# Graph attention-based neural collaborative filtering for item-specific recommendation system using knowledge graph

Ehsan Elahi [a], Sajid Anwar [b], Mousa Al-kfairy [c], Joel J.P.C. Rodrigues [d], Alladoumbaye Ngueilbaye [e], Zahid Halim [f,g,*], Muhammad Waqas [h,i]

[a] *Karlsruhe Institute of Technology, Karlsruhe 76131, Germany*
[b] *Institute of Management Sciences, Peshawar 25000, Pakistan*
[c] *College of Technological Innovation, Zayed University, Abu Dhabi, United Arab Emirates*
[d] *Amazonas State University, Manaus, AM, Brazil*
[e] *National Engineering Laboratory for Big Data System Computing Technology, and Big Data Institute, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, PR China*
[f] *Department of Information Management, National Yunlin University of Science and Technology, 123 University Road, Section 3, Douliou, Yunlin 64002, Taiwan*
[g] *Faculty of Computer Science and Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi 23460, Pakistan*
[h] *School of Computing and Mathematical Sciences, University of Greenwich, SE10 9LS London, UK*
[i] *School of Engineering, Edith Cowan University, 6027 Perth, WA, Australia*

*Keywords:*
Industry recommendation systems
Graph neural network
Knowledge graph
Attention mechanism

A B S T R A C T

Recently, the use of graph neural networks (GNNs) for leveraging knowledge graphs (KGs) has been on the rise due to their ability to encode both first-order and higher-order neighbor information. Most GNN-based models explicitly encode first-order information of an entity but may not effectively capture higher-order information. To address this, many existing methods overlook the impact of varying relations among neighboring nodes, leading to the integration of nodes with diverse semantics. This work propose an end-to-end recommendation model, named Item-Specific Graph Attention Network (IGAT), which jointly utilizes user-item interaction and KG information to predict user preferences. IGAT incorporates a knowledge-aware attention mechanism that assigns different weights to neighboring entities based on their relations and latent vector representations in the KG. Additionally, an item-specific attention mechanism is applied to measure the influence of the target item on the user's historical items. To mitigate biases from multi-layer propagation, IGAT utilizes contextualized representations of both users and items in the recommendation process. Extensive experiments on three benchmark datasets demonstrate the superior performance of IGAT compared to state-of-the-art KG-based recommendation models, with results showing that the proposed model outperforms the baselines.

## 1. Introduction

With the increase of online information in platforms such as social media, amazon and e-commerce websites, just to name a few, the need of recommender systems has increased. Due to this fact, almost every platform which engages the user is equipped with a recommendation system. This facilitates the users to reach out to the product of their interest. Furthermore, this assists businesses to expand their revenue by appealing their customers with the user-specific content. In the last decade, different recommendation strategies were proposed including collaborative filtering (CF) (Shi et al., 2014). CF works by assuming that similar users have similar preferences on items. Matrix Factorization (MF) (Koren et al., 2009) is one such approach that is based on CF, works by considering there is some latent relation between the users and items. MF has been extensively used in the literature for the recommendation system but MF has one shortcoming that it only relies on user-item interaction. Thus, it suffers from data sparsity problem as the information is only coming from user-item interaction.

Recently, deep learning is being utilized in the recommendation scenario to transfer the CF approach into deep neural network style. For example, neural collaborative filtering (NCF) (He et al., 2017) is the deep neural network style of CF which have two steps; one is embedding

* Corresponding author.
*E-mail addresses:* ehsan.elahi@kit.edu (E. Elahi), SAJID.ANWAR@IMSCIENCES.EDU.PK (S. Anwar), Mousa.Al-kfairy@zu.ac.ae (M. Al-kfairy), joeljr@ieee.org (J.J.P.C. Rodrigues), angueilbaye@szu.edu.cn (A. Ngueilbaye), zahidh@yuntech.edu.tw (Z. Halim), M.WAQAS@ECU.EDU.AU (M. Waqas).

component while the other is interaction modelling. In embedding component, vectorized representations of both the users and the items are made whereas in interaction modelling, vector form of user and item are utilized to rebuild the user-item interaction. Different interaction modelling approaches are proposed in the literature such as translation-based method (Tay et al., 2018) in which Euclidean distance is utilized as an alternative to inner product. One thing to highlight here is that these approaches are unfit for yielding satisfactory user and item embedding. This is because they rely only on descriptive features such as ID and thus, making the model overfit in deep neural network architecture. Consequently, data sparsity problem becomes even worse due to the capturing of complex relationship from the sparse data.

To overcome the problem of data sparsity, researchers have proposed to utilize side information in the recommendation problem. Hence, the use of KG is becoming popular and have attracted the attention of research community (Wang et al., 2019). KG provides fruitful facts and semantic relatedness between the connected items and entities and thus helps in capturing the latent connections to enhance the recommendation quality. By reducing the sparsity problem, KG provides reasoning ability to the recommendation system via KG links among items. The prior work done on KG based which is propagational based method: (1) path-based, (2) embedding based and (3) propagation-based recommendation. Path-based recommendation methods are intuitive in nature as they rely on hand-crafted paths, thus may not be appropriate to uncover new connectivity paths among items. Moreover, in path-based methods, domain knowledge is required to extract the long-range connectivity among items and entities (Shi et al., 2015; Hu et al., 2018). Embedding based recommendation methods enforce the regularization loss to preserve the KG structure. Furthermore, in these methods, KG relation is captured implicitly and hence may not be appropriate to capture semantic KG relation explicitly in the recommendation scenario. Propagation based recommendation methods stack multiple layers to capture the long-range connectivity among items and thus enhance the representation of each entity by encoding the neighboring nodes information (Wang et al., 2020). One problem with propagation-based methods is that they may allow noisy entities to be added into the aggregation. Furthermore, unique representation of each entity may be disappeared due to the over-smoothing of each entity in the KG.

Generally, KG is in the form of heterogeneous graph having items and entities, and this graph context includes the local (directly attached neighbors) as well as non-local (neighbors of neighbors) entities of the given target entity. Moreover, there are diverse relations among entities in KG and several node types, thus, it is heterogeneous in nature. As an example, KG is shown in the Fig. 1 which illustrates the items and entities connected to each other via different relations. Recently, the research community have proposed new paradigm of graph neural network (GNN) based recommendation methods (Wang et al., 2019) which try to capture the semantic information of local as well as non-local graph context of the target entity. Since, there is always the room for improvement, these GNN based recommendation models usually suffer from the limitations. Thus, these models may not be suitable for applications where the below mentioned challenges are critical.

**Challenge 1:** Non-local context of the target entity is the key thing as it assists in enhancing the representation of the entity in KG and thus provide the useful reasoning of the user preference for recommendation. Existing methods which utilize path-based approach may not be suitable to uncover the new connectivity and only rely on the hand-crafted paths (Dong et al., 2017). Propagation based approach exploit the layers to capture the long connectivity of items and entities in KG but they ignore the important relational aspect among entities. Since, entities are connected via different types of relation so ignoring them or treating all of them equally make these approaches inadequate to capture the user's preferences in the recommendation system.

**Challenge 2:** Generally, historical items of the user are used to predict her preference for the candidate item (Korean, 2008). Since user may have interacted with different items and each item may have different influence in predicting her preference on the candidate item. In the literature, most of the methods ignore this fact and directly aggregate the user's historical items to enhance the user representation without discriminating the varying importance of historical items. Hence, it is important to discriminate the varying importance of user's historical items while aggregating the historical items to learn the user preference.

In this paper, the aforementioned challenges are addressed as we propose an end to end recommendation model IGAT, which is a propagational based method in which local as well as non-local graph context is leveraged. The relations among items and entities are not ignored due to the usage of relational attention mechanism. Moreover, the proposed model assigns different importance to the user's historical items to determine the preference for the target item. In other words, item specific user preferences are presented to the user. In recent years, the utilization of KG into the recommendation system is becoming popular, but either they may not consider the diverse relation of KG into consideration, while aggregating the neighboring information, or they may ignore the importance of capturing the varying contributions of a user's historical items when aggregating their historical representations. In this paper, however, these considerations are examined to have better quality of recommendations, which are also user personalized. This a capability that existing GNN-based models may not fully capture. To sum up, this paper has the following main contributions;

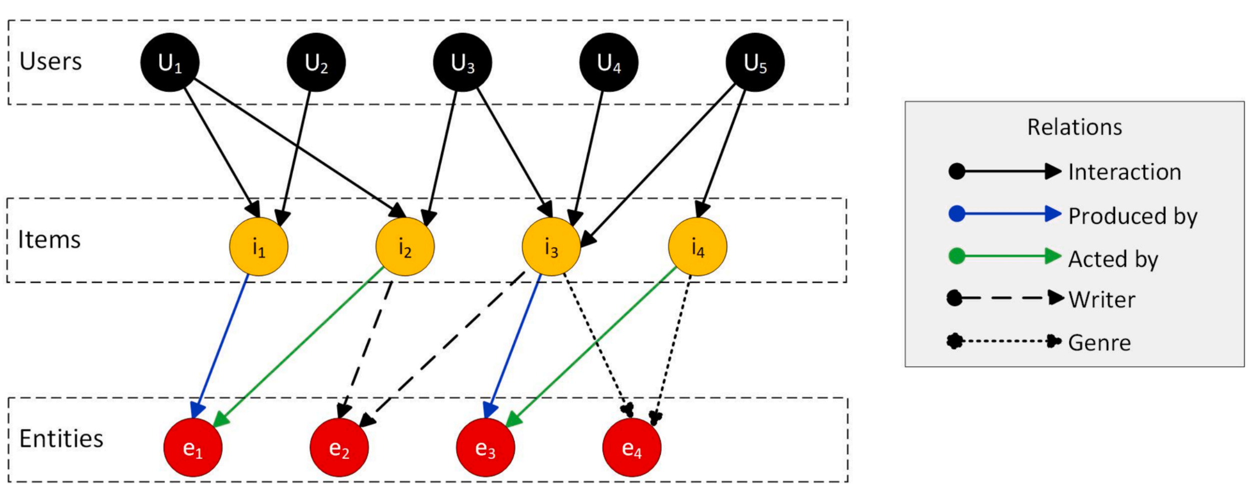


**Fig. 1.** KG with diverse entities and their relation.

- We propose an end-to-end recommendation model, IGAT, which effectively incorporate latent collaborative signals from user-item interactions with KG information.
- To mitigate bias introduced by multi-layer propagation, we have employed the contextualized representations for both users and items within the model.
- We emphasize the importance of capturing the varying contributions of a user's historical items when aggregating their historical representations.
- Extensive experiments have been conducted on three benchmark datasets from diverse domains to evaluate the effectiveness of IGAT. The results demonstrate that IGAT outperforms state-of-the-art KG-based recommendation models.

The remaining sections of the paper are organized as follows. Section 2 presents the most recent work of KG based and GNN based recommendation. In section 3, preliminaries of the problem to be solved is presented. Section 4 illustrate the proposed framework as well as detail methodology of the recommendation model whereas in section 5, experimental results along with the relevant discussion on the results is described. Section 6 conclude the paper and discuss some of the potential future directions of this study.

## 2. Related work

In this section, some recent methods proposed in the recommendation scenario are discussed. Here, we have divided them into three broad categories i.e. neural collaborative filtering, graph neural networks and KG-based recommendation. The details of each one is given in the following sub-sections.

### 2.1. Neural collaborative filtering

In the literature, neural collaborative filtering (NCF) model are implemented into two types; one of which is by representation learning and other one is by matching function learning. The detail of these are given in the following sections.

*Representation learning based CF:* In the last decade, matrix factorization (MF) and its variants remained popular in the recommendation research community (Koren et al., 2009; Hu et al., 2014). In these studies, auxiliary information is incorporated e.g. time, location, text description etc. which helps the matrix factorization models predict the user preference. With the advancement of deep learning models in the recent years, researchers are trying to utilize deep neural networks to learn the user-item representation along with collaborative filtering. The authors in (Sedhain et al., 2015) utilize the autoencoder and propose AutoRec model which learn user and item representation using autoencoder. DMF (Xue et al., 2017) utilize neural network architecture which is two pathways in a sense that it factorizes the rating matrix and then the user-item vectorized representation is learned. Generally speaking, representation learning based models try to learn user-item representations in various ways and flexible enough to incorporate the auxiliary data into the representation learning.

*Matching function learning based CF:* In recent years, NCF (He et al., 2017) is proposed which combines MF and multi-layer perceptron (MLP) in one unified model. In NCF, dot product of MF is replaced with the neural network and thus the matching function between users and items is learned through neural network. Another variant of NCF is proposed which is NNCF (Bai et al., 2017) in which neighbors of users and items are given as input to the model. ConvNCF (He et al., 2018) is proposed which uses an outer product as the matching function, to encode the pairwise correlation between user and item. Wide&Deep (Cheng, 2016) works by adapting the supervised learning and MLP to learn the matching function and uses the user-item categorical features. Besides these studies, others have also utilized auxiliary information for learning the matching function. In this paper, our focus is to utilize KG as an auxiliary information in neural collaborative filtering approach.

### 2.2. Graph neural network

In recent years, graph neural network (GNN) (Zhou et al., 2020); (Gallicchio and Micheli, 2010) and (Li et al., 2015) which is basically the extension of convolutional neural network (CNN) is emerging in the field of recommender systems. As the number of nodes in the graph is not fixed and thus they are irregular in nature, therefore it makes no sense to utilize CNN where there is no fixed size matrix as in the case of images. Consequently, the concept of GNN in recommendation scenario is proposed in the literature. Moreover, in recommendation system, there are users and items so it can be seen as a user-item bipartite graph where the edge between user and item depicts the interaction between the two. Graph convolutional network (GCN) (Kipf and Welling, 2016) works by aggregating the information of neighboring nodes to enhance each target node's representation. There are two broad categories of GCN methods; 1) spectral and 2) spatial methods.Spectral methods perform operation into the transformed Fourier domain where graph features are transformed first and then the convolution operation is performed into the Fourier domain. The authors in (Bruna et al., 2013) perform the eigen decomposition into the Fourier domain by utilizing the spectral GCN. In (Defferrard et al., 2016), authors have approximated the convolution operation by utilizing the Chebyshey polynomials, which is an effort to decrease the computational complexity. In this paper, the proposed model falls in the spatial methods category, so our more focus is in spatial methods.

On the other hand, spatial methods perform the convolution operation directly on the graph in such a way that each node gets the information from its neighboring nodes to enrich its own representation (Atwood and Towsley, 2016); (Zhang et al., 2018). The authors in (Micheli, 2009) proposed the basic spatial method in which a given node gets information from all of its neighboring nodes and then it is summed up to make the enriched representation of that node. Afterwards, residual connection is used which at each layer to ensure that information from the previous layer is preserved. For each node, its neighboring nodes may vary in number and in the literature, it is tackled by the usage of sampling approach. In these sampling approaches, fixed number of neighboring nodes are sampled for each node followed by the aggregator which is used to aggregate the information obtained from the neighboring nodes (Hamilton et al., 2017). One of the limitations of these methods is that they only consider homogeneous graph in which user-item information is being captured.

Graph Attention Network (GAT) (Velickovic et al., 2017) is introduced as an approach to assign varying weights to each neighboring node depending on its importance. In other words, rather than treating each node's neighbor equally, it is rational to aggregate them as per their weight. This is also realistic in real-world scenario as not each item in the user's historical item contribute equally in predicting her preference. Moreover, usage of these attention approaches helps to deal with the variable sized input nodes (Sang et al., 2021). In recent years, the concept of sequential recommendation system is also proposed e.g. RetaGNN (Hsu et al., 2021) which recommends to each user the next item based on the user's last interacted item. IMP-GCN (Liu et al., 2021) address the over-smoothing problem in high order information aggregation and thus preserve the node's uniqueness during information aggregation. The limitation of IMP-GCN is that it does not focus on attention mechanism and treat each neighboring node equally which may not be appropriate in the real-world scenario. In recent years, the concept of explainable recommendation is also gaining attention which assists in enhancing the user satisfaction. KEGNN (Lyu et al., 2023) is one such recommendation model which recommend items to the user along with useful explanation. It utilizes the knowledge from the external database to encode different aspects of the information. Since the recommendation system often encounter the data sparsity problem which leads to weak or poor generalizability. To mitigate this problem, some researchers have introduced the concept of tag-aware recommendation system (Wang et al., 2022). It leverages the personalized tags so that the modeling of user preferences as well as of item characteristics can be enriched. KGIC (Zou et al., 2022) leverages a multi-level interactive contrastive learning mechanism to enhance the coherence and sufficiency of information utilization from both CF and KG. KGRec (Yang et al., 2023) is a self-supervised rationalization method for knowledge-aware recommender systems. It employs an attentive knowledge rationalization mechanism which generate rational scores for knowledge triplets, that supports in identifying informative knowledge connections.

### *2.3. Knowledge graph-based recommendation*

In literature, KG is being widely adopted in the recommendation system scenario due to their benefit of providing fruitful facts and reasoning about items. KG based recommendation is classified into three main categories; 1) path-based, 2) embedding-based, and 3) propagation-based recommendation. The detail of each category is given in the following sections.

Path- based methods works by designing the hand-crafted paths to infer the user preferences. To select from the multiple paths in KG, selective approach (Wang et al., 2019); (Sun et al., 2018) is proposed which select the most significant paths and enrich the entity representation. Meta-paths pattern is another path-based approach which limit the paths of KG. CGAT (Liu et al., 2021) is proposed which present the concept of biased random walk where gated recurrent unit is employed, to capture the non-local context (higher order neighbors) of the target node. Biased random walk strategy is repeated several times to explore useful entities for the given target node. One of the limitations of the path-based method is that they require domain knowledge to extract the useful paths and also this is labor intensive. This factor may make these methods inefficient to design paths as KG size may reach up to millions with multiple entities and relations among them.

In embedding based methods, each entity of KG is converted into low dimensional vectorized form (embedding) whereas the KG structural proximity is kept preserved. The learning of user and item latent representation is regularized by this embedding. CKE (Zhang et al., 2016) and DKN (Wang et al., 2018) works by generating the semantic embedding of the KG nodes using knowledge graph embedding (KGE) methods. Subsequently, to regularize the user and item representation, these embedding are given to the recommender system as input. KGCN (Wang et al., 2019) works by embedding the items in KG using graph neural network. To capture the collaborative signal, this item embedding encodes the information from the neighboring entities of the KG, and thus enhance their capability for the recommendation.

Propagation-based methods works by propagating information in an iterative fashion to capture the supplementary information for recommendation (Wang et al., 2023). A lot of research has been conducted in this approach. RippleNet (Wang et al., 2018) is proposed which examines the users' potential preferences by propagating along KG links, although the significance of connections is ill-defined in it. KGNN-LS (Wang et al., 2019) is based on the graph convolutional network (GCN) in such a way that entity's representation is enriched by acquiring information from the neighborhood. Nevertheless, both KGCN and KGNN-LS have not paid attention to capture the explicit collaborative signal latent in KG, and thus, lead to inadequate item embeddings. To enrich item embedding, KGAT (Wang et al., 2019) present the collaborative knowledge graph (CKG) where user-item interaction and KG are combined to perform recursive propagation using GCN. KGAT assumes that items in user-item bipartite graph as well as related entities in KG as homogeneous nodes rather than considering them into different latent spaces. Moreover, as the user interaction with different types of items have diversity, going deep into the propagation may have different interpretation than that of the actual item's representation, thus noise is being introduced into the learned embedding. CKAN (Wang et al., 2020) is based on heterogeneous propagation in which collaboration propagation and KG propagation are integrated in a natural way such that they contribute with varying weights to the embedding learning. One thing to highlight here is that CKAN does not consider the user-specific component in their model which essentially means that recommendation generated may not be particular to each separate user. GACF (Elahi and Halim, 2022) proposes the relation aware recommendation system having user-specific component, which generates the recommendations for each user separately while not ignoring the relational context into the information propagation step. KGCAN (Elahi et al., 2024) is proposed which captures both relational and contextual entity information so that the representation of original entity is preserved. Moreover, it also leverages the user-specific attention mechanism to provide the user personalized recommendations to each user.

## 3. Task Formulation

Before we dive into the details of proposed recommendation model, it is necessary to first understand the preliminaries of KG based recommendation system. We have two forms of information which are being incorporated to learn the user preferences on an item. One of these is user-item interaction and the other is KG information. These two forms of information are discussed in the following sections.

### *3.1. User-item interaction*

We have a set of users $U$ and a set of items $I$ in the typical recommendation scenario. The user interact with an item and this interaction is denoted by the link or edge in the bipartite graph e.g. user $u$ has liked an item $i$ or viewed an item and the link between the user and item reflects their interaction. We have a user-item interaction matrix $Y \in \mathbb{R}^{M \times N}$ which is constructed from these interactions (where $M$ and $N$ represents the number of users and items respectively in the matrix). Each entry $y_{ui}$ in the matrix $Y$ is either 0 or 1 representing the implicit feedback such as;

$$y_{ui} = \begin{cases} 1 \text{ if user-item interaction } (u, i) \\ \qquad 0 \text{ otherwise.} \end{cases}$$

### *3.2. KG information*

KG is utilized as the auxiliary information which helps to overcome the sparsity of user-item interaction matrix. KG $G$ is a directed graph which have head and tail entities linked via relation $r$. Thus, it is in the form of triple fact $(h, r, t)$ representing the relation $r$ between head $h$ and tail $t$ entity. It is important to note here is that KG supplements the deep facts and semantic rich information about items.

By utilizing the user-item interaction $Y$ and KG information $G$, our recommendation task is to determine the probability score that a given user $u$ would like to interact with the given item $i$ having no interaction before (unobserved interaction). More formally, this probability score is given as $\widehat{y}_{ui} = \mathscr{F}(u, i | Y, G, \Theta)$, having $\Theta$ as all the parameters of the model. All the notations used in this paper are presented along with their description in the Table 1.

## 4. Proposed framework

The goal of utilizing KG in the recommendation system is to alleviate the sparsity issue which is inherent in the user-item interaction.

**Table 1**
Notations along with their interpretations.

| Notation | Interpretation |
|---|---|
| $U$ | Set of users |
| $I$ | Set of items |
| $G$ | Knowledge graph |
| $(h, r, t)$ | (head, relation, tail) a triplet of knowledge graph |
| $Y$ | User-item interaction information |
| $\hat{y}_{ui}$ | Probability score |
| $p_u$ | Contextualized user representation |
| $q_i$ | Contextualized item representation |
| $\Theta$ | Neural parameters of the model |
| $H$ | Learning rate |
| $I_u^+$ | Historical items of user $u$ |
| $\mathscr{J}$ | Cross entropy loss |
| $\mathscr{L}_{RS}$ | Loss function |
| $P$ | Dropout probability |

However, KG is equipped with many fruitful facts and relations, thus, extracting the relevant information from the KG is the key thing in the recommendation scenario. For this reason, we propose IGAT, an end-to-end recommendation model which jointly exploits user-item interaction and KG to predict the user preferences on items. We now present the IGAT model and the framework which is shown in Fig. 2. The proposed model consists of three main components: 1) Knowledge propagation-based item modeling, 2) Contextualized attention-aware user modeling, and 3) Model prediction. The probability that a user would prefer an item is predicted from the user-item representations learned from the first two components.

### *4.1. Knowledge propagation-based item modeling*

Based on GCN architecture, where the representation of each node is enhanced by propagating along the links, we build on this to capture the neighbors' information in the KG. As the item behaves like a bridge between the user and KG, so we aim to model the item by propagating along KG. In this way, the fruitful facts are encoded in the item's representation, thus providing reasoning about the user preferences. The single GCN layer encodes the information of directly attached neighbors of the given node. Thus, stacking $L$-layers assist to capture the $L$-hops away information of neighbors. Unlike path-based methods, no manual feature engineering and designing of paths are required in propagation-based methods.

KG has diverse types of entities and relations among them, so each entity may have a different meaning. It may not be appropriate to incorporate neighboring entity information while ignoring its relation and type. For example, two movies may have too many common things in terms of the cast but they may have a different genre. To incorporate such information in KG propagation, we propose a knowledge propagation-based model which assigns different weights to the neighboring tail entities when encountering different head entities. Consider a KG triplet (h, r, t) having head entity $h$, tail entity $t$, and relation $r$ between them. The attention mechanism which assigns varying weights to tail (neighboring) entities is given as;

$$k_i = \sum_{T^i} \pi(e_h, r) e_t \tag{1}$$

Here, $e_h$ is the embedding of head entity, $e_t$ is the tail entity's embedding while r is the relation embedding. $\pi(e_h, r)$ is the decay factor that assigns varying weights to each tail entity. It is defined as given;

$$\pi(e_h, r) = \sigma(W_2 \mathrm{ReLU}(W_1(e_h \| r) + b_1) + b_2) \tag{2}$$

$W_1$, $W_2$ are trainable weight matrices whereas $b_1$, $b_2$ are biases. $\|$ represents concatenation operation to aggregate two embeddings. $\sigma$ represents the sigmoid activation function. To normalize the coefficients across all the triples, we have utilized the softmax function (Memisevic et al., 2010) given as;

$$\pi(e_h, r) = \frac{\exp[\pi(e_h, r)]}{\sum_{(h, \tilde{r}, \tilde{t} \in T^i)} \exp[\pi(e_h, \tilde{r})]} \tag{3}$$

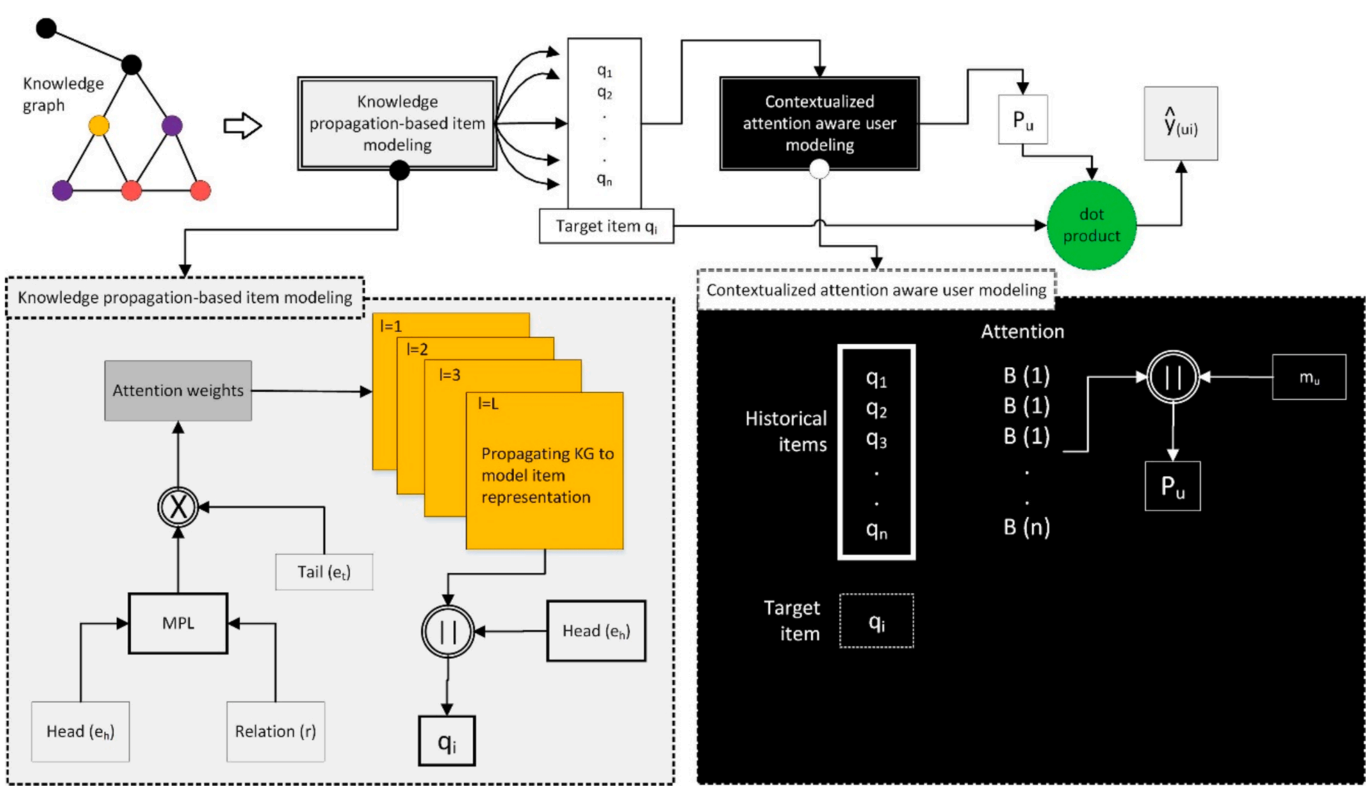


**Fig. 2.** Overall working of the proposed solution.

**Algorithm 1** IGAT Optimization

**Input:**
User-item interaction $Y$,
Knowledge graph $G$
**Output:**
Score function $\widehat{y}(u,i)$
All parameters are initialized using Xavier initializer
**for** epoch = 1 to max_epochs **do**
Sample of batch of size $b1$ from Y;
Sample of batch of size $b2$ from $G$;
Perform forward propagation to compute score $\widehat{y}(u,i)$;
Calculate the gradients using backpropagation of batch w.r.t $\Theta$;
Use Adam optimizer to update the parameters $\Theta$ having learning rate $\eta$;
**end for**
**return** $\widehat{y}(u,i)$

Hence, the generated attentive weight signifies the importance of relevant tail entities in the propagation step. By stacking L-layers, we can encode L-hops nodes information and enrich the item representation. The contextualized representation of the item is obtained by aggregating the original item representation $e_h$ with the encoded item representation from multiple layers. It is defined as follows;

$$q_i = e_h \| k_i \tag{4}$$

### 4.2. Contextualized attention-aware user modeling

For each user, her historical items are utilized to determine her potential preference (Shi et al., 2014). SVD++ model (Korean, 2008) is proposed which considers historical items of a user u as her implicit feedback and determines the significance of the user's historical items $I_u^+$ on the target item $i$. In the literature, most methods that utilize historical items for representing the user preference, do not consider the effect of the target item on her historical items. In this paper, an item-specific attention mechanism is proposed which assigns weights to each historical item based on its significance to the target item. Instead of treating each historical item equally, our item-specific attention mechanism assigns weight to each historical item $j$ of user $u$, based on their significance to the target item $i$. Given the user's historical item $I_u^+$, its embedding form is given as;

$$e_{I_u^+}^i = \sum_{j \in I_u^+} \beta(i,j) q_j \tag{5}$$

In equation (5), $\beta(i,j)$ factor controls the attentive weights and assigns importance to each historical item $j$ of the user. Formally it is given as;

$$\beta(i,j) = \frac{\exp\left[\tanh\left(\left(q_i \middle\| q_j\right) w^\top + b\right)\right]}{\sum_{k \in I_u^+} \exp[\tanh((q_i \| q_k) w^\top + b)]} \tag{6}$$

In equation (6), $q_i$ and $q_j$ are item $i$ and item $j$ representations, $w$ is the weight vector, while $b$ is the bias, and tanh is the non-linear activation function. $\|$ represents the concatenation operation which is used to aggregate $q_i$ and $q_j$. To construct the contextualized embedding representation of user u, we have concatenated the $\boldsymbol{e}_{I_u^+}$ and $\boldsymbol{e}_u$ (user embedding), followed by the non-linear transformation. Formally it is given as,

$$m_u = \mathrm{ReLU}\left[\left(e_u \middle\| e_{I_u^+}^i\right) W + b\right] \tag{7}$$

Here, $m_u$ is the contextualized embedding of user u and ReLU is the non-linear activation function. W and b are trainable weight matrix and bias respectively. Given $m_u$ and original user information $e_u$, the contextualized user representation is defined as;

$$p_u = \ (e_u \| m_u) \tag{8}$$

### 4.3. Model prediction

For item modeling, KG propagation is employed to learn item representation in a layered neural network architecture. Therefore, to aggregate the item representations from multiple layers, we have adopted three different aggregators as discussed in the following sections.

Sum aggregator: As the name suggests, it sums up item representations from multiple layers and then applies a non-linearity. Formally it is given as;

$$f_{sum}^i = \sigma\left(W \cdot \sum_{q_i \in R_i} q_i + b\right) \tag{9}$$

Where $\sigma$ represents the Sigmoid activation function, $W$ and $b$ are the weight matrix and bias vector respectively. $R_i$ is the set containing the item representations from multiple layers.

Pooling aggregator: In this aggregator, the maximum value is chosen from the representation set, and then a non-linear transformation is applied.

$$f_{\mathrm{pooling}}^i = \sigma(\mathrm{W} \cdot \mathrm{pool}_{max}(R_i) + \mathrm{b}) \tag{10}$$

Concatenation aggregator: It concatenates all the representations in the representation set and forms the single vector representation capturing information of all the vectors. It is given as;

$$f_{\mathrm{concatenation}}^i = \sigma\left(\mathrm{W} \cdot \left(\mathrm{q}_i^{(i_1)} \middle\| \cdots \middle\| \mathrm{q}_i^{(i_n)}\right) + \mathrm{b}\right) \tag{11}$$

Afterward, having aggregated representations of items and contextualized user representation, we apply the inner product of $p_u$ and $q_i$ to compute the probability score given as;

$$\widehat{y}(u,i) = \mathrm{p}_u^\top \mathrm{q}_i \tag{12}$$

$\widehat{y}(u,i)$ is the probability score that determines if a user $u$ would like to interact with an item $i$ with which she has not interacted previously.

### 4.4. Model training

In deep learning, model is trained on the given training data in the form of batches. After one epoch, all the training data complete its one pass and then model is given the data again for another epoch. In this way, model is trained and loss is calculated at each epoch, whereas model's parameters (weights) are updated to minimize the loss. The loss is being calculated using a loss function i.e. cross entropy loss (Zhang and Sabuncu, 2018) which measures the difference between the predicted and the actual score.

Now, we have positive interactions as well as negative interactions in the recommendation scenario. Positive interaction of a user $u$ describes those items with which user has interacted before. On the same ground, negative interactions of user u indicate those items with which user has not interacted before rather they are generated, for each user, by random sampling from the un-interacted or un-observed item list. To determine the effect of model training as well as to have a balance ratio, we consider, for each user, equal number of negative interactions as that of positive interactions. Cross entropy loss $\mathscr{J}$ is utilized in this study and it is given as;

$$\mathscr{J} = \frac{1}{N} \sum_{i=1}^{N} \quad (y_i * \log(p_i) + (1 \quad y_i) * \log(1 \quad p_i)) \tag{13}$$

Since the predicted score is basically the probability which is always within the range [0, 1] and log function in this range have negative values. That is why, to make loss values positive, it is multiplied with ( 1) as shown in the equation (13). Hence, the loss function of our recommendation system is as follows;

$$\mathscr{L}_{RS} = \sum_{u \in \mathscr{U}} \left( \sum_{(u,\ i) \in I^{+}} \mathscr{J}\left(y_{(u,\ i)}, \widehat{y}_{(u,\ i)}\right) \quad \sum_{(u,\ j) \in I} \mathscr{J}\left(y_{(u,j)}, \widehat{y}_{(u,j)}\right) \right) \tag{14}$$

Here, $I^{+}$ indicates the positive interactions with which user has interacted whereas $I$ indicates negative interactions obtained by negative sampling for each user. Thus, the objective function that learns and updates the model's parameters to minimize the loss function is given by;

$$\min_{\Theta} \mathscr{L}_{RS} + \lambda \left\|\Theta\right\|_2^2$$

Where $\Theta$ denotes all the parameters of the model and $\|\Theta\|_2^2$ indicates the L2-regularizer that is parameterized by $\lambda$. The given objective function is minimized using Adam (Kingma and Ba, 2014) optimizer. Algorithm 1 explains how the optimization is achieved in our model IGAT.

## 5. Experiments

In this section, we have conducted experiments to evaluate the proposed model on three benchmark datasets. The results obtained from the experiments answer the given research questions:

***RQ-1***: Whether the proposed IGAT perform better as compared to the state-of-the-art KG-based models on benchmark datasets?

***RQ-2***: How do different modules of IGAT influence the original model performance?

***RQ-3***: Whether the item-specific attention module models the influence of the target item on the user's historical items?

***RQ-4***: How do different settings of hyper-parameters (e.g. embedding size, depth of layer, and aggregation function) impact the proposed IGAT performance?

### 5.1. Datasets

In this paper, we have utilized three publicly available benchmark datasets for the experiment purpose. These benchmark datasets belong to different domains, the detail of which is given briefly as follows;

- **MovieLens-20 M**: This dataset is being widely used in movie recommendation and it comprises of almost 20 million ratings given by more than 138 thousand users. These ratings are given on the scale of 1 to 5.
- **Last.FM**: This dataset comprises of almost 2 thousand users whose music track count information is recorded. It is provided by Last.FM music system.
- **Book-Crossing**: This comprises of more than 17 thousand user's ratings on the scale of 0 to 10. Here, books are treated as items and it is provided by book crossing community.

In MovieLens-20 M,[1] Last.FM[2] and Book-Crossing[3] datasets (denoted by ML, FM and BC respectively), interactions are given as explicit feedback, which are converted into implicit feedback. In case of implicit feedback, 1 represents the positive interaction which essentially means that user have interacted with that item. Given the historical information of each user, her un-interacted items are randomly sampled from her historical information to have negative interactions of that user. The size of negative interactions is kept equal to the size of positive interactions for each user, so that the effect of biasedness is reduced. Since ML dataset is quite large in size having almost 20 million ratings, we have only considered those ratings which are greater than 4 as positive. FM and BC datasets are sparse in nature, so no threshold is considered in FM and BC. As the proposed recommendation model is utilizing user-item interactions as well as KG, so KG of these benchmark datasets are obtained from the public repository https://github.com/xiangwang1223. Each dataset has its own KG and given its whole KG, its sub-KG by considering those KG triples whose confidence level is greater than 0.9. Furthermore, we have removed those entities which have a match with other entities or items, to ensure the consistency. The statistics of these benchmark datasets are presented in the Table 2.

### 5.2. Baselines

NCF (He et al., 2017): It is collaborative filtering approach where neural network architecture is utilized to learn the user preferences on the given item. In this way, data is used to learn the matching function and the usage of neural network replaces the inner product of user and item.

RippleNet (Wang et al., 2018): It is embedding-based model which uses KG information to propagate and aggregate the user's potential preferences on the items.

KGCN (Wang et al., 2019): It is state of the art model based on non-spectral GCN approach having KG into the context. KG structural information is learnt through aggregation of neighborhood entities, thus encoding the neighborhood context for the given entity.

KGAT (Wang et al., 2019): It is the KG enhanced recommendation model which utilizes the attention network to discriminate the neighboring nodes in collaborative KG.

CKAN (Wang et al., 2020): This is propagation-based recommendation model that utilizes the relational attentive mechanism to discriminate the neighboring entities based on relational information.

GACF (Elahi and Halim, 2022): It is propagation-based model that utilizes user-specific attention mechanism for the user's personalized recommendation. It works by making user and item triple sets which propagates into the KG and thus enrich their representation.

### 5.3. Experimental settings

In the experiments, each dataset is distributed into training, validation, and testing sets. Firstly, the training set is chosen randomly from the whole data with a ratio of 60 %. Then, 20 % of data is chosen from the remaining data as the validation set, followed by the 20 % remaining data as the testing set. This distribution is adopted due to its wide acceptance in the recent literature (Elahi and Halim, 2022); (Liu et al., 2021). For the evaluation of IGAT, click-through rate (CTR) prediction is utilized in which the model's performance is measured through AUC and F1 scores. The CTR prediction score is determined by first using a training set to train the model, and afterward, the testing set is used to predict the probability score that how likely a user would interact with an item having no previous interaction. Xavier initializer (X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, 2010) is employed which initializes all the model's parameters. To optimize the model during training, Adam (Kingma and Ba,

**Table 2**
Datasets used along with their description.

| | ML | FM | BC |
|---|---|---|---|
| # Users | 138,159 | 1872 | 17,860 |
| # Items | 16,954 | 3846 | 14,967 |
| # Interactions | 13,501,622 | 42,346 | 139,746 |
| # Avg- interactions | 98 | 23 | 8 |
| Entities | 102,569 | 9,366 | 77,903 |
| Relations | 32 | 60 | 25 |
| Triples | 499,474 | 15,518 | 151,500 |

1 https://grouplens.org/datasets/movielens/20m/.
2 https://grouplens.org/datasets/hetrec-2011/.
3 https://www2.informatik.uni-freiburg.de/~cziegler/BX/.

2014) optimizer is used and the batch size of 1024 is selected. To prevent the model from overfitting, dropout technique is employed which randomly discards the influence of user and item in the training process. Here, node dropout technique is used which randomly discards the nodes (user and item) with $\rho$ probability, whereas $\rho$ is empirically chosen as 0.2.

IGAT is implemented in Pytorch,[4] a deep learning framework, and different hyperparameters settings are selected by the means of grid search. The learning rate we adopted is selected from $\{10^{\;3}, 4 \times 10^{\;3}, 10^{\;2}, 4 \times 10^{\;2}\}$ while embedding size is selected amongst {4, 8, 16, 32, 64, 128}. The dropout technique is also utilized to reduce the model's overfitting, and it is tuned amongst {0.0, 0.1, 0.2, … 0.6}. For item modeling, a propagation-based attention mechanism is utilized in which the optimal depth of layers is different for different datasets (it is reported in the upcoming results section). We kept the dimensions of both, the embedding and hidden layer equal. Moreover, the open source implementation of the comparison methods (baselines) is utilized to assess the IGAT performance on the publicly available datasets. The selection of hyper-parameters of the baselines is done either empirically or as done by their authors.

### 5.4. Comparison with baselines (RQ-1)

In this section, we have described the experimental results of the proposed IGAT model along with the state-of-the-art baselines. Table 3 states the experimental results where best performance is highlighted with bold face whereas second best is underlined. From the Table 3, we have made the following observations.

- On all datasets, IGAT significantly outperforms in comparison to state-of-the-art baselines with $p < 0.5$ by adopting Wilcoxon signed rank statistical test (Shani and Gunawardana, 2011). This highlights the importance of item specific attention mechanism where the influence of target item on historical items of the user is considered. Moreover, contextual representation of the user and the item also play the role as bias is reduced due to the multi-layer architecture. Formally, IGAT shows 0.61 %, 1.97 % and 1.84 % increase in ML, FM and BC dataset respectively w.r.t AUC in comparison to the second-best model as underlined in the Table 3.
- GACF depicts best performance w.r.t AUC from baselines, which implies the significance of utilizing semantic rich attention-aware entity's representation and hence, assigning varying importance to the neighbors of given entity. Moreover, GACF is expressive in nature as it models the latent relation between user and item in an explicit way. This highlights how incorporating collaborative signal in an explicit manner is significant to enhance the model performance.
- Among baselines, NCF consistently show worst performance w.r.t AUC in all the datasets. One possible reason of it is due to the lack of model's expressiveness in capturing the latent collaborative signal between the user-item interactions. Modelling the latent collaborative signal into the recommendation scenario gives more expressiveness to the model as well as to reason about the possible interacting items of the user's interests. This also verifies the importance of KG into the recommendation system as it provides additional fruitful facts and reasoning to the model which in turn helps to encode the useful information about the user and the item.
- In BC dataset, KGCN depicts worst performance w.r.t AUC and second worse in ML dataset. As KGCN enrich the entity representation by sampling from the receptive field which extends as layers are increased, thus more noisy entities may also introduce into the representation learning of an entity. Moreover, treating all the neighboring entities in KG equally is not rational due to the different types of entities, each one having its own semantics. This highlights the significance of attention mechanism which assign weights to the neighboring entities based on their importance.
- Among the benchmark datasets, one thing to note here is that ML dataset depicts best performance i.e. more than 96 % w.r.t AUC as compared to other datasets which shows up to 86 % and 77 % performance w.r.t AUC in FM and BC datasets respectively. This is due to the reason that ML is large dataset as compared to FM and BC having large number of user interactions and links per user. In this way, more information ML contains in itself and less information is being learnt from the latent entity's embedding.
- Compared with KGAT, CKAN shows best performance w.r.t AUC by 0.21 %, 2.43 % and 3.88 % increase in ML, FM and BC dataset respectively. This verifies the significance of incorporating the heterogeneous propagation where user-item interaction and KG are treated in different spaces i.e. items and entities are not treated homogeneously rather they are treated in separate spaces and are integrated in a natural manner. However, when CKAN is compared with proposed IGAT, CKAN depicts low performance w.r.t AUC in all datasets. The possible reason is due to the attention mechanism in CKAN as both the user and item are enriched with same attention mechanism thus less expressiveness the model contains.

**Table 3**
Experimental results of IGAT and the baselines w.r.t AUC and F1 score.

| Dataset | Model | AUC | F1 score |
|---|---|---|---|
| ML | NCF | 0.967 | 0.916 |
| | Ripple Net | 0.970 | 0.919 |
| | KGCN | 0.968 | 0.922 |
| | KGAT | 0.974 | 0.923 |
| | CKAN | 0.976 | 0.923 |
| | GACF | 0.983 | 0.925 |
| | **IGAT** | **0.989** | **0.927** |
| FM | NCF | 0.759 | 0.701 |
| | Ripple Net | 0.768 | 0.703 |
| | KGCN | 0.792 | 0.706 |
| | KGAT | 0.823 | 0.740 |
| | CKAN | 0.843 | 0.757 |
| | GACF | 0.861 | **0.787** |
| | **IGAT** | **0.878** | 0.785 |
| BC | NCF | 0.712 | 0.629 |
| | Ripple Net | 0.715 | 0.640 |
| | KGCN | 0.682 | 0.631 |
| | KGAT | 0.722 | 0.652 |
| | CKAN | 0.750 | 0.668 |
| | GACF | 0.762 | 0.683 |
| | **IGAT** | **0.776** | **0.691** |

### 5.5. Ablation study (RQ-2)

In this paper, the contribution of different components of IGAT is assessed by their removal from the original base model and making its variants. We have conducted experiments to verify their effectiveness, also the experimental results are reported in the Table 4 which make a comparison of original IGAT with that of its variants. Two variants of IGAT are given follows:

$\underline{\text{IGAT}_{/\text{contextual}}}$: Here, we have removed (disabled) the contextualized representation of user as well as item. For user, we consider the user

**Table 4**
Ablation study to compare IGAT with its variants (in terms of AUC).

| | $\text{IGAT}_{/\text{contextual}}$ | $\text{IGAT}_{/\text{attention}}$ | IGAT |
|---|---|---|---|
| ML | 0.971 | 0.978 | **0.989** |
| FM | 0.823 | 0.842 | **0.876** |
| BC | 0.748 | 0.757 | **0.774** |

[4] https://pytorch.org/.

representation in equation (7), and for item, we consider the item representation in equation (1), which is encoded representation from multiple layers. Through experiments, the performance of this variant is compared with original IGAT.

$\underline{\text{IGAT}_{/\text{attention}}}$: The attention mechanism which is utilized in IGAT for discriminating the importance of different neighboring nodes is disabled in this variant. More precisely, we took the average of the neighboring nodes and compare it with original IGAT.

In the Table 4, it can be seen that performance of original IGAT is superior to that of its both variants, which make it clear that contextual representation of the user and that of the item is necessary to be considered, as it plays significance role in boosting its performance. The following findings are drawn from the Table 4.

- $\text{IGAT}/_{\text{contextual}}$ is compared with IGAT and experimental results in Table 4 show the effectiveness of context-aware representation of user and item in the recommendation scenario. Since the uniqueness of each user and item is preserved as well as the bias due to the information capturing from large receptive field is reduced. Therefore, contextualized representation in IGAT show superior performance when compared with $\text{IGAT}/_{\text{contextual}}$.
- Since KG is heterogenous graph having multiple relations and entities, so it may not be rational to treat all the neighboring entities equally. Rather attention mechanism needs to be considered to avoid the ignorance of heterogeneity structure during aggregation. That is why, it can be seen that IGAT is superior in performance as compared to its variant $\text{IGAT}/_{\text{attention}}$. Moreover, as the number of neighboring entities for the given target node may vary in KG, therefore, attention mechanism is suitable in treating the varying number of neighboring nodes.

### 5.6. *Significance of item specific component (RQ-3)*

In the proposed model, we have modeled the influence of target item on user's historical items and thus, the user representation is learnt based on the significance of target item on the user's historical items (Fareed et al., 2024; Fareed et al., 2023; Saadat et al., 2024). In this way, each target item is not treated in the same manner and this discrimination helps in modeling the latent significance of each target item. For this purpose, experiments are conducted to determine its effect on the model performance on each dataset.

Although GAT is effective in modeling user-item interactions, they fail to adequately capture the dynamic influence of the target item on a user's historical items, which is essential for personalized recommendations. To address this gap, we introduce an item-specific attention layer. This layer enables the model to weigh historical items differently, depending on their relevance to the target item, thus providing a more refined and context-aware representation of user preferences. By incorporating this enhancement, the performance of recommendation model is improved, as it better captures the subtle interactions between users and items, a capability that GAT alone does not fully exploit.

The experimental results are reported and compared with original IGAT in the Table 5. It can be seen from the Table 5 that the item specific component is significant in modeling the discriminating effect of the target item on historical items of the user. By using item specific component, the performance is increased 1.54 % in ML, 1.74 % in FM, while 3.06 % in BC dataset. Hence the effectiveness of item specific component in IGAT is experimentally established.

**Table 5**
Performance of item specific component in IGAT (in terms of AUC).

| | Without item-specific | With item-specific |
|---|---|---|
| ML | 0.974 | **0.989** |
| FM | 0.861 | **0.876** |
| BC | 0.751 | **0.774** |

### 5.7. *Hyper-parameters study (RQ-4)*

To determine the impact of different hyper-parameters on the IGAT performance, extensive experiments have been performed. In the following sections, experimental results as well as their possible discussion is given.

#### 5.7.1. *Impact of aggregation function*

Through experiments, the impact of aggregation function on the performance of IGAT is assessed and the results are reported in the Table 6. It is clear from the Table 6 that concatenation aggregator is outperforming the other two aggregators, as highlighted by the bold face. One possible reason of it could be that the concatenation aggregator has the property of retaining and encoding more information in the entity embedding as compared to sum and pool aggregator. That is why, in all the datasets, concatenation aggregator is performing better when compared with the other two. Moreover, in the case of the sum and the pool aggregator, single value is generated from the aggregation of multiple layers.

#### 5.7.2. *Impact of layer's depth*

To evaluate the model's performance on varying depth of layers, experiments have been conducted on different datasets. Table 7 reports the experimental results on each dataset w.r.t AUC where the bold face entries depicts the best performance. Since each dataset belongs to a different domain, and thus, each one has different number of interactions per user. Therefore, the impact of depth of layer has different influence on these datasets. From the Table 7, it is clear that ML dataset, which is a large dataset having large number of interactions per user as compared to FM and BC, shows best performance on single layer neural network. In case of FM and BC dataset, best performance is reported when three-layer architecture is utilized. One possible explanation of it could be that ML dataset is so rich that little information is encoded from the neighboring entities during representation learning, rather neighboring entities are introducing noise as we increase the depth of layers. One thing to highlight here is that as we increase the depth of layers, the computational cost of the model also increases, whereas the performance decreases after reaching to a certain limit.

#### 5.7.3. *Impact of dimensions of embedding*

In our work, the dimensional parameters of both entity (head and tail) as well as relation embedding is same to avoid the bias which may inject into the different dimensional parameters. Table 8 reports the experimental results w.r.t AUC where the impact of different dimensions of embedding is recorded on different datasets. Table 8 demonstrate that by increasing the embedding dimension from 8 to 64, the performance increases. As we go from embedding dimension 64 to 128, the performance slightly decreases in case of ML and BC dataset. This decrease in performance is due to the model overfit where more information is being encoded than the model's capacity. Thus, the performance of our model slightly decreases. One thing to highlight here is that IGAT is robust enough to endure the varying dimensions of embedding as the minor change is observed in the performance. This indicate that the proposed IGAT is less reliant on different settings of hyper-parameters.

## 6. Conclusion and future work

This work presented an end-to-end recommendation model, Item-

**Table 6**
Impact of aggregation function on IGAT performance (w.r.t AUC).

| | Pool | Sum | Concatenation |
|---|---|---|---|
| ML | 0.971 | 0.983 | **0.989** |
| FM | 0.839 | 0.856 | **0.878** |
| BC | 0.747 | 0.761 | **0.776** |

**Table 7**
Impact of depth of layers (w.r.t AUC).

| No. of layers | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| ML | **0.989** | 0.976 | 0.967 | 0.959 |
| FM | 0.849 | 0.864 | **0.877** | 0.869 |
| BC | 0.762 | 0.768 | **0.775** | 0.769 |

**Table 8**
Impact of varying dimensions of embedding (w.r.t AUC).

| *d* | 8 | 16 | 32 | 64 | 128 |
|---|---|---|---|---|---|
| ML | 0.967 | 0.978 | 0.984 | **0.988** | 0.979 |
| FM | 0.862 | 0.869 | 0.874 | 0.876 | **0.879** |
| BC | 0.757 | 0.763 | 0.770 | **0.776** | 0.771 |

Specific Graph Attention Network (IGAT), which effectively integrated user-item interaction data with the KG information to encode user preferences on items. IGAT leverages a knowledge-aware attention mechanism to assign different weights to neighboring entities in the KG based on diverse relationships, thus enhancing latent vector representation. Furthermore, the item-specific attention mechanism models the influence of the target item on the user's historical items, contributing towards more accurate recommendations. To preserve the integrity of original representations and minimize propagation bias, the current proposal incorporated contextualized representations of both users and items within the model. All model parameters were jointly trained to capture the latent collaborative signals between the users and items. Through extensive experiments on three benchmark datasets, we demonstrated that the proposed IGAT approach outperforms several state-of-the-art KG-based recommendation models, indicating its effectiveness and robustness. The experimental results suggested that IGAT offers a promising approach for leveraging knowledge graphs in recommendation tasks, also it demonstrates that the IGAT outperforms the baselines.

In the future, we aim to develop an effective approach for extracting relevant higher-order neighbors while minimizing the inclusion of noisy entities. Additionally, we plan to design an aggregation function capable of naturally encoding both KG information and user-item interactions, with the added benefit of reducing model training time. Another potential research direction involves devising a random walk sampling method to efficiently extract relevant higher-order neighbors for aggregation.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.


## Acknowledgements

The authors would like to thank GIK Institute for providing research facilities. This work was supported by the GIK Institute graduate program research fund under the GA-F scheme.

### *Funding*

This work was sponsored by the GIK Institute graduate research fund under GA-F scheme.


### *Ethical approval*

No procedures performed in studies involved human participants.

## Data availability

Data will be made available on request.

## References


Shi, Y., Larson, M., & Hanjalic, A. (2014). Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys (CSUR), 47*(1), 1–45.
Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *IEEE Comput., 42*(8), 30–37.
He, X., Liao, L., Zhang, H., Nie, L., Hu, X., & Chua, T. S. (2017). Neural collaborative filtering, in. In *Proceedings of the 26th international conference on World Wide Web* (pp. 173–182).
Tay, Y., Tuan, L. A., & Hui, S. C. (2018). Latent relational metric learning via memory-based attention for collaborative ranking, in. In *Proceedings of the 2018 World Wide Web Conference* (pp. 729–739).
Wang, X., Wang, D., Xu, C., He, X., Cao, Y., & Chua, T. S. (2019). Explainable reasoning over knowledge graphs for recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence, 33*(1), 5329–5336.
Shi, C., Zhang, Z., Luo, P., Yu, P. S., Yue, Y., & Wu, B. (2015). Semantic path based personalized recommendation on weighted heterogeneous information networks. In *in: Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (pp. 453–462).
Hu, B., Shi, C., Zhao, W. X., & Yu, P. S. (2018). Leveraging meta-path based context for top-n recommendation with a neural co-attention model, in. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 1531–1540).
Z. Wang, G. Lin, H. Tan, Q. Chen, X. Liu, 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, 2020. pp. 219-228.
Wang, H., Zhang, F., Zhang, M., Leskovec, J., Zhao, M., Li, W., & Wang, Z. (2019). Knowledge-aware graph neural networks with label smoothness regularization for recommender systems. In *in: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 968–977).
Y. Dong, N.V. Chawla, A. Swami, metapath2vec: Scalable representation learning for heterogeneous networks, in: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining, 2017. pp. 135-144.
Korean, Y. (2008). Factorization meets the neighborhood: A multifaceted collaborative filtering model, in. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 426–434).
Hu, L., Sun, A., & Liu, Y. (2014). Your neighbors affect your ratings: On geographical neighborhood influence to rating prediction, in. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval* (pp. 345–354).
S. Sedhain, A. K. Menon, S. Sanner, L. Xie, Autorec: Autoencoders meet collaborative filtering, in: Proceedings of the 24th international conference on World Wide Web, 2015. pp. 111–112.
Xue, H. J., Dai, X., Zhang, J., Huang, S., & Chen, J. (2017). Deep matrix factorization models for recommender systems. *IJCAI, 17*, 3203–3209.
Bai, T., Wen, J. R., Zhang, J., & Zhao, W. X. (2017). A neural collaborative filtering model with interaction-based neighborhood, in. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (pp. 1979–1982).
He, X., Du, X., Wang, X., Tian, F., Tang, J., & Chua, T. S. (2018). Outer product-based neural collaborative filtering. *ArXiv e-prints*.
Cheng, H. T., et al. (2016). Wide & deep learning for recommender systems, in. In *Proceedings of the 1st workshop on deep learning for recommender systems* (pp. 7–10).
Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., & Sun, M. (2020). Graph neural networks: A review of methods and applications, in. *AI Open, 1*, 57–81.
C. Gallicchio, A. Micheli, Graph echo state networks, in: 2010 international joint conference on neural networks (IJCNN), 2010. pp. 1–8.
Li, Y., Tarlow, D., Brockschmidt, M., & Zemel, R. (2015). Gated graph sequence neural networks. *ArXiv e-prints*.
Kipf, T. N., & Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *ArXiv e-prints*.
Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2013). Spectral networks and locally connected networks on graphs. *ArXiv e-prints*.
M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in neural information processing systems, 29, 2016.
J. Atwood, D. Towsley, Diffusion-convolutional neural networks, in: Advances in Neural Information Processing Systems, 29, 2016.
Zhang, J., Shi, X., Xie, J., Ma, H., King, I., & Yeung, D. Y. (2018). Gaan: Gated attention networks for learning on large and spatiotemporal graphs. *ArXiv e-prints*.
Micheli, A. (2009). Neural network for graphs: A contextual constructive approach. *IEEE Trans. Neural Netw., 20*(3), 498–511.
Hamilton, W. L., Ying, R., & Leskovec, J. (2017). Inductive representation learning on large graphs. In *in: Proceedings of the 31st International Conference on Neural Information Processing Systems* (pp. 1025–1035).
Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., & Bengio, Y. (2017). *Graph attention networks, ArXiv e-prints*.
L. Sang, M. Xu, S. Qian, X. Wu, Knowledge graph enhanced neural collaborative recommendation, in: Expert Systems with Applications, vol. 164, 2021.

C. Hsu, C.T. Li, RetaGNN: Relational temporal attentive graph neural networks for holistic sequential recommendation, in: Proceedings of the Web Conference, 2021. pp. 2968-2979.

Liu, F., Cheng, Z., Zhu, L., Gao, Z., & Nie, L. (2021). Interest-aware message-passing gcn for recommendation, in. *Proceedings of the Web Conference*, 1296–1305.

Z. Sun, J. Yang, J. Zhang, A. Bozzon, L.K. Huang, C. Xu, Recurrent Knowledge Graph Embedding for Effective Recommendation, in: Proceedings of the 12th ACM Conference on Recommender Systems, 2018. pp. 297-305.

Liu, Y., Yang, S., Xu, Y., Miao, C., Wu, M., & Zhang, J. (2021). Contextualized graph attention network for recommendation with item knowledge graph, in. *IEEE Trans. Knowl. Data Eng.*

Zhang, F., Yuan, N. J., Lian, D., Xie, X., & Ma, W. Y. (2016). Collaborative knowledge base embedding for recommender systems, in. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 353–362).

H. Wang, F. Zhang, X. Xie, M. Guo, DKN: Deep knowledge-aware network for news recommendation, in: Proceedings of the 2018 world wide web conference, 2018, pp. 1835-1844.

Wang, H., Zhao, M., Xie, X., Li, W., & Guo, M. (2019). Knowledge graph convolutional networks for recommender systems. In *in: Proceedings of the World Wide Web Conference* (pp. 3307–3313).

H. Wang, F. Zhang, J. Wang, M. Zhao, W. Li, X. Xie, M. Guo, Ripplenet: Propagating user preferences on the knowledge graph for recommender systems, in: Proceedings of the 27th ACM International Conference on Information and Knowledge Management, 2018. pp. 417-426.

X. Wang, X. He, Y. Cao, M. Liu, T.S. Chua, Kgat: Knowledge graph attention network for recommendation, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019. pp. 950-958.

Elahi, E., & Halim, Z. (2022). Graph attention-based collaborative filtering for user-specific recommender system using knowledge graph and deep neural networks. *Knowl. Inf. Syst., 64*(9), 2457–2480.

R. Memisevic, C. Zach, M. Pollefeys, G.E. Hinton, Gated softmax classification, in: Advances in neural information processing systems, vol. 23, 2010. pp. 1603-1611.

Z. Zhang, M. Sabuncu, Generalized cross entropy loss for training deep neural networks with noisy labels, in: Advances in Neural Information Processing Systems, 31, 2018.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv e-prints*.

X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the thirteenth international conference on artificial intelligence and statistics, JMLR Workshop and Conference Proceedings, 2010. pp. 249-256.

G. Shani, A. Gunawardana, Evaluating recommendation systems, in: Recommender systems handbook Springer, Boston, MA, 2011. pp. 257-297.

Lyu, Z., Wu, Y., Lai, J., Yang, M., Li, C., & Zhou, W. (2023). Knowledge Enhanced Graph Neural Networks for Explainable Recommendation. *IEEE Trans. Knowl. Data Eng., 35* (05), 4954–4968.

B. Wang, H. Xu, C. Li, Y. Li, & M. Wang, TKGAT: Graph attention network for knowledge-enhanced tag-aware recommendation system in: Knowledge-Based Systems, 2022.

Wang, F., Zheng, Z., Zhang, Y., Li, Y., Yang, K., & Zhu, C. (2023). *To see further: Knowledge graph-aware deep graph convolutional network for recommender systems*. in: Information Sciences.

Elahi, E., Anwar, S., Shah, B., Halim, Z., Ullah, A., Rida, I., & Waqas, M. (2024). Knowledge Graph Enhanced Contextualized Attention-Based Network for Responsible User-Specific Recommendation, in. *ACM Trans. Intell. Syst. Technol.*

Zou, D., Wei, W., Wang, Z., Mao, X.-L., Zhu, F., Fang, R., & Chen, D. (2022). Improving Knowledge-aware Recommendation with Multi-level Interactive Contrastive Learning. In *in: ACM International Conference on Information and Knowledge Management* (pp. 2817–2826).

Y. Yang, C. Huang, L. Xia, & C. Huang, Knowledge Graph Self-Supervised Rationalization for Recommendation, in: ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'2023), 2023. pp. 3046-3056.

Fareed, A., Hassan, S., Belhaouari, S. B., & Halim, Z. (2024). Elevating recommender systems: Cutting-edge transfer learning and embedding solutions, in. *Appl. Soft Comput., 166*, Article 112140.

Fareed, A., Hassan, S., Belhaouari, S. B., & Halim, Z. (2023). A collaborative filtering recommendation framework utilizing social networks, in. *Machine Learning with Applications, 14*, Article 100495.

Saadat, H., Shah, B., Halim, Z., & Anwar, S. (2024). Knowledge Graph-Based Convolutional Network Coupled With Sentiment Analysis Towards Enhanced Drug Recommendation. *IEEE/ACM Trans. Comput. Biol. Bioinf., 21*(4), 983–994.