Expert Systems With Applications 212 (2023) 118806

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Learning knowledge graph embedding with a dual-attention embedding network

Haichuan Fang, Youwei Wang, Zhen Tian *, Yangdong Ye *

School of Computer and Artificial Intelligence, Zhengzhou University, Zhengzhou, 450001, China

ARTICLE INFO

Keywords:
Knowledge graph embedding
Knowledge graph
Graph convolutional network
Representation learning
Attention mechanism

ABSTRACT

Knowledge Graph Embedding (KGE) aims to retain the intrinsic structural information of knowledge graphs (KGs) via representation learning, which is critical for various downstream tasks including personalized recommendations, intelligent search, and relation extraction. The graph convolutional network (GCN), due to its remarkable performance in modeling graph data, has recently been studied extensively in the KGE field. However, when learning entity representations, most attention-based GCN approaches treat neighborhoods as a whole to measure their importance without considering the direction information of relations. Additionally, these approaches make relation representations perform self-update via a learnable matrix, resulting in ignoring the impact of neighborhood information on representation learning of relations. To this end, this study presents an innovative framework, namely learning knowledge graph embedding with a dual-attention embedding network (D-AEN), to jointly propagate and update the representations of both relations and entities via fusing neighborhood information. Here the dual attentions consist of a bidirectional attention mechanism and a relation-specific attention mechanism for jointly measuring the importance of neighborhoods in respectively learning entity and relation representations. Thus D-AEN enables elements like relations and entities to interact well semantically, which makes their learned representations retain more effective information of KGs. Extensive experimental results on three standard link prediction datasets demonstrate the superiority of D-AEN over several state-of-the-art approaches.

1. Introduction

Knowledge graphs (KGs) play a crucial role in various knowledge-driven intelligent applications including question answering (Hu, Zou, Yu, Wang, & Zhao, 2017; Huang, Zhang, Li, & Li, 2019), recommendation systems (Rosa, Schwartz, Ruggiero, & Rodríguez, 2018; Shao, Li, & Bian, 2021; Wang et al., 2019), information retrieval (Chen, Tu, Lv, & Chen, 2018; Li, Li, Shang, & Shen, 2019), etc. During the past decades, numerous types of KGs have been developed for facilitating these applications, such as NELL (Carlson et al., 2010), Freebase (Bollacker, Evans, Paritosh, Sturge, & Taylor, 2008), DBpedia (Auer et al., 2007), and YAGO3 (Mahdisoltani, Biega, & Suchanek, 2014). The knowledge information with multiple relations is currently stored in KGs as directed graphs whose edges and nodes indicate relations and entities respectively. Indeed, KGs are usually represented as multiple knowledge triples (i.e., facts) like (h, r, t), which denotes as a head entity h, a tail entity t, and a relation r connecting them, e.g., (*New York*, *located_in*, *The United States*).

Despite large amounts of relations, entities, and triples, KGs may still suffer from incompleteness with newly added knowledge, which has motivated massive studies on knowledge graph completion (i.e., Knowledge Graph Embedding, KGE). KGE learns distributed representations (embeddings) of relations and entities to preserve the intrinsic structural information of KGs for predicting missing facts. Current KGE methods can be generally categorized into translational models (Bordes, Usunier, Garcia-Duran, Weston, & Yakhnenko, 2013; Ji, He, Xu, Liu, & Zhao, 2015; Lin, Liu, Sun, Liu, & Zhu, 2015; Sun, Deng, Nie, & Tang, 2018; Wang, Zhang, Feng, & Chen, 2014; Zhang, Cai, Zhang and Wang, 2020), tensor factorization models (Nickel, Rosasco, & Poggio, 2016; Trouillon, Welbl, Riedel, Gaussier, & Bouchard, 2016; Yang, Yih, He, Gao, & Deng, 2015), and neural networks models (Dai Quoc Nguyen, Nguyen, & Phung, 2018; Dettmers, Minervini, Stenetorp, & Riedel, 2018; Jiang, Wang, & Wang, 2019; Vashishth, Sanyal, Nitin, Agrawal and Talukdar, 2020). These models embed relations and entities into a continuous vector space and validate triples based on the vector representations of relations and entities via a scoring function (Zeb, Haq, Zhang, Chen, & Gong, 2021). However, most of them treat knowledge facts independently and cannot utilize the

* Corresponding authors.
E-mail addresses: hcfang@gs.zzu.edu.cn (H. Fang), youweiwang@zzu.edu.cn (Y. Wang), ieztian@zzu.edu.cn (Z. Tian), yeyd@zzu.edu.cn (Y. Ye).

https://doi.org/10.1016/j.eswa.2022.118806
Received 12 November 2021; Received in revised form 27 June 2022; Accepted 7 September 2022
Available online 12 September 2022

structural information of given KGs to enforce the reliability of the embedded representations of relations and entities.

To incorporate the structural information of KGs into knowledge representation learning, a series of graph neural network-based (GNN-based) methods have been proposed. Graph convolutional network-based (GCN-based) methods include VR-GCN (Ye, Li, Fang, Zang, & Wang, 2019), WGCN (Shang et al., 2019), R-GCN (Schlichtkrull et al., 2018), and so on. They define a propagation function to recursively aggregate the information of neighbor entities, and implement the convolutional operation on KGs. Meanwhile, they stack multi-layer GCNs to capture the multi-hop relations for aggregating the information of higher-order neighbor entities. However, GCN-based methods always aggregate the information of neighbor entities with the same importance. To this end, graph attention network-based (GAT-based) methods (Li, Liu, Zhang, Liu and Xiong, 2021; Li, Wang, Feng, Niu and Zhang, 2021; Nathani, Chauhan, Sharma, & Kaul, 2019; Zhang et al., 2020; Zhao et al., 2021) incorporate the attention mechanism into GCN for selectively aggregating the information of neighbor entities, and greatly improving the performance. Nevertheless, to the best of our knowledge, few GNN-based works incorporate the direction information of relations into measuring the importance of neighbor entities for representation learning of entities. In addition, for updating relation representations, most GNN-based methods focus only on the relations themselves whereas ignoring the information of its related neighborhoods, which may lead to some semantic loss.

To overcome the obstacles above, we propose to fully capture the information of KGs for representation learning of both relations and entities. A KG contains massive semantic information of knowledge and depicts their linking relations in a visualized graph structure, which is beneficial for the process of representation learning. With the development of GNN in the KGE field, the neighborhood information of a central entity in KGs is utilized for the aggregation of its representation since different neighbor entities link to the central entity via a specific relation indicates that different knowledge facts about the central entity. As shown in Fig. 1(a), the central entity 'Christopher Nolan' is embraced by some neighbor entities with incoming and outgoing relations. Taking the example of learning the representation of 'Christopher Nolan', we can observe that: **(i)** 'Christopher Nolan' links to different neighbor entities, among which neighbor entity 'Director' may be more representative than others since 'Christopher Nolan' is known around the world as a film director, which illustrates that different neighborhoods may present different contributions for learning the representation of a central entity. **(ii)** Neighbor entities 'London' and 'UK' with outgoing relations 'Born_in' and 'Nationality' indicates that 'London' is probably a part of 'UK'. In addition, we can infer from the neighbor entity 'Inception' with an incoming relation 'Directed_by' that 'Christopher Nolan' may have directed more than one movie. This observation demonstrates that neighbor entities with different types of relations (incoming and outgoing) may present different semantic meanings for a central entity. Thus, a bidirectional attention mechanism is proposed to measure the importance of neighborhood for a central entity based on the directions of relations. Following Vashishth, Sanyal, Nitin and Talukdar (2020), we aggregate the information of neighborhoods with outgoing relations for a central entity and reverse the incoming relations. As illustrated in Fig. 1(b), the incoming relations ('Younger_brother_of' and 'Directed_by') from neighbor entities to the central entity 'Christopher Nolan' are reversed. The relations in KGs can therefore be intuitively divided into original and reversed types. Specifically, our proposed bidirectional attention mechanism for learning entity representations first splits the neighbor entities of a central entity into two sets based on the relation types (original or reversed types), then calculates the attention scores of neighbor entities belonging to the two sets respectively for aggregating the neighborhood information. Moreover, learning relation representations can also be inspired by some impressive observations from Fig. 1(a). For instance: **(i)** Because relations 'Has_gender' and 'Born_in' have different semantic

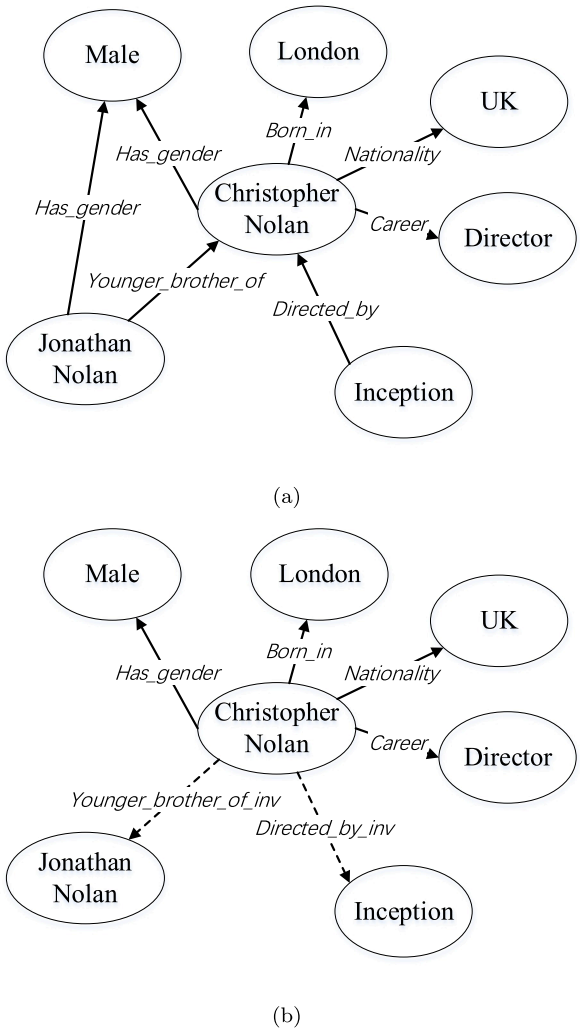

Fig. 1. A subgraph of a KG. (a) a central entity 'Christopher Nolan' connects with its neighbors by different original relations. (b) a variant of (a) where the neighbors with incoming relations of 'Christopher Nolan' are reversed to outgoing relations. The solid and dotted arrows denote the original and reversed relations respectively.

meanings in the KG, learning different representations can model their differences, which manifests the necessity of representation learning of relations. **(ii)** Because relations 'Nationality' and 'Career' are in different contexts (neighborhoods), their related neighborhoods may present some specific semantic information, thereby contributing to the representation learning of them. In turn, learning relation representations with neighborhood information can tremendously facilitate interactions between head and tail entities. **(iii)** Both 'Christopher Nolan' and 'Jonathan Nolan' are connected to entity 'Male' via relation 'Has_gender', which shows different neighborhoods of relation 'Has_gender'. This example suggests that different neighborhoods of a given relation may contribute diversely to the representation learning of the relation. From the aforementioned examples, a relation-specific attention mechanism is devised for measuring the importance of neighborhoods for learning relation representations. It is worth noting that the bidirectional attention mechanism for learning entity representations and the relation-specific attention mechanism for learning relation representations work jointly via an attention-based GCN encoder. We summarize our specific works as below:

- We propose D-AEN, an innovative encoder framework where the representations of both relations and entities are jointly learned to promote optimization of each other in an end-to-end way.

- A bidirectional attention mechanism is incorporated into learning entity representations by respectively measuring the importance of neighborhoods with different relation directions.
- A relation-specific attention mechanism is devised for measuring the importance of neighborhoods in learning relation representations.
- Extensive experimental results demonstrate that D-AEN significantly outperforms other representative baseline methods.

2. Related work

2.1. Graph neural networks

GNNs develop a unified framework for representation learning on arbitrary graphs with deep neural networks (DNNs). Hou et al. (2019), Prakash and Tucker (2021), Scarselli, Gori, Tsoi, Hagenbuchner, and Monfardini (2008), and Xu, Hu, Leskovec, and Jegelka (2018) show the expressive performance of GNNs on modeling various graph structures. Due to the successful developments of Convolutional Neural Networks (CNNs) on modeling euclidean data, Bruna, Zaremba, Szlam, and LeCun (2014) first propose a spectral method to implement convolutional operation on non-euclidean data (i.e. graphs). Along with this work, Kipf and Welling (2017) develops GCN by defining a convolutional operation on graphs in the spatial domain. Schlichtkrull et al. (2018) further keeps the idea of GCN and generalizes it on relational data and puts forward R-GCN. WGCN (Shang et al., 2019), CompGCN (Vashishth, Sanyal, Nitin and Talukdar, 2020), and VR-GCN (Ye et al., 2019) devise different propagation formulae to enrich the framework of R-GCN. In addition, considering that graph data can be divided into two parts according to certain rules, many GNN models based on dual graphs have been developed for various tasks. Li et al. (2020) addresses the issue of object-tag prediction by dividing a KG into an object graph and a tag graph to respectively encode high-order proximities for objects and tags. Splitting an entity's neighbors into two sets based on the direction of relations, Zeb et al. (2021) devises a dual weighted GCN framework based on WGCN to learn two different representations for entities in the KGE field. Guo et al. (2021) generates an attribute graph and a collaborative graph of users and items to respectively perform graph convolution for CTR prediction. Wu et al. (2021) takes into account both the topology and attributes of nodes in hypergraphs and proposes a Dual-view HyperGraph Neural Network for node classification. Concurrently, owing to the powerful advantages of the attention mechanism in Computer Vision (CV) and Natural Language Processing (NLP), Veličković et al. (2018) proposes GATs by incorporating an attention mechanism into GCNs for node classification, which selectively aggregates the information of neighbors for each node.

Recently, several advanced extensions of GAT have been developed to model KGs for the KGE task. For example, Nathani et al. (2019) assigns an attention value to each triple for fusing the neighborhood information for a central entity. Zhang, Zhuang et al. (2020) introduces a two-level hierarchical attention mechanism that corresponds to the relations and tail entities. Li, Wang et al. (2021) aggregates the information of both direct neighbors and multi-hop neighbors with a global attention mechanism. Zhao et al. (2021) incorporates the global information into the GAT model by estimating the entity importance based on an attention-based global random walk algorithm. Li, Liu et al. (2021) devises a relation-path-based attention mechanism to measure the importance of neighbor entities with different relations. However, these methods ignore the direction information of relations with measuring the importance of neighbors. In addition, none of them consider the impact of neighborhoods on representation learning of relations, thereby limiting the power of relation representations. Our proposed D-AEN not only incorporates the direction information into the measurement of neighbor importance for representation learning of entities but also fully considers the impact of neighborhoods on the representation learning of relations, which demonstrates that both entity embeddings and relation embeddings contribute to the optimization of each other.

2.2. Knowledge graph embedding

KGE focuses on learning distributed representations of relations and entities for deducing missing triples, and in turn, updating the representations of both relations and entities. KGE predicts the validity of given triples by a scoring function. According to the types of scoring functions, existing KGE methods can be generally divided into translational methods, tensor factorization methods, and neural network methods.

Translational methods define a scoring function on distance by regarding relations as a translation from heads to tails. TransE (Bordes et al., 2013), the most classic translational method, shows excellent performance on modeling 1-to-1 relations, while still suffering from some drawbacks on modeling complex relations. Consequently, a series of variants including TransR (Lin et al., 2015), TransD (Ji et al., 2015), and TransH (Wang et al., 2014) are developed to extend TransE by introducing different vector spaces to embed relations and entities. RotatE (Sun et al., 2018) introduces the idea of rotation to regard the relations as a rotation from heads to tails. Especially, it transforms relations and entities into a complex space and generates two representations for all relations and entities corresponding to real and imaginary parts. ModE (Zhang, Cai et al., 2020) preserves the phase and modulus information of relations and entities by modeling KGs in a polar coordinate system. Tensor factorization methods define a multiplicative function over the representations of relations and entities to score triples. DistMult (Yang et al., 2015) is a basic tensor factorization method that defines a bilinear operation on the embeddings of relations and entities to tackle the KGE task. ComplEx (Trouillon et al., 2016) generalizes DistMult using a complex embedding space to replace the real embedding space and can model asymmetric relations by a conjugate operation on tail entity embeddings. HolE (Nickel et al., 2016) applies a circular correlation operation to model asymmetric relations. TuckER (Balazevic, Allen, & Hospedales, 2019) adopts Tucker decomposition (Tucker et al., 1964) to model the binary representations of knowledge triples and then proposes a straightforward yet powerful linear model. Neural network methods capture the nonlinear features of relations and entities by applying neural networks. ConvE (Dettmers et al., 2018) implements a convolutional operation on reshaped representations of head entities and relations with the help of CNNs for matching tail entities. ConvKB (Dai Quoc Nguyen et al., 2018) subsequently performs a convolutional operation on all elements of a triple and extracts the complex features of triples jointly. ConvR (Jiang et al., 2019) takes relation embeddings as convolutional kernels to avoid over-parameterization in comparison with ConvE. InteractE (Vashishth, Sanyal, Nitin, Agrawal et al., 2020) generalizes ConvE by focusing on the feature permutation of the reshaped embeddings of head entities and relations. In contrast to translational methods and tensor factorization methods, neural network methods can learn more expressive features due to their deep and multilayer architectures.

Apart from the aforementioned three kinds of methods, GNN-based methods have attracted wide attention in recent years. R-GCN (Schlichtkrull et al., 2018) applies traditional GCN into multi-relational KGs. CompGCN (Vashishth, Sanyal, Nitin and Talukdar, 2020), WGCN (Shang et al., 2019), and VR-GCN (Ye et al., 2019) make some innovations based on R-GCN. KBGAT (Nathani et al., 2019) and RGHAT (Zhang, Zhuang et al., 2020) extend GNN methods by introducing an attention mechanism into representation learning. Following these works, we concentrate on a GNN-based method to address the KGE task.

3. Dual-attention Embedding Network (D-AEN)

The architecture of our D-AEN model with a single layer is shown in Fig. 2. Besides original triples, the reversed relations and reversed triples in KGs are taken into account for knowledge representation learning. Subsequently, it jointly learns the representations of both relations and entities by fusing the information of knowledge triples with proper importance. Before introducing D-AEN, we present the basic notations of this paper.

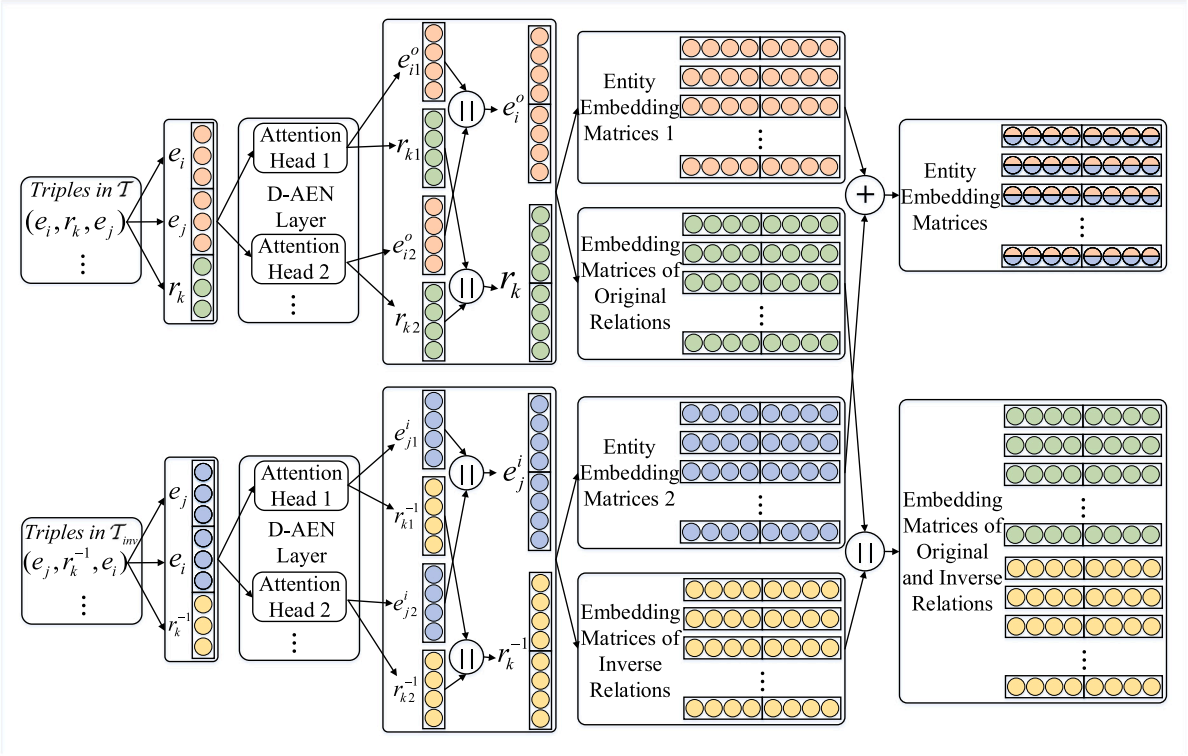

Fig. 2. The architecture of a single D-AEN layer where the representations of both relations and entities are learned jointly. $\mathcal{T}$ and $\mathcal{T}_{inv}$ respectively denote the set of original and reversed triples. Green and orange circles indicate relation and entity representations in $\mathcal{T}$ respectively. Yellow and blue circles indicate relation and entity representations in $\mathcal{T}_{inv}$ respectively. Circles with both orange and blue colors represent the final entity representations. $\|$ indicates concatenation and + indicates summation. Entity representations are learned separately based on two independent triple sets $\mathcal{T}$ and $\mathcal{T}_{inv}$, then are summed up as final entity representations. Homoplastically, D-AEN also learns the representations of both original and reversed relations separately and concatenates them as final relation representations.

Table 1
Notations and descriptions.

Notations	Descriptions
$\mathcal{G}$	Knowledge graphs
$\mathcal{E}$, $\mathcal{R}$, $\mathcal{T}$	Entity set, relation set, and triple set
$\mathcal{R}_{inv}$, $\mathcal{T}_{inv}$	Set of reversed relations and reversed triples
(e_i, r_k, e_j)	Triple with head entity, relation, and tail entity
t_{ij}^k	Triple (e_i, r_k, e_j)
r_k^{-1}	Reversed relation of relation r_k
(e_j, r_k^{-1}, e_i)	Reversed triple of triple (e_i, r_k, e_j)
h_i, g_k	Embedding of entity e_i and relation r_k
$\mathcal{N}_{e(i)}$	Set of neighbor entities of e_i for its outgoing edges
$\mathcal{N}_{r(k)}$	Set of the related head-tail entity pairs of r_k
$\mathbf{a}$	Attention vector
α	Attention scores of original triples for entities
β	Attention scores of reversed triples for entities
γ	Attention scores of triples for relations
$\varphi(x)$	Scoring function
y	Labels of triples
$\hat{y}$	Prediction scores of triples

3.1. Notations

The main notations of this paper and their corresponding descriptions are listed in Table 1. Some of them need to be explained in detail. Similar to Vashishth, Sanyal, Nitin and Talukdar (2020), we learn the representation for a central entity by fusing the information of neighborhoods with outgoing relations. Simultaneously, the incoming relations are converted into reversed ones. Specifically, for entity e_i, we denote $\mathcal{N}_{e(i)}$ as the set of neighbor entities of e_i for its outgoing edges (i.e., tail entity set of e_i). For the neighbor entities of e_i for its ingoing edges, we convert the relations between them into the reversed ones. Hence, we introduce $\mathcal{R}_{inv}$ and $\mathcal{T}_{inv}$ by incorporating reversed relations into representation learning, i.e. $\mathcal{R}_{inv} = \{r_k^{-1} | r_k \in \mathcal{R}\}$ and $\mathcal{T}_{inv} = \{(e_j, r_k^{-1}, e_i) | (e_i, r_k, e_j) \in \mathcal{T}\}$. In the end, the relations connecting with a central entity can be divided into two types: original and reversed types. Further, $\mathcal{R}$ and $\mathcal{T}$ can be extended as $\mathcal{R}' = \mathcal{R} \cup \mathcal{R}_{inv}$ and $\mathcal{T}' = \mathcal{T} \cup \mathcal{T}_{inv}$ respectively. Additionally, we refer to the triples related to an entity or a relation as their neighborhoods.

3.2. Representation learning of entities and relations

Following Nathani et al. (2019), we learn the representation of triple $t_{ij}^k = (e_i, r_k, e_j)$ in $\mathcal{T}'$ as below:

$$v_{ijk} = \mathbf{W_1} \cdot [h_i \| h_j \| g_k] \tag{1}$$

where $h_i, h_j \in \mathbb{R}^d$, and $g_k \in \mathbb{R}^d$ represent the initial embeddings of entities e_i, e_j, and relation r_k respectively. $\|$ denotes concatenation operation. $\mathbf{W_1} \in \mathbb{R}^{d \times 3d}$ indicates a linear transformation matrix.

We then learn the importance of each triple t_{ij}^k as:

$$b_{ijk} = \text{LeakeyReLU}(\mathbf{a} \cdot v_{ijk}) \tag{2}$$

where $\mathbf{a} \in \mathbb{R}^d$ is an attention vector. LeakeyReLU denotes a non-linear activation function with a negative slope value of 0.2. Next, the relative attention value of triple t_{ij}^k is computed by applying $softmax$ over b_{ijk}. To make the relations and entities interact semantically well, the representations of triples are used to jointly update the representations of both relations and entities, as shown in Fig. 2.

3.2.1. Representation learning of entities

We learn entity representations by aggregating the information of its related triples. Different from previous works, we utilize a bidirectional attention mechanism that encapsulates the direction information of relations to measure the importance of neighborhoods.

Based on the type of relation r_k in triple t_{ij}^k, we develop two different ways to measure its relative attention value for entity e_i as:

$$\alpha_{ijk} = softmax(b_{ijk}) = \frac{exp(b_{ijk})}{\sum_{(r_r,e_n)\in\mathcal{N}_{e(i)}} exp(b_{inr})}, \quad (r_k, r_r \in \mathcal{R}) \tag{3}$$

$$\beta_{ijk} = softmax(b_{ijk}) = \frac{exp(b_{ijk})}{\sum_{(r_r,e_n)\in\mathcal{N}_{e(i)}} exp(b_{inr})}, \quad (r_k, r_r \in \mathcal{R}_{inv}) \tag{4}$$

where $\mathcal{N}_{e(i)} = \{(r_k, e_j)|(e_i, r_k, e_j) \in \mathcal{T}'\}$ represents the set of the neighbor entities of e_i for its outgoing edges.

Consequently, the representation of entity e_i is updated by summing up the representation of its related triples with corresponding attention values, in which the triples with most representative neighbor entities may show more influence on the learned representation of e_i. Along with the above setting, we aggregate the representations of triples separately according to the type of relations, defined as follows:

$$h'_{i\_O} = f_e\left(\mathbf{W}_{\mathrm{O}} \sum_{(r_k,e_j)\in\mathcal{N}_{e(i)}} \alpha_{ijk} v_{ijk}\right), r_k \in \mathcal{R} \tag{5}$$

$$h'_{i\_I} = f_e\left(\mathbf{W}_{\mathrm{I}} \sum_{(r_k,e_j)\in\mathcal{N}_{e(i)}} \beta_{ijk} v_{ijk}\right), r_k \in \mathcal{R}_{inv} \tag{6}$$

where h'_{i_O} and h'_{i_I} represent the sum of the representations of the triples with original and reversed relation respectively. f_e denotes a non-linear activation function of entities. $\mathbf{W}_{\mathrm{O}} \in \mathbb{R}^{d'\times d}$ and $\mathbf{W}_{\mathrm{I}} \in \mathbb{R}^{d'\times d}$ are the graph convolutional kernels corresponding to the original and reversed relations respectively. Additionally, to preserve the information from e_i itself, we then incorporate its transformed initial representation into its final representation as:

$$h'_i = h'_{i\_O} + h'_{i\_I} + \mathbf{W}_{\mathrm{e}} \cdot h_i \tag{7}$$

where h'_i denotes the learned representation of e_i, and $\mathbf{W}_{\mathrm{e}} \in \mathbb{R}^{d'\times d}$ represents a linear transformation matrix.

To aggregate more information of neighborhoods and stabilize the training process, we apply a multi-head attention mechanism like GAT (Veličković et al., 2018) where M attention mechanisms learn M independent representations. Then we concatenate them to generate the final representation as:

$$h'_i = \Big\|_{m=1}^{M} h'_{i_m} \tag{8}$$

3.2.2. Representation learning of relations

Owing to the heterogeneous types of relations in KGs, we need to learn independent representations for them to model their differences, instead of ignoring their representation learning. Further, since these relations are in different contexts (neighborhoods), focusing only on their self-updating may suffer from some semantic loss. To this end, we encapsulate neighborhood information with proper importance into the learned relation representations.

Same as the process of learning entity representations, for relation $r_k \in \mathcal{R}'$, we selectively fuse the information of its related triples to update its representation. First, we calculate the relative attention of triple t_{ij}^k for relation r_k.

$$\gamma_{ijk} = softmax(b_{ijk}) = \frac{exp(b_{ijk})}{\sum_{(e_m,e_n)\in\mathcal{N}_{r(k)}} exp(b_{mnk})} \tag{9}$$

where $\mathcal{N}_{r(k)} = \{(e_i, e_j)|(e_i, r_k, e_j) \in \mathcal{T}'\}$ denotes the set of related head-tail entity pairs of r_k.

Then, the new representation of r_k is expressed accordingly as:

$$g'_k = f_r\left(\mathbf{W}_{\mathrm{R}} \sum_{(e_i,e_j)\in\mathcal{N}_{r(k)}} \gamma_{ijk} v_{ijk}\right) \tag{10}$$

Algorithm 1: Learning entity and relation embeddings in the encoder model.

Input: KGs $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$;
Number of D-AEN layers L;
Attention vector of each layer;
Weight matrices of each layer;
Embedding size d^0, d^1, ..., d^L ;
Number of attention heads M_1, M_2, ..., M_L.
Output: New embeddings of entities, original relations, and reversed relations.

```
Extend relation set R' ← R ∪ R_inv, R_inv ← {r_k^{-1} | r_k ∈ R};
Extend triple set T' ← T ∪ T_inv, T_inv ← {(e_j, r_k^{-1}, e_i) | (e_i, r_k, e_j) ∈ T};
Initialize embeddings of entities and relations, h_i^0 ∈ R^{d^0}, ∀e_i ∈ E;
  g_k^0 ∈ R^{d^0}, ∀r_k ∈ R';
for L = 1, 2, ..., L do
    for m = 1, 2, ..., M_l do
        for t_ij^k ∈ T' do
            Learn the embeddings of triples by (1);
            Calculate the importance of triples by (2);
        for e_i ∈ E do
            for (e_i, r_k, e_j) ∈ T do
                Calculate the attention value α_ijk^l by (3);
            Fuse original neighborhoods for e_i: h_im_O^l by (5) (r_k ∈ R);
            for (e_i, r_k, e_j) ∈ T_inv do
                Calculate the attention value β_ijk^l by (4);
            Fuse reversed neighborhoods for e_i: h_im_I^l by (6)
              (r_k ∈ R_inv);
            Update entity embedding h_im^l by (7);
        for r_k ∈ R' do
            for (e_i, r_k, e_j) ∈ T' do
                Calculate the attention value γ_ijk^l by (9);
            Update relation embedding g_km^l by (11) and (10);
    Concatenate the learned embeddings of entities and relations
      from all attention heads: h_i^l, ∀e_i ∈ E and g_k^l, ∀r_k ∈ R' by (8) and
      (12) respectively;
return h_i^L ∈ R^{d^L}, ∀e_i ∈ E; g_k^L ∈ R^{d^L}, ∀r_k ∈ R'.
```

where f_r denotes a non-linear activation function of relations. $\mathbf{W}_{\mathrm{R}} \in \mathbb{R}^{d'\times d}$ is the relation-specific graph convolutional kernel.

To keep initial relation information in the updated representation, we add the transformed initial representation of r_k into its updated representation.

$$g'_k = g'_k + \mathbf{W}_{\mathrm{r}} \cdot g_k \tag{11}$$

where $\mathbf{W}_{\mathrm{r}} \in \mathbb{R}^{d'\times d}$ is a linear transformation matrix.

Moreover, we incorporate M attention heads into the process of the representation learning of r_k to encapsulate more neighborhood information.

$$g'_k = \Big\|_{m=1}^{M} g'_{k_m} \tag{12}$$

4. Encoder–decoder architecture

We follow an encoder–decoder framework to address the KGE task. The encoder contains L GCN layers. A scoring function is leveraged by the decoder to predict the validity of given triples.

4.1. Encoder

After the single D-AEN layer introduced above, we develop the encoder framework with two D-AEN layers in practice, in which the

Table 2
Dataset statistics.

Datasets	Entities	Relations	Edges			
			Train	Valid	Test	Total
FB15k-237	14,541	237	272,115	17,535	20,466	310,116
WN18RR	40,943	11	86,835	3034	3134	93,003
Kinship	104	25	8544	1068	1074	10,686

first layer consists of M attention heads for generating M different output representations of relations and entities and concatenating them, followed by a non-linear activation. Whereafter, the second layer adopts a single attention head and a non-linear activation to learn the final embeddings of relations and entities. Furthermore, we set $f_e = f_r = \mathbf{tanh}$ for simplicity. Algorithm 1 presents the overall process of our encoder model.

4.2. Decoder

We choose ConvE (Dettmers et al., 2018) as the decoder to predict the validity of a given triple. Note that we also try to utilize other representative models, such as DistMult (Yang et al., 2015) and TransE (Bordes et al., 2013), but find ConvE performs the best. For triple t_{ij}^k, the scoring function of ConvE is formally defined as:

$$\varphi(e_i, r_k, e_j) = f(vec(f([\overline{h_i}; \overline{g_k}] * w))\mathbf{W})h_j \tag{13}$$

where f represents the activation function. $\overline{h_i}, \overline{g_k} \in \mathbb{R}^{d_1 d_2}$ indicate the 2D reshaping of $h_i, g_k \in \mathbb{R}^{d'}$ in which $d' = d_1 d_2$. $*$ and w denote 2D convolutional operation and a set of convolutional kernels respectively. $vec(\cdot)$ is a vectorization operation that converts a tensor into a vector. $\mathbf{W}$ is a linear transformation matrix. In practice, $h_i, g_k, h_j \in \mathbb{R}^{d'}$ are the output of the encoder model.

4.3. Optimization

We train the encoder–decoder model jointly. Specifically, for triple t_{ij}^k, we train the model and update corresponding parameters by minimizing the Binary Cross Entropy (BCE) loss as:

$$\mathcal{L} = -\frac{1}{N}\sum_{o=1}^{N}\left(y_{t_{io}^k} \cdot log(\hat{y}_{t_{io}^k}) + (1 - y_{t_{io}^k}) \cdot log(1 - \hat{y}_{t_{io}^k})\right) \tag{14}$$

where N denotes the number of candidates of tail entities. $y_{t_{io}^k}$ (1 or 0) is the label of triple t_{io}^k, and $\hat{y}_{t_{io}^k} = sigmoid(\varphi(e_i, r_k, e_o))$ corresponds to its prediction score. What's more, we use label smoothing (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016) and dropout (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014) techniques to lessen overfitting and improve generalization. Batch normalization (Ioffe & Szegedy, 2015) is adopted for stabilizing, regularizing, and speeding the rate of convergence. The Adam optimizer (Kingma & Ba, 2015) is employed to optimize the loss function.

5. Experiments

We experiment with the link prediction task to evaluate D-AEN against several baselines. We further investigate the power of our model on modeling different categories of relations. In addition, the impact of hyperparameters and model components on prediction performance is also explored.

5.1. Datasets

WN18RR, FB15k-237, and Kinship are utilized to conduct experiments in the link prediction task. These three benchmark datasets contain a certain number of relations and entities, and their basic statistics are listed in Table 2. We also present their detailed information below.

- FB15k-237 (Dettmers et al., 2018) contains 14 541 entities and 237 relations, which is built by removing reversible relations from the FB15k dataset (Bordes et al., 2013).
- WN18RR (Dettmers et al., 2018) contains 40 943 entities and 11 relations, which is built by removing reversible relations from the WN18 dataset (Bordes et al., 2013).
- Kinship (Lin, Socher, & Xiong, 2018) contains 104 entities and 25 relations, which describes the personal relationships of Alyawarra tribe.

5.2. Baselines

We compare D-AEN with several representative baselines. Their detailed descriptions are listed below.

- TransE (Bordes et al., 2013): a classic translational method that devises a scoring function based on the distance to model relations between heads and tails with a translation operation.
- RotatE (Sun et al., 2018): an advanced extension of TransE. It treats relations as a rotation from heads to tails, thereby modeling various relation patterns.
- ModE (Zhang, Cai et al., 2020): an innovative translational model that preserves the phase and modulus information of relations and entities by introducing a polar coordinate system.
- DistMult (Yang et al., 2015): a representative tensor factorization model that calculates the scores of triples by employing a bilinear scoring function.
- ComplEx (Trouillon et al., 2016): a marked extension of DistMult which introduces a complex space to model antisymmetric relations.
- Tucker (Balazevic et al., 2019): a recent tensor factorization model which adopts Tucker decomposition (Tucker et al., 1964) to model the binary representations of knowledge triples.
- R-GCN (Schlichtkrull et al., 2018): a generalization of GCN which uses GCN to model multi-relational data.
- ConvE (Dettmers et al., 2018): the first CNN model that addresses relations and entities by applying convolutional operation.
- ConvKB (Dai Quoc Nguyen et al., 2018): another CNN model that can capture global relationships and transitional properties of triples.
- ConvR (Jiang et al., 2019): an advanced extension of ConvE that regards relations as convolutional kernels to lessen model parameters.
- InteractE (Vashishth, Sanyal, Nitin, Agrawal et al., 2020): an excellent extension of ConvE that captures more numbers of interactions between relations and entities.
- CompGCN (Vashishth, Sanyal, Nitin and Talukdar, 2020): a recent GCN-based model that performs a compositional operation between relations and tail entities.
- WGCN (Shang et al., 2019): a novel GCN-based model that introduces a weighted GCN to model the differences of relations.
- KBGAT (Nathani et al., 2019): a powerful GNN-based model applying graph attention mechanism into capturing the information of multi-hop neighborhoods.
- KGEL (Zeb et al., 2021): an advanced extension of WGCN that proposes a dual-weighted GCN framework for respectively aggregating the information of head entities and tail entities.
- HRAN (Li, Liu et al., 2021): a novel GNN-based model that devises a relation-path-based attention mechanism to measure the importance of neighbor entities with different relations.

5.3. Evaluation protocol

Following most baseline methods, we use the ranks of triples for evaluation. Specifically, we perform head evaluation and tail evaluation for all correct triples in the testing dataset. Taking the tail

Table 3
Hyperparameter settings.

Datasets	WN18RR	FB15k-237	Kinship
Learning rate	1e−4	1e−4	1e−4
Epochs	800	500	500
Batch size	128	128	128
Label smooth	0.1	0.1	0.1
Initial embedding size	200	300	300
GCN embedding size	200	200	200
LeakeyReLU	0.2	0.2	0.2
GCN dropout	0.5	0.6	0.4
Attention heads	1	3	3
Embedding dropout	0.0	0.0	0.0
Hidden dropout	0.0	0.0	0.0
Feature dropout	0.0	0.0	0.0
Kernel size	5	5	5
Number of filters	300	300	300
Negative samples	40	40	40

evaluation of a triple as an example, we first build its correct triples by replacing its head and tail entities with other entities. The scores of the correct triple and corrupted triples are then predicted by the proposed model and sorted in descending order for generating corresponding ranks. Based on the ranks, we finally evaluate our proposed model with several rank-based metrics including mean reciprocal rank (MRR), mean rank (MR), and Hits@N (N = 1,3,10). We expect to achieve higher MRR and Hits@N, and lower MR. To obtain more reasonable results, we follow the 'Filter' setting in TransE (Bordes et al., 2013), i.e., the corrupted triples in the training dataset do not involve rankings. Like the process of tail evaluation, the results of head evaluation can be obtained in the same way. The final evaluation results are the average results of head evaluation and tail evaluation.

5.4. Experimental setting

We conduct all experiments by training the encoder and decoder model jointly, rather than a separate training procedure (Li, Wang et al., 2021; Nathani et al., 2019; Zhao et al., 2021). The hyperparameter settings are selected as follows: learning rate [0.001, 0.0005, 0.0001], batch size [64, 128, 256], label smooth [0.1, 0.2, 0.3], initial embedding size of relations and entities [100, 200, 300, 400], the number of attention heads [1, 2, 3, 4], kernel size [3, 5, 7], the number of filters [100, 200, 300]. The dropout includes GCN dropout, embedding dropout, feature dropout, and hidden dropout sampled from 0.0 to 0.7. The number of negative samples for each triple are is selected from [20, 30, 40, 50]. Additionally, the final embedding size of relations and entities updated by GCN layers is set to 200. Table 3 lists the details of hyperparameter settings for different datasets. We implement D-AEN using the software library PyTorch, and all experiments are conducted over a PC server equipped with an Intel i7 8700K CPU and an NVIDIA GTX 1080Ti GPU.

5.5. Results and analysis

To demonstrate the effectiveness of D-AEN, we intuitively analyze the experimental results from two aspects. On the one hand, the overall performance of D-AEN is evaluated against several baseline models, shown in Tables 4 and 5. On the other hand, to evaluate the power of modeling diverse relation types, we compare D-AEN with some baselines in Hits@10 by relation categories, shown in Tables 6 and 7.

5.5.1. Overall results

The overall results on FB15k-237 and WN188RR are shown in Table 4. The results on Kinship are shown in Table 5. Results with ♣ and ¶ are reported from Nathani et al. (2019) and Sun et al. (2018) respectively. '-' denotes missing values and the rest are extracted from the original works. We summarize the following observations from the results. **(i)** The results indicate that D-AEN achieves profound performance compared with the baselines except for KBGAT on FB15k-237 and significantly outperforms all the state-of-the-art baselines on 2 metrics for WN18RR and 5 metrics for Kinship. KBGAT expands the training dataset by converting the two-hop neighbor entities of all entities to one-hop, which can encapsulate more neighborhood information in learned entity representations. Therefore, KBGAT achieves the best performance on FB15k-237 because it is a more complicated KG that contains many relations and edges against WN18RR and Kinship, which makes the entities in FB15k-237 contain more two-hop neighbor entities. It is worth noting that although the results of KBGAT on FB15k-237 are better than those of D-AEN, D-AEN outperforms KBGAT on 3 metrics for WN18RR and 5 metrics for Kinship. In consequence, the overall results elucidate the effectiveness of our proposed D-AEN. **(ii)** Compared with the decoder model ConvE only, D-AEN improves dramatically on all metrics for the three benchmark datasets. For example, D-AEN performs better than ConvE on the MRR metric with an improvement of 11.3% for FB15k-237. This strongly confirms the effectiveness of our encoder model, and also indicates that the neighborhood information aggregated by D-AEN is valuable. D-AEN also significantly outperforms the recent GCN model CompGCN, which adopts ConvE as the decoder model like D-AEN. This result demonstrates the superiority of our encoder model and further verifies the effectiveness of D-AEN.

5.5.2. Results of hits@10 based on relation categories

In this part, we conduct experiments by relation categories for evaluating the power of modeling different types of relations. Intuitively, the relations in a KG can be generally categorized into 1-to-1 (one specific head entity and one specific tail entity are connected with the relation), N-to-1 (many different head entities and one specific tail entity are connected with the relation), 1-to-N (one specific head entity and many different tail entities are connected with the relation) and N-to-N (many different head entities and many different tail entities are connected with the relation). The statistics show that the FB15k-237 dataset contains 7.2% 1-to-1, 34.2% N-to-1, 11.0% 1-to-N, and 47.6% N-to-N relations, and the WN18RR dataset contains 18.2% 1-to-1, 27.3% N-to-1, 36.3% 1-to-N, and 18.2% N-to-N relations. Tables 6 and 7 present the results of Hits@10 based on relation categories on the two datasets. Because the link prediction task focuses on the average results of the tail evaluation and head evaluation, we mainly compare the average scores here. Results with ♠ are taken from Li, Wang et al. (2021). On FB15k-237, D-AEN significantly outperforms all the baselines on four categories of relations, which illuminates the great power of D-AEN for modeling multi-relational KGs. For example, D-AEN performs better than the decoder ConvE with an improvement of 21.2% on N-to-1 relations. On WN18RR, D-AEN also achieves the best results on four metrics. It is worth noting that TransE, ComplEx, and ConvE achieve the same performance on 1-to-1 relations. And D-AEN has a slight improvement in N-to-N relations. The reason is that WN18RR contains only two 1-to-1 relations and two N-to-N relations, which makes the prediction relatively easy.

5.6. Convergence analysis

Here, we investigate the convergence of D-AEN with metric MRR on FB15k-237 and WN18RR. As Fig. 3 shows, the results of the head evaluation and tail evaluation are denoted by green and blue lines respectively, and red lines indicate their average values. We can observe that these three values increase rapidly in the first 50 epochs, then stabilize after approximately 350 epochs and achieve satisfactory performance on FB15k-237. Homoplastically, these three values rise rapidly in the first 100 epoch, then gradually increase on WN18RR. These observations illustrate that our model converges quickly, is not prone to overfitting, and also is reliable in practical applications.

Table 4
Link prediction results of D-AEN and several baselines on two benchmark datasets evaluated by MR, MRR, and Hits@N.

Datasets	WN18RR					FB15k-237				
Metrics	Hits			MR↓	MRR↑	Hits			MR↓	MRR↑
	@1↑	@3↑	@10↑			@1↑	@3↑	@10↑		
TransE ♣	–	–	0.501	3384	0.226	–	–	0.465	357	0.294
RotatE	0.428	0.492	0.571	3340	0.476	0.241	0.375	0.533	177	0.338
ModE	0.427	0.486	0.564	–	0.472	0.244	0.380	0.534	–	0.341
DistMult ♣	0.39	0.44	0.49	5110	0.43	0.155	0.263	0.419	254	0.241
ComplEx ♣	0.41	0.46	0.51	5261	0.44	0.158	0.275	0.428	339	0.247
TuckER	0.443	0.482	0.526	–	0.470	0.266	0.394	0.544	–	0.358
ConvE	0.39	0.43	0.48	5277	0.46	0.239	0.350	0.491	246	0.316
ConvKB	–	–	0.525	2544	0.248	–	–	0.517	257	0.396
ConvR	0.433	0.489	0.537	–	0.475	0.261	0.385	0.528	–	0.350
InteractE	0.430	–	0.528	5202	0.463	0.263	–	0.535	172	0.354
R-GCN	–	–	–	–	–	0.151	0.264	0.417	–	0.249
WGCN	0.43	0.48	0.54	–	0.47	0.26	0.39	0.54	–	0.35
CompGCN	0.443	0.494	0.546	3533	0.479	0.264	0.390	0.535	197	0.355
KBGAT	0.361	0.483	**0.581**	**1940**	0.440	**0.460**	**0.540**	**0.626**	210	**0.518**
KGEL	0.446	0.467	0.547	–	0.476	0.317	0.462	0.593	–	0.414
HRAN	**0.450**	0.494	0.542	2113	0.479	0.263	0.390	0.541	**156**	0.355
D-AEN	0.443	**0.500**	0.561	2248	**0.484**	0.337	0.471	0.611	164	0.429

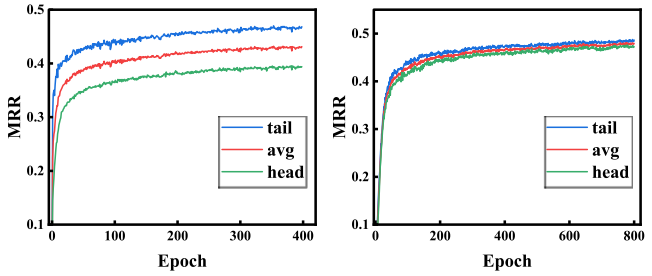

Fig. 3. The convergence of D-AEN with metric MRR on (a) FB15K-237, (b) WN18RR. The green and blue lines correspond to the results of the head evaluation and tail evaluation respectively, with the red lines indicating average values.

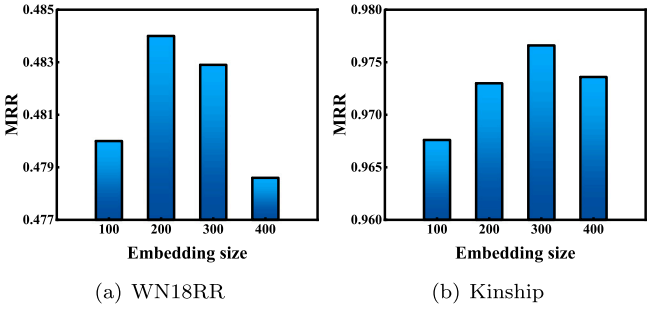

Fig. 4. The investigation of parameter sensitivity to the initial embedding size on (a) WN18RR, and (b) Kinship.

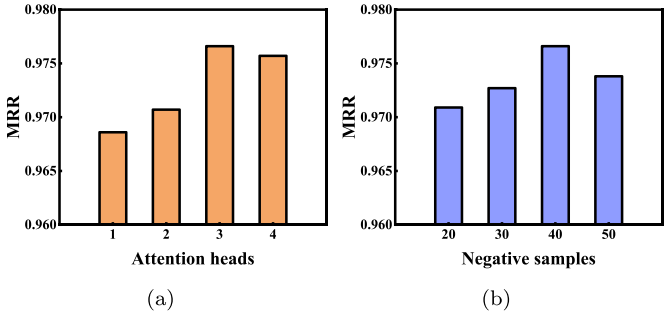

Fig. 5. The investigation of parameter sensitivity to (a) Number of attention heads, and (b) Number of negative samples on the Kinship dataset.

Table 5
Link prediction results of D-AEN and several baselines on the Kinship dataset evaluated by MR, MRR, and Hits@N.

Datasets	Kinship				
Metrics	Hits			MR↓	MRR↑
	@1↑	@3↑	@10↑		
TransE ¶	0.009	0.643	0.841	6.8	0.309
RotatE	–	–	–	–	–
ModE	–	–	–	–	–
DistMult ¶	0.367	0.581	0.867	5.26	0.516
ComplEx ¶	0.733	0.899	0.971	2.48	0.823
TuckER	–	–	–	–	–
ConvE	0.73	0.91	0.98	2	0.83
ConvKB ¶	0.436	0.755	0.953	3.3	0.614
ConvR	–	–	–	–	–
InteractE	–	–	–	–	–
R-GCN ¶	0.03	0.088	0.239	25.92	0.109
WGCN	–	–	–	–	–
CompGCN	–	–	–	–	–
KBGAT	0.859	0.941	0.980	1.94	0.904
KGEL	0.764	0.919	0.983	–	0.844
HRAN	–	–	–	–	–
D-AEN	**0.968**	**0.984**	**0.990**	**1.52**	**0.977**

5.7. Parameter sensitivity

5.7.1. Initial embedding size

To explore the impact of the initial embedding size of entities and relations on the model performance, we experiment with the embedding size sampled from 100, 200, 300, and 400 on WN18RR and Kinship. The results with the MRR metric are shown in Fig. 4, which shows that the best performance is achieved with the initial embedding size of 200 and 300 on WN18RR and Kinship respectively, while the other embedding sizes leads to worse performance. The reason is that the initial entity and relation representations with a smaller initial embedding size cannot retain enough intrinsic structural information of KGs. A larger initial embedding size, on the other hand, may make the learned entity and relation representations nosier and worsen the generalization ability of the model.

5.7.2. Number of attention heads

We investigate the performance of D-AEN with the number of attention heads $m \in [1, 2, 3, 4]$ on Kinship. As shown in Fig. 5(a), the performance of D-AEN with 3 attention heads has a great improvement against the one with 1 or 2 attention heads and is also better than the one with 4 attention heads. The reason is that incorporating a proper

Table 6
Link prediction results of D-AEN and several baselines by relation categories on the FB15k-237 dataset.

Models (Hits@10)↑	1-to-1			N-to-1			1-to-N			N-to-N		
	Tail	Head	Avg	Tail	Head	Avg	Tail	Head	Avg	Tail	Head	Avg
TransE ♠	0.521	0.537	0.529	0.833	0.070	0.452	0.052	0.573	0.312	0.508	0.347	0.428
DistMult ♠	0.182	0.193	0.188	0.793	0.031	0.412	0.039	0.514	0.277	0.485	0.320	0.403
ComplEx ♠	0.411	0.411	0.411	0.818	0.050	0.434	0.050	0.551	0.300	0.533	0.379	0.456
ConvE ♠	0.258	0.250	0.254	0.865	0.147	0.506	0.132	0.603	0.368	0.581	0.426	0.504
D-AEN	0.563	0.589	**0.576**	0.872	0.564	**0.718**	0.211	0.625	**0.418**	0.640	0.557	**0.598**

Table 7
Link prediction results of D-AEN and several baselines by relation categories on the WN18RR dataset.

Models (Hits@10)↑	1-to-1			N-to-1			1-to-N			N-to-N		
	Tail	Head	Avg	Tail	Head	Avg	Tail	Head	Avg	Tail	Head	Avg
TransE ♠	0.976	0.976	**0.976**	0.190	0.022	0.106	0.061	0.276	0.169	0.941	0.942	0.942
DistMult ♠	0.929	0.952	0.941	0.334	0.047	0.191	0.051	0.269	0.160	0.944	0.948	0.946
ComplEx ♠	0.976	0.976	**0.976**	0.309	0.053	0.181	0.086	0.288	0.187	0.950	0.951	0.951
ConvE ♠	0.976	0.976	**0.976**	0.303	0.107	0.205	0.190	0.451	0.321	0.948	0.947	0.948
D-AEN	0.976	0.976	**0.976**	0.387	0.242	**0.345**	0.229	0.505	**0.367**	0.952	0.952	**0.952**

Table 8
Ablation study of D-AEN in the link prediction task on two benchmark datasets with metrics MRR and Hits@N.

Datasets	WN18RR				Kinship			
Metrics	Hits			MRR↑	Hits			MRR↑
	@1↑	@3↑	@10↑		@1↑	@3↑	@10↑	
RemoveBR	0.424	0.497	0.554	0.472	0.922	0.967	0.985	0.946
RemoveR	0.434	0.493	0.555	0.476	0.932	0.969	0.988	0.952
RemoveRA	0.439	0.497	0.553	0.479	0.960	0.978	0.988	0.970
D-AEN	**0.443**	**0.500**	**0.561**	**0.484**	**0.968**	**0.984**	**0.990**	**0.977**

number of attention heads into D-AEN can encapsulate more neighborhood information to learn both entity and relation representations, but more attention heads may encapsulate some useless neighborhood information thereby resulting in worse performance.

5.7.3. Number of negative samples

To investigate the significance of the number of negative samples, we experiment with the number of negative samples $[20, 30, 40, 50]$ on Kinship. Fig. 5(b) presents the results with the MRR metric. The best performance is achieved with 40 negative samples. The performance becomes worse as the number of negative samples decreases or increases. The main reason is that a smaller number of negative samples may not be enough to train the model effectively, and a larger one may contain some unhelpful negative samples to train the model.

5.8. Ablation study

To test the validity of model components, three variants of D-AEN are introduced to conduct an ablation study on WN18RR and Kinship as follows: (1) RemoveRA: Remove the relation-specific attention mechanism from D-AEN. In this trial, the neighborhood information is fused with equal importance for representation learning of relations. (2) RemoveR: Remove neighborhood information from RemoveRA in representation learning of relations. In this setting, relation representations are self-updated via a learnable matrix. (3) RemoveBR: Remove the bidirectional attention mechanism from RemoveR. In this scenario, we treat neighborhoods as a whole to measure their importance when learning entity representations without encapsulating the direction information of relations.

As shown in Table 8, firstly, RemoveR achieves better performance than RemoveBR on 7 out of 8 metrics, indicating that the bidirectional mechanism has a profound impact on the experimental results. It illustrates the effectiveness of incorporating the direction information of relations into measuring the importance of neighborhoods. Secondly,

Table 9
Predicted examples on the FB15k-237 dataset.

Head entity and relation	Predicted tail entities
(James Madison, organization_founder)	(1) **Democratic Party**; (2) United States Military Academy; (3) Democratic-Republican Party; (4) Episcopal Church.
(National Football League, team)	(1) **Los Angeles Chargers**; (2) Carolina Panthers; (3) New York Jets; (4) Detroit Lions.
(The X-Files, actor)	(1) William B. Davis; (2) **Gillian Anderson**; (3) Cary Elwes; (4) Robert Patrick.
(marriage, location_of_ceremony)	(1) Paris; (2) Sydney; (3) **Las Vegas**; (4) London.

the experimental results manifest that RemoveRA achieves significant improvements over RemoveR on 6 out of 8 metrics. The reason is that incorporating neighborhood information into representation learning of relations can learn more effective relation representations and further make entities and relations interact well semantically. Finally, the performance of D-AEN improves dramatically on all metrics compared with RemoveRA, which confirms that aggregating neighborhood information with different importance benefits representation learning of relations. These observations illustrate the reasonability of all components in D-AEN.

5.9. Case study

To intuitively testify the prediction ability of D-AEN, we leverage the FB15k-237 dataset to conduct a case study. As Table 9 shows, given some head entities and relations, D-AEN predicts four tail entities with the highest scores. The bold indicates the true tail entities in the testing dataset and the underlined represents the correct tails in the training dataset. The results illuminate that D-AEN successfully predicts the correct tail entities even if the true tails are not always at the best rank.

6. Conclusion

This article presents an innovative GCN-based encoder named D-AEN to tackle the KGE task. D-AEN simultaneously learns the representations of relations and entities by aggregating the information

of neighborhoods where both entity and relation representations are utilized to boost the representation learning of each other. Specifically, a bidirectional attention mechanism and a relation-specific attention mechanism are devised to jointly measure the importance of neighborhoods for selectively aggregating neighborhood information, which can make elements in KGs like relations and entities interact well semantically. Extensive experimental results elucidate the superiority of D-AEN against state-of-the-art models. Moreover, the impact of hyperparameters and model components on prediction performance is investigated.

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.

Data availability

The authors are unable or have chosen not to specify which data has been used.

Acknowledgments

This work was supported by the National Natural Science Foundation of China [grant No. 62176239 and No. 61801432], in part by the Key Research and Development and Promotion Special Project of Henan Province (scientific and technological research, China) [grant No. 212102210548].

References

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., & Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *The semantic web* (pp. 722–735). Springer.

Balazevic, I., Allen, C., & Hospedales, T. (2019). TuckER: Tensor factorization for knowledge graph completion. In *2019 conference on empirical methods in natural language processing and 9th international joint conference on natural language processing* (pp. 5184–5193). Association for Computational Linguistics.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T., & Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on management of data* (pp. 1247–1250).

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., & Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. *Advances in Neural Information Processing Systems*, *26*.

Bruna, J., Zaremba, W., Szlam, A., & LeCun, Y. (2014). Spectral networks and deep locally connected networks on graphs. In *2nd international conference on learning representations, ICLR 2014*.

Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E. R., & Mitchell, T. M. (2010). Toward an architecture for never-ending language learning. In *Twenty-fourth AAAI conference on artificial intelligence*.

Chen, L., Tu, D., Lv, M., & Chen, G. (2018). A knowledge-based semisupervised hierarchical online topic detection framework. *IEEE Transactions on Cybernetics*, *49*(9), 3307–3321.

Dai Quoc Nguyen, T. D. N., Nguyen, D. Q., & Phung, D. (2018). A novel embedding model for knowledge base completion based on convolutional neural network. In *Proceedings of NAACL-HLT* (pp. 327–333).

Dettmers, T., Minervini, P., Stenetorp, P., & Riedel, S. (2018). Convolutional 2d knowledge graph embeddings. In *Thirty-second AAAI conference on artificial intelligence*.

Guo, W., Su, R., Tan, R., Guo, H., Zhang, Y., Liu, z., et al. (2021). Dual graph enhanced embedding neural network for CTR prediction. In *Proceedings of the 27th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 496–504). ACM.

Hou, Y., Zhang, J., Cheng, J., Ma, K., Ma, R. T., Chen, H., et al. (2019). Measuring and improving the use of graph information in graph neural networks. In *International conference on learning representations*.

Hu, S., Zou, L., Yu, J. X., Wang, H., & Zhao, D. (2017). Answering natural language questions by subgraph matching over knowledge graphs. *IEEE Transactions on Knowledge and Data Engineering*, *30*(5), 824–837.

Huang, X., Zhang, J., Li, D., & Li, P. (2019). Knowledge graph embedding based question answering. In *Proceedings of the twelfth ACM international conference on web search and data mining* (pp. 105–113).

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448–456). PMLR.

Ji, G., He, S., Xu, L., Liu, K., & Zhao, J. (2015). Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (Volume 1: Long papers)* (pp. 687–696).

Jiang, X., Wang, Q., & Wang, B. (2019). Adaptive convolution for multi-relational learning. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: Human language technologies, Volume 1 (Long and short papers)* (pp. 978–987).

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *ICLR (Poster)*.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International conference on learning representations*.

Li, C., Chen, X., Zhang, Y., Chen, S., Lv, D., & Wang, Y. (2020). Dual graph embedding for object-tag link prediction on the knowledge graph. In *2020 IEEE international conference on knowledge graph* ICKG, (pp. 283–290). IEEE.

Li, F., Li, Y., Shang, C., & Shen, Q. (2019). Fuzzy knowledge-based prediction through weighted rule interpolation. *IEEE Transactions on Cybernetics*, *50*(10), 4508–4517.

Li, Z., Liu, H., Zhang, Z., Liu, T., & Xiong, N. N. (2021). Learning knowledge graph embedding with heterogeneous relation attention networks. *IEEE Transactions on Neural Networks and Learning Systems*.

Li, Q., Wang, D., Feng, S., Niu, C., & Zhang, Y. (2021). Global graph attention embedding network for relation prediction in knowledge graphs. *IEEE Transactions on Neural Networks and Learning Systems*.

Lin, Y., Liu, Z., Sun, M., Liu, Y., & Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *Twenty-ninth AAAI conference on artificial intelligence*.

Lin, X. V., Socher, R., & Xiong, C. (2018). Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 conference on empirical methods in natural language processing* (pp. 3243–3253).

Mahdisoltani, F., Biega, J., & Suchanek, F. (2014). Yago3: A knowledge base from multilingual wikipedias. In *7th Biennial conference on innovative data systems research*. CIDR Conference.

Nathani, D., Chauhan, J., Sharma, C., & Kaul, M. (2019). Learning attention-based embeddings for relation prediction in knowledge graphs. In *Proceedings of the 57th annual meeting of the association for computational linguistics* (pp. 4710–4723).

Nickel, M., Rosasco, L., & Poggio, T. (2016). Holographic embeddings of knowledge graphs. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 30*.

Prakash, S. K. A., & Tucker, C. S. (2021). Node classification using kernel propagation in graph neural networks. *Expert Systems with Applications*, *174*, Article 114655.

Rosa, R. L., Schwartz, G. M., Ruggiero, W. V., & Rodríguez, D. Z. (2018). A knowledge-based recommendation system that includes sentiment analysis and deep learning. *IEEE Transactions on Industrial Informatics*, *15*(4), 2124–2135.

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2008). The graph neural network model. *IEEE Transactions on Neural Networks*, *20*(1), 61–80.

Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., & Welling, M. (2018). Modeling relational data with graph convolutional networks. In *European semantic web conference* (pp. 593–607). Springer.

Shang, C., Tang, Y., Huang, J., Bi, J., He, X., & Zhou, B. (2019). End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 33* (pp. 3060–3067).

Shao, B., Li, X., & Bian, G. (2021). A survey of research hotspots and frontier trends of recommendation systems from the perspective of knowledge graph. *Expert Systems with Applications*, *165*, Article 113764.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, *15*(1), 1929–1958.

Sun, Z., Deng, Z.-H., Nie, J.-Y., & Tang, J. (2018). RotatE: Knowledge graph embedding by relational rotation in complex space. In *International conference on learning representations*.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).

Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., & Bouchard, G. (2016). Complex embeddings for simple link prediction. In *International conference on machine learning* (pp. 2071–2080). PMLR.

Tucker, L. R., et al. (1964). The extension of factor analysis to three-dimensional matrices. *Contributions to Mathematical Psychology*, *110119*.

Vashishth, S., Sanyal, S., Nitin, V., Agrawal, N., & Talukdar, P. (2020). Interacte: Improving convolution-based knowledge graph embeddings by increasing feature interactions. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34* (pp. 3009–3016).

Vashishth, S., Sanyal, S., Nitin, V., & Talukdar, P. (2020). Composition-based multi-relational graph convolutional networks. In *International conference on learning representations*.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., & Bengio, Y. (2018). Graph attention networks. In *International conference on learning representations*.

Wang, Z., Zhang, J., Feng, J., & Chen, Z. (2014). Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 28*.

Wang, H., Zhang, F., Wang, J., Zhao, M., Li, W., Xie, X., et al. (2019). Exploring high-order user preference on the knowledge graph for recommender systems. *ACM Transactions on Information Systems (TOIS)*, *37*(3), 1–26.

Wu, L., Wang, D., Song, K., Feng, S., Zhang, Y., & Yu, G. (2021). Dual-view hypergraph neural networks for attributed graph learning. *Knowledge-Based Systems*, Article 107185.

Xu, K., Hu, W., Leskovec, J., & Jegelka, S. (2018). How powerful are graph neural networks? In *International conference on learning representations*.

Yang, B., Yih, W., He, X., Gao, J., & Deng, L. (2015). Embedding entities and relations for learning and inference in knowledge bases. In *International conference on learning representations*.

Ye, R., Li, X., Fang, Y., Zang, H., & Wang, M. (2019). A vectorized relational graph convolutional network for multi-relational network alignment. In *IJCAI* (pp. 4135–4141).

Zeb, A., Haq, A. U., Zhang, D., Chen, J., & Gong, Z. (2021). KGEL: A novel end-to-end embedding learning framework for knowledge graph completion. *Expert Systems with Applications*, *167*, Article 114164.

Zhang, Z., Cai, J., Zhang, Y., & Wang, J. (2020). Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34* (pp. 3065–3072).

Zhang, Z., Zhuang, F., Zhu, H., Shi, Z., Xiong, H., & He, Q. (2020). Relational graph neural network with hierarchical attention for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence, Vol. 34* (pp. 9612–9619).

Zhao, Y., Zhou, H., Xie, R., Zhuang, F., Li, Q., & Liu, J. (2021). Incorporating global information in local attention for knowledge representation learning. In *Findings of the association for computational linguistics: ACL-IJCNLP 2021* (pp. 1341–1351).