multiple datasets allows for a holistic evaluation of their effectiveness and generalizability. Models that consistently perform well indicate their potential for wider application and adoption. Second, identifying their strengths and weaknesses can help choose the most appropriate model for the specific scenario. Furthermore, model evaluation provides valuable guidance for parameter tuning. Each model may have several hyperparameters that need to be optimized, such as learning rate, dropout rate, embedding dimension, etc. Researchers can identify the factors that influence the model performance to fine-tune the parameters. Comparing model performance in Table 4, we have the following observations.

First, advanced KG-based CTR prediction models demonstrate superior performance compared with matrix factorization models (e.g., BPRMF and LibFM). Specifically, among 42 studies conducted on 17 datasets, 41 studies indicate that KG-based CTR prediction models outperform matrix factorization models in terms of AUC except (Dai et al., 2022); among 21 studies conducted on 11 datasets, 20 studies indicate that KG-based CTR prediction models perform better than matrix factorization models in terms of Accuracy except (Wang et al., 2020a); among 16 studies conducted on 7 datasets, 15 studies report that KG-based CTR prediction models showcase competitive performance than matrix factorization models in F1 except (Dai et al., 2022).

Second, we analyze the performance of embedding-based models and that of path-based models. To be specific, among 6 studies conducted on 7 datasets, all the studies show that path-based models yield better results in AUC; among 4 studies conducted on 6 datasets, 3 studies report that path-based models gain better performance in Accuracy except (Li et al., 2023c) in which the embedding-based method (Wang et al., 2019b) performs better. It is worth noting that embedding-based models are early research and show modest competitiveness.

Third, we explore the performance between path-based models and propagation-based models. Concretely, among 4 studies conducted on 6 datasets, 3 studies show that path-based models yield better results in AUC apart from (Li et al., 2023c) in which the propagation-based method (Wang et al., 2019c) performs better; among 4 studies conducted on 6 datasets, all the studies report that path-based models reveal superior performance in Accuracy. Although it seems that path-based models are advantageous, that is only when compared to the earliest propagation model (Wang et al., 2018a). In fact, path-based models face the challenge of explicitly encoding path information, which may hinder their ability to excel over other models.

Fourth, we discuss the performance of embedding-based models and propagation-based models. Concretely, among 32 studies conducted on 13 datasets, 27 studies show that propagation-based models are superior to embedding-based models in AUC while 5 studies (Huang et al., 2022; Shu and Huang, 2021; Sun and Shagar, 2020; Wang et al., 2019b; Wang et al., 2021) hold opposing results; among 21 studies conducted on 9 datasets, 16 studies report that propagation-based models gain better performance in Accuracy, with the exception of 5 studies (Huang et al., 2022; Sun and Shagar, 2020; Wang et al., 2019b; Wang et al., 2021; Wang et al., 2022) showing contrasting findings; among 29 studies conducted on 11 datasets, 27 studies demonstrate the superiority of propagation-based models over embedding-based models in F1 except (Shu and Huang, 2021; Wang et al., 2022). Our analysis reveals that propagation-based models generally exhibit superior performance compared to embedding-based models.

Moreover, we find that in the embedding-based models, all 7 studies conducted on 4 datasets consistently exhibit the inferiority of the alternate learning model (i.e., Wang et al., 2019b) compared to the one-by-one learning model (i.e., Wang et al., 2018b). The model in (Wang et al., 2018b) is more adept at the news domain, which may be due to the fact that other domains (e.g., movie, book), have too little text data to provide sufficient information. In the propagation-based models, among 34 studies conducted on 15 datasets, a majority of 26 studies present that refining user representations (Wang et al., 2018a) outperforms refining item representations (Wang et al., 2019c) in AUC, Accuracy, F1, MSE and mAP. In addition, comparing the models with refinement of user or item representations (i.e., Wang et al., 2018b; Wang et al., 2019a; Wang et al., 2019c) with the model with both refinements (Wang et al., 2020c), we observe that among 15 studies conducted on 6 datasets, 9 studies show that (Wang et al., 2020c) does not contribute to a further improvement in AUC; among 11 studies conducted on 5 datasets, 6 studies indicate that (Wang et al., 2020c) fails to enhance the results in F1. We also observe that the relative performance of models remains generally consistent within individual studies. Therefore, we speculate that discrepancies in results across different studies may be influenced by factors such as dataset preprocessing and parameter settings.

In summary, the evaluation results across multiple studies and datasets consistently highlight the advantages of propagation-based models, which undergo significant improvements and show superior potential in CTR prediction. Embedding-based models lag behind the other two models in capturing high-order relationships. Path-based models appear to have a relative disadvantage in large-scale and sparse datasets. It is notable that the performance comparison is not unanimous. This is evident in the observation that a model exhibiting superior performance in terms of AUC does not necessarily guarantee better performance in F1. Therefore, when choosing a model for CTR prediction, it is crucial to consider the specific dataset, task requirements, and evaluation metrics.

Table 4. Comparison for model performance on different datasets.

| Dataset | Reference | Model Performance |
|---|---|---|
| MovieLens-1M | Tang et al. (2019) | AUC: DKN<CKE<LibFM<RippleNet<Wide&Deep<AKUPM<br>Accuracy: DKN<CKE<LibFM<Wide&Deep<RippleNet<AKUPM |
| | Li et al. (2023c) | AUC: DKN<PER<CKE<CAFSE<LibFM<Wide&Deep<RippleNet<MKR<AKUPM<KGCN<DISL<br>Accuracy: DKN<PER<CKE<LibFM<Wide&Deep<RippleNet<MKR<AKUPM<KGCN<DISL |
| | Wang et al. (2019b) | AUC: DKN<PER<CKE<LibFM<Wide&Deep<MKR<RippleNet<br>Accuracy: DKN<PER<CKE<LibFM<Wide&Deep<RippleNet<MKR |
| | Ong et al. (2023) | AUC: PER<BPRMF<CKE<KGNN-LS<KGCN<RippleNet<KGIN<KGAT<MCCLK<KGIC<QTEF-CRL<br>F1: PER<BPRMF<CKE<KGCN<KGNN-LS<RippleNet<KGAT<KGIN<MCCLK<KGIC<QTEF-CRL |
| | Wang et al. (2021) | AUC: DKN<PER<Wide&Deep<MKR<RippleNet<Ripp-MKR<br>Accuracy: DKN<PER<Wide&Deep<RippleNet<MKR<Ripp-MKR |
| | Zou et al. (2022a) | AUC: PER<BPRMF<CKE<KGCN<KGNN-LS<KGAT<RippleNet<KGIN<MCCLK<br>F1: PER<BPRMF<CKE<KGCN<KGNN-LS<RippleNet<KGAT<KGIN<MCCLK |
| | Xiao et al. (2022) | AUC: MKR<FM_MKR<DFM-GCN<br>F1: MKR<FM_MKR<DFM-GCN |
| | Wang et al. (2018a) | AUC: DKN<PER<SHINE<CKE<LibFM<Wide&Deep<RippleNet<br>Accuracy: DKN<PER<SHINE<CKE<LibFM<Wide&Deep<RippleNet |
| | Li et al. (2024) | AUC: DKN<CKE<KANR-A<MCRec<KANR-K<KGCN<GMCF<RippleNet<Cn-RippleNet<RKGCN<br>Accuracy: DKN<CKE<MCRec<KANR-K<KANR-A<KGCN<RippleNet<GMCF<Cn-RippleNet<RKGCN |
| | Hui et al. (2022) | AUC: DKN<PER<SHINE<CKE<LibFM<Wide&Deep<PHGR<ReBKC<br>Accuracy: DKN<PER<CKE<SHINE<LibFM<Wide&Deep<ReBKC |
| | Dong et al. (2022) | AUC: DKN<PER<CKE<LibFM<Wide&Deep<KGCN<MKR<RippleNet<HRS<br>Accuracy: DKN<PER<CKE<LibFM<Wide&Deep<KGCN<RippleNet<MKR<HRS |
| | Guo et al. (2020) | AUC: DKN<PER<LibFM<Wide&Deep<RippleNet<MKR<KGCN<KGE-DNN<DKEN<br>Accuracy: DKN<PER<LibFM<Wide&Deep<RippleNet<MKR<KGCN<KGE-DNN<DKEN |
| | Wang et al. (2023a) | AUC: CKE<KGAT<KGCN<HGCF<LGCF<MVIN<GDCF<KGCL<LKGR<CurvRec<br>Accuracy: CKE<KGAT<KGCN<HGCF<LGCF<GDCF<MVIN<KGCL<LKGR<CurvRec<br>F1: CKE<KGCN<KGAT<HGCF<LGCF<GDCF<MVIN<KGCL<LKGR<CurvRec |

| | | |
|---|---|---|
| | Wang et al. (2022) | AUC: KGCN<MKR<RippleNet<GRE<br>Accuracy: KGCN<MKR<RippleNet<GRE<br>F1: KGCN<MKR<RippleNet<GRE |
| | Zou et al. (2022b) | AUC: PER<BPRMF<CKE<CKAN<KGCN<CG-KGR<KGNN-LS<KGAT<RippleNet<KGIN<KGIC<br>F1: PER<BPRMF<CKE<CG-KGR<KGCN<CKAN<KGNN-LS<RippleNet<KGAT<KGIN<KGIC |
| | Wang et al. (2020a) | AUC: PER<CKE<LibFM<KGCN<RippleNet<KCER<br>Accuracy: PER<CKE<LibFM<KGCN<RippleNet<KCER |
| | Yu et al. (2020) | AUC: DKN<PER<CKE<LibFM<Wide&Deep<RippleNet<GFEN<br>Accuracy: DKN<PER<CKE<LibFM<Wide&Deep<RippleNet<GFEN |
| | Sun and Shagar (2020) | AUC: DKN<PER<CKE<LibFM<Wide&Deep<MUKG<RippleNet<br>Accuracy: DKN<PER<CKE<LibFM<Wide&Deep<RippleNet<MUKG |
| | Wang et al. (2019d) | AUC: DKN<PER<SHINE<CKE<BPRMF<DMCM |
| | Huang et al. (2022) | AUC: PER<CKE<MKR<RippleNet<Ripp-MKR<CAKR<br>Accuracy: PER<CKE<RippleNet<MKR<Ripp-MKR<CAKR |
| | Wang et al. (2024b) | AUC: PER<BPRMF<CKE<KGCN<CKAN<KGAT<RippleNet<KFGAN<br>F1: PER<BPRMF<CKE<KGCN<RippleNet<CKAN<KGAT<KFGAN |
| | Tu et al. (2021) | AUC: CKE<NFM<KGAT<RippleNet<KCAN |
| | Qu et al. (2019) | AUC: PER<DKN<LibFM<CKE<Wide&Deep<MCRec<RippleNet<PinSage<KNI<br>Accuracy: PER<DKN<LibFM<CKE<Wide&Deep<MCRec<RippleNet<PinSage<KNI |
| | Ong et al. (2023) | AUC: PER<BPRMF<CKE<KGCN<KGNN-LS<KGAT<RippleNet<KGIN<KGIC<MCCLK<QTEF-CRL<br>F1: PER<BPRMF<CKE<KGCN<KGNN-LS<RippleNet<KGAT<KGIN<KGIC<MCCLK<QTEF-CRL |
| | Tai et al. (2020) | AUC: MCRec<GC-MC<KGNN<CKE<FM<NFM<RippleNet<KGAT<MVIN<br>Accuracy: GC-MC<MCRec<FM<KGNN<CKE<NFM<RippleNet<KGAT<MVIN |
| | Wang et al. (2020b) | AUC: DKN<PER<CKE<CoFM<LibFM<Wide&Deep<RippleNet<MRP2Rec<br>Accuracy: DKN<PER<CoFM<CKE<LibFM<Wide&Deep<RippleNet<MRP2Rec |
| | Ma et al. (2021) | AUC: DKN<PER<CKE<LibFM<Wide&Deep<MKR<KNI<MNI<br>Accuracy: DKN<PER<CKE<LibFM<Wide&Deep<MKR<KNI<MNI |
| | Hu et al. (2023) | AUC: KGAT<LightGCN<MMGCN<KGCN<CKE<CKAN<MGAT<CRMMAN<br>NDCG@5: CKE<KGCN<CKAN<KGAT<LightGCN<MMGCN<MGAT<CRMMAN<br>NDCG@10: CKE<KGCN<CKAN<KGAT<LightGCN<MMGCN<MGAT<CRMMAN |
| | Tu et al. (2023) | AUC: CKE<NFM<KGAT<GAT<DisenGCN<DisenKGAT<KGIN<KCAN<DIKGNN |
| | Wang et al. (2023b) | AUC: SVD++<MCRec<CKE<KGAT<KGCN<MKR<RippleNet<MVIN<KGCL<LKGR<UPIACM<br>Accuracy: SVD++<CKE<MCRec<KGAT<KGCN<MKR<RippleNet<MVIN<KGCL<LKGR<UPIACM<br>F1: SVD++<CKE<KGCN<KGAT<MKR<RippleNet<MVIN<KGCL<LKGR<UPIACM |
| | Peng et al. (2023) | AUC: YoutubeNet<Wide&Deep<DIN<DIEN<DS-KGAT<br>Logloss: DS-KGAT<DIEN<DIN<Wide&Deep<YoutubeNet<br>RI: YoutubeNet<Wide&Deep<DIN<DIEN<DS-KGAT |
| | Hai and Hongyan (2023) | AUC: SHINE<CKE<LibFM<Wide&Deep<Ripple-DF<br>Accuracy: SHINE<CKE<LibFM<Wide&Deep<Ripple-DF |

| | | |
|---|---|---|
| | Wang et al. (2024a) | Accuracy: VRKG4Rec<BroadCF<KGNN-LS<KGIN<KGCN<CKAN<MKNBL<br>MAE: MKNBL<CKAN<KGIN<KGCN<KGNN-LS<BroadCF<VRKG4Rec<br>RMSE: MKNBL<CKAN<KGCN<KGNN-LS<KGIN<BroadCF<VRKG4Rec |
| | Wang et al. (2023c) | AUC: DIN<DIEN<BST<LightSANs<KGDIE<br>Accuracy: DIN<DIEN<BST<LightSANs<KGDIE |
| | Elahi et al. (2024) | AUC: NCF<RippleNet<KGNN-LS<KGAT<KGCN<CKAN<KGCAN<br>F1: NCF<RippleNet<KGCAN<KGCN<KGAT<KGNN-LS<CKAN |
| | Ma et al. (2023) | AUC: KGCN<LibFM<CKAN<KGBPR<br>Accuracy: KGCN<LibFM<CKAN<KGBPR |
| MovieLens-20M | Qu et al. (2019) | AUC: PER<DKN<LibFM<Wide&Deep<MCRec<CKE<RippleNet<PinSage<KNI<br>Accuracy: PER<DKN<LibFM<Wide&Deep<MCRec<CKE<RippleNet<PinSage<KNI |
| | Duan et al. (2023a) | AUC: DKN<CKE<BPRMF<LibFM<RippleNet<CKAN<KGCN<CG-KGR<DCRAN<br>F1: DKN<CKE<LibFM<BPRMF<CKAN<RippleNet<KGCN<DCRAN<CG-KGR |
| | Yang et al. (2023a) | AUC: MKR<LibFM<CFKG<KGCN<FIRE<RippleNet<HKJPN<KGAT<CG-KGR<KEMIM<br>F1: CFKG<MKR<LibFM<FIRE<KGCN<RippleNet<KGAT<HKJPN<CG-KGR<KEMIM |
| | Li et al. (2023b) | AUC: CKE<RippleNet<KGAT<KGIN<KARN<AGRE<CG-KGR<SEKGAT<br>F1: CKE<KGAT<RippleNet<KARN<AGRE<KGIN<CG-KGR<SEKGAT |
| | Liu et al. (2023) | AUC: PER<CKE<RippleNet<KGCN<KRAN<SKGCR<K-NCR<br>F1: PER<CKE<RippleNet<KGCN<SKGCR<KRAN<K-NCR |
| | Shu and Huang (2023) | AUC: PER<CKE<LibFM<SVD<KGCN<KGNN-LS<LightGCN<Multi-Rec<br>F1: PER<CKE<LibFM<SVD<KGCN<KGNN-LS<LightGCN<Multi-Rec |
| | Shu and Huang (2021) | AUC: PER<CKE<LibFM<SVD<LibFM+TransE<RippleNet<KGNN-LS<RKG<br>F1: PER<CKE<LibFM<RippleNet<LibFM+TransE<SVD<KGNN-LS<RKG |
| | Li et al. (2019) | AUC: PER<CKE<LibFM<SVD<RippleNet<KGCN<KGDAM<br>F1: PER<CKE<LibFM<RippleNet<SVD<KGCN<KGDAM |
| | Cui et al. (2024) | AUC: BPRMF<CFKG<RippleNet<KGCN<CKE<KGAT<KGCL<LightGCN<RAKCR<br>F1: BPRMF<CFKG<CKE<RippleNet<KGCN<KGAT<KGCL<LightGCN<RAKCR |
| | Duan et al. (2023b) | AUC: DKN<CKE<LibFM<RippleNet<KGCN<RFAN<br>Accuracy: DKN<CKE<LibFM<RippleNet<RFAN<KGCN |
| | Tang et al. (2022) | AUC: PER<CKE<LibFM<SVD<LibFM+TransE<RippleNet<KGCN<KGNN-LS<KE-GCN<br>F1: PER<CKE<LibFM<RippleNet<LibFM+TransE<SVD<KGCN<KGNN-LS<KE-GCN |
| | Xiao et al. (2022) | AUC: MKR<FM_MKR<DFM-GCN<br>F1: MKR<FM_MKR<DFM-GCN |
| | Qian et al. (2022) | AUC: PER<CKE<KGNN-LS<RippleNet<KGAT<CKAN<KGCN<RKAC<br>F1: PER<CKE<RippleNet<KGAT<KGNN-LS<CKAN<KGCN<RKAC |
| | Dai et al. (2022) | AUC: PER<NGCF<RippleNet<KGCN<NFM<KGNN-LS<KGAT<FM<CKE<COAT<GCMC<br>F1: PER<NGCF<RippleNet<KGCN<NFM<KGNN-LS<KGAT<FM<CKE<COAT<GCMC |
| | Wang et al. (2019a) | AUC: PER<CKE<LibFM<RippleNet<SVD<LibFM+TransE<KGNN-LS |
| | Ma et al. (2022a) | AUC: PER<LibFM<CKE<LibFM+TransE<RippleNet<KGCN<KNCR<KRNN<br>F1: PER<CKE<LibFM<LibFM+TransE<RippleNet<KGCN<KNCR<KRNN |
| | Wang et al. (2019c) | AUC: PER<CKE<LibFM<SVD<LibFM+TransE<RippleNet<KGCN<br>F1: PER<CKE<LibFM<RippleNet<LibFM+TransE<SVD<KGCN |
| | Fan et al. (2022) | AUC: AKGE<KGAT<KNCR<PRSKG<CGAT<KGFER<br>F1: AKGE<KGAT<KNCR<PRSKG<CGAT<KGFER |
| | He et al. (2023) | AUC: GC-MC<NGCF<RippleNet<KGAT<CKAN<CIEPA<KGNN-LS<KGIN<ERSIF-KR<br>F1: GC-MC<NGCF<RippleNet<KGAT<CKAN<CIEPA<KGNN-LS<KGIN<ERSIF-KR |
| | Elahi and Halim (2022) | AUC: KGCN<RippleNet<K-NCR<KGNN-LS<KGAT<CKAN<GACF<br>F1: RippleNet<KGCN<KGAT<CKAN<KGNN-LS<GACF<K-NCR |
| | Wang et al. (2020c) | AUC: PER<CKE<BPRMF<KGNN-LS<RippleNet<KGAT<CKAN<KGCN<br>F1: PER<CKE<BPRMF<RippleNet<KGAT<KGNN-LS<CKAN<KGCN |

| | | |
|---|---|---|
| | Lyu et al. (2022) | AUC: KGAT<BKANE<RippleNet<CKAN<KGCN<br>F1: KGAT<BKANE<RippleNet<CKAN<KGCN |
| | Chen et al. (2022b) | AUC: NFM<CKAN<BPRMF<RippleNet<CKE<KGCN<KGNN-LS<KGAT<CG-KGR<br>F1: NFM<BPRMF<CKE<RippleNet<KGCN<KGAT<CKAN< KGNN-LS<CG-KGR |
| MovieLens-Latest | Wu et al. (2022) | AUC: KGNN-LS<RippleNet<SASRec<LSAN<KGCN<Caser<GRU4Rec<CUIKG<LightGCN<GCM<HAGERec<UBAR<br>F1: KGNN-LS<SASRec<KGCN<LSAN<GCM<RippleNet<GRU4Rec<Caser<CUIKG<LightGCN<HAGERec<UBAR |
| IMDb | Liu and Miyazaki (2022) | AUC (2-core[1]): MKR<GMF<PMF<MPCN<DeepCoNN<KANN<TransNets<br>Accuracy (2-core): MKR<GMF<PMF<TransNets<MPCN<DeepCoNN<KANN<br>RMSE (2-core): KANN<TransNets<DeepCoNN<MKR<MPCN<GMF<PMF<br>AUC (10-core[2]): MKR<GMF<PMF<MPCN<DeepCoNN<TransNets<KANN<br>Accuracy (10-core): MKR<GMF<DeepCoNN<TransNets<MPCN<PMF<KANN<br>RMSE (10-core): KANN<PMF<TransNets<GMF<DeepCoNN<MKR<MPCN |
| Amazon-Movies&TV-A | Liu and Miyazaki (2022) | AUC (2-core): GMF<MKR<PMF<KANN<MPCN<DeepCoNN<TransNets<br>Accuracy (2-core): PMF<GMF<MKR<MPCN<DeepCoNN<TransNets<KANN<br>RMSE (2-core): KANN<DeepCoNN<TransNets<MPCN<MKR<GMF<PMF<br>AUC (10-core): GMF<MKR<PMF<KANN<DeepCoNN<MPCN<TransNets<br>Accuracy (10-core): PMF<MKR<MPCN<GMF<DeepCoNN<TransNets<KANN<br>RMSE (10-core): KANN<MPCN<DeepCoNN<TransNets<MKR<PMF<GMF |
| Book-Crossing | Hou et al. (2021) | AUC: PER<CKE<LibFM+TransE<RippleNet<KGCN<KGCN-CF<br>F1: PER<CKE<LibFM+TransE<RippleNet<KGCN<KGCN-CF |
| | Duan et al. (2023a) | AUC: DKN<BPRMF<CKE<KGCN<LibFM<RippleNet<CKAN<CG-KGR<DCRAN<br>F1: DKN<BPRMF<CKE<KGCN<LibFM<RippleNet<CKAN<CG-KGR<DCRAN |
| | Elahi et al. (2024) | AUC: KGNN-LS<KGCN<NCF<RippleNet<KGAT<CKAN<KGCAN<br>F1: NCF<RippleNet<KGNN-LS<KGCN<KGAT<CKAN<KGCAN |
| | Wang et al. (2024a) | Accuracy: VRKG4Rec<KGCN<KGNN-LS<CKAN<KGIN<MKNBL<br>MAE: MKNBL<KGIN<CKAN<KGNN-LS<VRKG4Rec<KGCN<br>RMSE: MKNBL<KGIN<CKAN<KGNN-LS<VRKG4Rec<KGCN |
| | Wang et al. (2024b) | AUC: PER<BPRMF<CKE<KGCN<RippleNet<KGAT<CKAN<COAT<KFGAN<br>F1: PER<BPRMF<CKE<KGCN<RippleNet<KGAT<CKAN<COAT<KFGAN |
| | Li et al. (2023b) | AUC: CKE<AGRE<RippleNet<KGAT<KARN<KGIN<CG-KGR<SEKGAT<br>F1: CKE<AGRE<RippleNet<KGAT<KGIN<CG-KGR<KARN<SEKGAT |
| | Yang et al. (2023a) | AUC: LibFM<CFKG<FIRE<MKR<KGCN<KGAT<RippleNet<CG-KGR<HKJPN<KEMIM<br>F1: CFKG<LibFM<FIRE<MKR<KGCN<RippleNet<HKJPN<KGAT<CG-KGR<KEMIM |
| | Wang et al. (2023b) | AUC: SVD++<KGAT<CKE<RippleNet<MKR<KGCN<MVIN<KGCL<LKGR<UPIACM<br>Accuracy: CKE<SVD++<KGAT<RippleNet<MVIN<KGCL<LKGR<MKR<KGCN<UPIACM<br>F1: MVIN<KGAT<LKGR<RippleNet<KGCN<MKR<CKE<SVD++<KGCL<UPIACM |
| | Hai and Hongyan (2023) | AUC: SHINE<CKE<LibFM<Wide&Deep<Ripple-DF<br>Accuracy: Wide&Deep<SHINE<CKE<LibFM<Ripple-DF |
| | Shu and Huang (2023) | AUC: PER<SVD<CKE<KGCN<LibFM<KGNN-LS<LightGCN<MKR<KCRec<Multi-Rec<br>F1: PER<CKE<LibFM<SVD<KGNN-LS<LightGCN<KGCN<KCRec<MKR<Multi-Rec |
| | Wang et al. (2023a) | AUC: KGAT <CKE<KGCN<MVIN<HGCF<LGCF<KGCL<GDCF<LKGR<CurvRec<br>Accuracy: CKE<KGAT<MVIN<HGCF<LGCF<GDCF<KGCL<LKGR<KGCN<CurvRec<br>F1: MVIN<KGAT<LKGR<KGCN<GDCF<CKE<HGCF<LGCF<KGCL<CurvRec |
| | Shu and Huang (2021) | AUC: PER<SVD<CKE<LibFM<KGNN-LS<LibFM+TransE<RippleNet<MKR<RKG<br>F1: PER<CKE<LibFM<LibFM+TransE<SVD<KGNN-LS<RippleNet<MKR<RKG |

[1] 2-core denotes each user in the dataset had at least two reviews for items.
[2] 10-core denotes each user in the dataset had at least ten reviews for items.

| | |
|---|---|
| Khan et al. (2022) | AUC: PER<CKE<KGAT<MCRec<AKGE<RippleNet<NACF<SAGE<br>Accuracy: PER<CKE<MCRec<RippleNet<KGAT<AKGE<NACF<SAGE |
| Wang et al. (2018a) | AUC: DKN<PER<SHINE<CKE<LibFM<Wide&Deep<RippleNet<br>Accuracy: PER<DKN<Wide&Deep<SHINE<CKE<LibFM<RippleNet |
| Liu et al. (2023) | AUC: PER<CKE<KGCN<KRAN<RippleNet<K-NCR<SKGCR<br>F1: PER<CKE<KGCN<RippleNet<KRAN<SKGCR<K-NCR |
| Qian et al. (2022) | AUC: PER<CKE<KGNN-LS<KGCN<RippleNet<KGAT<CKAN<RKAC<br>F1: PER<CKE<KGCN<KGNN-LS<RippleNet<KGAT<CKAN<RKAC |
| Zhang and Yang (2020) | AUC: DKN<PER<SHINE<CKE<LibFM<LI<RippleNet<LAGI<br>Accuracy: PER<DKN<LI<SHINE<CKE<LibFM<RippleNet<LAGI |
| Zhang et al. (2023) | AUC: DKN<BPR<FM<RippleNet<KGNN-LS<KGAN<CKAN |
| Duan et al. (2023b) | AUC: DKN<CKE<KGCN<LibFM<RippleNet<RFAN<br>Accuracy: DKN<CKE<LibFM<KGCN<RippleNet<RFAN |
| Xie et al. (2022) | AUC: CKE<LibFM<AFM<RippleNet<KGCN<KGAFM<br>Accuracy: CKE<LibFM<RippleNet<AFM<KGCN<KGAFM |
| Li et al. (2024) | AUC: DKN<MCRec<KGCN<RippleNet<RKGCN<CKE<KANR-K<KANR-A<Cn-RippleNet<CKAN<ATKGRM<GMCF<br>Accuracy: DKN<MCRec<KGCN<RippleNet<CKE<RKGCN<KANR-K<CKAN<KANR-A<Cn-RippleNet<GMCF<ATKGRM |
| Dai et al. (2022) | AUC: PER<NGCF<CKE<GCMC<KGNN-LS<KGCN<KGAT<RippleNet<FM<COAT<NFM<br>F1: PER<NGCF<CKE<KGNN-LS<KGCN<GCMC<RippleNet<KGAT<COAT<FM<NFM |
| Li et al. (2019) | AUC: PER<SVD<CKE<LibFM<RippleNet<KGCN<KGDAM<br>F1: PER<CKE<LibFM<SVD<RippleNet<KGDAM<KGCN |
| Hui et al. (2022) | AUC: DKN<PER<SHINE<CKE<LibFM<Wide&Deep<PHGR<ReBKC<br>Accuracy: PER<DKN<Wide&Deep<SHINE<CKE<LibFM<ReBKC |
| Wang et al. (2021) | AUC: DKN<PER<Wide&Deep<RippleNet<MKR<Ripp-MKR<br>Accuracy: RippleNet<PER<DKN<Wide&Deep<MKR<Ripp-MKR |
| Zou et al. (2022a) | AUC: PER<BPRMF<CKE<KGNN-LS<KGCN<RippleNet<KGIN<KGAT<MCCLK<br>F1: PER<BPRMF<CKE<KGCN<KGNN-LS<RippleNet<KGAT<KGIN<MCCLK |
| Ma et al. (2021) | AUC: DKN<PER<CKE<LibFM<Wide&Deep<MKR<KNI<MNI<br>Accuracy: PER<DKN<Wide&Deep<CKE<LibFM<MKR<KNI<MNI |
| Wang et al. (2019b) | AUC: DKN<PER<CKE<LibFM<Wide&Deep<RippleNet<MKR<br>Accuracy: PER<DKN<Wide&Deep<CKE<LibFM<RippleNet<MKR |
| Tang et al. (2022) | AUC: RippleNet<KGCN=KGNN-LS<KE-GCN<br>F1: RippleNet< KGNN-LS<KGCN<KE-GCN |
| Wang et al. (2020a) | AUC: PER<CKE<LibFM<KGCN<RippleNet<KCER<br>Accuracy: PER<CKE<KGCN<RippleNet<KCER<LibFM |
| Wang et al. (2019a) | AUC: PER<SVD<CKE<LibFM<LibFM+TransE<RippleNet<KGNN-LS |
| Sun and Shagar (2020) | AUC: DKN<PER<CKE<LibFM<Wide&Deep<RippleNet<MUKG<br>Accuracy: PER<DKN<Wide&Deep<CKE<LibFM<RippleNet<MUKG |
| Ma et al. (2022b) | AUC: KGCN<KGNN-LS<RippleNet<BPRMF<CKE<CKAN<KGAT<KGET<br>F1: CKE<BPRMF<RippleNet<KGAT<KGCN<KGNN-LS<CKAN<KGET |
| Wang et al. (2019c) | AUC: PER<SVD<CKE<LibFM<LibFM+TransE<RippleNet<KGCN<br>F1: PER<CKE<LibFM<LibFM+TransE<SVD<RippleNet<KGCN |
| Wang et al. (2020b) | AUC: DKN<PER<CKE<LibFM<CoFM<Wide&Deep<RippleNet<MRP2Rec<br>Accuracy: PER<DKN<LibFM<CKE<CoFM<Wide&Deep<RippleNet<MRP2Rec |
| Zhang et al. (2021) | AUC: PER<DKN<CKE<KGCN<RippleNet<KCRec<br>Accuracy: PER<DKN<CKE<KGCN<RippleNet<KCRec |

| | | |
|---|---|---|
| | | F1: PER<DKN<CKE<KGCN<RippleNet<KCRec |
| | Zou et al. (2022b) | AUC: PER<BPRMF<CKE<KGNN-LS<KGCN<RippleNet<KGIN<KGAT<CKAN<CG-KGR<KGIC<br>F1: PER<BPRMF<CKE<KGCN<KGNN-LS<RippleNet<KGAT<KGIN<CKAN<CG-KGR<KGIC |
| | Elahi and Halim (2022) | AUC: K-NCR<RippleNet<KGCN<KGNN-LS<KGAT<CKAN<GACF<br>F1: K-NCR<RippleNet<KGCN<KGNN-LS<KGAT<CKAN<GACF |
| | Dong et al. (2022) | AUC: DKN<PER<CKE<LibFM<Wide&Deep<RippleNet<MKR<KGCN<HRS<br>Accuracy: PER<DKN<Wide&Deep<CKE<LibFM<RippleNet<KGCN<HRS<MKR |
| | Yu et al. (2020) | AUC: DKN<PER<CKE<LibFM<Wide&Deep<RippleNet<GFEN<br>Accuracy: PER<DKN<Wide&Deep<CKE<LibFM<RippleNet<GFEN |
| | Guo et al. (2020) | AUC: DKN<PER<KGCN<LibFM<Wide&Deep<RippleNet<MKR<KGE-DNN<DKEN<br>Accuracy: PER<DKN<KGCN<Wide&Deep<LibFM<RippleNet<KGE-DNN<DKEN<MKR |
| | Huang et al. (2022) | AUC: PER<CKE<RippleNet<MKR<Ripp-MKR<CAKR<br>Accuracy: PER<CKE<RippleNet<MKR<CAKR<Ripp-MKR |
| | Wang et al. (2020c) | AUC: PER<BPRMF<CKE<KGNN-LS<KGCN<RippleNet<KGAT<CKAN<br>F1: PER<BPRMF<CKE<KGNN-LS<KGCN<RippleNet<KGAT<CKAN |
| | Lyu et al. (2022) | AUC: KGCN<RippleNet<KGAT<CKAN<BKANE<br>F1: KGCN<RippleNet<KGAT<CKAN<BKANE |
| | Qu et al. (2019) | AUC: PER<DKN<CKE<LibFM<PinSage<Wide&Deep<MCRec<RippleNet<KNI<br>Accuracy: PER<Wide&Deep<DKN<LibFM<CKE<PinSage<RippleNet<MCRec<KNI |
| | Tang et al. (2019) | AUC: DKN<CKE<RippleNet<LibFM<Wide&Deep<AKUPM<br>Accuracy: DKN<CKE<Wide&Deep<LibFM<RippleNet<AKUPM |
| | Hu et al. (2023) | AUC: CKE<LightGCN<MMGCN<KGAT<MGAT<CKAN<KGCN<CRMMAN<br>NDCG@5: MMGCN<MGAT<CKE<KGCN<LightGCN<KGAT<CKAN<CRMMAN<br>NDCG@10: MMGCN<MGAT<CKE<KGCN<LightGCN<KGAT<CKAN<CRMMAN |
| | Chen et al. (2022b) | AUC: BPRMF<CKE<KGCN<KGAT<KGNN-LS<RippleNet<NFM<CKAN<CG-KGR<br>F1: CKE<BPRMF<KGCN<NFM<KGNN-LS<RippleNet<KGAT<CKAN<CG-KGR |
| Amazon-Book-A | Khan et al. (2023) | AUC: PER<CKE<MCRec<AKGE<RippleNet<KGAT<NACF<DKEN<H-SAGE<br>Accuracy: PER<CKE<MCRec<AKGE<RippleNet<KGAT<NACF<DKEN<H-SAGE |
| Amazon-Book-B | Fan et al. (2022) | AUC: AKGE<KGAT<KNCR<PRSKG<CGAT<KGFER<br>F1: AKGE<KGAT<KNCR<PRSKG<CGAT<KGFER |
| Amazon-Book-C | Tai et al. (2020) | AUC: FM<MCRec<KGNN<CKE<GC-MC<NFM<RippleNet<KGAT<MVIN<br>Accuracy: FM<MCRec<CKE<KGNN<GC-MC<NFM<RippleNet<KGAT<MVIN |
| Amazon-Book-E | Qu et al. (2019) | AUC: PER<LibFM<Wide&Deep<DKN<CKE<PinSage<MCRec<RippleNet<KNI<br>Accuracy: PER<LibFM<DKN<Wide&Deep<PinSage<CKE<MCRec<RippleNet<KNI |
| Amazon-Book-F | Cui et al. (2024) | AUC: BPRMF<CKE<CFKG<RippleNet<KGCN<KGCL<LightGCN<KGAT<RAKCR<br>F1: BPRMF<CKE<RippleNet<CFKG<KGCN<KGCL<LightGCN<KGAT<RAKCR |
| Amazon-Book-G | Wang et al. (2023c) | AUC: DIN<DIEN<BST<LightSANs<KGDIE<br>Accuracy: DIN<DIEN<LightSANs<BST<KGDIE |
| Amazon-Electronics | Wu et al. (2022) | AUC: KGNN-LS<RippleNet<SASRec<GRU4Rec<LSAN<Caser<CUIKG<KGCN<HAGERec<LightGCN<GCM<UBAR<br>F1: KGNN-LS<RippleNet<SASRec<LSAN<KGCN<CUIKG<LightGCN<HAGERec<Caser<GRU4Rec<GCM<UBAR |
| Last.FM | Hou et al. (2021) | AUC: PER<CKE<RippleNet<LibFM+TransE<KGCN<KGCN-CF<br>F1: PER<CKE<RippleNet<LibFM+TransE<KGCN<KGCN-CF |
| | Duan et al. (2023a) | AUC: DKN<CKE<BPRMF<RippleNet<LibFM<KGCN<CG-KGR<CKAN<DCRAN<br>F1: DKN<CKE<BPRMF<LibFM<RippleNet<KGCN<CG-KGR<DCRAN<CKAN |
| | Elahi et al. (2024) | AUC: NCF<RippleNet<KGNN-LS<KGCN<KGAT<CKAN<KGCAN<br>F1: CKAN<NCF<RippleNet<KGCN<KGNN-LS<KGAT<KGCAN |

| | | |
|---|---|---|
| | Yang et al. (2023a) | AUC: LibFM<CFKG<MKR<KGCN<RippleNet<HKJPN<CG-KGR<KEMIM<KGAT<FIRE<br>F1: CFKG<LibFM<KGCN<MKR<HKJPN<CG-KGR<KEMIM<RippleNet<KGAT<FIRE |
| | Ong et al. (2023) | AUC: PER<CKE<BPRMF<RippleNet<KGCN<KGNN-LS<KGAT<KGIN<KGIC<MCCLK<QTEF-CRL<br>F1: PER<CKE<BPRMF<RippleNet<KGCN<KGNN-LS<KGAT<KGIN<KGIC<MCCLK<QTEF-CRL |
| | Wang et al. (2023a) | AUC: CKE<KGAT<KGCN<MVIN<HGCF<LGCF<KGCL<GDCF<LKGR<CurvRec<br>Accuracy: CKE<KGAT<HGCF<MVIN<KGCL<LGCF<GDCF<KGCN<LKGR<CurvRec<br>F1: CKE<KGAT<KGCN<MVIN<HGCF<KGCL<LGCF<GDCF<LKGR<CurvRec |
| | Wang et al. (2024a) | Accuracy: VRKG4Rec<KGNN-LS<KGCN<KGIN<CKAN<BroadCF<MKNBL<br>MAE: MKNBL<KGIN<BroadCF<CKAN<KGNN-LS<KGCN<VRKG4Rec<br>RMSE: MKNBL<KGIN<BroadCF<CKAN<KGNN-LS<KGCN<VRKG4Rec |
| | Li et al. (2024) | AUC: KANR-K<GMCF<DKN<KGCN<KANR-A<CKAN<ATKGRM<MCRec<CKE<RippleNet<RKGCN<br>Accuracy: GMCF<KGCN<KANR-K<KANR-A<CKAN<ATKGRM<DKN<MCRec<CKE<RippleNet<RKGCN |
| | Wang et al. (2024b) | AUC: PER<CKE<BPRMF<RippleNet<KGCN<COAT<KGAT<CKAN<KFGAN<br>F1: PER<CKE<BPRMF<RippleNet<KGCN<KGAT<COAT<CKAN<KFGAN |
| | Liu et al. (2023) | AUC: PER<CKE<RippleNet<K-NCR<KGCN<KRAN<SKGCR<br>F1: PER<CKE<RippleNet<KGCN<K-NCR<KRAN<SKGCR |
| | Shu and Huang (2023) | AUC: PER<CKE<SVD<LibFM<KGCN<MKR<KGNN-LS<KCRec<LightGCN<Multi-Rec<br>F1: PER<CKE<SVD<LibFM<KCRec<KGCN<MKR<KGNN-LS<LightGCN<Multi-Rec |
| | Li et al. (2023b) | AUC: KARN<CKE<AGRE<RippleNet<KGAT<CG-KGR<KGIN<SEKGAT<br>F1: CKE<RippleNet<KGAT<KARN<CG-KGR<AGRE<KGIN<SEKGAT |
| | Shu and Huang (2021) | AUC: PER<CKE<SVD<LibFM+TransE<LibFM<RippleNet<MKR<KGNN-LS<RKG<br>F1: PER<CKE<SVD<RippleNet<LibFM+TransE<LibFM<KGNN-LS<MKR<RKG |
| | Dai et al. (2022) | AUC: PER<KGCN<KGNN-LS<CKE<FM<NFM<RippleNet<GCMC<NGCF<KGAT<COAT<br>F1: PER<KGCN<KGNN-LS<CKE<NGCF<RippleNet<GCMC<NFM<FM<KGAT<COAT |
| | Zhang and Yang (2020) | AUC: SHINE<CKE<PER<LibFM<DKN<LI<RippleNet<LAGI<br>Accuracy: CKE<SHINE<LibFM<PER<DKN<LI<RippleNet<LAGI |
| | Khan et al. (2022) | AUC: PER<CKE<MCRec<RippleNet<AKGE<KGAT<NACF<SAGE<br>Accuracy: PER<CKE<MCRec<KGAT<RippleNet<AKGE<NACF<SAGE |
| | Li et al. (2019) | AUC: PER<CKE<SVD<LibFM<RippleNet<KGCN<KGDAM<br>F1: PER<CKE<SVD<RippleNet<LibFM<KGCN<KGDAM |
| | Sun and Li (2022) | AUC: CKE<KGNN-LS<MKR<RippleNet<KGHR<br>Accuracy: MKR<KGNN-LS<RippleNet<CKE<KGHR<br>F1: CKE<KGNN-LS<MKR<RippleNet<KGHR |
| | Duan et al. (2023b) | AUC: DKN<CKE<RippleNet<LibFM<KGCN<RFAN<br>Accuracy: DKN<CKE<RippleNet<LibFM<KGCN<RFAN |
| | Qian et al. (2022) | AUC: PER<CKE<RippleNet<KGCN<KGNN-LS<KGAT<CKAN<RKAC<br>F1: PER<CKE<RippleNet<KGCN<KGNN-LS<KGAT<CKAN<RKAC |
| | Tai et al. (2020) | AUC: MCRec<CKE<FM<KGNN<GC-MC<NFM<KGAT<RippleNet<MVIN<br>Accuracy: MCRec<CKE<FM<KGNN<GC-MC<KGAT<NFM<RippleNet<MVIN |
| | Wang et al. (2019b) | AUC: DKN<PER<CKE<Wide&Deep<RippleNet<LibFM<MKR<br>Accuracy: DKN<PER<CKE<Wide&Deep<RippleNet<LibFM<MKR |
| | Wang et al. (2021) | AUC: DKN<PER<Wide&Deep<RippleNet<MKR<Ripp-MKR<br>Accuracy: DKN<PER<Wide&Deep<RippleNet<MKR<Ripp-MKR |
| | Yang et al. (2021) | AUC: DKN<PER<CKE<SHINE<Wide&Deep<RippleNet<LibFM<MKR<SYT-RippleNet<br>Accuracy: DKN<PER<CKE<SHINE<Wide&Deep<RippleNet<LibFM<MKR<SYT-RippleNet |
| | Zou et al. (2022a) | AUC: PER<CKE<BPRMF<RippleNet<KGCN<KGNN-LS<KGAT<KGIN<MCCLK<br>F1: PER<CKE<BPRMF<RippleNet<KGCN<KGNN-LS<KGAT<KGIN<MCCLK |

| | | |
|---|---|---|
| | Ma et al. (2021) | AUC: DKN<PER<CKE<Wide&Deep<LibFM<MKR<KNI<MNI<br>Accuracy: DKN<PER<CKE<Wide&Deep<LibFM<MKR<KNI<MNI |
| | Wang et al. (2020a) | AUC: PER<CKE<RippleNet<KGCN<LibFM<KCER<br>Accuracy: PER<CKE<RippleNet<KGCN<LibFM<KCER |
| | Sun and Shagar (2020) | AUC: DKN<PER<CKE<Wide&Deep<RippleNet<LibFM<MUKG<br>Accuracy: DKN<PER<CKE<Wide&Deep<RippleNet<LibFM<MUKG |
| | Li et al. (2023c) | AUC: DKN<PER<CKE<Wide&Deep<RippleNet<LibFM<MKR<DISL<KGCN<br>Accuracy: DKN<PER<CKE<Wide&Deep<RippleNet<LibFM<KGCN<DISL<MKR |
| | Tang et al. (2022) | AUC: PER<CKE<SVD<LibFM+TransE<LibFM<RippleNet<KGCN<KE-GCN<KGNN-LS<br>F1: PER<CKE<SVD<RippleNet<LibFM+TransE<LibFM<KGNN-LS<KGCN<KE-GCN |
| | Wang et al. (2019a) | AUC: PER<CKE<SVD<RippleNet<LibFM+TransE<LibFM<KGNN-LS |
| | Ma et al. (2022a) | AUC: PER<LibFM<LibFM+TransE<RippleNet<KGCN<CKE<KNCR<KRNN<br>F1: PER<LibFM<LibFM+TransE<CKE<RippleNet<KGCN<KNCR<KRNN |
| | Ma et al. (2022b) | AUC: KGNN-LS<RippleNet<KGCN<CKE<BPRMF<CKAN<KGAT<KGET<br>F1: CKE<BPRMF<RippleNet<KGNN-LS<KGCN<KGAT<CKAN<KGET |
| | Wang et al. (2019c) | AUC: PER<CKE<SVD<LibFM+TransE<LibFM<RippleNet<KGCN<br>F1: PER<CKE<SVD<RippleNet<LibFM+TransE<LibFM<KGCN |
| | Zhang et al. (2021) | AUC: DKN<PER<RippleNet<CKE<KGCN<KCRec<br>Accuracy: DKN<PER<CKE<RippleNet<KGCN<KCRec<br>F1: PER<DKN<CKE<RippleNet<KGCN<KCRec |
| | Fan et al. (2022) | AUC: AKGE<KGAT<KNCR<PRSKG<CGAT<KGFER<br>F1: KGAT<AKGE<KNCR<PRSKG<CGAT<KGFER |
| | Zou et al. (2022b) | AUC: PER<CKE<BPRMF<RippleNet<KGCN<KGNN-LS<KGAT<CG-KGR<CKAN<KGIN <KGIC<br>F1: PER<CKE<BPRMF<RippleNet<KGCN<KGNN-LS<KGAT<CG-KGR<CKAN<KGIN<KGIC |
| | Khan et al. (2023) | AUC: PER<CKE<MCRec<RippleNet<KGAT<AKGE<NACF<DKEN<H-SAGE<br>Accuracy: PER<CKE<MCRec<RippleNet<KGAT<AKGE<NACF<DKEN<H-SAGE |
| | Dong et al. (2022) | AUC: DKN<PER<CKE<Wide&Deep<RippleNet<LibFM<KGCN<MKR<HRS<br>Accuracy: DKN<PER<CKE<Wide&Deep<RippleNet<LibFM<KGCN<MKR<HRS |
| | He et al. (2023) | AUC: RippleNet<CIEPA<GC-MC<NGCF<KGAT<KGNN-LS<CKAN<KGIN<ERSIF-KR<br>F1: RippleNet<GC-MC<CIEPA<NGCF<KGAT<KGNN-LS<CKAN<KGIN<ERSIF-KR |
| | Elahi and Halim (2022) | AUC: KGNN-LS<KGCN<RippleNet<K-NCR<KGAT<CKAN<GACF<br>F1: KGNN-LS<KGCN<RippleNet<KGAT<CKAN<K-NCR<GACF |
| | Yu et al. (2020) | AUC: DKN<PER<CKE<Wide&Deep<RippleNet<LibFM<GFEN<br>Accuracy: DKN<PER<CKE<Wide&Deep<RippleNet<LibFM<GFEN |
| | Guo et al. (2020) | AUC: DKN<PER<Wide&Deep<RippleNet<LibFM<KGCN<MKR<KGE-DNN<DKEN<br>Accuracy: DKN<PER<Wide&Deep<RippleNet<LibFM<KGCN<MKR<KGE-DNN<DKEN |
| | Tu et al. (2021) | AUC: CKE<RippleNet<NFM<KGAT<KCAN |
| | Wang et al. (2020c) | AUC: PER<CKE<BPRMF<RippleNet<KGCN<KGNN-LS<KGAT<CKAN<br>F1: PER<CKE<BPRMF<RippleNet<KGCN<KGNN-LS<KGAT<CKAN |
| | Chen et al. (2022a) | AUC: GC-MC<RippleNet<MKR<FM-KGE<DeepFM_GCN<br>F1: GC-MC<MKR<RippleNet<FM-KGE<DeepFM_GCN |
| | Chen et al. (2022b) | AUC: KGNN-LS<NFM<BPRMF<CKE<KGCN<RippleNet<KGAT<CG-KGR<CKAN<br>F1: KGNN-LS<NFM<BPRMF<CKE<KGCN<RippleNet<KGAT<CG-KGR<CKAN |
| | Lyu et al. (2022) | AUC: RippleNet<KGCN<KGAT<CKAN<BKANE<br>F1: RippleNet<KGCN<KGAT<CKAN<BKANE |

| | | |
|---|---|---|
| | Huang et al. (2022) | AUC: PER<CKE<RippleNet<MKR<Ripp-MKR<CAKR<br>Accuracy: PER<CKE<RippleNet<MKR<CAKR<Ripp-MKR |
| | Yang et al. (2023b) | AUC: DCBVN<GHCF<Pinsage<KGBIN<MetaKG<CKGE |
| | Tu et al. (2023) | AUC: GAT<CKE<DisenGCN<NFM<KGAT<DisenKGAT<KCAN<KGIN<DIKGNN |
| | Wang et al. (2023b) | AUC: SVD++<CKE<KGAT<MKR<KGCN<RippleNet<MVIN<KGCL<LKGR<UPIACM<br>Accuracy: CKE<SVD++<RippleNet<KGAT<MVIN<KGCL<MKR<KGCN<LKGR<UPIACM<br>F1: CKE<RippleNet<MKR<KGAT<SVD++<KGCN<MVIN<KGCL<LKGR<UPIACM |
| | Wang et al. (2022) | AUC: MKR<KGCN<RippleNet<GRE<br>Accuracy: KGCN<RippleNet<GRE<MKR<br>F1: KGCN<MKR<RippleNet<GRE |
| Dianping-Food | Hou et al. (2021) | AUC: PER<CKE<RippleNet<LibFM+TransE<KGCN<KGCN-CF<br>F1: PER<CKE<RippleNet<LibFM+TransE<KGCN<KGCN-CF |
| | Dai et al. (2022) | AUC: PER<KGNN-LS<KGCN<CKE<NGCF<GCMC<RippleNet<FM<NFM<COAT<br>F1: PER<CKE<RippleNet<KGNN-LS<KGCN<NGCF<GCMC<COAT<FM<NFM |
| | Qian et al. (2022) | AUC: PER<CKE<KGCN<KGAT<KGNN-LS<RippleNet<CKAN<RKAC<br>F1: PER<CKE<KGCN<KGNN-LS<RippleNet<KGAT<CKAN<RKAC |
| | Wang et al. (2019a) | AUC: PER<CKE<RippleNet<LibFM<SVD<LibFM+TransE<KGNN-LS |
| | Wang et al. (2019b) | AUC: CKE<LibFM<Wide&Deep<DKN<RippleNet<MKR<br>Accuracy: CKE<LibFM<Wide&Deep<DKN<RippleNet<MKR |
| | Wang et al. (2020c) | AUC: PER<CKE<BPRMF<KGCN<KGAT<KGNN-LS<RippleNet<CKAN<br>F1: PER<CKE<BPRMF<KGCN<KGNN-LS<RippleNet<KGAT<CKAN |
| | Shu and Huang (2023) | AUC: PER<CKE<LibFM<SVD<KGCN<KGNN-LS<LightGCN<MKR<Multi-Rec<br>F1: PER<CKE<SVD<LibFM<KGCN<KGNN-LS<LightGCN<MKR<Multi-Rec |
| | He et al. (2023) | AUC: GC-MC<NGCF<RippleNet<KGAT<CIEPA<KGNN-LS<CKAN<KGIN<ERSIF-KR<br>F1: GC-MC<NGCF<RippleNet<CIEPA<KGAT<KGNN-LS<CKAN<KGIN<ERSIF-KR |
| | Bai et al. (2023) | AUC: SPrank<Caser<CKE<GeoPFM<GeoMF<P-GCN<br>F1: SPrank<Caser<CKE<GeoPFM<GeoMF<P-GCN |
| | Chen et al. (2022b) | AUC: KGAT<KGNN-LS<CKE<BPRMF<KGCN<NFM<RippleNet<CKAN<CG-KGR<br>F1: CKE<BPRMF<KGAT< KGNN-LS<KGCN<NFM <RippleNet<CKAN<CG-KGR |
| Yelp | Feng et al. (2020a) | AUC: YoubtubeNet<RippleNet<DeepFM<DIN<DIEN<DSIN<KGAT<KPRN<ATBRG<br>RI: KPRN<KGAT<DSIN<DIEN<DIN<DeepFM<RippleNet<YoubtubeNet |
| | Wang et al. (2023b) | AUC: SVD++<CKE<MKR<KGCN<RippleNet<KGAT<MVIN<KGCL<LKGR<UPIACM<br>Accuracy: KGCN<SVD++<MKR<CKE<RippleNet<MVIN<KGAT<KGCL<LKGR<UPIACM<br>F1: CKE<MKR<KGCN<SVD++<RippleNet<MVIN<KGAT<KGCL<LKGR<UPIACM |
| | Bai et al. (2023) | AUC: Caser<SPrank<CKE<GeoMF<GeoPFM<P-GCN<br>F1: Caser<SPrank<GeoMF<CKE<GeoPFM<P-GCN |
| | Fan et al. (2022) | AUC: AKGE<KGAT<KNCR<PRSKG<CGAT<KGFER<br>F1: AKGE<KGAT<KNCR<PRSKG<CGAT<KGFER |
| | Khan et al. (2022) | AUC: PER<CKE<MCRec<KGAT<RippleNet<AKGE<NACF<SAGE<br>Accuracy: PER<CKE<MCRec<KGAT<RippleNet<AKGE<NACF<SAGE |
| | Cui et al. (2024) | AUC: CFKG<RippleNet<BPRMF<CKE<KGCN<KGAT<KGCL<LightGCN<RAKCR<br>F1: RippleNet<CFKG<CKE<BPRMF<KGCN<KGCL<KGAT<LightGCN<RAKCR |
| | Feng et al. (2020b) | AUC: YoutubeNet<DeepFM<DIN<DIEN<GIN<DSIN<MTBRN<br>Logloss: MTBRN<DSIN<GIN<DIEN<DIN<DeepFM<YoutubeNet |
| | Tu et al. (2023) | AUC: NFM<GAT<CKE<KGAT<DisenGCN<DisenKGAT<KCAN<KGIN<DIKGNN |
| | Tu et al. (2021) | AUC: NFM<CKE<RippleNet<KGAT<KCAN |

| | | |
|---|---|---|
| Official-Account | Yao et al. (2023) | AUC: DeepFM<NFM<CCPM<AFM<IFM<FiBiNET<DCN<WDL<AFN<DCNMix<AutoInt<DIFM<KNIFE<br>Logloss: KNIFE<DIFM<WDL<DCNMix<FiBiNET<AutoInt<DCN<IFM<AFM<CCPM<NFM<DeepFM<AFN |
| Mini-Program | Yao et al. (2023) | AUC: WDL<IFM<NFM<DeepFM<AFM<CCPM<DCN<AutoInt<FiBiNET<AFN<DCNMix<DIFM<KNIFE<br>Logloss: KNIFE<DIFM<DCNMix<FiBiNET<AutoInt<CCPM<AFN<DCN<DeepFM<IFM<NFM<AFM<WDL |
| Taobao-A | Wu et al. (2022) | AUC: KGCN<KGNN-LS<Caser<SASRec<RippleNet<GRU4Rec<LSAN<CUIKG<LightGCN<HAGERec<GCM<UBAR<br>F1: KGCN<KGNN-LS<RippleNet<Caser<SASRec<LSAN<CUIKG<HAGERec<LightGCN<GRU4Rec<GCM<UBAR |
| Taobao-B | Feng et al. (2020a) | AUC: RippleNet<YoubtubeNet<DeepFM<DIN<DIEN<KGAT<DSIN<KPRN<ATBRG<br>RI: KPRN<DSIN<KGAT<DIEN<DIN<DeepFM<YoubtubeNet<RippleNet |
| Tmall | Wu et al. (2022) | AUC: RippleNet<KGNN-LS<KGCN<CUIKG<SASRec<LSAN<LightGCN<GCM<GRU4Rec<Caser<HAGERec<UBAR<br>F1: RippleNet<KGNN-LS<KGCN<CUIKG<GCM<LSAN<SASRec<LightGCN<HAGERec<GRU4Rec<Caser<UBAR |
| Fund | Tu et al. (2023) | AUC: CKE<NFM<GAT<DisenGCN<DisenKGAT<KGAT<KGIN<KCAN<DIKGNN |
| E-commerce | Feng et al. (2020b) | AUC: YoutubeNet<DeepFM<DIN<DIEN<DSIN<GIN<MTBRN<br>Logloss: MTBRN<DSIN<GIN<DIEN<DIN<DeepFM<YoutubeNet |
| MaFengWo | Gao et al. (2023) | AUC: CF<Rank-GeoFM<Wide&Deep<Flashback<RippleNet<KGIN<KGDAE<br>Accuracy: CF<Rank-GeoFM<Wide&Deep<Flashback<RippleNet<KGIN<KGDAE |
| TripAdvisor | Gao et al. (2023) | AUC: CF<Rank-GeoFM<Wide&Deep<Flashback<RippleNet<KGIN<KGDAE<br>Accuracy: CF<Rank-GeoFM<Flashback<Wide&Deep<RippleNet<KGDAE<KGIN |
| Bing-News | Wang et al. (2018b) | AUC: DMF<LibFM<DeepFM<YouTubeNet<Wide&Deep<DSSM<KPCNN<DKN<br>F1: DMF<LibFM<DeepFM<YouTubeNet<Wide&Deep<DSSM<KPCNN<DKN |
| | Sun and Shagar (2020) | AUC: CKE<LibFM<Wide&Deep<DKN<RippleNet<MUKG<br>Accuracy: CKE<LibFM<Wide&Deep<DKN<RippleNet<MUKG |
| | Wang et al. (2018a) | AUC: SHINE<CKE<LibFM<Wide&Deep<DKN<RippleNet<br>Accuracy: CKE<SHINE<LibFM<Wide&Deep<DKN<RippleNet |
| | Khan et al. (2023) | AUC: PER<CKE<MCRec<RippleNet<AKGE<KGAT<NACF<DKEN<H-SAGE<br>Accuracy: PER<CKE<MCRec<RippleNet<AKGE<KGAT<NACF<DKEN<H-SAGE |
| | Hui et al. (2022) | AUC: SHINE<CKE<LibFM<Wide&Deep<DKN<ReBKC<br>Accuracy: CKE<SHINE<LibFM<Wide&Deep<DKN<ReBKC |
| | Lee et al. (2020) | AUC: LibFM<DeepFM<LSTUR<DKN<TEKGR<br>F1: LibFM<DeepFM<LSTUR<DKN<TEKGR |
| | Chen et al. (2024) | AUC: NRMS<NPA<KGCN<DKN<KGAT<KHGT<KGIN<RippleNet<KGCL<CKLF<br>F1: KGCN<DKN<KGAT<NRMS<NPA<KHGT<KGCL<RippleNet<KGIN<CKLF<br>MSE: CKLF<KGCL<KHGT<KGIN<RippleNet<KGAT<DKN<KGCN<NRMS<NPA<br>mAP: NPA<NRMS<KGCN<DKN<KGAT<KGIN<KHGT<RippleNet<KGCL<CKLF |
| MIND | Chen et al. (2024) | AUC: NPA<NRMS<KGCN<DKN<KGAT<RippleNet<KHGT<KGIN<KGCL<CKLF<br>F1: KGCN<DKN<KGAT<NRMS<RippleNet<KHGT<KGIN<KGCL<NPA<CKLF<br>MSE: KGCN<DKN<KGAT<NRMS<RippleNet<KHGT<KGIN<KGCL<NPA<CKLF<br>mAP: NPA<NRMS<KGCN<DKN<KGAT<RippleNet<KHGT<KGIN<KGCL<CKLF |
| Adressa-News | Lee et al. (2020) | AUC: LibFM<DKN<DeepFM<LSTUR<TEKGR<br>F1: DeepFM<LibFM<DKN<LSTUR<TEKGR |
| Weibo | Yang et al. (2020) | AUC: KG-M<KG-SNM<KG-RWSNM<br>F1: KG-M<KG-SNM<KG-RWSNM |

| | | |
|---|---|---|
| | | Recall: KG-SNM<KG-RWSNM<KG-M<br>Precision: KG-M<KG-SNM<KG-RWSNM |
| Nudge | Tangruamsub et al. (2022) | AUC: Deep-and-cross<Caregraph+TextEmb<Caregraph<Caregraph+TextSim |
| Food.com | Li et al. (2023a) | AUC: KGNN<GCN<KGNN-LS<GAT<HRR |
| MyFitnessPal | Li et al. (2023a) | AUC: KGNN<GCN<GAT<KGNN-LS<HRR |
| StackOverflow | Tang et al. (2024) | AUC: CF<KPCNN<DKN<CKE<KGAT<RippleNet<KGQR<EPAN-SERec<br>F1: CF<KPCNN<DKN<CKE<KGAT<KGQR<RippleNet<EPAN-SERec<br>Accuracy: CF<KPCNN<DKN<CKE<KGAT<RippleNet<KGQR<EPAN-SERec |
| Agricultural-Products | Xie et al. (2022) | AUC: CKE<RippleNet<LibFM<AFM<KGCN<KGAFM<br>Accuracy: CKE <LibFM<AFM<RippleNet<KGCN<KGAFM |
| MOOCCube | Zhang et al. (2023) | AUC: LR<DKN<FM<BPR<CKAN<KGNN-LS<RippleNet<KGAN |
| ML | Cao et al. (2021) | AUC: CKE(g)<BPR<CKE(g+t)<KGCN<RippleNet<KGAT<DEKR<br>Accuracy: CKE(g)<BPR<CKE(g+t)<RippleNet<KGCN<KGAT<DEKR<br>F1: CKE(g)<BPR<CKE(g+t)<KGCN<RippleNet<KGAT<DEKR |
| Amazon-Cross | Wang et al. (2024a) | Accuracy: VRKG4Rec<KGCN<KGNN-LS<CKAN<KGIN<MKNBL<br>MAE: MKNBL<KGIN<CKAN<KGNN-LS<KGCN<VRKG4Rec<br>RMSE: MKNBL<KGIN<CKAN<KGNN-LS<KGCN<VRKG4Rec |

# 6. Research Perspectives

In this section, we identify current research trends and challenges and describe potential research directions in KG-based CTR prediction.

## 6.1 Current Research Trends

### 6.1.1 Integration of User-item Interactions and KG

Previous works predominantly focus on leveraging KG information but overlook the collaborative signals in user-item interactions. In recent years, there has been a growing interest in integrating user-item interactions and KG for CTR prediction. They typically model user-item interactions and KG in different latent spaces or integrate them into a unified graph. Furthermore, researchers not only consider the existence of a relationship between a user and an item but also different types of user behaviors, such as purchasing, browsing and adding-to-cart.

### 6.1.2 Attention Mechanism in Propagation

Most propagation-based models focus on how to effectively design attention mechanisms to characterize the importance of different neighboring nodes. Main aggregation strategies have considered relationships between entities. Further studies emphasize personalized information and assert that entity aggregation should not be isolated from target user-item pairs. Therefore, they also incorporate user features into the attention mechanism.

## 6.2 Main Challenges

### 6.2.1 Data Sparsity

In the research of CTR prediction, a stubborn challenge arises from sparse user-item interactions. With limited supervised signals, stacking multiple GNN layers prone to overfitting. Wang et al. (2019a) proposed a label smoothness technique with the assumption of similar user preferences for nearby items in the KG. Zou et al. (2022a) explored cross-view contrastive learning including structural view, collaborative view and semantic view.

A further challenge exists in unbalanced information utilization between sparse user-item interactions and redundant knowledge. The overemphasis on external knowledge can potentially overshadow the importance of basic user-item interactions, leading to trivialization of information source of reasoning user preferences. Zou et al. (2022b) developed intra-graph and inter-graph interactive contrastive learning to provide balanced information utilization. Chen et al. (2022b) encoded historical interactions as guidance signals for knowledge extraction. Indeed, achieving a balanced information aggregation still requires a thoughtful and ongoing research effort.

### 6.2.2 Noise in Aggregation

Most of the existing studies use uniform sampling to select neighbors over the entire graph in information aggregation. Although KG contains rich semantic information, it is inevitable to introduce irrelevant nodes, which may lead to performance degradation. To solve this issue, Tang et al. (2019) employed an attention mechanism to weigh the neighbor nodes. It can help reduce the influence of irrelevant nodes on the representation of the target node but fails to filter out the unrelated entities. Additionally, random walks or graph search algorithms can be considered to create subgraphs, but they tend to overlook structural relationships and damage the stability of the model performance (Feng et al., 2020b). Another alternative is utilizing graph prune strategies to adjust the graph structure adaptively considering the mutual effect between user behaviors and target items (Feng et al., 2020a). It is still challenging to design non-uniform samplers while balancing the preservation of important relationships and minimizing introduced bias.

## 6.3 Future Directions

### 6.3.1 Dynamic Modeling

Previous studies hardly explore dynamic modeling in the field of KG-based CTR prediction. On the one hand, they roughly assume static KG that does not capture the dynamic nature of real-world scenarios, such as the news scenario. Future research can focus on developing techniques to continuously update and refine KG based on new information to reflect real-time changes. On the other hand, user preferences change over time and are influenced by external factors, such as current events or trends. Studying how to adapt to temporal dynamics of user behaviors could also be a promising avenue for future research.

### 6.3.2 Alternate Learning

The models within the framework of alternate learning have demonstrated improved performance in CTR prediction. The KGE module and CTR prediction module are not mutually independent but are highly correlated shared features between items and entities in (Wang et al., 2019b) and the extensions. They suffer from information loss caused by data compression and dimensional transformation and insufficient learning with TransE and MLP. It is worth exploring how to efficiently transfer and share information between two modules and investigating advanced networks in the alternate learning framework. What's more, researchers could also consider introducing other relevant tasks, such as entity recognition or relation extraction into multi-task learning, making the supervision for CTR prediction.

### 6.3.3 Multi-domain KG

While existing studies have incorporated domain-specific KG, a critical gap in the existing research pertains to the lack of absorbing multi-domain information. It is important to note that knowledge from different domains is often unbalanced. This means that leveraging the data from a source domain with rich information can complete the target domain by means of techniques such as transfer learning. In particular, there is less exploration of user-side supplementary information. For instance, social network data, such as friends, followers, or connections enables a holistic understanding of user preferences and behaviors. Therefore, the exploration of innovative techniques for the integration of user-centric knowledge and multi-domain KG holds immense potential for advancing CTR prediction.

### 6.3.4 Multi-modal Knowledge

A promising future research direction lies in the exploration of multi-modal knowledge. Multi-modal knowledge refers to the integration of diverse types of information, including text, images, audio, and even videos, to provide rich information for CTR prediction.

There has been a growing interest in leveraging knowledge-enhanced language modeling to improve the accuracy of CTR prediction models. By linking words in the text to related entities in the KG, text representations can be enriched with additional semantics and connections. Wang et al. (2018b) fused semantic-level and knowledge-level representations by convolutional operations to obtain final news representations. Lee et al. (2020) and Yang et al. (2020) made further improvements and extracted topic features in news titles for precise news representations. The application of knowledge-enhanced language modeling is not limited to the news domain and can be extended to general domains. A solution is to incorporate user reviews, which often contain detailed discussions about specific aspects of an item and effectively reflect attention factors that users consider when selecting items.

Future research in this area can focus on developing techniques to effectively extract, represent, and fuse multi-modal knowledge within KG. The synergy between KG and ongoing advancements in natural language processing and computer vision techniques offers a potential research direction to drive progress in CTR prediction.

### 6.3.5 Self-supervised Learning

Future research can investigate the potential of self-supervised learning techniques to enhance prediction performance. Self-supervised learning can leverage unlabeled data to learn high-quality representations. Designing effective self-supervised pre-training tasks tailored for CTR prediction using KG is crucial. This could involve developing novel pretext tasks that leverage the inherent structure and semantics of KG to generate supervisory signals for representation learning. Researchers could focus on developing contrastive learning models that distinguish between similar and dissimilar user-item interaction patterns, or generative models that predict missing links within the graph. Additionally, there is a need to investigate how to effectively combine self-supervised learning with supervised learning objectives to achieve optimal performance in CTR prediction.

### 6.3.6 Interpretability of CTR Prediction

In general, KG can boost the interpretability of CTR prediction models, making them more trustworthy and transparent. An interpretable CTR prediction model should be able to understand and explain why a user is likely to click on a specific item. Existing studies (Duan et al., 2023b; Liu and Miyazaki, 2022; Zhu et al., 2020) designed or mined crucial paths from the user to the item within the KG. Wang et al. (2018a) and Lyu et al. (2022) presented visualized connection examples to illustrate the reasons underlying prediction results. However, interpretation solely based on common relationships or attributes is not sufficient. It is essential to consider a broader range of factors and fine-grained features that contribute to prediction results. Future research can focus on conducting in-depth and systematic studies to enhance interpretability. It is desirable to design quantitative methods such as causal inference and conduct theoretical analysis to measure interpretability.

## 7. Conclusion

In this literature review, we systematically explore the domain of KG-based CTR prediction and discuss how KG can be incorporated to improve CTR prediction. Through an in-depth analysis, we identify and categorize three primary categories of models: embedding-based, path-based and propagation-based models and summarize their strengths and limitations. We delve into these models, discussing their underlying principles, methodologies, and key contributions. Moreover, we present the datasets and KG used in this paper and evaluate the performance of various models on these datasets. Finally, we discuss current research trends, such as integrating user-item interactions and KG and designing effective attention mechanisms during propagation, and highlight the main challenges that require further exploration, including how to balance information utilization between sparse user-item interactions and rich knowledge and reduce noise in aggregation. We also provide future research directions in this domain, such as the integration of multi-modal knowledge and exploitation of self-supervised learning to enhance CTR prediction. We hope this survey sheds light on the progress made in KG-based CTR prediction and provides valuable insights for future research endeavors.

## References


Bai, Z., Zhang, S., Li, P., and Chang, Y. (2023). Personalized Point-of-Interest Recommendation with Relation-Enhanced Graph Convolutional Network. In Proceedings of the 2022 11th International Conference on Networks, Communication and Computing (ICNCC 2022), Beijing, China. Association for Computing Machinery, 254–260.

Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., and Yakhnenko, O. (2013). Translating Embeddings for Modeling Multi-relational Data. In Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS 2013), Lake Tahoe, Nevada. Curran Associates Inc., 2787–2795.

Cao, X., Shi, Y., Yu, H., Wang, J., Wang, X., Yan, Z., and Chen, Z. (2021). DEKR: Description Enhanced Knowledge Graph for Machine Learning Method Recommendation. In Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021), Virtual Event, Canada. Association for Computing Machinery, New York, NY, USA, 203–212.

Chen, L., Bi, X., Fan, G., and Sun, H. (2022a). A Multitask Recommendation Algorithm Based on DeepFM and Graph Convolutional Network. Concurrency and Computation: Practice and Experience, 35.

Chen, X., Liang, Q., Chen, Y., Wang, P., Yu, H., and Luo, X. (2024). Cognitive-based Knowledge Learning Framework for Recommendation. Knowledge-Based Systems, 287, 111446.

Chen, Y., Yang, Y., Wang, Y., Bai, J., Song, X., and King, I. (2022b). Attentive Knowledge-aware Graph Convolutional Networks with Collaborative Guidance for Personalized Recommendation. In 2022 IEEE 38th International Conference on Data Engineering (ICDE 2022),

Cui, Y., Yu, H., Guo, X., Cao, H., and Wang, L. (2024). RAKCR: Reviews Sentiment-aware based Knowledge Graph Convolutional Networks for Personalized Recommendation. Expert Systems with Applications, 248, 123403.

Dai, Q., Wu, X.-M., Fan, L., Li, Q., Liu, H., Zhang, X., Wang, D., Lin, G., and Yang, K. (2022). Personalized Knowledge-aware Recommendation with Collaborative and Attentive Graph Convolutional Networks. Pattern Recognition, 128, 108628.

Dong, C., Ju, X., and Ma, Y. (2022). HRS: Hybrid Recommendation System based on Attention Mechanism and Knowledge Graph Embedding. In IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2022), Melbourne, VIC, Australia. Association for Computing Machinery, New York, NY, USA, 406–413.

Duan, H., Liang, X., Zhu, Y., Zhu, Z., and Liu, P. (2023a). Reducing Noise-triplets via Differentiable Sampling for Knowledge-enhanced Recommendation with Collaborative Signal Guidance. Neurocomputing, 558, 126771.

Duan, H., Liu, P., and Ding, Q. (2023b). RFAN: Relation-fused Multi-head Attention Network for Knowledge Graph Enhanced Recommendation. Applied Intelligence, 53(1), 1068-1083.

Elahi, E., Anwar, S., Shah, B., Halim, Z., Ullah, A., Rida, I., and Waqas, M. (2024). Knowledge Graph Enhanced Contextualized Attention-Based Network for Responsible User-Specific Recommendation. ACM Transactions on Intelligent Systems and Technology.

Elahi, E.and Halim, Z. (2022). Graph Attention-based Collaborative Filtering for User-specific Recommender System Using Knowledge Graph and Deep Neural Networks. Knowledge and Information Systems, 64(9), 2457-2480.

Fan, H., Zhong, Y., Zeng, G., and Ge, C. (2022). Improving Recommender System via Knowledge Graph Based Exploring User Preference. Applied Intelligence, 52(9), 10032-10044.

Feng, Y., Hu, B., Lv, F., Liu, Q., Zhang, Z., and Ou, W. (2020a). ATBRG: Adaptive Target-Behavior Relational Graph Network for Effective Recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020), Virtual Event, China. Association for Computing Machinery, New York, NY, USA, 2231–2240.

Feng, Y., Lv, F., Hu, B., Sun, F., Kuang, K., Liu, Y., Liu, Q., and Ou, W. (2020b). MTBRN: Multiplex Target-Behavior Relation Enhanced Network for Click-Through Rate Prediction. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (ICKM 2020), Virtual Event, Ireland. Association for Computing Machinery, New York, NY, USA, 2421–2428.

Gao, J., Peng, P., Lu, F., Claramunt, C., and Xu, Y. (2023). Towards Travel Recommendation Interpretability: Disentangling Tourist Decision-making Process via Knowledge Graph. Information Processing & Management, 60(4), 103369.

Gao, Y., Li, Y.-F., Lin, Y., Gao, H., and Khan, L. (2020). Deep Learning on Knowledge Graph for Recommender System: A Survey. arXiv preprint arXiv:2004.00387.

Guo, Q., Zhuang, F., Qin, C., Zhu, H., Xie, X., Xiong, H., and He, Q. (2020a). A Survey on Knowledge Graph-based Recommender Systems. IEEE Transactions on Knowledge and Data Engineering, 34(8), 3549-3568.

Guo, X., Lin, W., Li, Y., Liu, Z., Yang, L., Zhao, S., and Zhu, Z. (2020b). DKEN: Deep Knowledge-enhanced Network for Recommender Systems. Information Sciences, 540, 263-277.

Hai, Z.and Hongyan, H. (2023). Research and Implementation of Recommendation System Based on Neural Network. In 2023 IEEE 7th Information Technology and Mechatronics Engineering Conference (ITOEC 2023), 2460-2464.

He, Z., Hui, B., Zhang, S., Xiao, C., Zhong, T., and Zhou, F. (2023). Exploring Indirect Entity Relations for Knowledge Graph Enhanced Recommender System. Expert Systems with Applications, 213, 118984.

Hou, Z., Li, T., Fu, H., Liu, Q., Zhang, Z., and Hu, M. (2021). A Model Hybrid Recommendation Approach Based on Knowledge Graph Convolution Networks. In 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD 2021), Chengdu, China. IEEE, 283-288.

Hu, Z., Cai, S.-M., Wang, J., and Zhou, T. (2023). Collaborative Recommendation Model Based on Multi-modal Multi-view Attention Network: Movie and Literature Cases. Applied Soft Computing, 144, 110518.

Huang, W., Wu, J., Song, W., and Wang, Z. (2022). Cross Attention Fusion for Knowledge Graph Optimized Recommendation. Applied Intelligence, 52(9), 10297-10306.

Hui, B., Zhang, L., Zhou, X., Wen, X., and Nian, Y. (2022). Personalized Recommendation System Based on Knowledge Embedding and Historical Behavior. Applied Intelligence, 52(1), 954-966.

Ji, G., He, S., Xu, L., Liu, K., and Zhao, J. (2015). Knowledge Graph Embedding via Dynamic Mapping Matrix. In Annual Meeting of the Association for Computational Linguistics (ACL 2015), Beijing, China. Association for Computational Linguistics,

Khan, N., Ma, Z., Ullah, A., and Polat, K. (2022a). Categorization of Knowledge Graph Based Recommendation Methods and Benchmark Datasets from the Perspectives of Application Scenarios: A comprehensive survey. Expert Systems with Applications, 206, 117737.

Khan, N., Ma, Z., Ullah, A., and Polat, K. (2022b). Similarity Attributed Knowledge Graph Embedding Enhancement for Item Recommendation. Information Sciences, 613, 69-95.

Khan, N., Ma, Z., Yan, L., and Ullah, A. (2023). Hashing-based Semantic Relevance Attributed Knowledge Graph Embedding Enhancement for Deep Probabilistic Recommendation. Applied Intelligence, 53(2), 2295-2320.

Lee, D., Oh, B., Seo, S., and Lee, K.-H. (2020). News Recommendation with Topic-Enriched Knowledge Graphs. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management (ICKM 2020), Virtual Event, Ireland. Association for Computing Machinery, New York, NY, USA, 695–704.

Li, C., Cao, Y., Zhu, Y., Cheng, D., Li, C., and Morimoto, Y. (2024). Ripple Knowledge Graph Convolutional Networks for Recommendation Systems. Machine Intelligence Research.

Li, D., Zaki, M. J., and Chen, C.-H. (2023a). Health-guided Recipe Recommendation over Knowledge Graphs. Journal of Web Semantics, 75, 100743.

Li, J.-L., Du, Z.-J., and Zhou, J.-T. (2019). Recommendation Algorithm Based on Dual Attention Mechanism and Explicit Feedback.

Li, Y., Fu, S., Feng, H., Zeng, Y., Wang, J., Jiang, Z., and Zhang, L. (2023b). Simple and Efficient Knowledge Graph Attention Network for Recommendation. In 2023 International Conference on Cyber-Physical Social Intelligence (ICCSI 2023), 333-338.

Li, Y., Guo, X., Lin, W., Zhong, M., Li, Q., Liu, Z., Zhong, W., and Zhu, Z. (2023c). Learning Dynamic User Interest Sequence in Knowledge Graphs for Click-Through Rate Prediction. IEEE Transactions on Knowledge and Data Engineering, 35(1), 647-657.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning Entity and Relation Embeddings for Knowledge Graph Completion. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI 2015), Austin, Texas. AAAI Press, Palo Alto, California USA, 2181–2187.

Liu, X., Yang, B., and Xu, J. (2023). SKGCR: Self-supervision Enhanced Knowledge-aware Graph Collaborative Recommendation. Applied Intelligence, 53(17), 19872–19891.

Liu, Y.and Miyazaki, J. (2022). Knowledge-aware Attentional Neural Network for Review-based Movie Recommendation with Explanations. Neural Computing and Applications, 35(3), 2717–2735.

Luo, X., Liu, L., Yang, Y., Bo, L., Cao, Y., Wu, J., Li, Q., Yang, K., and Zhu, K. Q. (2020). AliCoCo: Alibaba E-commerce Cognitive Concept Net. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data (SIGMOD 2020), Portland, OR, USA. Association for Computing Machinery, New York, NY, USA, 313–327.

Lyu, Y., Su, G., Wang, J., and Xing, Y. (2022). Bidirectional Knowledge-Aware Attention Network over Knowledge Graph for Explainable Recommendation. In Proceedings of the 2022 5th International Conference on Machine Learning and Natural Language Processing (MLNLP 2022), Sanya, China. Association for Computing Machinery, New York, NY, USA, 170–174.

Ma, R., Guo, F., Li, Z., and Zhao, L. (2022a). Knowledge Graph Random Neural Networks for Recommender Systems. Expert Systems with Applications, 201, 117120.

Ma, R., Guo, F., Zhao, L., Mei, B., Bu, X., Wu, H., and Song, E. (2022b). Knowledge Graph Extrapolation Network with Transductive Learning for Recommendation. Applied Sciences, 12(10).

Ma, R., Yang, X., Li, J., and Gao, F. (2023). Bayesian Personalized Ranking based on Knowledge Graph. In Proceedings of the 2022 11th International Conference on Computing and Pattern Recognition (ICCPR 2022), Beijing, China. Association for Computing Machinery, New York, NY, USA, 545–550.

Ma, X., Dong, L., Wang, Y., Li, Y., and Zhang, H. (2021). MNI: An Enhanced Multi-task Neighborhood Interaction Model for Recommendation on Knowledge Graph. PLoS One, 16, e0258410.

Noy, N., Gao, Y., Jain, A., Narayanan, A., Patterson, A., and Taylor, J. (2019). Industry-Scale Knowledge Graphs: Lessons and Challenges: Five Diverse Technology Companies Show How It's Done. Queue, 17(2), 48–75.

Ong, R. K., Qiu, W., and Khong, A. W. H. (2023). Quad-Tier Entity Fusion Contrastive Representation Learning for Knowledge Aware Recommendation System. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM 2023), Birmingham, United Kingdom. Association for Computing Machinery, 1949–1959.

Peng, W., Cheng, J., Wang, Z., Zhao, M., and Wu, X. (2023). DS-KGAT: A Deep Session GAT with Knowledge Enhancement for CTR Prediction. In 2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA 2023), Chongqing, China, 1590-1594.

Qian, F., Zhu, Y., Chen, H., Chen, J., Zhao, S., and Zhang, Y. (2022). Reduce Unrelated Knowledge through Attribute Collaborative Signal for Knowledge Graph Recommendation. Expert Systems with Applications, 201, 117078.

Qu, Y., Bai, T., Zhang, W., Nie, J., and Tang, J. (2019). An End-to-end Neighborhood-based Interaction Model for Knowledge-enhanced Recommendation. In Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data (DLP-KDD 2019), Anchorage, Alaska. Association for Computing Machinery, New York, NY, USA, Article 8.

Shu, H.and Huang, J. (2021). User-Preference Based Knowledge Graph Feature and Structure Learning for Recommendation. In 2021 IEEE International Conference on Multimedia and Expo (ICME 2021), 1-6.

Shu, H.and Huang, J. (2023). Multi-task Feature and Structure Learning for User-preference based Knowledge-aware Recommendation. Neurocomputing, 532, 43-55.

Song, X., Qin, J., and Ren, Q. (2023). A Recommendation Algorithm Combining Local and Global Interest Features. Electronics, 12(8), 1857.

Sun, J.and Shagar, M. M. B. (2020). MUKG: Unifying Multi-Task and Knowledge Graph Method for Recommender System. In Proceedings of the 2020 2nd International Conference on Image Processing and Machine Vision (IPMV 2020), Bangkok, Thailand. Association for Computing Machinery, New York, NY, USA, 14–21.

Sun, Y.and Li, H. (2022). Personalized Recommendation Based on Knowledge Map and Multi Feature Fusion. In 2022 11th International Conference on Communications, Circuits and Systems (ICCCAS 2022), Singapore, Singapore. IEEE, 301-306.

Tai, C.-Y., Wu, M.-R., Chu, Y.-W., Chu, S.-Y., and Ku, L.-W. (2020). MVIN: Learning Multiview Items for Recommendation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020), Virtual Event, China. Association for Computing Machinery, New York, NY, USA, 99–108.

Tang, M., Wu, D., Zhang, S., and Gao, W. (2024). EPAN-Serec: Expertise Preference-aware Networks for Software Expert Recommendations with Knowledge Graph. Expert Systems with Applications, 244, 122985.

Tang, X., Wang, T., Yang, H., and Song, H. (2019). AKUPM: Attention-Enhanced Knowledge-Aware User Preference Model for Recommendation. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD 2019), Anchorage, AK, USA. Association for Computing Machinery, New York, NY, USA, 1891–1899.

Tang, X., Yang, J., Xiong, D., Luo, Y., Wang, H., and Peng, D. (2022). Knowledge-enhanced Graph Convolutional Network for Recommendation. Multimedia Tools and Applications, 81(20), 28899-28916.

Tangruamsub, S., Kappaganthu, K., O'donovan, J., and Madan, A. (2022). CareGraph: A Graph-based Recommender System for Diabetes Self-Care.

Tu, K., Cui, P., Wang, D., Zhang, Z., Zhou, J., Qi, Y., and Zhu, W. (2021). Conditional Graph Attention Networks for Distilling and Refining Knowledge Graphs in Recommendation. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM 2021), Virtual Event, Queensland, Australia. Association for Computing Machinery, New York, NY, USA, 1834–1843.

Tu, K., Qu, W., Wu, Z., Zhang, Z., Liu, Z., Zhao, Y., Wu, L., Zhou, J., and Zhang, G. (2023). Disentangled Interest Importance Aware Knowledge Graph Neural Network for Fund Recommendation. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (ICKM 2023), Birmingham, United Kingdom. Association for Computing Machinery, New York, NY, USA, 2482–2491.

Uyar, A.and Aliyu, F. M. (2015). Evaluating Search Features of Google Knowledge Graph and Bing Satori. Online Information Review, 39(2), 197-213.

Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., and Guo, M. (2018a). RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management (ICKM 2018), Torino, Italy. Association for Computing Machinery, New York, NY, USA, 417–426.

Wang, H., Zhang, F., Xie, X., and Guo, M. (2018b). DKN: Deep Knowledge-Aware Network for News Recommendation. In Proceedings of the 2018 World Wide Web Conference (WWW 2018). International World Wide Web Conferences Steering, Committee, Republic and Canton of Geneva, Switzerland, 1835-1844.

Wang, H., Zhang, F., Zhang, M., Leskovec, J., Zhao, M., Li, W., and Wang, Z. (2019a). Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems. In Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD 2019), Anchorage, AK, USA. Association for Computing Machinery, New York, NY, USA, 968–977.

Wang, H., Zhang, F., Zhao, M., Li, W., Xie, X., and Guo, M. (2019b). Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation. In The World Wide Web Conference (WWW 2019), San Francisco, CA, USA. Association for Computing Machinery, New York, NY, USA, 2000–2010.

Wang, H., Zhao, M., Xie, X., Li, W., and Guo, M. (2019c). Knowledge Graph Convolutional Networks for Recommender Systems. In Proceedings of the 2019 World Wide Web Conference (WWW 2019), San Francisco, CA, USA. Association for Computing Machinery, New York, NY, USA, 3307-3313.

Wang, J., Kou, Y., Zhang, Y., Gao, N., and Tu, C. (2020a). Leveraging Knowledge Context Information to Enhance Personalized Recommendation. In Neural Information Processing Bangkok, Thailand. Springer International Publishing (ICONIP 2020), 467-478.

Wang, J., Shi, Y., Cheng, L., Zhang, K., and Chen, Z. (2022). GRE: A GAT-Based Relation Embedding Model of Knowledge Graph for Recommendation. In Computer Supported Cooperative Work and Social Computing (ChineseCSCW 2021), Xiangtan, China. Springer, Singapore, 77-91.

Wang, J., Shi, Y., Yu, H., Wang, X., Yan, Z., and Kong, F. (2023a). Mixed-Curvature Manifolds Interaction Learning for Knowledge Graph-aware Recommendation. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2023), Taipei, Taiwan. Association for Computing Machinery, 372–382.

Wang, J., Shi, Y., Yu, H., Yan, Z., Li, H., and Chen, Z. (2023b). A Novel KG-based Recommendation Model via Relation-aware Attentional GCN. Knowledge-Based Systems, 275, 110702.

Wang, L.-E., Qi, Y., Sun, Z., and Li, X. (2024a). MKNBL: Joint Multi-Channel Knowledge-Aware Network and Broad Learning for Sparse Knowledge Graph-Based Recommendation. Neurocomputing, 575, 127277.

Wang, T., Shi, D., Wang, Z., Xu, S., and Xu, H. (2020b). MRP2Rec: Exploring Multiple-Step Relation Path Semantics for Knowledge Graph-Based Recommendations. IEEE Access, 8, 134817-134825.

Wang, W., Shen, X., Yi, B., Zhang, H., Liu, J., and Dai, C. (2024b). Knowledge-aware Fine-grained Attention Networks with Refined Knowledge Graph Embedding for Personalized Recommendation. Expert Systems with Applications, 249, 123710.

Wang, X., Gao, M., Lu, Z., Wang, Z., Zhang, J., and Zhang, Y. (2019d). DMCM: A Deep Multi-Channel Model for Dynamic Movie Recommendation. In Neural Information Processing (ICONIP 2019), Sydney, NSW, Australia. Springer International Publishing, 425-432.

Wang, Y., Dong, L., Li, Y., and Zhang, H. (2021). Multitask Feature Learning Approach for Knowledge Graph Enhanced Recommendations with RippleNet. PLoS One, 16(5), e0251162.

Wang, Z., Lin, G., Tan, H., Chen, Q., and Liu, X. (2020c). CKAN: Collaborative Knowledge-aware Attentive Network for Recommender Systems. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020), Virtual Event, China. Association for Computing Machinery, New York, NY, USA, 219–228.

Wang, Z., Xu, Y., Wang, Z., Fan, R., Guo, Y., and Li, W. (2023c). Knowledge Graph-aware Deep Interest Extraction Network on Sequential Recommendation.

Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014). Knowledge Graph Embedding by Translating on Hyperplanes. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI 2014), Québec City, Québec, Canada. AAAI Press, Palo Alto, California USA, 1112–1119.

Wong, C. M., Feng, F., Zhang, W., Vong, C. M., Chen, H., Zhang, Y., He, P., Chen, H., Zhao, K., and Chen, H. (2021). Improving Conversational Recommender System by Pretraining Billion-scale Knowledge Graph. In 2021 IEEE 37th International Conference on Data Engineering (ICDE 2021), Chania, Greece, 2607-2612.

Wu, X., Li, Y., Wang, J., Qian, Q., and Guo, Y. (2022). UBAR: User Behavior-Aware Recommendation with Knowledge Graph. Knowledge-Based Systems, 254, 109661.

Xiao, Y., Li, C., and Liu, V. (2022). DFM-GCN: A Multi-Task Learning Recommendation Based on a Deep Graph Neural Network. Mathematics, 10(5).

Xie, H., Yang, J., Huang, C., Wang, Z., and Liu, Y. (2022). Recommendation Algorithm for Agricultural Products Based on Attention Factor Decomposer and Knowledge Graph. In 2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML 2022), Hangzhou, China. IEEE, 626-631.

Yang, B., Yih, S. W.-T., He, X., Gao, J., and Deng, L. (2015). Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In Proceedings of the International Conference on Learning Representations (ICLR) 2015,

Yang, F., Yue, Y., Li, G., Payne, T. R., and Man, K. L. (2023a). KEMIM: Knowledge-Enhanced User Multi-Interest Modeling for Recommender Systems. IEEE Access, 11, 55425-55434.

Yang, J., Wan, J., Wang, Y., and Mao, Y. (2020). Social Network-based News Recommendation with Knowledge Graph. In 2020 IEEE International Conference on Information Technology,Big Data and Artificial Intelligence (ICIBA 2020), Chongqing, China, 1255-1260.

Yang, Y.and Zhai, P. (2022). Click-Through Rate Prediction in Online Advertising: A Literature Review. Information Processing & Management, 59(2), 102853.

Yang, Y., Zhang, C., Song, X., Dong, Z., Zhu, H., and Li, W. (2023b). Contextualized Knowledge Graph Embedding for Explainable Talent Training Course Recommendation. ACM Trans. Inf. Syst., 42(2), Article 33.

Yang, Z., Cheng, J., Zhou, Y., Deng, H., Sun, Z., and Dong, A. (2021). Music Recommendation Algorithm Based on Knowledge graph Propagation User Preference. In 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE 2021), Kyoto, Japan. IEEE, 1-6.

Yao, Y., Zhang, Z., Yang, K., Liang, H., Yan, Q., Zhuang, F., Xu, Y., Diao, B., and Li, C. (2023). A Knowledge Enhanced Hierarchical Fusion Network for CTR Prediction under Account Search Scenario in WeChat. In Companion Proceedings of the ACM Web Conference 2023 (WWW 2023). Association for Computing Machinery, New York, NY, USA, 475–479.

Yu, M., Zhen, C., Yu, R., Li, X., Xu, T., Zhao, M., Liu, H., Yu, J., and Dong, X. (2020). GFEN: Graph Feature Extract Network for Click-Through Rate Prediction. In Neural Information Processing (ICONIP 2020), Bangkok, Thailand. Springer International Publishing, 444-454.

Zhang, H., Shen, X., Yi, B., Wang, W., and Feng, Y. (2023). KGAN: Knowledge Grouping Aggregation Network for Course Recommendation in MOOCs. Expert Systems with Applications, 211, 118344.

Zhang, L., Kang, Z., Sun, X., Sun, H., Zhang, B., and Pu, D. (2021). KCRec: Knowledge-aware Representation Graph Convolutional Network for Recommendation. Knowledge-Based Systems, 230, 107399.

Zhang, X.and Yang, Y. (2020). User Local and Global Interest Depth Interaction for Recommendation Algorithm. In 2020 7th International Conference on Information Science and Control Engineering (ICISCE 2020), Changsha, China, 1499-1504.

Zhao, W. X., He, G., Yang, K., Dou, H., Huang, J., Ouyang, S., and Wen, J.-R. (2019). KB4Rec: A Data Set for Linking Knowledge Bases with Recommender Systems. Data Intelligence, 1(2), 121-136.

Zhu, Q., Zhou, X., Wu, J., Tan, J., and Guo, L. (2020). A Knowledge-Aware Attentional Reasoning Network for Recommendation. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2020), New York, New York, USA. AAAI Press, Palo Alto, California USA, 6999-7006.

Zou, D., Wei, W., Mao, X.-L., Wang, Z., Qiu, M., Zhu, F., and Cao, X. (2022a). Multi-level Cross-view Contrastive Learning for Knowledge-aware Recommender System. In Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022), Madrid, Spain. Association for Computing Machinery, New York, NY, USA, 1358–1368.

Zou, D., Wei, W., Wang, Z., Mao, X.-L., Zhu, F., Fang, R., and Chen, D. (2022b). Improving Knowledge-aware Recommendation with Multi-level Interactive Contrastive Learning. In Proceedings of the 31st ACM International Conference on Information & Knowledge Management (ICKM 2022), Atlanta, GA, USA. Association for Computing Machinery, New York, NY, USA, 2817–2826.

## A.1 Research Articles included in This Survey

Table A1. The list of reviewed articles.

| No. | Title | Year | Authors | Publication Outlet |
|---|---|---|---|---|
| 1 | DKN: Deep Knowledge-Aware Network for News Recommendation | 2018 | Wang et al. | Proceedings of the 2018 World Wide Web Conference (WWW 2018) |
| 2 | RippleNet: Propagating User Preferences on the Knowledge Graph for Recommender Systems | 2018 | Wang et al. | Proceedings of the 27th ACM International Conference on Information and Knowledge Management (ICKM 2018) |
| 3 | AKUPM: Attention-Enhanced Knowledge-Aware User Preference Model for Recommendation | 2019 | Tang et al. | Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD 2019) |
| 4 | An end-to-end neighborhood-based interaction model for knowledge-enhanced recommendation | 2019 | Qu et al. | Proceedings of the 1st International Workshop on Deep Learning Practice for High-Dimensional Sparse Data (DLP-KDD 2019) |
| 5 | DMCM: A Deep Multi-Channel Model for Dynamic Movie Recommendation | 2019 | Wang et al. | Neural Information Processing (ICONIP 2019) |
| 6 | Knowledge Graph Convolutional Networks for Recommender Systems | 2019 | Wang et al. | Proceedings of the 2019 World Wide Web Conference (WWW 2019) |
| 7 | Knowledge-aware Graph Neural Networks with Label Smoothness Regularization for Recommender Systems | 2019 | Wang et al. | Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (SIGKDD 2019) |
| 8 | Multi-Task Feature Learning for Knowledge Graph Enhanced Recommendation | 2019 | Wang et al. | The World Wide Web Conference (WWW 2019) |
| 9 | Recommendation Algorithm Based on Dual Attention Mechanism and Explicit Feedback | 2019 | Li et al. | Pre-print |
| 10 | A Knowledge-Aware Attentional Reasoning Network for Recommendation | 2020 | Zhu et al. | Proceedings of the AAAI Conference on Artificial Intelligence (AAAI 2020) |
| 11 | ATBRG: Adaptive Target-Behavior Relational Graph Network for Effective Recommendation | 2020 | Feng et al. | Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020) |
| 12 | DKEN: Deep knowledge-enhanced network for recommender systems | 2020 | Guo et al. | Information Sciences |
| 13 | GFEN: Graph Feature Extract Network for Click-Through Rate Prediction | 2020 | Yu et al. | Neural Information Processing (ICONIP 2020) |
| 14 | Leveraging Knowledge Context Information to Enhance Personalized Recommendation | 2020 | Wang et al. | Neural Information Processing |
| 15 | MRP2Rec: Exploring Multiple-Step Relation Path Semantics for Knowledge Graph-Based Recommendations | 2020 | Wang et al. | IEEE Access |
| 16 | MTBRN: Multiplex Target-Behavior Relation Enhanced Network for Click-Through Rate Prediction | 2020 | Feng et al. | Proceedings of the 29th ACM International Conference on Information & Knowledge Management (ICKM 2020) |
| 17 | MUKG: Unifying Multi-Task and Knowledge Graph Method for Recommender System | 2020 | Sun and Shagar | Proceedings of the 2020 2nd International Conference on Image Processing and Machine Vision (IPMV 2020) |
| 18 | MVIN: Learning Multiview Items for Recommendation | 2020 | Tai et al. | Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020) |

| | | | | |
|---|---|---|---|---|
| 19 | News Recommendation with Topic-Enriched Knowledge Graphs | 2020 | Lee et al. | Proceedings of the 29th ACM International Conference on Information & Knowledge Management (ICKM 2020) |
| 20 | Social network-based News Recommendation with Knowledge Graph | 2020 | Yang et al. | 2020 IEEE International Conference on Information Technology,Big Data and Artificial Intelligence (ICIBA 2020) |
| 21 | User Local and Global Interest Depth Interaction for Recommendation Algorithm | 2020 | Zhang and Yang | 2020 7th International Conference on Information Science and Control Engineering (ICISCE 2020) |
| 22 | A Model Hybrid Recommendation Approach Based on Knowledge Graph Convolution Networks | 2021 | Hou et al. | 2021 4th International Conference on Artificial Intelligence and Big Data (ICAIBD 2021) |
| 23 | Conditional Graph Attention Networks for Distilling and Refining Knowledge Graphs in Recommendation | 2021 | Tu et al. | Proceedings of the 30th ACM International Conference on Information & Knowledge Management (CIKM 2021) |
| 24 | DEKR: Description Enhanced Knowledge Graph for Machine Learning Method Recommendation | 2021 | Cao et al. | Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2021) |
| 25 | Improving Conversational Recommender System by Pretraining Billion-scale Knowledge Graph | 2021 | Wong et al. | 2021 IEEE 37th International Conference on Data Engineering (ICDE 2021) |
| 26 | KCRec: Knowledge-aware representation Graph Convolutional Network for Recommendation | 2021 | Zhang et al. | Knowledge-Based Systems |
| 27 | MNI: An enhanced multi-task neighborhood interaction model for recommendation on knowledge graph | 2021 | Ma et al. | PLoS One |
| 28 | Multitask feature learning approach for knowledge graph enhanced recommendations with RippleNet | 2021 | Wang et al. | PLoS One |
| 29 | Music Recommendation Algorithm Based on Knowledge graph Propagation User Preference | 2021 | Yang et al. | 2021 IEEE 30th International Symposium on Industrial Electronics (ISIE 2021) |
| 30 | User-Preference Based Knowledge Graph Feature and Structure Learning for Recommendation | 2021 | Shu and Huang | 2021 IEEE International Conference on Multimedia and Expo (ICME 2021) |
| 31 | A multitask recommendation algorithm based on DeepFM and Graph Convolutional Network | 2022 | Chen et al. | Concurrency and Computation: Practice and Experience |
| 32 | Attentive Knowledge-aware Graph Convolutional Networks with Collaborative Guidance for Personalized Recommendation | 2022 | Chen et al. | 2022 IEEE 38th International Conference on Data Engineering (ICDE 2022) |
| 33 | Bidirectional Knowledge-Aware Attention Network over Knowledge Graph for Explainable Recommendation | 2022 | Lyu et al. | Proceedings of the 2022 5th International Conference on Machine Learning and Natural Language Processing (MLNLP 2022) |
| 34 | CareGraph: A Graph-based Recommender System for Diabetes Self-Care | 2022 | Tangruamsub et al. | Pre-print |
| 35 | CKAN: Collaborative Knowledge-aware Attentive Network for Recommender Systems | 2022 | Wang et al. | Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2020) |
| 36 | Cross attention fusion for knowledge graph optimized recommendation | 2022 | Huang et al. | Applied Intelligence |
| 37 | DFM-GCN: A Multi-Task Learning Recommendation Based on a Deep Graph Neural Network | 2022 | Xiao et al. | Mathematics |
| 38 | Graph attention-based collaborative filtering for user-specific | 2022 | Elahi and | Knowledge and Information Systems |

| | | | | |
|---|---|---|---|---|
| | recommender system using knowledge graph and deep neural networks | | Halim | |
| 39 | GRE: A GAT-Based Relation Embedding Model of Knowledge Graph for Recommendation | 2022 | Wang et al. | Computer Supported Cooperative Work and Social Computing (ChineseCSCW 2021) |
| 40 | HRS: Hybrid Recommendation System based on Attention Mechanism and Knowledge Graph Embedding | 2022 | Dong et al. | IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT 2022) |
| 41 | Improving Knowledge-aware Recommendation with Multi-level Interactive Contrastive Learning | 2022 | Zou et al. | Proceedings of the 31st ACM International Conference on Information & Knowledge Management (ICKM 2022) |
| 42 | Improving recommender system via knowledge graph based exploring user preference | 2022 | Fan et al. | Applied Intelligence |
| 43 | Knowledge Graph Extrapolation Network with Transductive Learning for Recommendation | 2022 | Ma et al. | Applied Sciences |
| 44 | Knowledge Graph Random Neural Networks for Recommender Systems | 2022 | Ma et al. | Expert Systems with Applications |
| 45 | Knowledge-aware attentional neural network for review-based movie recommendation with explanations | 2022 | Liu and Miyazaki | Neural Computing and Applications |
| 46 | Knowledge-enhanced graph convolutional network for recommendation | 2022 | Tang et al. | Multimedia Tools and Applications |
| 47 | Multi-level Cross-view Contrastive Learning for Knowledge-aware Recommender System | 2022 | Zou et al. | Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2022) |
| 48 | Personalized knowledge-aware recommendation with collaborative and attentive graph convolutional networks | 2022 | Dai et al. | Pattern Recognition |
| 49 | Personalized Recommendation Based on Knowledge Map and Multi Feature Fusion | 2022 | Sun and Li | 2022 11th International Conference on Communications, Circuits and Systems (ICCCAS 2022) |
| 50 | Personalized recommendation system based on knowledge embedding and historical behavior | 2022 | Hui et al. | Applied Intelligence |
| 51 | Recommendation algorithm for agricultural products based on attention factor decomposer and knowledge graph | 2022 | Xie et al. | 2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML 2022) |
| 52 | Reduce unrelated Knowledge through Attribute Collaborative signal for knowledge graph recommendation | 2022 | Qian et al. | Expert Systems with Applications |
| 53 | Similarity attributed knowledge graph embedding enhancement for item recommendation | 2022 | Khan et al. | Information Sciences |
| 54 | UBAR: User Behavior-Aware Recommendation with knowledge graph | 2022 | Wu et al. | Knowledge-Based Systems |
| 55 | A Knowledge Enhanced Hierarchical Fusion Network for CTR Prediction under Account Search Scenario inWeChat | 2023 | Yao et al. | Companion Proceedings of the ACM Web Conference 2023 (WWW 2023) |
| 56 | A novel KG-based recommendation model via relation-aware attentional GCN | 2023 | Wang et al. | Knowledge-Based Systems |
| 57 | A Recommendation Algorithm Combining Local and Global Interest Features | 2023 | Song et al. | Electronics |
| 58 | Bayesian Personalized Ranking based on Knowledge Graph | 2023 | Ma et al. | Proceedings of the 2022 11th International Conference on Computing and Pattern |

| | | | | |
|---|---|---|---|---|
| | | | | Recognition (ICCPR 2022) |
| 59 | Collaborative Recommendation Model Based on Multi-modal Multi-view Attention Network: Movie and literature cases | 2023 | Hu et al. | Applied Soft Computing |
| 60 | Contextualized Knowledge Graph Embedding for Explainable Talent Training Course Recommendation | 2023 | Yang et al. | ACM Transactions on Information Systems |
| 61 | Disentangled Interest importance aware Knowledge Graph Neural Network for Fund Recommendation | 2023 | Tu et al. | Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (ICKM 2023) |
| 62 | DS-KGAT: A Deep Session GAT with Knowledge Enhancement for CTR Prediction | 2023 | Peng et al. | 2023 IEEE 3rd International Conference on Information Technology, Big Data and Artificial Intelligence (ICIBA 2023) |
| 63 | Exploring indirect entity relations for knowledge graph enhanced recommender system | 2023 | He et al. | Expert Systems with Applications |
| 64 | Hashing-based semantic relevance attributed knowledge graph embedding enhancement for deep probabilistic recommendation | 2023 | Khan et al. | Applied Intelligence |
| 65 | Health-guided recipe recommendation over knowledge graphs | 2023 | Li et al. | Journal of Web Semantics |
| 66 | KEMIM: Knowledge-Enhanced User Multi-Interest Modeling for Recommender Systems | 2023 | Yang et al. | IEEE Access |
| 67 | KGAN: Knowledge Grouping Aggregation Network for course recommendation in MOOCs | 2023 | Zhang et al. | Expert Systems with Applications |
| 68 | Knowledge Graph-aware Deep Interest Extraction Network on Sequential Recommendation | 2023 | Wang et al. | Pre-print |
| 69 | Learning Dynamic User Interest Sequence in Knowledge Graphs for Click-Through Rate Prediction | 2023 | Li et al. | IEEE Transactions on Knowledge and Data Engineering |
| 70 | Mixed-Curvature Manifolds Interaction Learning for Knowledge Graph-aware Recommendation | 2023 | Wang et al. | Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2023) |
| 71 | Multi-task feature and structure learning for user-preference based knowledge-aware recommendation | 2023 | Shu and Huang | Neurocomputing |
| 72 | Personalized Point-of-Interest Recommendation with Relation-Enhanced Graph Convolutional Network | 2023 | Bai et al. | Proceedings of the 2022 11th International Conference on Networks, Communication and Computing (ICNCC 2022) |
| 73 | Quad-Tier Entity Fusion Contrastive Representation Learning for Knowledge Aware Recommendation System | 2023 | Ong et al. | Proceedings of the 32nd ACM International Conference on Information and Knowledge Management (CIKM 2023) |
| 74 | Reducing noise-triplets via differentiable sampling for knowledge-enhanced recommendation with collaborative signal guidance | 2023 | Duan et al. | Neurocomputing |
| 75 | Research and Implementation of Recommendation System Based on Neural Network | 2023 | Hai and Hongyan | 2023 IEEE 7th Information Technology and Mechatronics Engineering Conference (ITOEC 2023) |
| 76 | RFAN: Relation-fused multi-head attention network for knowledge graph enhanced recommendation | 2023 | Duan et al. | Applied Intelligence |
| 77 | Simple and Efficient Knowledge Graph Attention Network for Recommendation | 2023 | Li et al. | 2023 International Conference on Cyber-Physical Social Intelligence (ICCSI 2023) |

| | | | | |
|---|---|---|---|---|
| 78 | SKGCR: self-supervision enhanced knowledge-aware graph collaborative recommendation | 2023 | Liu et al. | Applied Intelligence |
| 79 | Towards travel recommendation interpretability: Disentangling tourist decision-making process via knowledge graph | 2023 | Gao et al. | Information Processing & Management |
| 80 | Cognitive-based knowledge learning framework for recommendation | 2024 | Chen et al. | Knowledge-Based Systems |
| 81 | EPAN-SERec: Expertise preference-aware networks for software expert recommendations with knowledge graph | 2024 | Tang et al. | Expert Systems with Applications |
| 82 | Knowledge Graph Enhanced Contextualized Attention-Based Network for Responsible User-Specific Recommendation | 2024 | Elahi et al. | ACM Transactions on Intelligent Systems and Technology |
| 83 | Knowledge-aware fine-grained attention networks with refined knowledge graph embedding for personalized recommendation | 2024 | Wang et al. | Expert Systems with Applications |
| 84 | MKNBL: Joint multi-channel knowledge-aware network and broad learning for sparse knowledge graph-based recommendation | 2024 | Wang et al. | Neurocomputing |
| 85 | RAKCR: Reviews sentiment-aware based knowledge graph convolutional networks for Personalized Recommendation | 2024 | Cui et al. | Expert Systems with Applications |
| 86 | Ripple Knowledge Graph Convolutional Networks for Recommendation Systems | 2024 | Li et al. | Machine Intelligence Research |

### A.2 Datasets for KG-based CTR Prediction.

Table A2. Summary of datasets for KG-based CTR prediction.

| Dataset | URL | Description | Details | | References |
|---|---|---|---|---|---|
| MovieLens-1M | https://grouplens.org/datasets/movielens/ | MovieLens contains movie rating data collected from the MovieLens web site with different sizes, e.g., MovieLens-1M, MovieLens-20M and MovieLens-Latest. | Scenario | Movie | Tang et al. (2019); Huang et al. (2022); Chen et al. (2022a); Xiao et al. (2022); Li et al. (2023c); Guo et al. (2020); Wang et al. (2019d); He et al. (2023); Yu et al. (2020); Wang et al. (2022); Dong et al. (2022); Tu et al. (2021); Wang et al. (2020a); Zou et al. (2022b); Qu et al. (2019); Zou et al. (2022a); Wang et al. (2019b); Ma et al. (2021); Wang et al. (2020b); Sun and Shagar (2020); Tai et al. (2020); Hui et al. (2022); Wang et al. (2018a); Wang et al. (2021); Song et al. (2023); Hu et al. (2023); Tu et al. (2023); Ma et al. (2023); Wang et al. (2023b); Peng et al. (2023); Wang et al. (2023a); Ong et al. (2023); Hai and Hongyan (2023); Wang et al. (2024a); Wang et al. (2024b); Wang et al. (2023c); Li et al. (2024) |
| | | | # Users | 6,036 | |
| | | | # Items | 2,445 | |
| | | | # Interactions | 753,772 | |
| MovieLens-20M | | | Scenario | Movie | Lyu et al. (2022); Chen et al. (2022b); Xiao et al. (2022); Wang et al. (2020c); Dai et al. (2022); Elahi and Halim (2022); Tang et al. (2022); Wang et al. (2019c); Li et al. (2019); Fan et al. (2022); Wang et al. (2019a); Qu et al. (2019); Ma et al. (2022a); Duan et al. (2023b); Qian et al. (2022); Shu and Huang (2021); Yang et al. (2023a); Shu and Huang (2023); Duan et al. (2023a); Li et al. (2023b); Elahi et al. (2024); Cui et al. (2024); Liu et al. (2023) |
| | | | # Users | 138,159 | |
| | | | # Items | 16,954 | |
| | | | # Interactions | 13,501,622 | |
| MovieLens-Latest | | | Scenario | Movie | Wu et al. (2022) |
| | | | # Users | 610 | |
| | | | # Items | 9,742 | |
| | | | # Interactions | 100,836 | |
| Book-Crossing | http://www2.informatik.uni-freiburg.de/~cziegler/BX/ | Book-Crossing consists of user ratings and book information collected from the online platform Book-Crossing community. | Scenario | Book | Tang et al. (2019); Lyu et al. (2022); Huang et al. (2022); Chen et al. (2022b); Wang et al. (2020c); Dai et al. (2022); Guo et al. (2020); Elahi and Halim (2022); Yu et al. (2020); Dong et al. (2022); Wang et al. |
| | | | # Users | 17,860 | |
| | | | # Items | 14,967 | |

| | | | | | |
|---|---|---|---|---|---|
| | | | # Interactions | 139,746 | (2020a); Zhang et al. (2021); Tang et al. (2022); Xie et al. (2022); Wang et al. (2019c); Hou et al. (2021); Li et al. (2019); Ma et al. (2022b); Wang et al. (2019a); Qu et al. (2019); Zhang and Yang (2020); Zou et al. (2022a); Wang et al. (2019b); Ma et al. (2021); Wang et al. (2020b); Sun and Shagar (2020); Hui et al. (2022); Duan et al. (2023b); Wang et al. (2018a); Wang et al. (2021); Qian et al. (2022); Shu and Huang (2021); Khan et al. (2022); Song et al. (2023); Hu et al. (2023); Wang et al. (2023b); Yang et al. (2023a); Wang et al. (2023a); Shu and Huang (2023); Ong et al. (2023); Duan et al. (2023a); Hai and Hongyan (2023); Li et al. (2023b); Elahi et al. (2024); Wang et al. (2024a); Wang et al. (2024b); Liu et al. (2023); Li et al. (2024) |
| Last.FM | https://grouplens.org/datasets/hetrec- 2011/ | Last.FM contains social networking, tagging and music artist listening information from the online music platform Last.FM. | Scenario | Music | Lyu et al. (2022); Huang et al. (2022); Chen et al. (2022b); Wang et al. (2020c); Dai et al. (2022); Chen et al. (2022a); Li et al. (2023c); Guo et al. (2020); He et al. (2023); Elahi and Halim (2022); Yu et al. (2020); Wang et al. (2022); Dong et al. (2022); Khan et al. (2023); Tu et al. (2021); Wang et al. (2020a); Zhang et al. (2021); Tang et al. (2022); Wang et al. (2019c); Hou et al. (2021); Li et al. (2019); Ma et al. (2022b); Fan et al. (2022); Sun and Li (2022); Zou et al. (2022b); Wang et al. (2019a); Ma et al. (2022a); Zhang and Yang (2020); Zou et al. (2022a); Wang et al. (2019b); Ma et al. (2021); Sun and Shagar (2020); Tai et al. (2020); Duan et al. (2023b); Wang et al. (2021); Qian et al. (2022); Shu and Huang (2021); Khan et al. (2022); Yang et al. (2021); Song et al. (2023); Tu et al. (2023); Wang et al. (2023b); Yang et al. (2023b); Yang et al. (2023a); Wang et al. (2023a); Shu and Huang (2023); Ong et al. (2023); Duan et al. (2023a); Li et al. (2023b); Elahi et al. (2024); Wang et al. (2024a); Wang et al. (2024b); Liu et al. (2023); Li et al. (2024) |
| | | | # Users | 1,872 | |
| | | | # Items | 3,846 | |
| | | | # Interactions | 42,346 | |
| Yelp | https://www.yelp.com/dataset | Yelp records users' reviews for local businesses, user profiles (e.g., id, name and review count) and item profiles (e.g., id, address and business). | Scenario | Business | Feng et al. (2020a); Tu et al. (2021); Khan et al. (2022); Fan et al. (2022); Feng et al. (2020b); Tu et al. (2023); Wang et al. (2023b); Bai et al. (2023); Cui et al. (2024) |
| | | | # Users | $4.5\times10^4$ | |
| | | | # Items | $4.5\times10^4$ | |
| | | | # Interactions | $1.0\times10^6$ | |

| | | | | | |
|---|---|---|---|---|---|
| Dianping-Food | https://github.com/hwwang55/KGNN-LS/tree/master/data/restaurant | Dianping-Food provides interaction behaviors (e.g., clicking and purchasing) between users and restaurants from Dianping.com. | Scenario | Food | Chen et al. (2022b); Wang et al. (2020c); Dai et al. (2022); He et al. (2023); Hou et al. (2021); Qian et al. (2022); Wang et al. (2019a); Shu and Huang (2023); Bai et al. (2023) |
| | | | # Users | 2,298,698 | |
| | | | # Items | 1,362 | |
| | | | # Interactions | 23,416,418 | |
| Bing-News | https://github.com/hwwang55/DKN/tree/master/data | Bing-News records user id, news URL, news title, timestamp, and implicit feedback from the server logs of Bing News. | Scenario | News | Wang et al. (2018b); Wang et al. (2019b); Sun and Shagar (2020); Hui et al. (2022); Wang et al. (2018a); Lee et al. (2020); Chen et al. (2024) |
| | | | # Users | 141,487 | |
| | | | # Items | 535,145 | |
| | | | # Interactions | 1,025,192 | |
| IMDb | https://www.dropbox.com/s/0oea49j7j30y671/data.json?dl=0 | IMDb provides user's ratings, reviews and sentiments for different aspects of a movie, such as actors, plot and visual effects. | Scenario | Movie | Liu and Miyazaki (2022) |
| | | | # Users | 2,088 | |
| | | | # Items | 4,668 | |
| | | | # Interactions | 126,874 | |
| Amazon-Movies&TV-A | http://jmcauley.ucsd.edu/data/amazon/ | Amazon is a commonly used dataset, which collects multiple categories of products, including movies&TV, book, music, clothing and electronics. Each category includes user reviews and ratings, product descriptions, and other relevant information. | Scenario | Movie | Liu and Miyazaki (2022) |
| | | | # Users | 244,782 | |
| | | | # Items | 59,652 | |
| | | | # Interactions | 1,588,922 | |
| Amazon-Movies&TV-B | | | Scenario | Movie | Zhu et al. (2020) |
| | | | # Users | 2,088,620 | |
| | | | # Items | 498,117 | |
| | | | # Interactions | 6,304,580 | |
| Amazon-Book-A | | | Scenario | Book | Khan et al. (2023) |
| | | | # Users | 55,255 | |
| | | | # Items | 20,235 | |
| | | | # Interactions | 232,562 | |
| Amazon-Book-B | | | Scenario | Book | Fan et al. (2022) |
| | | | # Users | 17,860 | |
| | | | # Items | 14,910 | |
| | | | # Interactions | 69,873 | |

| | | | | | |
|---|---|---|---|---|---|
| Amazon-Book-C | | | Scenario | Book | Tai et al. (2020) |
| | | | # Users | 6,969 | |
| | | | # Items | 9,854 | |
| | | | # Interactions | 552,706 | |
| Amazon-Book-D | | | Scenario | Book | Zhu et al. (2020) |
| | | | # Users | 8,026,324 | |
| | | | # Items | 3,979,373 | |
| | | | # Interactions | 31,405,196 | |
| Amazon-Book-E | | | Scenario | Book | Qu et al. (2019) |
| | | | # Users | 78,809 | |
| | | | # Items | 32,389 | |
| | | | # Interactions | 1,181,684 | |
| Amazon-Book-F | | | Scenario | Book | Cui et al. (2024) |
| | | | # Users | 839,644 | |
| | | | # Items | 24,915 | |
| | | | # Interactions | 1,902,124 | |
| Amazon-Book-G | | | Scenario | Book | Wang et al. (2023c) |
| | | | # Users | 42,715 | |
| | | | # Items | 29,231 | |
| | | | # Interactions | 323,689 | |
| Amazon-Music | | | Scenario | Music | Zhu et al. (2020) |
| | | | # Users | 478,235 | |
| | | | # Items | 599,080 | |
| | | | # Interactions | 900,712 | |
| Amazon-Clothing | | | Scenario | Clothing | Zhu et al. (2020) |
| | | | # Users | 3,117,268 | |
| | | | # Items | 3,040,742 | |
| | | | # Interactions | 6,027,597 | |

| | | | | | |
|---|---|---|---|---|---|
| Amazon-Electronics | | | Scenario | Electronics | Wu et al. (2022) |
| | | | # Users | 941,861 | |
| | | | # Items | 9,490 | |
| | | | # Interactions | 1,048,575 | |
| Taobao-A | https://tianchi.aliyun.com/dataset/dataDetail?dataId=649 | Taobao-A contains user behaviors from China's e-commerce platform Taobao. | Scenario | E-commerce | Wu et al. (2022) |
| | | | # Users | 10,202 | |
| | | | # Items | 412,369 | |
| | | | # Interactions | 848,449 | |
| Tmall | https://tianchi.aliyun.com/dataset/dataDetail?dataId=47 | Tmall spans May to November 2015 and captures user behaviors on the Tmall online shopping platform, provided by Alibaba Group. | Scenario | E-commerce | Wu et al. (2022) |
| | | | # Users | 6,876 | |
| | | | # Items | 237,700 | |
| | | | # Interactions | 690,267 | |
| Food.com | https://www.kaggle.com/shuyangli94/food-com-recipes-and-user-interactions | Food.com contains over 180,000 recipes and 700,000 recipe reviews scraped from the Food.com website, spanning 18 years of user interactions from 2000 to 2018. | Scenario | Food | Li et al. (2023a) |
| | | | # Users | 24,957 | |
| | | | # Items | 102,717 | |
| | | | # Interactions | 673,321 | |
| MyFitnessPal | https://www.kaggle.com/zvikinozadze/myfitnesspal-dataset | MyFitnessPal includes 587K+ food log records for 9K+ users, recorded between September 2014 and April 2015. | Scenario | Food | Li et al. (2023a) |
| | | | # Users | 9,897 | |
| | | | # Items | 23,341 | |
| | | | # Interactions | 156,322 | |
| MIND | https://msnews.github.io/ | MIND is a large-scale dataset from October 2019 to November 2019 for news recommendation, collected from anonymized behavior logs of the Microsoft News platform. | Scenario | News | Chen et al. (2024) |
| | | | # Users | 156,965 | |
| | | | # Items | 51,282 | |
| | | | # Interactions | 1,816,594 | |
| Adressa-News | http://reclab.idi.ntnu.no/dataset/ | Adressa-News is a news reading dataset gathered from Adresseavisen's news portal, including session time, news title, user profile, etc. | Scenario | News | Lee et al. (2020) |
| | | | # Users | 561,733 | |
| | | | # Items | 11,207 | |
| | | | # Interactions | – | |

| | | | | | |
|---|---|---|---|---|---|
| MOOCCube | http://moocdata.cn/data/MOOCCube | MOOCCube is a publicly available online learning dataset that combines courses, concepts, student behaviors, relationships, and external resources, sourced from MOOC websites. | Scenario | Education | Zhang et al. (2023) |
| | | | # Users | 7,156 | |
| | | | # Items | 219 | |
| | | | # Interactions | 32,091 | |
| Taobao-B | Private | Taobao-B is sourced from the online shopping platform Taobao, ranging from 2019/08/22 to 2019/08/29. | Scenario | E-commerce | Feng et al. (2020a) |
| | | | # Users | $2.2\times10^{8}$ | |
| | | | # Items | $1.1\times10^{8}$ | |
| | | | # Interactions | $7.2\times10^{9}$ | |
| E-commerce | Private | E-commerce is collected from a popular E-commerce Mobile App. It captures impression and click logs spanning 8 consecutive days, with positive instances identified as clicked items and negative instances as non-clicked ones. | Scenario | E-commerce | Feng et al. (2020b) |
| | | | # Users | 0.2 Billion | |
| | | | # Items | 0.1 Billion | |
| | | | # Interactions | 7.2 Billion | |
| Official-Account | Private | Official-Account sources from official account retrieval in WeChat. | Scenario | Business | Yao et al. (2023) |
| | | | # Users | 433,221 | |
| | | | # Items | 900,755 | |
| | | | # Interactions | 4,815,922 | |
| Mini-Program | Private | Mini-Program is selected from mini program retrieval in WeChat. | Scenario | Business | Yao et al. (2023) |
| | | | # Users | 128,192 | |
| | | | # Items | 280,631 | |
| | | | # Interactions | 4,336,346 | |
| Alipay-A | Private | Alipay contains timestamp, user id and item id collected from the Alipay APP with two online local service scenarios. Alipay-A includes services in the fields such as medical insurance, fitness and exercise, health, etc. Alipay-B includes local services, such as hiking and eating activities for daily life based on the | Scenario | Business | Li et al. (2023c) |
| | | | # Users | 212,089 | |
| | | | # Items | 311 | |
| | | | # Interactions | 2,121,437 | |
| Alipay-B | Private | | Scenario | Business | Li et al. (2023c) |
| | | | # Users | 2,963,532 | |
| | | | # Items | 25,386 | |

| | | | | | |
|---|---|---|---|---|---|
| | | user's location, interests, and personality traits. | # Interactions | 68,018,027 | |
| Fund | Private | Fund is collected from a real-world FinTech platform to recommend funds for users, containing online exposure samples over a period of two weeks. The click samples are considered positive samples, while the non-click samples serve as negative samples. | Scenario | Finance | Tu et al. (2023) |
| | | | # Users | 266,902 | |
| | | | # Items | 20,019 | |
| | | | # Interactions | 861,159 | |
| MaFengWo | Private | MaFengWo is collected from a tourism platform MaFengWo in China, spanning from January 2014 to April 2019. | Scenario | Tourism | Gao et al. (2023) |
| | | | # Users | 57,843 | |
| | | | # Items | 18,710 | |
| | | | # Interactions | 6,505,559 | |
| TripAdvisor | Private | TripAdvisor is an international tourism dataset from the TripAdvisor platform, which exhibits greater sparsity, limitations, and non-identical distribution compared with the MaFengWo dataset. | Scenario | Tourism | Gao et al. (2023) |
| | | | # Users | 10,952 | |
| | | | # Items | 273 | |
| | | | # Interactions | 40,732 | |
| Weibo | Private | Weibo provides a collection of user-generated content and social connections from September 2012 to October 2012 from Sina Weibo.com. | Scenario | News | Yang et al. (2020) |
| | | | # Users | 1,776,950 | |
| | | | # Items | – | |
| | | | # Interactions | – | |
| StackOverflow | Private | StackOverflow is collected from StackOverflow , a software knowledge community, which includes experts in the fields such as software tool and software framework, questions and corresponding reputation scores. | Scenario | Knowledge Q&A | Tang et al. (2024) |
| | | | # Users | 54 | |
| | | | # Items | 10,152 | |
| | | | # Interactions | – | |
| Nudge | Private | Nudge provides information about suggestion actions to users in a digital health platform for diabetes management. | Scenario | Health | Tangruamsub et al. (2022) |
| | | | # Users | – | |
| | | | # Items | – | |

| | | | | | |
|---|---|---|---|---|---|
| | | | # Interactions | – | |
| Agricultural-Products | Private | Agricultural-Products records user ratings of agricultural products and product details such as price, title and review number. | Scenario | Agriculture | Xie et al. (2022) |
| | | | # Users | 194,202 | |
| | | | # Items | 1,951 | |
| | | | # Interactions | 1,098,723 | |
| ML | Private | ML contains machine learning datasets, methods, and their properties, as well as other related entities collected from open academic platforms (e.g., Paperswithcode and Github), covering 19 areas (e.g., computer vision, natural language processing, and graphs). | Scenario | Machine Learning | Cao et al. (2021) |
| | | | # Users | 2,092 | |
| | | | # Items | 6,239 | |
| | | | # Interactions | 13,732 | |
| Amazon-Cross | Private | Amazon-Cross includes three domains: movies and TV, kindle store and digital music. | Scenario | Cross Domains | Wang et al. (2024a) |
| | | | # Users | 8,943 | |
| | | | # Items | 68,458 | |
| | | | # Interactions | 107,606 | |

## A3. KG

Table A3. KG and descriptions.

| KG | URL | Descriptions |
|---|---|---|
| Microsoft Satori | https://searchengineland.com/library/bing/bing-satori | Microsoft Satori is a KG developed by Microsoft to enhance Bing search capabilities. Satori spans a wide range of domains and provides a comprehensive understanding of the world's knowledge. It encompasses about 2 billion primary entities and 55 billion facts from public online resources, digital books, encyclopedias, and other relevant sources that might be of interest to Internet users (Noy et al., 2019; Uyar and Aliyu, 2015). |
| Freebase | http://www.freebase.com/ | Freebase is a scalable collaborative knowledge base serving as a centralized repository of information, allowing users to contribute and access data on a wide range of subjects. It was developed and run publicly since March 2007 by Metaweb, a company acquired by Google in 2010. In 2014, Google announced that Freebase would be retired and migrated to Wikidata, another collaborative knowledge base. |
| Wikidata | https://www.wikidata.org/wiki/Wikidata:Main_Page | Wikidata is a free and open knowledge base launched in October 2012, with more than 100 million data items. It allows anyone to edit, even without an account. It serves as a central repository of structured data that can be used by various Wikimedia projects. |
| FoodKG | https://foodkg.github.io/ | FoodKG encompasses recipes, food ontologies, and nutritional information on a vast scale. It contains approximately one million recipes, 7,700 nutrient records, and 7,300 distinct food types, totaling over 67 million triples. |
| Tourism-KG | https://gaojialianglreis.github.io/KG_demo_2023/ | Tourism-KG provides abundant information, facilitating the tourist decision-making process. Its physical dimension encompasses various types of leisure resources; while the psychological dimension captures the intangible cognitive features of attractions. |
| Meituan Brain | Private | Meituan Brain is an internal KG developed by Meituan-Dianping Group, designed for dining and entertainment. It includes billions of entities and tens of billions of triplets, embodying knowledge associations between users, merchants, products and scenarios. |
| Antfin Digital Local Service KG | Private | Antfin Digital Local Service KG is a large-scale KG to provide accurate representations of digital local services. It encompasses over 30 different types of entities and 70 types of relationships, with a collection of 370 million entities and 29.4 billion relations. |
| SWKG | Private | SWKG provides a comprehensive and interconnected view on software knowledge entities and their relations. |