

RESEARCH ARTICLE

# KGRec: A knowledge graph attention-based model for recommender system

Trinh Duong Hoan , Bui Thanh Hung \*

Data Science Laboratory, Faculty of Information Technology, Industrial University of Ho Chi Minh city, Ho Chi Minh city, Vietnam

\* [buihanhhung@iuh.edu.vn](mailto:buihanhhung@iuh.edu.vn)



## Abstract

Recommender systems have recently gained significant traction as powerful tools for personalized content delivery. While accuracy remains a key focus, users now expect more than precise suggestions. To meet diverse preferences, these systems must also ensure recommendation diversity. They are widely applied in domains such as e-commerce, social media, and online entertainment platforms. Conventional approaches like collaborative filtering mainly emphasize user–item interactions, often overlooking contextual and attribute information, which results in limited performance, especially under sparse data conditions. To address this, we present the KGRec- Knowledge Graph Attention Network Recommendation model, the novel KGRec model integrates knowledge graphs to capture higher-order relationships among users, items, and their associated attributes. KGRec applies multi-layer embedding propagation combined with an attention mechanism to model indirect user–item connections through intermediate attributes, enabling the model to assess the significance of each relation and thus improve recommendation quality. Empirical evaluations on four benchmark datasets—Yelp2018, Last-FM, Amazon-Book and MovieLen-1M—demonstrate the effectiveness of KGRec. The proposed model consistently outperforms all baseline methods across every evaluation metric. These improvements highlight the model's robustness and its effectiveness in capturing richer semantic representations for recommendation.

## OPEN ACCESS

**Citation:** Hoan TD, Hung BT (2026) KGRec: A knowledge graph attention-based model for recommender system. PLoS One 21(3): e0344585. <https://doi.org/10.1371/journal.pone.0344585>

**Editor:** Chi Ho Yeung, Education University of Hong Kong, CHINA

**Received:** August 23, 2025

**Accepted:** February 23, 2026

**Published:** March 11, 2026

**Copyright:** © 2026 Hoan, Hung. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Data availability statement:** The datasets are publicly available as follows: Yelp2018 dataset [32]: <https://www.yelp.com/dataset/> Last-FM dataset [33]: <http://millionsongdataset.com/lastfm/> Amazon-Book dataset [34]: <https://nijianmo.github.io/amazon/> MovieLen-1M dataset [35]: <https://grouplens.org/datasets/movielens/1m/>.

## Introduction

In the digital era, recommender systems have become an indispensable component of online platforms, supporting a wide range of applications in e-commerce, social networks, content streaming services, and news portals. By analyzing vast amounts of user behavior data, these systems enable platforms to deliver highly personalized content, helping individuals save time, discover new products, and engage more effectively with the services they use [1–4]. Such personalization not only enhances user experience but also drives business growth by improving customer satisfaction,

**Funding:** The author(s) received no specific funding for this work.

**Competing interests:** The authors have declared that no competing interests exist.

retention, and sales conversions [5–7]. Despite their success, however, traditional recommendation models—most notably Collaborative Filtering (CF) [8]—still face notable challenges. While CF and related supervised learning approaches have proven effective in capturing similarities between users or items, they often perform poorly when data is sparse, when new users or items are introduced, or when contextual factors play a critical role in shaping preferences. More importantly, these models generally treat each user–item interaction as an isolated event, overlooking the broader web of relationships among entities such as directors, genres, brands, product categories, or other rich attributes that influence user decisions. This independence assumption limits the system’s ability to incorporate domain knowledge and complex semantic structures, resulting in a shallow understanding of user interests. Consequently, the recommendations generated may fail to capture deeper collaborative signals hidden in the data, leading to suboptimal outcomes in terms of both accuracy and diversity. Furthermore, these limitations hinder scalability, making it increasingly difficult for platforms to maintain effective recommendations as the volume and variety of data grow in today’s fast-changing digital ecosystems.

To address these limitations, an emerging and increasingly popular approach is to incorporate Knowledge Graphs (KGs) into the recommendation process [9–11]. A knowledge graph is a structured representation of information, where entities are modeled as nodes and their relationships are represented as edges, forming a rich network of interconnected knowledge. For instance, in the domain of movies, a KG might capture facts such as “Movie A is directed by Director B”, “Movie A belongs to Genre C”, or “Actor D stars in Movie A.” Similarly, in the context of books or e-commerce, KGs can represent associations such as “Book E is written by Author F” or “Product G is manufactured by Brand H.” By encoding this information in a graph structure, KGs provide a natural way to integrate heterogeneous data sources into recommendation models.

Integrating knowledge graphs with traditional user–item interaction data opens new opportunities for overcoming the sparsity and independence issues that plague conventional recommendation methods. Instead of relying solely on observed user–item interactions, the system can exploit high-order relationships among entities, discovering hidden connections that may not be directly visible in the interaction matrix. For example, if a user has shown interest in movies directed by a certain filmmaker, the system can recommend other films associated with similar directors, genres, or production companies through multi-hop reasoning over the KG. This allows the model to capture deeper semantic associations, enrich the contextual information available for decision-making, and provide users with recommendations that are not only more accurate but also more diverse and explainable.

In essence, knowledge graphs serve as a powerful auxiliary source of structured knowledge that bridges the gap between raw interaction data and the rich semantic world underlying user preferences. Their ability to encode domain knowledge, uncover higher-order connectivity, and enhance interpretability makes them a promising foundation for next-generation recommender systems that aim to go beyond accuracy and offer more meaningful, scalable, and personalized recommendations.

In this study, we introduce the Knowledge Graph Attention Network Recommendation model (KGRec), a novel deep learning model designed to fully exploit high-order relationships in knowledge graphs through the perspective of Graph Neural Networks (GNNs) [12,13]. KGRec operates by propagating embeddings from neighboring nodes across multiple layers, while employing an attention mechanism to automatically determine the importance of each relation during learning. This allows the model to prioritize more informative paths, thereby enhancing recommendation quality for each prediction. Unlike path-based methods [14,15], which require manual specification of meta-paths and face challenges in global optimization, KGRec models high-order relations in an end-to-end fashion, offering both computational efficiency and seamless integration into modern machine learning pipelines. Compared to regularization-based methods, KGRec directly incorporates knowledge information into the prediction process, enabling the model to learn better parameters for recommendation objectives.

We conducted evaluations of KGRec on four real-world datasets from different domains: Yelp2018 (local businesses), Last-FM (music listening), Amazon-Book (book purchases) and MovieLens-1M (movies rating). Experimental results demonstrate that KGRec outperforms state-of-the-art methods not only in terms of accuracy (Precision, Recall, F1, and NDCG) but also in robustness under sparse data conditions. Overall, KGRec represents a significant step forward in integrating semantic knowledge into recommender systems, while also opening new directions for building recommendation models that are context-aware and adaptive to data-scarce environments—an area where traditional methods remain limited.

## Related work

In recent years, the field of recommender systems has made significant progress, with numerous methods proposed to enhance personalization, improve prediction accuracy, and increase transparency in recommendations for users. In addition to classical models such as collaborative filtering, many studies have expanded their approaches by incorporating auxiliary information and leveraging structured knowledge to further improve performance.

Shashidhar Reddy Javaji and Krutika Sarode proposed a hybrid recommender system that integrates Graph Neural Networks (GNNs) and the BERT model to enhance the effectiveness of anime recommendations for users [16]. The model adopts the GraphSAGE architecture [17] to represent the relationships between users and anime series as a heterogeneous graph, while also extracting semantic features from anime descriptions through sentence embeddings generated by SentenceTransformer (based on BERT). By combining structural information from GNNs with semantic representations from text, the model is able to capture user–anime interactions as well as distinguish between anime that share similar genres but differ in content. The system was evaluated on the Anime Recommendation Database 2020, which contains more than 78 million interactions between 320,000 users and 16,000 anime titles. Experimental results show that the model can predict potential links (i.e., unknown ratings) and recommend top-ranked anime aligned with user preferences. However, the accuracy on the test set remains relatively low (around 37%) and the error rate is high (RMSE ~0.66), indicating that the model has yet to fully exploit its representational potential. Moreover, processing sentence embeddings and training GNNs on large-scale graphs [3] require substantial computational resources, making it challenging to scale the model to other domains or larger datasets. Nevertheless, the integration of textual semantics and graph structures remains a highly promising direction, particularly for recommender systems that must handle unstructured and diverse data such as anime or books.

Chuang Ma et al. proposed a novel recommendation model named FedGR, which combines Graph Neural Networks (GNNs) with Federated Learning to simultaneously address three key challenges: improving recommendation accuracy, leveraging social information, and preserving user privacy [18]. This is among the first models to jointly integrate GNNs, social network information, and privacy-preserving mechanisms in a distributed environment. FedGR employs a split-model architecture, where user representations are trained locally on edge devices, while item representations are trained on a central server. In addition, the model adopts the Graph Attention Network (GAT) [19] to learn user embeddings from

two graphs: the user–item interaction graph and the social relation graph. To enhance privacy protection, FedGR introduces two mechanisms: Noise Injection, which adds dummy interactions to obfuscate true user histories, and Encryption–Decryption, a dual encryption strategy combining symmetric and asymmetric cryptography to secure both input data and model transmission. Experiments on two widely used datasets, Ciao and Epinions, demonstrate that FedGR significantly improves accuracy (reducing MAE and RMSE) compared to previous federated approaches such as FedMF [20] and FeSoG [21], while maintaining robust performance even with the added security mechanisms. However, the model still has limitations, such as not incorporating temporal factors in user behavior and relying on the FedAVG [22] parameter aggregation algorithm, which may cause information loss when averaging parameters from heterogeneous devices. Moreover, although the proposed privacy mechanisms are effective, they also increase computational complexity and communication overhead between the server and edge devices.

Wenyi Liu et al. proposed a novel recommendation algorithm based on deep Graph Neural Networks (GNNs), aiming to address the problem of over-smoothing—a phenomenon where information becomes indistinguishable across layers as network depth increases, leading to reduced recommendation accuracy [23]. The model integrates collaborative filtering with GNNs and incorporates initial residual connections and identity mapping during the propagation process, thereby preserving user–item interaction signals and enhancing model performance. While the approach achieves outstanding accuracy on benchmark datasets such as Amazon-Book, Yelp2018, and Gowalla, it still demands substantial computational resources due to its multi-layer, complex architecture and has yet to fully address the challenge of providing intuitive and interpretable recommendations for general users.

Begum Ozbay et al. proposed a novel session-based recommendation model leveraging Graph Neural Networks (GNNs), with the aim of improving the accuracy of predicting user behavior through an adaptive weighting mechanism [24]. The model builds upon the SR-GNN architecture [25] and extends it by introducing a component that dynamically adjusts the importance of each item within a session, based on its relative position and its similarity to the last item in the session. In addition, the study incorporates side information in two forms: information from the last item and the averaged information of the entire session, which enhances global session representation and improves the understanding of users' short-term behavior. The model was evaluated on the Dressipi dataset with various training ratios (1/128, 1/64, 1/32, 1/4, and the full dataset), achieving significant improvements under sparse-data (cold-start) scenarios. In particular, when using adaptive weights at low ratios such as 1/64 and 1/128, the model improved recommendation accuracy (Recall@20) by more than 13% compared to the baseline SR-GNN. However, on the full dataset, a slight decline in performance was observed, suggesting that the adaptive weighting mechanism may not be as effective in dense-data environments. Moreover, the model introduces additional computational complexity, as it requires calculating weights based on session order and item similarity, which poses challenges for real-time deployment in large-scale, high-frequency user interaction settings.

Overall, previous studies have made significant progress in enhancing the effectiveness of recommender systems, particularly in exploiting the complex relationships between users and items. However, most approaches remain largely academic and are not yet well-suited for real-world deployment due to limitations in flexibility, computational cost, or scalability. Motivated by this gap, the present study proposes a novel method that integrates graph neural networks with an attention mechanism, aiming to leverage deeper semantic knowledge within the data while ensuring scalability and stable performance in practical applications.

## Materials and methods

### The proposed KGRec model

Our proposed KGRec model is composed of four major components: the Collaborative Knowledge Graph, the Embedding Layer, the Attentive Embedding Propagation Layer, and the Prediction Layer. The Collaborative Knowledge Graph is first constructed from the input data, where both users and items are represented as nodes, and their interactions, together

with external knowledge, form the edges of the graph. Subsequently, the Embedding Layer parameterizes each node by mapping it into a low-dimensional vector representation, which serves as the foundation for learning meaningful semantic features. Building on this, the Attentive Embedding Propagation Layer enhances the expressiveness of node representations by iteratively aggregating information not only from direct neighbors but also from higher-order neighbors, while the attention mechanism assigns different weights to neighbors according to their relative importance. Finally, the Prediction Layer computes the compatibility score between a user and an item based on their learned embeddings, and this score is used to generate the final recommendation. An overview of the overall architecture of our proposed KGRec model is illustrated in Fig 1.

### Collaborative knowledge graph

In this study, we use the Collaborative Knowledge Graph (CKG) and emphasize the importance of high-order connections between nodes as well as composite relations [9]. The structure of a CKG is defined as follows:

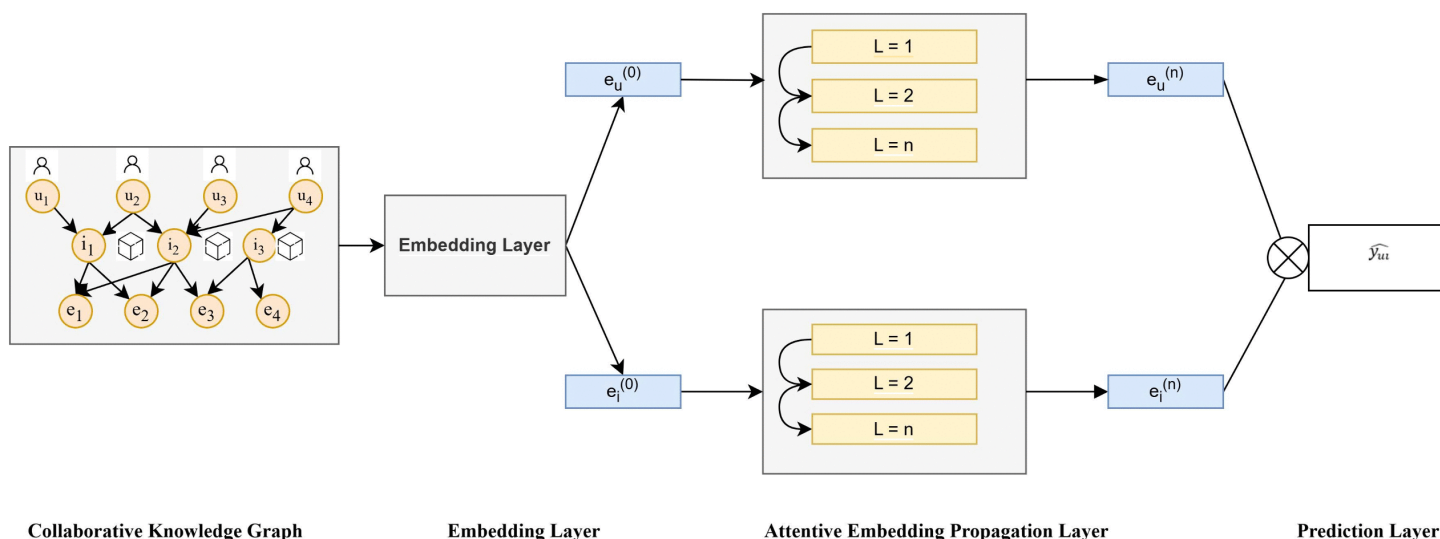
**User–item graph.** In the context of recommender systems, historical interaction data between users and items (e.g., purchases or clicks) is represented as a bipartite graph  $G_1$ . This graph is formally defined as:

$$G_1 = \{(u, y_{ui}, i) \mid u \in U, i \in I\} \tag{1}$$

Where  $U$  and  $I$  denote the sets of users and items, respectively. The variable  $y_{ui}$  indicates the interaction between a user and an item, where  $y_{ui} = 1$  if user  $u$  has interacted with item  $i$ , and  $y_{ui} = 0$  otherwise.

**Knowledge graph.** Beyond user–item interactions, additional information about items (such as item attributes and external knowledge) is also incorporated. This supplementary data consists of real-world entities and the relationships among them to describe an item (for example, a movie can be characterized by its director, actors, and genres). Such information is typically organized as triples (head, relation, tail), or  $(h, r, t)$ . In this study, we define the knowledge graph  $G_2$  as a directed graph constructed from these triples  $(h, r, t)$ , specifically:

$$G_2 = \{(h, r, t) \mid h, t \in E, r \in R\} \tag{2}$$



**Fig 1. The overall architecture of our proposed KGRec model.**

<https://doi.org/10.1371/journal.pone.0344585.g001>

Here,  $E$  denotes the set of entities and  $R$  represents the set of relations. For instance, the triple  $(Hugh\ Jackman, ActorOf, Logan)$  indicates that *Hugh Jackman* is an actor in the movie *Logan*. In addition, we define a set of item–entity links:  $A = \{(i, e) \mid i \in I, e \in E\}$  which specifies that an item  $i$  is associated with an entity  $e$  in the knowledge graph. Fig 2 describes an example of a knowledge graph.

**Collaborative knowledge graph (CKG).** The CKG is a unified structure that integrates user behaviors and item knowledge into a single relational graph. Each user behavior  $(u, y_{ui}, i)$  is represented as a triple  $(u, Interact, i)$ , where  $y_{ui} = 1$  is treated as an additional *Interact* relation between user  $u$  and item  $i$ . Based on the set of item–entity links  $A$ , the user–item graph can be seamlessly integrated with the knowledge graph to form a unified structure:

$$G = \{(h, r, t) \mid h, t \in E', r \in R'\} \tag{3}$$

Where  $E' = E \cup U$  and  $R' = R \cup \{Interact\}$ .

### Embedding layer

The embedding layer is responsible for parameterizing each node in the CKG into a representation vector such that the structure of the graph—including users, items, entities, and the relationships among them—is preserved. In this study, we employ the TransR method [26], a popular knowledge graph embedding approach, to embed nodes in the CKG into embedding vectors through the translation principle. Each entity  $h$  (head entity) and  $t$  (tail entity) in the CKG is represented by a vector in the entity space:  $e_h, e_t \in \mathbb{R}^d$  where  $d$  is the dimension of the entity space. Each relation  $r$  (relation) is represented by a translation vector  $e_r \in \mathbb{R}^k$  where  $k$  is the dimension of the relation space (which may differ from  $d$ ). Additionally, each relation  $r$  is associated with a projection matrix  $W_r \in \mathbb{R}^{k \times d}$  that maps entities from the entity space to the relation space. TransR assumes that for each triple  $(h, r, t)$  existing in the CKG, in the relation space  $r$ , the projected vector of the head entity ( $e'_h$ ) added to the relation vector ( $e_r$ ) approximates the projected vector of the tail entity ( $e'_t$ ). Formally, this can be expressed as follows:

$$e'_h + e_r \approx e'_t \tag{4}$$

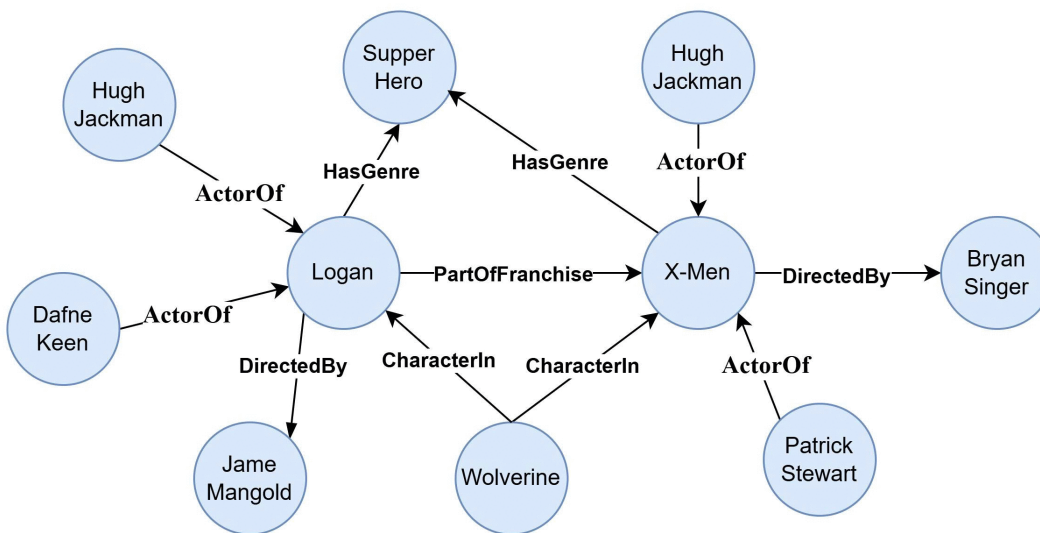


Fig 2. An example of a knowledge graph.

<https://doi.org/10.1371/journal.pone.0344585.g002>

Where:

- $e_h^r = W_r e_h$ : the representation of  $h$  in the relation space.
- $e_t^r = W_r e_t$ : the representation of  $t$  in the relation space.

For each triple  $(h, r, t)$ , we compute its plausibility score using the squared Euclidean distance [27]. The plausibility score is defined as:

$$g(h, r, t) = \|e_h^r + e_r - e_t^r\|_2^2 \quad (5)$$

When the value of  $g(h, r, t)$  is lower, it means that  $e_h^r + e_r$  is closer to  $e_t^r$  in the relation space, indicating that the triple  $(h, r, t)$  is more likely to be valid. Conversely, a higher score suggests that the triple is less plausible.

The training of TransR employs a pairwise ranking loss function [28] to distinguish between valid and invalid triples. This facilitates modeling entities and relations at the triple level, serving as a regularizer and incorporating direct connections into the representations, thereby enhancing the model's representational capacity.

### Attentive embedding propagation layer

After obtaining the initial vector representations of entities, items, and users through the TransR embedding layer, we implement a multi-layer attentive embedding propagation mechanism to extend the representation scope of each node from its direct neighbors to higher-order neighbors. This mechanism enables the model to leverage not only the direct connections between users and items but also indirect relational paths through intermediate entities such as genres, directors, or brands. Specifically, for each entity  $h \in E$ , we define its direct neighborhood in the CKG as:

$$N_h = \{(h, r, t) \mid (h, r, t) \in G\} \quad (6)$$

The objective is to compute a new embedding vector for node  $h$  by aggregating information from its neighbors  $t \in N_h$ , while simultaneously adjusting the contribution of each neighbor through a knowledge-aware attention mechanism.

At each layer, we implement three main components:

**Information propagation.** The information from neighboring nodes is propagated to the target node through a weighted summation, specifically:

$$e_{N_h} = \sum_{(h,r,t) \in N_h} \pi(h, r, t) e_t \quad (7)$$

Where:

- $e_t$ : the representation of the tail node  $t$ .
- $\pi(h, r, t)$ : the attention coefficient reflecting the importance of the edge  $(h, r, t)$ .

**Knowledge-aware attention mechanism.** The attention coefficient  $\pi(h, r, t)$  is determined based on the interaction between the target node, the relation, and its neighboring node, and is computed as follows:

$$\pi(h, r, t) = (W_r e_t)^\top \tanh(W_r e_h) \quad (8)$$

Here, we choose the activation function to be  $\tanh(\cdot)$ , whose values range from  $[-1, 1]$ , enabling the model to effectively capture complex relationships. Subsequently, to normalize the attention weights, we employ the softmax function [29]:

$$\pi(h, r, t) = \frac{\exp(\pi(h, r, t))}{\sum_{(h,r,t') \in N_h} \exp(\pi(h, r, t'))} \quad (9)$$

This mechanism enables the model to focus on semantic relations with higher contributions while simultaneously reducing noise from less relevant relations.

The incorporation of the attention mechanism plays a crucial role in enabling adaptive and knowledge-aware information aggregation. In a collaborative knowledge graph, a node is typically connected to multiple neighbors through heterogeneous relations. Treating all neighbors equally during aggregation may introduce noise and dilute important semantic signals. By using Eq 9, KGRec dynamically assigns different importance weights to neighboring entities and relations. This allows the model to prioritize more informative and semantically relevant connections while suppressing less useful ones. Such selective aggregation is particularly beneficial in sparse and heterogeneous graphs, where irrelevant or weakly related neighbors can otherwise negatively affect representation learning.

**Information aggregation.** After obtaining the propagated information  $e_{N_h}$ , the embedding vector of  $h$  is combined with the aggregated embedding vectors from its neighboring nodes to generate a new representation of  $h$  at the current layer. This process aims to integrate both the intrinsic information of node  $h$  and the propagated information from its neighbors, thereby providing a more comprehensive characterization of node  $h$ . The general formula for this process is:

$$e_h^{(l)} = f(e_h^{(l-1)}, e_{N_h}) \tag{10}$$

Where  $f(e_h^{(l-1)}, e_{N_h})$  denotes the aggregator function, which determines how these two vectors are combined. In this study, we conduct experiments with three different aggregators:

- **Graph Convolutional Network (GCN) [30]:** This is a simple yet effective method directly inherited from the traditional GCN architecture. In this approach, the new embedding vector is computed by performing a linear summation of the central node's vector and the aggregated vector from its neighbors, followed by a transformation matrix and a nonlinear activation function. The formula is given as:

$$e_h^{(l)} = \text{LeakyReLU}(W(e_h^{(l-1)} + e_{N_h})) \tag{11}$$

Where  $W \in \mathbb{R}^{d \times d}$  is the weight matrix, and LeakyReLU is a nonlinear activation function. This method assumes that the information from the central node and its neighbors contributes equally and is combined in a linear manner, which limits its ability to capture the nonlinear interactions between the central node and its neighbors.

- **GraphSAGE [17]:** GraphSAGE employs vector concatenation by combining the embedding vector of the central node with the aggregated vector from its neighbors before passing it through a weighted transformation layer. The formula is given as:

$$e_h^{(l)} = \text{LeakyReLU}(W[e_h^{(l-1)} || e_{N_h}]) \tag{12}$$

This method leverages concatenation to preserve distinct information from both the central node and its neighbors, thereby allowing the model to learn separate weights for each type of signal. However, its limitation lies in the higher computational cost, as the input vector dimension is doubled.

- **Bi-Interaction [13]:** This is the primary method adopted in KGRec. Unlike GCN, which only performs linear summation, this approach introduces an additional nonlinear component by applying element-wise multiplication to explicitly model the detailed interactions between the central node and its neighbors. The outputs from both branches are then combined after being transformed through two separate weight matrices. The formula is given as:

$$e_h^{(l)} = \text{LeakyReLU}(W_1(e_h^{(l-1)} + e_{N_h})) + \text{LeakyReLU}(W_2(e_h^{(l-1)} \odot e_{N_h})) \tag{13}$$

Where:

- $\odot$ : the element-wise multiplication (Hadamard product).
- $W_1, W_2 \in \mathbb{R}^{d \times d}$ : two separate learnable matrices for each branch.

This method provides a better modeling of the nonlinear dependencies between a node and its neighbors, thereby enhancing the representational capacity in heterogeneous graphs. However, it requires higher computational cost and longer training time due to the need for learning multiple parameters.

### Prediction layer

After completing the embedding propagation through  $L$  layers, each user  $u$  and each item  $i$  will have a set of representations from layer 0 to layer  $L$ :  $\{e_u^{(0)}, e_u^{(1)}, \dots, e_u^{(L)}\}$  and similarly for item  $i$ . We then concatenate all the embeddings from different layers to form a comprehensive embedding vector. Specifically, the formula for aggregating the embeddings is as follows:

$$e_u^* = e_u^{(0)} \parallel e_u^{(1)} \parallel \dots \parallel e_u^{(L)} \quad (14)$$

$$e_i^* = e_i^{(0)} \parallel e_i^{(1)} \parallel \dots \parallel e_i^{(L)} \quad (15)$$

Subsequently, to evaluate the compatibility score between a user and an item, we compute the inner product of their two vectors  $e_u^*$  and  $e_i^*$ , specifically as follows:

$$\hat{y}_{ui} = (e_u^*)^\top e_i^* \quad (16)$$

This score reflects the probability that user  $u$  will be interested in item  $i$ . A higher score indicates a greater likelihood that the user will engage with the item, while a lower score suggests the opposite.

### Loss function

In this study, we train the model using a joint optimization strategy, which includes:

- The knowledge graph embedding loss  $\mathcal{L}_{KG}$  from TransR.
- The collaborative filtering loss  $\mathcal{L}_{CF}$ , which employs Bayesian Personalized Ranking (BPR) [31]. The formula is:

$$\mathcal{L}_{CF} = \sum_{(u,i,j) \in \mathcal{O}} -\ln \sigma(\hat{y}_{ui} - \hat{y}_{uj}) \quad (17)$$

In addition, we employ a regularization coefficient to prevent the model from overfitting. Finally, the overall loss function of the model is formulated as follows:

$$\mathcal{L}_{KGR\text{ec}} = \mathcal{L}_{KG} + \mathcal{L}_{CF} + \lambda \|\Theta\|_2^2 \quad (18)$$

Where  $\lambda$  is the regularization coefficient, and  $\theta$  denotes the set of model parameters.

### Experiments

We design experiments to address two key research questions:

- **RQ1:** How does KGRec perform in comparison with state-of-the-art knowledge-aware recommendation methods?
- **RQ2:** How do different architectural components—such as the attention, the adjacency matrix normalization mechanisms, the number of layers and the choice of aggregator—affect the performance of the model?

The following section provides a detailed explanation of these experiments and their results.

## Dataset

To evaluate the effectiveness of KGRec in knowledge graph-based recommendation, we employ four publicly available benchmark datasets: Yelp2018, Last-FM, Amazon-Book and MovieLens-1M. These datasets represent different application domains (e-commerce, music, and local businesses), with varying scales and levels of sparsity, making them suitable for assessing the generalizability and scalability of the model. For each of the above datasets, we split the data into 80% for training, 10% for validation, and 10% for testing. Dataset statistics are shown in [Table 1](#).

- **Yelp2018 [32]:** The Yelp2018 dataset is derived from the 2018 Yelp Challenge, and is one of the standard benchmarks in location and service recommendation research. The original data consist of user reviews of local businesses such as restaurants, cafés, and fitness centers. We apply the 10-core criterion to ensure that each user and each business has a minimum level of interaction.
- **Last-FM [33]:** The Last-FM dataset is constructed from users’ music listening histories on the Last.fm platform, covering the period from January to June 2015. Each user can interact with one or more music artists through listening behavior. Similar to Amazon-Book, we apply the 10-core criterion to filter users and artists with a sufficient number of interactions.
- **Amazon-Book [34]:** The Amazon-Book dataset is extracted from the Amazon product review dataset (2018), widely used in product recommendation research. In this dataset, users interact with items (books) through ratings and reviews. To ensure data quality and stability, we apply the 10-core filtering strategy, which retains only users and items with at least 10 interactions.
- **MovieLens-1M [35]:** The MovieLens-1M dataset is a widely used benchmark in recommender-system research, containing 1 million movie ratings collected from real users on the MovieLens platform. It provides rich information including user profiles, movie metadata, and timestamped ratings, making it suitable for building and evaluating collaborative-filtering and hybrid models. Similar to other datasets, we apply the 10-core criterion to filter users and movies with a sufficient number of interactions.

**Table 1. Dataset Statistics.**

Type of Recommender System	Type	Yelp2018	Last-FM	Amazon-Book	MovieLens-1M
User-Item Interaction	Users	45,919	23,566	70,679	6,041
	Items	45,538	48,123	24,915	3,261
	Interactions	1,185,068	3,034,796	847,733	18,120
Knowledge Graph	Entities	90,961	58,266	88,572	14,709
	Relations	42	9	39	21
	Triplets	1,853,704	464,567	2,557,746	434,189

<https://doi.org/10.1371/journal.pone.0344585.t001>

## Model evaluation

### Ranking strategy

During evaluation, the model computes the predicted compatibility score  $\hat{y}_{ui}$  for all items  $i \in I$  with respect to each user  $u$ . The items are then ranked in descending order according to their  $\hat{y}_{ui}$ . From this ranked list, the model selects the top- $K$  items with the highest scores, which are regarded as the recommendation results for user  $u$ . In this study, we set  $K = 20$ , meaning that the model returns the 20 items with the highest predicted scores for each user.

### Evaluation metrics

In this study, to evaluate the performance of the model, we employ four widely used metrics: Precision, Recall, F1-score [36] and NDCG [37].

- **Precision@K:** This metric measures the proportion of recommended items that are relevant, defined as:

$$\text{Precision@K} = \frac{|\text{Recommended} \cap \text{Relevant}|}{|\text{Recommended}|} \quad (19)$$

- **Recall@K:** This metric measures the ability of the model to correctly recommend the items that a user is truly interested in. For each user, the model generates a ranked list of the top- $K$  items with the highest predicted scores, and then checks how many of the items the user actually interacted with appear in this list. The formula is given as:

$$\text{Recall@K} = \frac{\text{Number of relevant items at } K}{\text{Total number of relevant items}} \quad (20)$$

- **F1-score:** This metric is the harmonic mean of Precision and Recall, defined as:

$$F1 = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (21)$$

- **NDCG@K:** This metric not only evaluates the presence of relevant items in the recommendation list but also takes into account their positions within the list. Items appearing at higher ranks are assigned greater importance. This provides a more accurate reflection of the model's ranking quality. The general formula is given as:

$$\text{NDCG@K} = \frac{1}{\text{IDCG@K}} \sum_{i=1}^K \frac{rel_i}{\log_2(i+1)} \quad (22)$$

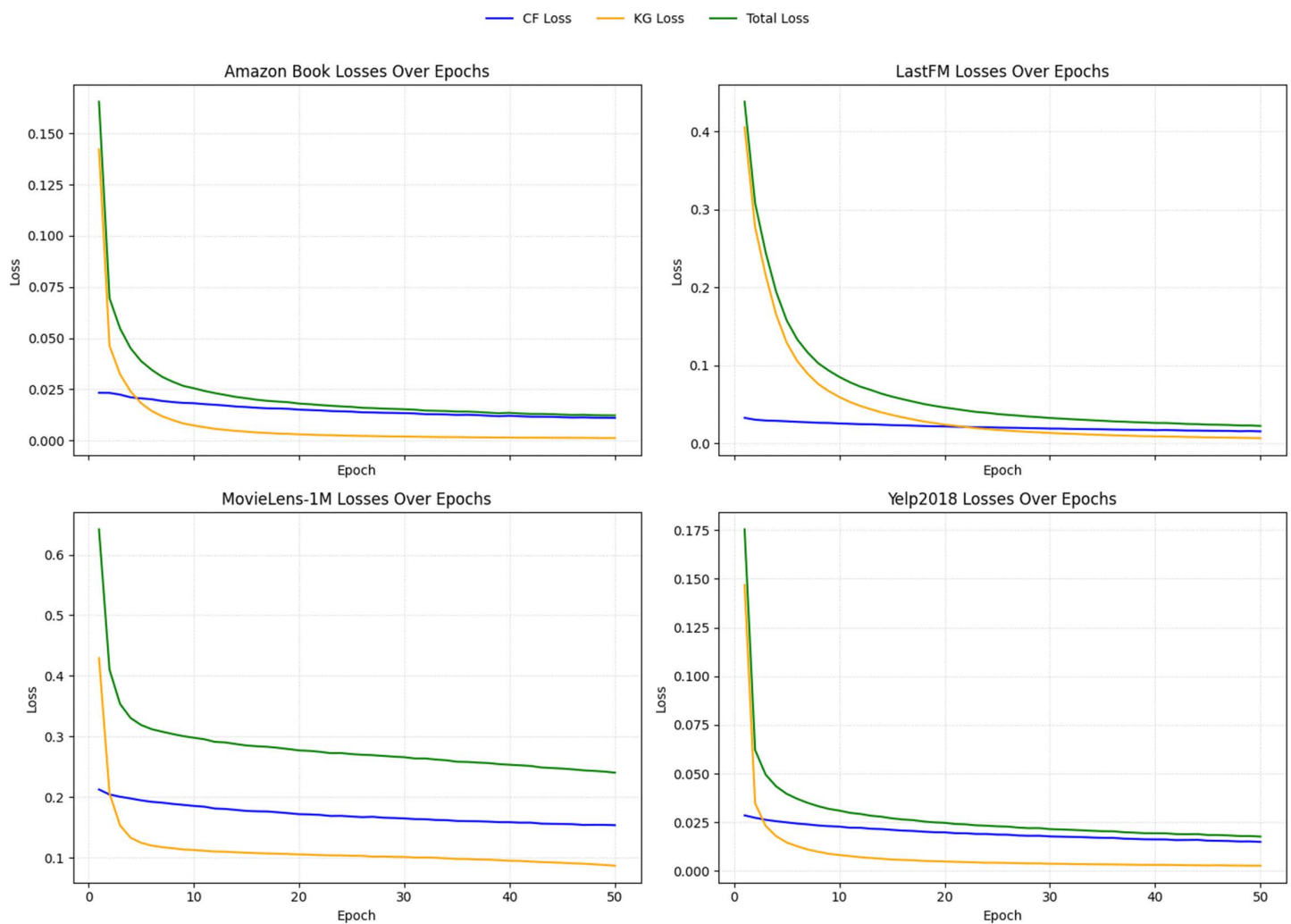
Where  $rel_i$  denotes the relevance of the item at position  $i$ ,  $IDCG$  is the ideal DCG value (i.e., the maximum value achieved if all relevant items appear at the top positions).

Note that Precision@K, Recall@K, and F1@K are ranking-based top-K metrics. In large-scale and sparse recommendation settings, their absolute values are typically small (often below 0.1) and are not directly comparable to classification-based F1 scores. For instance, while some baseline models like CKAN may report F1 scores higher than 0.7 in binary classification tasks (such as CTR prediction), their performance in top-K ranking tasks on sparse datasets like Amazon-Book is significantly lower due to the increased difficulty of the ranking objective. In this study, we strictly follow the top-K ranking protocol to ensure a fair and consistent comparison across all models.

## Results

The behavior of the loss function throughout the training process is presented in Fig 3 for each of the four benchmark datasets. Specifically, for the Amazon-Book, Last-FM and MovieLens-1M datasets, the number of training epochs varies from 10 up to 100, whereas for the Yelp2018 dataset, the training epoch range is slightly shorter, spanning from 10 to 60. By examining the training curves, we observe a consistent trend across all datasets. In the early phase of training—approximately the first 40% of the total epochs—the loss associated with both the collaborative filtering (CF) component and the knowledge graph (KG) component decreases rapidly, indicating that the model is learning useful representations and adjusting its parameters effectively during this stage.

After this initial steep decline, however, the rate of loss reduction gradually diminishes, and the curves begin to flatten out. In the later part of training—roughly the remaining 60% of epochs—the loss values become much more stable and exhibit only minor fluctuations, suggesting that the model has reached a point of convergence and that further training yields relatively limited improvements. This stabilization of the loss is consistent across the Yelp2018, Last-FM,



**Fig 3. Training loss curves on the four datasets.**

<https://doi.org/10.1371/journal.pone.0344585.g003>

Amazon-Book and MovieLens-1M datasets, highlighting that the proposed training strategy enables the model to converge efficiently within a reasonable number of epochs.

### Model performance (RQ1)

We train KGRec and compare its performance with six baseline models such as:

- **MCR**ec [38]: Exploits meta-paths in the knowledge graph.
- **AMIE** [39]: A model that combines embeddings with the knowledge graph, but without propagation.
- **CBFM** [40]: A linear model that learns user–item feature interactions through embedding vectors.
- **KGAT** [41]: A knowledge graph attention network.
- **CKAN** [42]: Collaborative knowledge-aware attentive network.
- **KGNN\_LS** [43]: Knowledge-aware graph neural networks with label smoothness regularization.

The experimental results obtained from the four benchmark datasets are summarized in Table 2. As shown in Table 2, the experimental results across the four datasets consistently demonstrate that KGRec delivers substantially better performance than the other state-of-the-art knowledge-graph-based recommendation models, including MCR, AMIE, CBFM, KGAT, CKAN, and KGNN\_LS. When evaluated using four commonly used metrics—Precision, Recall, F1-score, and NDCG—KGRec achieves the highest or co-highest performance in nearly all configurations, highlighting its robustness and adaptability across datasets of different sizes and sparsity levels.

In particular, on the three relatively sparse datasets, Amazon-Book, Last-FM and Yelp2018, KGRec exhibits a pronounced advantage, especially in Recall and NDCG. This improvement underscores KGRec’s ability to more effectively capture high-order semantic relationships and leverage structural signals in the knowledge graph, allowing it to identify relevant items even in scenarios where user–item interactions are limited. On the denser MovieLens-1M dataset, KGRec continues to demonstrate strong superiority by maintaining stable gains over all baselines. Its improvements in both

**Table 2. Comparison results with other baseline models.**

Dataset	Metric	MCR	AMIE	CBFM	KGAT	CKAN	KGNN_LS	KGRec
<b>Amazon Book</b>	Precision	0.011	0.010	0.012	0.013	0.006	0.012	<b>0.014</b>
	Recall	13.04%	11.04%	12.04%	13.05%	6.1%	12.18%	<b>14.13%</b>
	F1	0.020	0.012	0.018	0.023	0.011	0.022	<b>0.025</b>
	NDCG	0.060	0.041	0.051	0.068	0.030	0.063	<b>0.074</b>
<b>LastFM</b>	Precision	0.028	0.025	0.027	0.030	0.004	0.029	<b>0.032</b>
	Recall	6.85%	5.50%	6.20%	7.1%	1.00%	7.28%	<b>7.9%</b>
	F1	0.032	0.020	0.028	0.034	0.005	0.034	<b>0.037</b>
	NDCG	0.058	0.045	0.052	0.061	0.009	0.061	<b>0.068</b>
<b>Yelp2018</b>	Precision	0.015	0.012	0.014	0.015	0.005	0.014	<b>0.016</b>
	Recall	6.64%	5.10%	5.85%	6.26%	2.29%	5.96%	<b>6.78%</b>
	F1	0.023	0.015	0.020	0.022	0.008	0.021	<b>0.024</b>
	NDCG	0.042	0.030	0.038	0.040	0.014	0.037	<b>0.043</b>
<b>MovieLens-1M</b>	Precision	0.215	0.185	0.205	0.221	0.103	0.217	<b>0.227</b>
	Recall	19.80%	17.50%	18.90%	21.01%	10.00%	20.41%	<b>21.82%</b>
	F1	0.165	0.140	0.158	0.170	0.079	0.166	<b>0.176</b>
	NDCG	0.280	0.245	0.270	0.291	0.134	0.281	<b>0.300</b>

<https://doi.org/10.1371/journal.pone.0344585.t002>

ranking- and relevance-based metrics indicate that the model’s enhanced feature aggregation and deep interaction reasoning contribute directly to more accurate top-*K* recommendation performance.

Overall, the expanded experimental analysis provides clear evidence that KGRec is the most effective and reliable model in the comparison group. Its consistent and significant improvements across all datasets illustrate its capacity to generalize well and deliver high-quality recommendations under diverse conditions, further validating the advantages of our proposed framework.

We conducted multiple experimental runs to ensure the reliability of our results. Specifically, we trained the KGRec model five times using different random initialization seeds to verify that the performance improvements are consistent and not merely artifacts of stochastic initialization. The results are shown in [Table 3](#).

All training configurations and hyperparameters used in our experiments are documented in the configuration files, which are publicly available in the GitHub repository provided in the link.

The results obtained from training the KGRec model across five independent runs demonstrate a remarkably high level of stability. For key evaluation metrics such as Precision, F1-score, and NDCG, the variation between different training runs is virtually zero. This consistency indicates that the model’s performance is not influenced by minor perturbations introduced through random parameter initialization. In other words, the core mechanisms of KGRec—its graph reasoning, feature aggregation, and interaction modeling—operate reliably regardless of small stochastic differences at the start of training.

For the *Recall* metric, although a slight variation is observed across runs, the range remains extremely small, fluctuating only between approximately 0.1% and 0.24%. Importantly, this difference is statistically insignificant and does not suggest any instability in model behavior. All four datasets used in the experiments—Amazon-Book, Last-FM, Yelp2018 and MovieLens-1M—exhibit the same overall trend: KGRec consistently produces nearly identical outcomes with low variance and no observable anomalies across repeated tests.

Taken together, these findings demonstrate that KGRec exhibits excellent robustness and reproducibility. The model is largely insensitive to random fluctuations and can reliably reproduce its performance even when trained multiple times

**Table 3. Evaluation of multiple experimental runs.**

Dataset	Metric	1	2	3	4	5	std
Amazon Book	Precision	0.014	0.014	0.014	0.014	0.014	0
	Recall	13.99%	14.01%	13.96%	14.03%	14.13%	0.17%
	F1	0.025	0.025	0.025	0.025	0.025	0
	NDCG	0.074	0.074	0.074	0.074	0.075	0.001
LastFM	Precision	0.032	0.032	0.032	0.032	0.032	0
	Recall	7.9%	7.9%	8.0%	7.9%	8.0%	0.1%
	F1	0.037	0.037	0.038	0.038	0.038	0.001
	NDCG	0.068	0.068	0.069	0.068	0.068	0.001
Yelp2018	Precision	0.016	0.016	0.016	0.016	0.016	0
	Recall	6.72%	6.77%	6.72%	6.72%	6.75%	0.05%
	F1	0.024	0.024	0.024	0.024	0.024	0
	NDCG	0.042	0.043	0.043	0.042	0.043	0.001
MovieLens-1M	Precision	0.226	0.226	0.227	0.226	0.227	0.001
	Recall	21.69%	21.71%	21.86%	21.62%	21.74%	0.24%
	F1	0.175	0.175	0.176	0.175	0.175	0.001
	NDCG	0.298	0.298	0.299	0.299	0.299	0.001

<https://doi.org/10.1371/journal.pone.0344585.t003>

under independent initialization conditions. This strong stability further confirms the reliability of the model and supports its ability to generalize effectively across different learning scenarios and datasets.

We also conducted experiments on training time and memory usage for all compared models. The results are shown in [Table 4](#) and [Table 5](#). Following these tables, we provide a detailed analysis of the model complexity and efficiency.

**Complexity and Efficiency Analysis:** We acknowledge that, as shown in [Table 5](#), KGRec has the highest or nearly highest number of parameters among the compared models. This is primarily due to the deliberate architectural design of KGRec, which integrates a knowledge-aware attention mechanism, multi-layer embedding propagation, and a Bi-Interaction aggregator. These components introduce additional trainable parameters but are essential for capturing high-order semantic relations in the collaborative knowledge graph. Importantly, the parameter scale of KGRec is comparable to other advanced KG-based models such as KGAT and CKAN, and does not represent a significant increase relative to their model capacities. In contrast, lighter models (e.g., AMIE and CBFM) employ simpler architectures without deep propagation or attention mechanisms, which explains both their smaller size and inferior recommendation performance.

Moreover, as demonstrated in [Table 4](#), the increased number of parameters does not lead to prohibitive computational costs. We agree that model size is an important factor affecting training time; however, training efficiency is not determined solely by the number of model parameters. In graph-based recommender systems, training time is also strongly influenced by dataset characteristics such as the number of entities and triples in the knowledge graph, graph sparsity, neighborhood size, and convergence behavior. This explains why KGRec exhibits relatively higher training time on the Amazon-Book dataset, which contains a large-scale and sparse knowledge graph with over 2.5 million triples, leading to more expensive neighbor aggregation during propagation. In contrast, the LastFM dataset has a much smaller knowledge graph with fewer relations, and MovieLens-1M is both compact and relatively dense, allowing KGRec to converge faster despite having a comparable number of parameters. KGRec maintains competitive training time and acceptable

**Table 4. Evaluation on time and memory of all compared models.**

Dataset	Runtime/Memory	MCRRec	AMIE	CBFM	KGAT	CKAN	KGNN_LS	KGRec
Amazon Book	Training time (s)	9500.25	4200.15	6800.50	11475.40	8064.40	5769.44	<b>15172.66</b>
	Memory (MB)	1850.5	520.4	1100.2	2435.9	669.7	413.6	<b>7252.1</b>
LastFM	Training time (s)	10200.40	5100.88	7500.30	12886.61	15980.97	20551.07	<b>11383.38</b>
	Memory (MB)	2100.6	650.2	1350.8	3916.3	1109.8	545.2	<b>4723.6</b>
Yelp2018	Training time (s)	18500.10	8500.60	13200.45	26727.24	11575.90	12036.29	<b>15821.15</b>
	Memory (MB)	1200.3	450.5	850.9	1648.8	1090.6	583.3	<b>6538.8</b>
MovieLens-1M	Training time (s)	9200.55	3800.20	6100.75	11877.61	8429.03	12836.84	<b>6650.44</b>
	Memory (MB)	420.5	150.2	310.6	548.5	303.2	60.7	<b>2866.8</b>

<https://doi.org/10.1371/journal.pone.0344585.t004>

**Table 5. Model summary of all compared models.**

Model	Model parameters	Model size (MB)
KGAT	12146920	46.34
CKAN	11804096	45.03
KGNN_LS	11795904	45.00
MCRRec	6611458	25.22
AMIE	2105400	8.15
CBFM	4500800	17.20
<b>KGRec</b>	<b>12132848</b>	<b>46.28</b>

<https://doi.org/10.1371/journal.pone.0344585.t005>

memory consumption across all datasets, while consistently achieving superior performance in terms of Precision, Recall, F1-score, and NDCG. This indicates a favorable trade-off between model complexity and recommendation effectiveness.

### Architectural components affecting the performance of KGRec (RQ2)

To gain deeper insights into the core components of KGRec and to assess the contribution of each factor to the overall performance, we conduct a series of in-depth analyses on the four aforementioned datasets, with the following specific objectives:

- **Impact of Attention:** We investigate how integrating attention influences the recommendation performance of the model. The results are presented in [Table 6](#).

The experimental results obtained from the four benchmark datasets —Amazon-Book, Last-FM, Yelp2018 and MovieLens-1M—demonstrate a consistent and substantial performance improvement when the attention mechanism is incorporated into the KGRec architecture. Across all evaluation metrics, the attention-enhanced version of KGRec outperforms the variant that excludes attention, indicating that this component plays a critical role in strengthening the model’s ability to capture meaningful relational patterns within the knowledge graph.

More specifically, the attention mechanism enables the model to dynamically weigh the importance of different neighboring entities and relations, allowing KGRec to focus on the most informative semantic signals during the feature aggregation process. This selective weighting results in richer and more discriminative user–item representations, which in turn leads to more accurate recommendation outcomes. The consistent performance gains across diverse datasets—ranging from highly sparse (Amazon-Book, Last-FM, Yelp2018) to relatively dense (MovieLens-1M)—further highlight the robustness and general applicability of the attention mechanism.

Overall, these findings affirm that the attention mechanism is a crucial and indispensable component within KGRec. By enabling the model to learn enhanced semantic representations and prioritize relevant information more effectively, attention contributes significantly to the model’s ability to achieve strong and reliable recommender performance across multiple data conditions and domains.

**Table 6. Impact of attention mechanism on model performance.**

Dataset	Metric	With Attention	Without Attention
Amazon Book	Precision	0.014	0.012
	Recall	14.13%	14.03%
	F1	0.025	0.024
	NDCG	0.074	0.070
LastFM	Precision	0.032	0.030
	Recall	8.00%	7.9%
	F1	0.037	0.034
	NDCG	0.068	0.064
Yelp2018	Precision	0.016	0.014
	Recall	6.78%	6.72%
	F1	0.024	0.022
	NDCG	0.043	0.040
MovieLens-1M	Precision	0.227	0.225
	Recall	21.82%	21.74%
	F1	0.176	0.173
	NDCG	0.300	0.295

<https://doi.org/10.1371/journal.pone.0344585.t006>

- **Impact of Adjacency Matrix Normalization:** We evaluate how different adjacency matrix normalization mechanisms in the graph affect the model. The results are presented in [Table 7](#).

The experimental results obtained from the four benchmark datasets indicate that adopting the random-walk adjacency matrix normalization leads to consistently better performance compared to using symmetric normalization. Although the magnitude of improvement is relatively small, it remains highly stable across nearly all evaluation metrics. This effect is especially pronounced on the three sparse datasets—Amazon-Book, Last-FM, and Yelp2018—where the quality of graph-based signal propagation plays a more critical role due to limited user–item interactions.

The superior performance of the random-walk normalization can be attributed to its ability to model personalized transition probabilities by normalizing each node according to its outgoing degree. This formulation allows information to flow through the graph in a direction-sensitive manner, enabling the model to better capture asymmetric structural relationships and prioritize more informative neighbors during message passing. In contrast, symmetric normalization imposes an equalized structure that may dilute important signals, particularly in graphs with high sparsity or skewed degree distributions.

Across all datasets, the results consistently demonstrate that random-walk normalization enhances KGRec’s capacity to propagate semantic information more effectively through the knowledge graph, leading to more expressive user and item embeddings. This improvement ultimately contributes to higher recommendation accuracy and stronger ranking performance. Overall, the findings validate random-walk normalization as a more suitable and advantageous choice for the KGRec model, ensuring more reliable and robust recommendation quality across diverse data conditions.

- **Impact of the number of propagation layers:** We investigate how the number of embedding propagation layers influences the recommendation performance of the model. Each additional layer corresponds to expanding the receptive field to higher-order neighbors in the knowledge graph. The results are presented in [Table 8](#).

The experimental results across all four datasets indicate that the number of propagation layers in KGRec does influence recommendation quality, although the magnitude of change remains relatively moderate. For the Amazon-Book, Last-FM and Yelp2018 datasets, we observe a small but consistent improvement in performance when increasing the number

**Table 7. Performance comparison of adjacency matrix normalization mechanisms.**

Dataset	Metric	Random-walk	Symmetric
Amazon Book	Precision	0.014	0.013
	Recall	14.13%	14.09%
	F1	0.025	0.024
	NDCG	0.074	0.072
Last-FM	Precision	0.032	0.030
	Recall	8.00%	7.97%
	F1	0.037	0.036
	NDCG	0.068	0.065
Yelp2018	Precision	0.016	0.015
	Recall	6.78%	6.74%
	F1	0.024	0.022
	NDCG	0.043	0.042
MovieLens-1M	Precision	0.227	0.226
	Recall	21.82%	21.80%
	F1	0.176	0.175
	NDCG	0.300	0.297

<https://doi.org/10.1371/journal.pone.0344585.t007>

**Table 8. Impact of the number of propagation layers on the performance of KGRec.**

Dataset	Metric	1 Layer	2 Layers	3 Layers	4 Layers
Amazon Book	Precision	0.014	0.014	0.014	0.014
	Recall	14.07%	14.08%	14.16%	14.11%
	F1	0.025	0.025	0.025	0.026
	NDCG	0.074	0.074	0.075	0.074
Last-FM	Precision	0.032	0.032	0.032	0.032
	Recall	7.8%	7.9%	7.9%	7.98%
	F1	0.037	0.037	0.037	0.038
	NDCG	0.067	0.067	0.068	0.068
Yelp2018	Precision	0.016	0.016	0.016	0.016
	Recall	6.78%	6.85%	6.90%	6.71%
	F1	0.024	0.024	0.024	0.023
	NDCG	0.043	0.043	0.044	0.042
MovieLens-1M	Precision	0.227	0.227	0.227	0.225
	Recall	21.70%	21.72%	21.82%	21.63%
	F1	0.175	0.175	0.176	0.174
	NDCG	0.295	0.298	0.300	0.297

<https://doi.org/10.1371/journal.pone.0344585.t008>

of layers to three. Metrics such as Recall, F1-score, and NDCG all show noticeable gains compared to the 1-layer and 2-layer configurations, suggesting that deeper propagation at this level helps the model capture richer relational information from the knowledge graph. However, when the number of layers is further increased to four, the performance declines slightly, indicating that excessive propagation introduces noise and may distort the learned representations.

A similar trend appears in the MovieLens-1M dataset. The model achieves its highest performance at three layers, with NDCG reaching 0.300—the best among all tested configurations. When the depth is increased to four layers, the performance again drops, reinforcing the observation that overly deep propagation can lead to over-smoothing, where node representations become less distinguishable and the model’s discriminative power weakens.

Overall, these findings consistently demonstrate that a depth of three propagation layers is the most appropriate setting for KGRec. This configuration provides a balance between extracting useful high-order semantic information and avoiding the negative effects associated with overly deep message passing, allowing the model to fully leverage the structure of the knowledge graph while maintaining strong and stable recommendation performance.

- **Impact of the aggregator function:** To evaluate the effectiveness of different strategies for aggregating information from neighbors, we compare the three aggregators introduced in the methodology section: GCN [30], GraphSAGE [17], and Bi-Interaction [13]. The results are presented in Table 9.

The experimental results indicate that the three information aggregation mechanisms evaluated in KGRec—GraphSAGE, GCN, and Bi-Interaction—provide broadly comparable levels of recommendation performance. However, Bi-Interaction consistently delivers the strongest results across all four benchmark datasets. On Amazon-Book, Last-FM and Yelp2018, Bi-Interaction demonstrates clear improvements in key metrics such as Recall, F1-score, and NDCG. These gains suggest that its ability to combine both “sum-based aggregation” and explicit “pairwise feature interaction” enables the model to capture richer relational dependencies and more expressive semantic signals from the knowledge graph. By simultaneously modeling first-order and second-order feature interactions, Bi-Interaction generates more discriminative user and item embeddings, contributing to its superior performance.

**Table 9. Impact of the aggregator function on the performance of KGRec.**

Dataset	Metric	GraphSAGE	GCN	Bi-Interaction
Amazon Book	Precision	0.014	0.013	0.014
	Recall	13.8%	14.10%	14.13%
	F1	0.025	0.023	0.025
	NDCG	0.073	0.074	0.075
Last-FM	Precision	0.031	0.030	0.032
	Recall	7.86%	7.8%	7.9%
	F1	0.036	0.035	0.037
	NDCG	0.067	0.066	0.068
Yelp2018	Precision	0.016	0.016	0.016
	Recall	6.72%	6.77%	6.78%
	F1	0.023	0.024	0.025
	NDCG	0.042	0.043	0.044
MovieLens-1M	Precision	0.226	0.225	0.227
	Recall	21.76%	21.72%	21.82%
	F1	0.175	0.175	0.176
	NDCG	0.297	0.298	0.300

<https://doi.org/10.1371/journal.pone.0344585.t009>

In contrast, GCN tends to produce the weakest performance among the three methods. This is largely due to its simple symmetric normalization and linear message-passing process, which may not fully capture complex relational patterns, especially in sparse or highly heterogeneous graph structures. GraphSAGE performs moderately well, generally achieving results between GCN and Bi-Interaction. Its sampling-based aggregation helps capture localized neighborhood structure more effectively than GCN, but it still lacks the richer interaction modeling that gives Bi-Interaction its advantage.

A similar pattern emerges in the MovieLens-1M dataset. Although the differences between the aggregation methods are somewhat smaller due to the dataset’s relative density, Bi-Interaction still achieves the highest scores across all evaluation metrics. This consistency across both sparse and dense datasets confirms that Bi-Interaction is better equipped to leverage graph connectivity and semantic relationships, enabling more accurate top-*K* recommendations.

Overall, the results strongly support Bi-Interaction as the most suitable and effective information aggregation mechanism for the KGRec model. Its ability to capture multi-level relational information and generate more expressive feature representations leads to consistently superior recommendation quality across diverse datasets, making it the preferred choice among the evaluated alternatives.

## Conclusion

In this study, we introduced and implemented the Knowledge Graph Attention Network Recommendation model (KGRec) for the top-*K* recommendation task, with the goal of effectively exploiting semantic knowledge from knowledge graphs to improve recommendation quality. Unlike traditional recommendation methods that rely solely on user–item interactions, KGRec deeply integrates both knowledge information and graph structure through multi-layer weighted embedding propagation and a knowledge-aware attention mechanism. The propagation process is performed iteratively, combining multi-level representations and employing the Bi-Interaction aggregator to enhance expressive capacity.

Experimental results on three real-world datasets—Yelp2018, Last-FM, and Amazon-Book—demonstrated that KGRec outperforms existing baseline models in terms of both accuracy (Recall@20) and ranking quality (NDCG@20). Through detailed analyses, we also showed that the attention mechanism, the number of propagation layers, and the choice of aggregator function are key factors influencing system performance.

Although KGRec has proven highly effective for knowledge graph-based recommendation, several promising research directions remain for future work. First, it is necessary to optimize computational cost in the embedding propagation process, especially when applied to large-scale knowledge graphs, by adopting techniques such as selective propagation (sampling) or compressed propagation. Second, incorporating temporal and sequential information in user behavior is essential to better capture context and preference dynamics over time. Third, leveraging pre-trained deep learning models such as K-BERT, BERT4Rec, or KG-BERT could further enhance the representational capacity of entities and relations in the KG. Moreover, extending KGRec to other tasks such as sequential recommendation, rating prediction, or real-time personalization represents practical directions. These avenues will serve as a foundation for advancing and deploying KGRec in large-scale intelligent recommender systems in the future.

## Author contributions

**Conceptualization:** Trinh Duong Hoan, Bui Thanh Hung.

**Data curation:** Trinh Duong Hoan, Bui Thanh Hung.

**Formal analysis:** Trinh Duong Hoan, Bui Thanh Hung.

**Funding acquisition:** Bui Thanh Hung.

**Investigation:** Bui Thanh Hung.

**Methodology:** Trinh Duong Hoan, Bui Thanh Hung.

**Project administration:** Bui Thanh Hung.

**Supervision:** Bui Thanh Hung.

**Validation:** Bui Thanh Hung.

**Visualization:** Trinh Duong Hoan, Bui Thanh Hung.

**Writing – original draft:** Trinh Duong Hoan, Bui Thanh Hung.

**Writing – review & editing:** Bui Thanh Hung.

## References

1. Xu J, Chen Z, Yang S, Li J, Wang W, Hu X, et al. A Survey on Multimodal Recommender Systems: Recent Advances and Future Directions. arXiv preprint arXiv:250215711. 2025. <http://dx.doi.org/10.48550/arXiv.2502.15711>
2. Zhu X, Wang Y, Gao H, Xu W, Wang C, Liu Z, et al. Recommender Systems Meet Large Language Model Agents: A Survey. *Foundations and Trends® in Privacy and Security*. 2025;7(4):247–396. <https://doi.org/10.1561/33000000050>
3. Anand V, Maurya AK. A Survey on Recommender Systems Using Graph Neural Network. *ACM Trans Inf Syst*. 2024;43(1):1–49. <https://doi.org/10.1145/3694784>
4. Liu Q, Hu J, Xiao Y, Zhao X, Gao J, Wang W, et al. Multimodal Recommender Systems: A Survey. *ACM Comput Surv*. 2024;57(2):1–17. <https://doi.org/10.1145/3695461>
5. Hung BT, Duy HVH. ExVQA: a novel stacked attention networks with extended long short-term memory model for visual question answering. *Computers and Electrical Engineering*. 2025;126:110439. <https://doi.org/10.1016/j.compeleceng.2025.110439>
6. Li Y, Liu K, Satapathy R, Wang S, Cambria E. Recent Developments in Recommender Systems: A Survey [Review Article]. *IEEE Comput Intell Mag*. 2024;19(2):78–95. <https://doi.org/10.1109/mci.2024.3363984>
7. Hung BT, Huy VQ. MTFIC: enhanced fashion image captioning via multi-transformer architecture with contrastive and bidirectional encodings. *Vis Comput*. 2025;41(13):10841–55. <https://doi.org/10.1007/s00371-025-04072-8>
8. Aljunid MF, Manjaiah H. M, Hooshmand MK, Ali WA, Shetty AM, Alzoubah SQ. A collaborative filtering recommender systems: Survey. *Neurocomputing*. 2025;617:128718. <https://doi.org/10.1016/j.neucom.2024.128718>
9. Hogan A, Blomqvist E, Cochez M, D'amato C, Melo GD, Gutierrez C, et al. Knowledge Graphs. *ACM Comput Surv*. 2021;54(4):1–37. <https://doi.org/10.1145/3447772>
10. Liang W, Meo PD, Tang Y, Zhu J. A Survey of Multi-modal Knowledge Graphs: Technologies and Trends. *ACM Comput Surv*. 2024;56(11):1–41. <https://doi.org/10.1145/3656579>

11. Hung BT, Nhan NVP, Sy NT. DCARES: deep convolutional neural network with neural-based optimization for image-based product recommender system. *Multimed Tools Appl.* 2025;84(30):36693–723. <https://doi.org/10.1007/s11042-025-20655-y>
12. Wu S, Sun F, Zhang W, Xie X, Cui B. Graph Neural Networks in Recommender Systems: A Survey. *ACM Comput Surv.* 2022;55(5):1–37. <https://doi.org/10.1145/3535101>
13. Ho Vo Hoang D, Vo Quoc H, Hung BT. ConBGAT: a novel model combining convolutional neural networks, transformer and graph attention network for information extraction from scanned image. *PeerJ Comput Sci.* 2024;10:e2536. <https://doi.org/10.7717/peerj-cs.2536> PMID: 39650481
14. Xiao X, Jin B, Zhang C. Personalized paper recommendation for postgraduates using multi-semantic path fusion. *Appl Intell.* 2022;53(8):9634–49. <https://doi.org/10.1007/s10489-022-04017-x>
15. Markchom T, Liang H, Ferryman J. Explainable Meta-Path Based Recommender Systems. *ACM Trans Recomm Syst.* 2024;3(2):1–28. <https://doi.org/10.1145/3625828>
16. Javaji SR, Sarode K. Hybrid recommendation system using graph neural network and bert embeddings. arXiv preprint. 2023. <https://doi.org/10.48550/arXiv.2310.04878>
17. Liu Y, Qi L, Liu W, Xu X, Zhang X, Dou W. GraphSAGE-based POI Recommendation via Continuous-Time Modeling. In: Companion Proceedings of the ACM Web Conference 2024, 2024. 585–8. <http://dx.doi.org/10.1145/3589335.3651515>
18. Ma C, Ren X, Xu G, He B. FedGR: Federated Graph Neural Network for Recommendation Systems. *Axioms.* 2023;12(2):170. <https://doi.org/10.3390/axioms12020170>
19. Vrahatis AG, Lazaros K, Kotsiantis S. Graph Attention Networks: A Comprehensive Review of Methods and Applications. *Future Internet.* 2024;16(9):318. <https://doi.org/10.3390/fi16090318>
20. Zheng X, Jia X, Cheng X, He W, Sun L, Guo L, et al. DM-FedMF: A Recommendation Model of Federated Matrix Factorization With Detection Mechanism. *IEEE Trans Netw Sci Eng.* 2025;12(4):2679–93. <https://doi.org/10.1109/tNSE.2025.3551923>
21. Liu Z, Yang L, Fan Z, Peng H, Yu PS. Federated Social Recommendation with Graph Neural Network. *ACM Trans Intell Syst Technol.* 2022;13(4):1–24. <https://doi.org/10.1145/3501815>
22. Qu Z, Lin K, Li Z, Zhou J. Federated learning's blessing: Fedavg has linear speedup. In: ICLR 2021-Workshop on Distributed and Private Machine Learning (DPML); 2021.
23. Liu W, Zhang Z, Li X. Enhancing Recommendation Systems with GNNs and Addressing Over-Smoothing. 2024. <https://doi.org/10.48550/arXiv.2412.03097>
24. Ozbay B, Tugay R, Oguducu SG. A GNN Model with Adaptive Weights for Session-Based Recommendation Systems. In: Proceedings of the 2024 9th International Conference on Machine Learning Technologies, 2024. 258–64. <http://dx.doi.org/10.48550/arXiv.2408.05051>
25. Bera A, Wharton Z, Liu Y, Bessis N, Behera A. SR-GNN: Spatial Relation-aware Graph Neural Network for Fine-Grained Image Categorization. *IEEE Trans Image Process.* 2022. <https://doi.org/10.1109/TIP.2022.3205215> PMID: 36103441
26. Yang X-H, Ma G-F, Jin X, Long H-X, Xiao J, Ye L. Knowledge graph embedding and completion based on entity community and local importance. *Appl Intell.* 2023;53(19):22132–42. <https://doi.org/10.1007/s10489-023-04698-y>
27. Hohmann M, Devriendt K, Coscia M. Quantifying ideological polarization on a network using generalized Euclidean distance. *Sci Adv.* 2023;9(9):eabq2044. <https://doi.org/10.1126/sciadv.abq2044> PMID: 36857460
28. Durmus F, Saribas H, Aldemir S, Yang J, Cevikalp H. Pairwise ranking loss for multi-task learning in recommender systems. 2024. <https://doi.org/10.48550/arXiv.2406.02163>
29. Franke M, Degen J. The softmax function: Properties, motivation, and interpretation. Center for Open Science. 2023. <https://doi.org/10.31234/osf.io/vsw47>
30. Wu Z, Chen Z, Du S, Huang S, Wang S. Graph Convolutional Network with elastic topology. *Pattern Recognition.* 2024;151:110364. <https://doi.org/10.1016/j.patcog.2024.110364>
31. Hu Y, Xiong F, Pan S, Xiong X, Wang L, Chen H. Bayesian personalized ranking based on multiple-layer neighborhoods. *Information Sciences.* 2021;542:156–76. <https://doi.org/10.1016/j.ins.2020.06.067>
32. Yelp. Yelp 2018 Dataset; 2018. Distributed by Yelp. Available from: <https://www.yelp.com/dataset/>
33. Bertin-Mahieux T, Ellis DP, Whitman B, Lamere P. The Million Song Dataset; 2011. Available from: <http://millionsongdataset.com/lastfm/>
34. Kang WC, Kim E, Leskovec J, Rosenberg C, McAuley J. Complete the look: Scene-based complementary product recommendation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2019. p. 1053241. <http://dx.doi.org/10.1109/cvpr.2019.01078>
35. Harper FM, Konstan JA. The MovieLens Datasets. *ACM Trans Interact Intell Syst.* 2015;5(4):1–19. <https://doi.org/10.1145/2827872>
36. Zangerle E, Bauer C. Evaluating Recommender Systems: Survey and Framework. *ACM Comput Surv.* 2022;55(8):1–38. <https://doi.org/10.1145/3556536>
37. Brama H, Dery L, Grinshpoun T. Evaluation of neural networks defenses and attacks using NDCG and reciprocal rank metrics. *Int J Inf Secur.* 2022;22(2):525–40. <https://doi.org/10.1007/s10207-022-00652-0>

38. Ning W, Cheng R, Shen J, Haldar NAH, Kao B, Yan X, et al. Automatic Meta-Path Discovery for Effective Graph-Based Recommendation. In: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022. 1563–72. <http://dx.doi.org/10.1145/3511808.3557244>
39. Meilicke C, Fink M, Wang Y, et al. Fine-grained evaluation of rule-and embedding-based systems for knowledge graph completion. In: International Semantic Web Conference. Springer; 2018. p. 3-20.
40. Madani R, Idrissi A, Ez-Zahout A. A new context-based factorization machines for context-aware recommender systems. Modern artificial intelligence and data science: Tools, techniques and systems. Springer Nature Switzerland. 2023. p. 15–23.
41. Wang X, He X, Cao Y, Liu M, Chua T-S. KGAT. In: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019. 950–8. <http://dx.doi.org/10.1145/3292500.3330989>
42. Wang Z, Lin G, Tan H, Chen Q, Liu X. CKAN. In: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020. 219–28. <http://dx.doi.org/10.1145/3397271.3401141>
43. Wang H, Zhang F, Zhang M, Leskovec J, Zhao M, Li W, et al. 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, 2019. 968–77. <http://dx.doi.org/10.1145/3292500.3330836>