Knowledge-Based Systems 233 (2021) 107498

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

HackRL: Reinforcement learning with hierarchical attention for cross-graph knowledge fusion and collaborative reasoning

Linyao Yang [a,b], Xiao Wang [a,c,*], Yuxin Dai [d], Kejun Xin [c,e], Xiaolong Zheng [a], Weiping Ding [f], Jun Zhang [a,c,d], Fei-Yue Wang [a,c]

[a] *The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China*
[b] *School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China*
[c] *Qingdao Academy of Intelligent Industries, Qingdao 266000, China*
[d] *School of Electrical Engineering and Automation, Wuhan University, Wuhan 430072, China*
[e] *Nanjing Joinmap Data Research Institute, Nanjing 211100, China*
[f] *School of Information Science and Technology, Nantong University, Nantong 226019, China*

ARTICLE INFO

Article history:
Received 11 March 2021
Received in revised form 20 July 2021
Accepted 14 September 2021
Available online 20 September 2021

Keywords:
Knowledge fusion
Knowledge reasoning
Decision-making
Hierarchical graph attention
Reinforcement learning

ABSTRACT

Reasoning aiming at inferring implicit facts over knowledge graphs (KGs) is a critical and fundamental task for various intelligent knowledge-based services. With multiple distributed and complementary KGs, the effective and efficient capture and fusion of knowledge from different KGs is becoming an increasingly important topic, which has not been well studied. To fill this gap, we propose to explore cross-KG relation paths with the anchor links identified by entity alignment for the knowledge fusion and collaborative reasoning of multiple KGs. To address the heterogeneity of different KGs, this paper proposes a novel reasoning model named HackRL based on the reinforcement learning framework, which incorporates the long short-term memory and hierarchical graph attention in the policy network to infer indicative cross-KG relation paths from the history trajectory and the heterogeneous environment for predicting corresponding relations. Meanwhile, an entity alignment-oriented representation learning method is utilized to embed different KGs into a unified vector space based on the anchor links to reduce the impact of distinct vector spaces, and two training mechanisms, action mask and retrain with sampled paths, are proposed to optimize the training process to learn more successful indicative paths. The proposed HackRL is validated on three cross-lingual datasets built from DBpedia on the link prediction and fact prediction tasks. Experimental results demonstrate that HackRL achieves better performance on most tasks than existing methods. This work provides an industrially-applicable framework for fusing distributed KGs to make better decisions.

1. Introduction

Knowledge graphs (KGs) encode unstructured knowledge into structured triples, which provide an effective scheme for describing the complex relationships between concepts and entities. In recent years, with the developments of big data [1,2] and natural language processing, a large number of KGs, including YAGO [3], DBpedia [4], and Freebase [5], have been constructed, which contain millions of facts about real-world entities and relations, e.g., (Joe Biden, president of, USA). Due to the explainable characteristics of KGs [6], they have been widely used in various tasks, such as information retrieval [7], question answering [8, 9], and recommender systems [10–12]. Original KGs are usually constructed with data collected from the Internet or gathered manually; consequently, KGs always encounter the knowledge incompleteness issue even though they are large in size. Links among entities can be lost in a KG, which greatly affects the performances of downstream tasks [13]. Predicting the missing links is crucial to various knowledge-based services, which is referred to as the knowledge reasoning task.

So far, numerous methods have been proposed to cope with the knowledge reasoning problem, which can be classified into three categories: rule-based, embedding-based, and path-based [14–16]. The rule-based methods use first-order predicate logic or ontology to represent concepts, which heavily rely on expert knowledge and are not suitable for large-scale KGs. Therefore, many recent works have focused on reasoning based on

* Corresponding author at: The State Key Laboratory for Management and Control of Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China.

E-mail addresses: yanglinyao2017@ia.ac.cn (L. Yang), x.wang@ia.ac.cn (X. Wang), YuxinDai@whu.edu.cn (Y. Dai), xinkj@joinmap.ai (K. Xin), xiaolong.zheng@ia.ac.cn (X. Zheng), ding.wp@ntu.edu.cn (W. Ding), jun.zhang.ee@whu.edu.cn (J. Zhang), feiyue.wang@ia.ac.cn (F. Wang).

https://doi.org/10.1016/j.knosys.2021.107498
0950-7051/

distributed representations, which embed entities and relations into low-dimensional vector spaces and predict links based on their embeddings. Researchers have proposed a large number of embedding methods based on tensor decomposition [17], translational models [18–20], and semantic matching, which have achieved great improvements in various applications. Despite their impressive results, embedding-based methods are usually hard to explain and cannot model multi-hop relation paths [21]. In contrast, path-based reasoning methods find equivalent relation paths of a relation and use them as features to predict the existence of corresponding relations, which is easier to explain and can handle multi-hop reasoning problems. Recently, researchers have proposed exploring the most indicative relation paths with deep reinforcement learning (DRL) [22–24], which achieves excellent results on the knowledge reasoning task.

However, all of the existing methods conduct reasoning on a single KG. In reality, there are usually multiple KGs regarding a specific domain. A better solution is to associate and fuse multiple KGs to improve the reasoning performance [25,26]. Nevertheless, the only existing research on reasoning over multiple KGs predicts links based on the ensemble of inference results over different aligned KGs [27], which suffers from poor flexibility as it requires the predicted entities to both have pairs in other KGs. Therefore, how to integrate the complementary knowledge contained in distributed KGs effectively and efficiently to help improve each KG's completeness remains a problem, which involves identifying the equivalent knowledge and integrating the complementary knowledge.

To fill this gap, we propose to explore equivalent cross-KG relation paths to integrate complementary relation paths based on the anchor links from the entity alignment, which aims to predict the missing links between different KGs by identifying equivalent entities. The difficulties of learning useful relation paths over multiple KGs are that the feature spaces of different KGs are heterogeneous [25] and the action space is expanded significantly. Existing DRL-based reasoning methods can hardly learn indicative cross-KG relation paths effectively and efficiently under such conditions. To tackle these issues, we propose a novel model for the knowledge fusion and collaborative reasoning of multiple KGs named hierarchical graph attention-enabled cross-knowledge graph reinforcement learning (HackRL), which incorporates long short-term memory (LSTM) [28] and hierarchical graph attention (HGA) to form the policy network to make history-dependent policies based on the comprehensive learning of the heterogeneous environment. To reduce the impacts of feature heterogeneity among different KGs, we first use the graph attention network (GAT) and TransE model [29] to embed entities and relations of different KGs into a unified vector space based on the identified anchor links from entity alignment. The agent is built with the LSTM and HGA-based policy network to find the most indicative cross-KG relation paths. An action mask mechanism is designed to filter out unreasonable actions at each step and increase the success rate of exploration. Meanwhile, to optimize the training process to learn indicative cross-KG paths, we design a novel reward function that jointly encourages accuracy, diversity, and efficiency, and we sample both intra-KG and inter-KG paths for demonstrations [30] to guide the agent's learning when it fails. Our contributions are summarized as follows:

- We are the first to propose exploring cross-KG relation paths for the knowledge fusion and collaborative reasoning of multiple KGs. Our proposed model provides an effective and flexible solution for integrating, unifying, and enhancing KGs to make better knowledge-based decisions.
- A novel reinforcement learning model is proposed, where the collaborative reasoning with fused knowledge is driven by an agent based on the embeddings in the same vector space and well-designed training mechanisms.
- The proposed model is evaluated on three cross-lingual KGs with downstream tasks. Experimental results indicate that the proposed HackRL model outperforms other baseline models and demonstrate the effectiveness of the proposed policy module and optimization method.

The remainder of the paper is organized as follows: Section 2 reviews the related works of entity alignment and knowledge reasoning for KGs; Section 3 provides the details of the proposed HackRL model; Section 4 describes the experimental datasets, settings, and results; Section 5 concludes the paper and shows our future work directions.

2. Related work

The emergence of many heterogeneous, distributed, varied, unconnected, yet complementary KGs has emphasized the importance of associating and integrating multiple KGs to achieve refined and complete KGs. Entity alignment and knowledge reasoning are two fundamental and crucial techniques for achieving this goal. In this section, recent advances in entity alignment and knowledge reasoning are briefly reviewed.

2.1. Entity alignment

The entity alignment problem has been researched for years. The earliest entity alignment approaches utilize hand-crafted features [31] or external sources based on crowdsourcing, which suffer from heavy human efforts and insufficient scalability. Recently, embedding-based methods have become the most popular solutions, which assign counterparts based on the distance between their embedding vectors [32]. Most embedding-based entity alignment approaches can be divided into two groups [33], one of which encodes each KG into separated vector spaces and learns transitions between different vector spaces and the other of which encodes multiple KGs into a unified vector space.

Many embedding-based entity alignment methods [34,35] employ translational models to learn entity representations based on the attribute and relation triples. For example, Chen et al. [34] proposed encoding entities and relations of different KGs into separate vector spaces based on the TransE model [19] and provided translations for each embedding vectors to their counterparts in the other space. Recently, graph convolutional neural network (GCN)-based entity alignment methods have been widely investigated, e.g., GCN-Align [36], KECG [29], and AliNet [37], which use GCNs to capture the semantic relations between entities and encode different KGs into a unified vector space by sharing the weights of different GCNs.

In addition to structural proximity, some approaches [35,38] also incorporate attribute and semantic information to boost entity alignment performance. Trisedya et al. [35] proposed learning attribute embeddings from attribute triples based on the TransE model to capture entities' attribute similarities. Wu et al. [39] proposed a relation-aware dual-graph convolutional network to incorporate relation information via attentive interactions between the KG and its dual KG. To mitigate the limitations of structural information for long-tail entities, Zeng et al. [40] proposed a degree-aware co-attention network to dynamically adjust the importance of structural and name embeddings while calculating entities' similarities. Entity alignment is a typical semi-supervised learning problem with little labeled data. To tackle the issue of insufficient pre-alignment seeds, Sun et al. [41] proposed a bootstrapping method that uses highly confident prediction results to enrich the pre-alignment seeds and iteratively train the model. Zeng et al. [38] proposed adopting the deferred acceptance algorithm (DAA) to solve the one-to-one mapping problem faced by most entity alignment problems. Wu et al. [42] proposed estimating entities' similarities according to the structural and neighborhood differences based on the graph matching method. Tang et al. [43] proposed a BERT-based interaction model (BERT-INT) to leverage the side information, including entities' names, descriptions, and attributes to calculate entities' similarities, which achieves the best performance on public datasets. Based on these prior studies, we can obtain accurate aligned entity pairs as anchor links to connect different KGs.

2.2. Knowledge graph reasoning

To date, many methods have been proposed to tackle the knowledge reasoning problem [44]. First-order inductive learner [45] is a typical rule-based reasoning method that acquires the Horn clauses of specific relations as features for predicting whether a relation exists. Jiang et al. [46] proposed a Markov logic network-based system to combine manually defined rules with probabilistic inference. Yang et al. [47] proposed neural logic programming to learn first-order logical rules in an end-to-end differentiable fashion. Traditional rule-based algorithms achieve high inference accuracy on small-scale KGs; however, they can hardly handle reasoning on large-scale KGs, since the number of inference patterns is exponential to the scale of entities.

In recent years, embedding approaches have also gained a great deal of attention in the knowledge reasoning field. Embedding-based methods capture the distance between entities and relations while projecting them into a continuous and low-dimensional vector space. TransE [19] is a commonly used embedding model that interprets a relation as the translation from its head entity to its tail entity, i.e., $\mathbf{h}+\mathbf{r} \approx \mathbf{t}$ is satisfied for a triple (h, r, t), where $\mathbf{h}$, $\mathbf{r}$, and $\mathbf{t}$ are the embeddings of h, r, and t. TransE has motivated many embedding models due to its simplicity and efficiency; these include TransH [48], TransR [20], and TransD [49]. Tensor decomposition models such as DistMult [17], ComplEx [50], and SimplE [51] and deep learning models such as ConvE [52] are also widely used for knowledge reasoning. However, embedding-based methods can only handle single-hop reasoning [13] and are hard to explain.

For the path-based reasoning methods, Lao et al. [53,54] proposed the classic path-ranking algorithm (PRA) for the link prediction problem. To learn an inference model for a certain relation type, PRA utilizes a random walk-based inference method to find paths that frequently link instance entities and uses these paths as features to train a bilinear regression model. Although the PRA model achieves excellent performance on multi-hop reasoning, it operates in a fully discrete space and lacks the ability to distinguish similar entities and relations. To overcome this problem, a few DRL-based path inference models have been proposed. DeepPath [22] is the first work to bring DRL into the missing link prediction task, which uses a pre-trained policy-based agent to sample the most promising paths. MINERVA [23] improves on DeepPath by incorporating LSTM into the policy network to memorize the path traversed and stop at the right entity. ADRL [55] also incorporates LSTM with an attention module into the DRL-based reasoning model to improve efficiency, generalization, and interpretability. Lin et al. [56] proposed improving these two models by introducing action dropout and reward shaping. Li et al. [21] proposed a multi-agent reinforcement learning method that uses two agents to conduct relation selection and entity selection iteratively for path-finding. AttnPath [24] incorporates LSTM with a graph attention network as the memory component in the policy network.

All of the above reasoning methods learn and predict relation links within a single KG. So far, although there are many entity alignment methods that significantly improve the ability to identify equivalent knowledge objects, very little attention has been paid to integrating complementary knowledge to boost knowledge reasoning.

3. The proposed HackRL

This section describes the concept, framework, and learning process of the proposed HackRL model. An overview of the task formulation and the related notations is given. The components of the proposed reinforcement learning framework are then introduced. Finally, the training and optimization approach is designed on the basis of the reinforcement learning framework.

3.1. Problem formulations and notations

A knowledge graph $\mathcal{G}=(\mathcal{E},\mathcal{R},\mathcal{T})$ consists of a set of entities $\mathcal{E}$, a set of relations $\mathcal{R}$, and a set of triples $\mathcal{T}$. $e\in\mathcal{E}$ is an entity, $r\in\mathcal{R}$ is a relation, and $(e_o, r, e_t)\in\mathcal{T}$ is a triple that points the head entity e_o to the tail entity e_t. Without loss of generality, we consider the knowledge fusion and collaborative reasoning of two KGs, i.e., $\mathcal{G}_1=(\mathcal{E}_1,\mathcal{R}_1,\mathcal{T}_1)$ and $\mathcal{G}_2=(\mathcal{E}_2,\mathcal{R}_2,\mathcal{T}_2)$, aiming to predict the missing element of ? given a query among three cases, $(?, r, e_t)$, $(e_o, ?, e_t)$, and $(e_o, r, ?)$. e_o and e_t are the head entity and the tail entity not directly connected within a single KG. Instead, there may be some long intra-KG and inter-KG inference paths $\left(e_o \xrightarrow{r_1} e_1 \xrightarrow{r_2} e_2 \cdots \xrightarrow{r_m} e_t\right)$, where e_i is the i-th entity and r_i is the i-th relation in the path.

Since the primary goal of this paper is to develop a model to resolve the path inference problem over multiple heterogeneous KGs, i.e., to automatically infer promising cross-KG relation paths to indicate the existence of specific relations, we identify equivalent entities based on the entity alignment and add the highly confident predicted entity pairs to the anchor links to connect the KGs. Based on this, we can fill the missing link *athletePlaysInLeague* between X and Z if both $(X, \mathit{atheletePlaysForTeam}, Y)$ and $(Y, \mathit{teamPlaysInLeague}, Z)$ exist in $\mathcal{G}_1$, which can also be inferred from $\left(X, \mathit{Identical}, X'\right)$, $\left(Z, \mathit{Identical}, Z'\right)$ between $\mathcal{G}_1$ and $\mathcal{G}_2$, and $\left(X', \mathit{atheletePlaysInLeague}, Z'\right)$ in $\mathcal{G}_2$. Recently, DRL-based methods have achieved impressive performance on the knowledge reasoning task, in which the path-finding problem is formulated as a Markov decision process (MDP). The MDP is defined as a 4-tuple $(\mathcal{S},\mathcal{A},\mathcal{P},\mathcal{R})$ [57–59], where $\mathcal{S}$ is the continuous states, $\mathcal{A}$ is the available actions, $\mathcal{P}\left(\mathcal{S}_{i+1}=s'|\mathcal{S}_i=s,\mathcal{A}_i=a\right)$ is the state transition function, and $\mathcal{R}(s,a)$ is the reward function of each (s,a) pair.

3.2. RL framework of HackRL

The overview of the proposed HackRL model is shown in Fig. 1. HackRL is framed on reinforcement learning, where a policy network is formed to infer indicative paths for a relation throughout the environment. At each step, the agent takes an action to choose a relation from the action space based on the history trajectory and current state encoded by LSTM and HGA network. If the agent selects a valid action, it will move forward to extend the relation path; otherwise, it will stay at the origin and be punished. The detailed framework, the policy network, and the training method are described as follows.

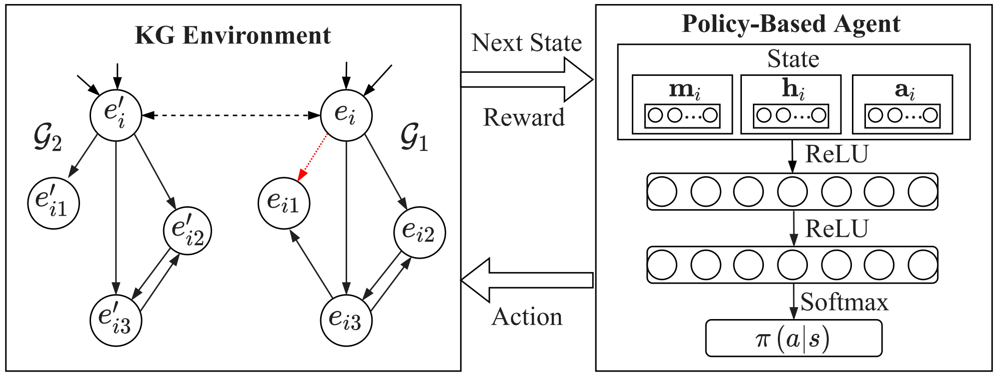


Fig. 1. The reinforcement learning framework of the proposed HackRL.

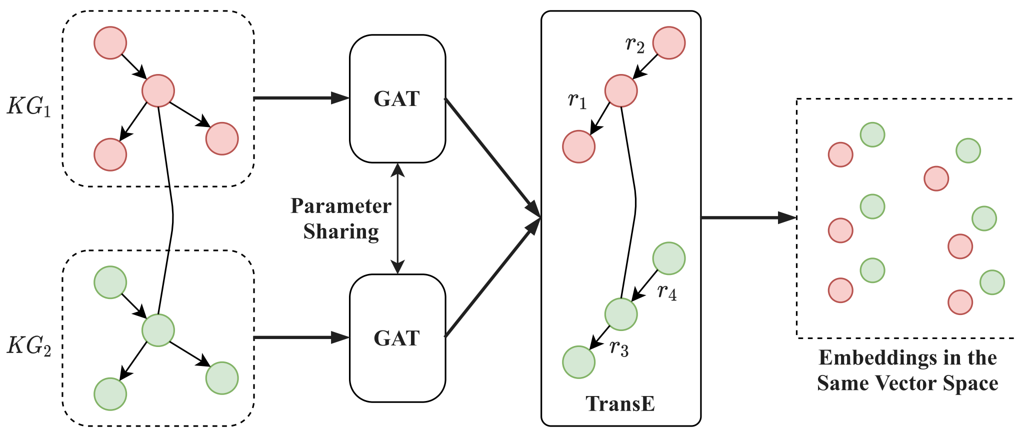


Fig. 2. The embedding module for learning representations of entities and relations.

3.2.1. KG environment

For the case of two aligned KGs, the environment refers to the whole $\mathcal{G}_1$, $\mathcal{G}_2$ as well as their anchor links, excluding the query relation triples. We identify equivalent entities based on the state-of-the-art entity alignment method BERT-INT [43]. Specifically, BERT-INT uses a BERT model as a basic representation unit to embed the name, description, attribute, and value of an entity, which is highly discriminative information for distinguishing entities. The embeddings are then inputted into an interaction module to compute the name/description-view, neighbor-view, and attribute-view interactions. The matching scores between different entity pairs are aggregated based on the interaction features. Based on the matching scores, an entity is assigned to the candidate that possesses the highest matching score in another KG. We refer the reader to [43] for more information about the implementation details. Based on this method, we achieve more than 96% alignment accuracy for the datasets used in this study. Therefore, the prediction results are all added into the anchor links. However, this method may be not suitable for entity alignment of KGs without explicit and informative side information, in such cases, the alignment results may need to be carefully checked.

To facilitate path-finding, for each triple (e_o, r, e_t), we add the inverse triple $\left(e_t, r^{-1}, e_o\right)$ to the datasets. Additionally, each aligned entity pair $\left(e, e'\right) \in \mathcal{I}$ is formed as a triple with the relation *Identical*, where $\mathcal{I}$ denotes all of the aligned entity pairs. The environment remains the same throughout the training process for a given query relation r_q.

Since the agent works on continuous vector space, we need to learn continuous representations for entities and relations for the learning of the DRL model. Since the feature spaces of different KGs are distinct and heterogeneous in nature, we need to map entities and relations of different KGs into the same vector space. For the inter-connected KG pair, we learn embeddings in a unified vector space based on the graph attention network (GAT) and TransE model, following [29]. The framework of the embedding module is shown in Fig. 2. We first embed entities of different KGs into a unified vector space based on a pair of GAT models with parameter sharing. The loss function of the GAT models is as follows:

$$\mathcal{L}_G = \sum_{(e_i, e_j \in S)} \sum_{\left(e'_i, e'_j \in S'\right)} \max\left(0, \left\|\mathbf{e}_i - \mathbf{e}_j\right\|_2 + \gamma_1 - \left\|\mathbf{e}'_i - \mathbf{e}'_j\right\|_2\right), \quad (1)$$

where $\|\cdot\|_2$ is the L_2 distance, S is the set of positive aligned entity pairs, S' is the set of negative entity pairs generated by negative sampling [29], and γ_1 is a margin hyper parameter. Since the GAT model cannot learn embeddings of relations, we then utilize the TransE model to embed relations with the objective of minimizing the following loss function:

$$\mathcal{L}_T = \sum_{(e_o, r, e_t) \in T} \sum_{\left(e'_o, r', e'_t\right) \in T'} \max\left(0, f\left(e_o, r, e_t\right) + \gamma_2 - f\left(e'_o, r', e'_t\right)\right), \quad (2)$$

where $f\left(e_o, r, e_t\right) = \|\mathbf{e}_o + \mathbf{r} - \mathbf{e}_t\|_2$, T is the positive triple set, and T' is the negative triple set generated from corrupted T following TransE [19]. With the above process, the embeddings of entities and relations that encode their semantic relations in a unified vector space are obtained.

3.2.2. MDP

The MDP contains state, action, transition, and reward. Given the KG environment, at each step, the agent learns to select a

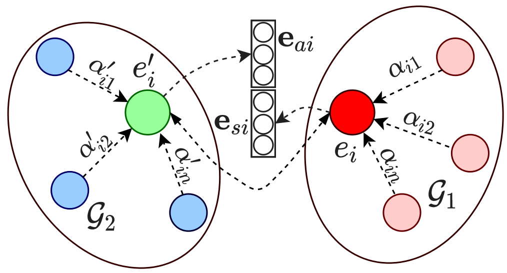


Fig. 3. The hierarchical graph attention mechanism.

promising relation r_i to extend the relation path based on the current state s_i. Then, the agent may walk from entity e_{i-1} to e_i according to the transition rules $\mathcal{P}$. At the end of each episode, the agent will receive the rewards $\mathcal{R}$ for its actions. The details of each part are as follows.

State: In the proposed reinforcement learning framework, the state encodes the agent's location information in the environment with a fixed-length vector, which is composed of three parts: the entity embeddings, the LSTM-encoded history trajectory embedding, and the hierarchical graph attention embedding. Thus, the state vector at step i is defined as follows:

$$\boldsymbol{s}_i = [\boldsymbol{m}_i; \boldsymbol{h}_i; \boldsymbol{a}_i], \tag{3}$$

where $\boldsymbol{h}_i$ is the history trajectory embedding, $\boldsymbol{a}_i$ is the hierarchical graph attention embedding, and $\boldsymbol{m}_i$ is the entity embedding part, which is defined as

$$\boldsymbol{m}_i = [\mathbf{e}_i; \mathbf{e}_t - \mathbf{e}_i], \tag{4}$$

where $\mathbf{e}_i$ and $\mathbf{e}_t$ are the embeddings of the current entity and the target entity. $\mathbf{e}_t - \mathbf{e}_i$ is used to capture the distance between the target entity and the current entity. [;] denotes the concatenation operation.

We adopt LSTM in the policy network so that the agent can retain its experience and history trajectory. In the proposed model, a three-layer LSTM is utilized. The search history consists of the sequence of states and actions taken up to step i, which is defined as follows:

$$\boldsymbol{h}_i = \text{LSTM}(\boldsymbol{h}_{i-1}, [\mathbf{r}_{i-1}; \mathbf{m}_i]), \tag{5}$$

where $\boldsymbol{h}_i$ denotes the hidden state at step i, the initial hidden state $\boldsymbol{h}_0$ is set to a zero vector, and the input of LSTM at step i is composed of the embedding of the action taken at step $i-1$ (i.e., $\mathbf{r}_{i-1}$) and the entity embedding at step i, respectively.

KGs consist of entities and semantic relations, and multiple aligned KGs also consist of many anchor links across them. Therefore, multiple aligned KGs have a hierarchical structure. For an entity e_i in $\mathcal{G}_1$ with aligned entity e'_i in $\mathcal{G}_2$, e_i and e'_i are semantically identical, and neighbors of e'_i in $\mathcal{G}_2$ are differentially informative, since they have different characteristics. Therefore, we utilize the hierarchical graph attention embedding to make the agent focus more on relations and connected entities highly related to the query relations in the heterogeneous environment. The illustration of the hierarchical graph attention is shown in Fig. 3. The hierarchical graph attention includes two levels of attentions, i.e., the node level and the network level, to make the agent pay more or less attention to the neighbors in the same KG or the equivalent entity in another KG while choosing actions. These two levels of attention mechanism are formally called intra-KG graph attention and inter-KG graph attention. The intra-KG graph attention encodes the one-hop neighbors' information by means of the following equation:

$$\mathbf{e}_{si} = \sum_{j \in N_s} \alpha_{ij} \cdot \boldsymbol{W}\mathbf{e}_j, \tag{6}$$

where $\boldsymbol{W}$ is a linear transformation matrix and N_s denotes the intra-KG neighbors of e_i. α_{ij} is the weight of the j-th neighbor, which is calculated using a single-layer self-attention neural network. Specifically, following the graph attention model [60], the attention weight between the i-th entity and the j-th entity is calculated as follows:

$$a_{ij} = \text{LeakyReLU}\left(\boldsymbol{q}^T\left[\boldsymbol{W}\mathbf{e}_i; \boldsymbol{W}\mathbf{e}_j\right]\right), \tag{7}$$

where $\boldsymbol{q}$ is a learnable weight vector that is shared by all entities. LeakyReLU is the nonlinear activation function with a negative input slope of 0.2. After obtaining the attention weight for entity e_i to all its intra-KG direct-connected neighbors, the normalized attention weight is calculated by a Softmax function as follows:

$$\alpha_{ij} = \frac{\exp(a_{ij})}{\sum_{k \in N_s} \exp(a_{ik})}. \tag{8}$$

With the intra-KG attention mechanism, HackRL can pay more attention to the neighbors that are more promising and extend the relation path within the same KG.

Unlike the intra-KG graph attention that targets the direct neighbors of e_i in the same KG, the inter-KG graph attention pays attention to the neighbors of e_i's aligned entity to judge the gains of jumping to the aligned KG to learn cross-KG relation paths. The anchor links connecting different KGs play a crucial role in the fusion of cross-KG knowledge. In a similar way to the intra-KG graph attention, the weighted combination of the neighbors of e'_i, $\mathbf{e}_{ai}$, is computed by Eqs. (6)–(8) with a different transformation matrix and weight vector. If the current entity does not have aligned pair in the other KG, the $\mathbf{e}_{ai}$ is set as a zero vector. To retain more information of the heterogeneous environment, the intra-KG graph attention vector and the inter-KG graph attention vector are directly concatenated to form the hierarchical graph attention embedding, i.e., $\boldsymbol{h}_i = [\mathbf{e}_{si}; \mathbf{e}_{ai}]$.

Action: For the collaborative reasoning task, an action refers to a relation to forward the path, which can be taken from all the relation types across the environment except the query relation and its inverse relation in our framework. The agent chooses the most promising relations according to its observation of the current state. The action r_i taken by the agent at step i can be valid or invalid. If e_i has outgoing edges with type r_i, the action is valid; otherwise, it is an invalid action.

Transition: The transition $\mathcal{P}$ is used to model the probability distribution of the next state, which is defined as a mapping function, $\mathcal{P} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$. In our DRL framework, the transition involves randomly selecting a tail entity to forward given a valid action. That is, if there are m tail entities connected with the current entity by the valid chosen relation, the agent will randomly choose one of them with equal probabilities to forward the relation path.

Reward: Reward is an indicator of the chosen action's effectiveness and evaluates the qualities of inferred relation paths in reinforcement learning. The reward function in HackRL is a weighted sum of the global accuracy, path efficiency, path diversity, and cross-KG efficiency to encourage the agent to learn more indicative paths.

Following the reward shaping mechanism proposed by [56], the global accuracy reward R_{acc} is defined as follows: if the path reaches e_t, the reward is $+1$; if the path does not reach the ground truth, we borrow the idea of TransE to calculate the reward shaping function, that is, $R_{acc} = -\left\|\mathbf{e}_o + \mathbf{r}_q - \mathbf{e}_i\right\|_1$, where $\|\cdot\|_1$ denotes the L_1 norm; if the agent chose an invalid action, the reward for this action is -1. For the actions in a successful episode, the path efficiency reward R_{eff} and the path diversity reward R_{div} follow the settings of DeepPath [22]. Specifically, if

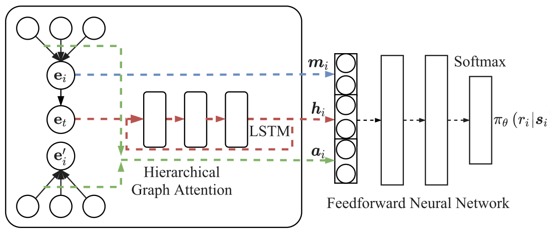


Fig. 4. The architecture of the policy network.

we consider a successful relation path p composed of a sequence of relations, the efficiency reward is defined as follows:

$$R_{eff} = \frac{1}{length\,(p)}. \tag{9}$$

To encourage the agent to learn more different paths, the diversity reward function is defined as follows:

$$R_{div} = -\frac{1}{|F|}\sum_{i=1}^{F} \cos\left(\mathbf{p}, \mathbf{p}_i\right), \tag{10}$$

where $\cos(\cdot)$ is the cosine function, $|F|$ is the number of successful inferred paths, and $\mathbf{p} = \sum_{i=1}^{n} \mathbf{r}_i$ denotes the sum of all embeddings of the relations in the path.

In order to avoid long paths and duplicated paths caused by multiple jumps between different KGs, we design a cross-KG path efficiency reward to balance the gains from cross-KG paths and the losses from long paths, which is defined as follows:

$$R_{cp} = \frac{1}{number\,(CKG)}, \tag{11}$$

where $number\,(CKG)$ is the number of the *Identical* action in a successful path.

Thus, the total reward for a successful epoch is:

$$R_{total} = \lambda_1 R_{acc} + \lambda_2 R_{eff} + \lambda_3 R_{div} + \lambda_4 R_{cp} \tag{12}$$

where λ_i is the weight parameter with $\sum \lambda_i = 1$. In contrast, the reward for a unsuccessful epoch that reaches a false entity is the global accuracy reward with reward shaping, i.e., $R_{shaping} = -\left\|\mathbf{e}_o + \mathbf{r}_q - \mathbf{e}_t\right\|_1$.

Neural policy network: In the proposed HackRL model, the agent needs to select promising relations based on the state. Therefore, to generate the combined state vector from the observable environment, we incorporate the LSTM and HGA modules into the policy component. The overall architecture of the policy component is shown in Fig. 4. In each step, the LSTM network encodes the traversed relation paths and generates the history trajectory embedding, while the HGA network calculates the hierarchical graph attention embedding of the current entity. These two embeddings are then concatenated with $\boldsymbol{m}_i$ to generate the state vector $\boldsymbol{s}_i$. The inputs of the policy component are all explicit embeddings from the environment.

The state vector is then forwarded to a three-layer fully connected neural network with two hidden layers and a Softmax layer to parameterize the policy function $\pi_\theta(r_i|\boldsymbol{s}_i)$ to map the state vector to a probability distribution over all possible relations. The action is then taken based on the output probability. At the end of each epoch, the parameters of the three kinds of neural modules are collectively trained with the supervision from the rewards.

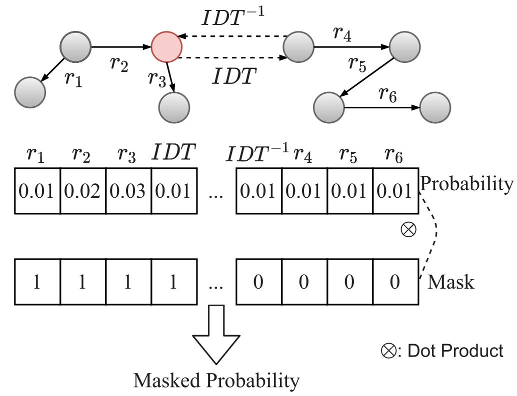


Fig. 5. Illustration of the action mask mechanism.

3.3. Training and optimization

For each path with a sequence of relations, we want to find parameters θ that maximize the expected reward:

$$J(\theta) = \mathbb{E}_{r_i \sim \pi_\theta} \sum_{i}^{T} R\left(r_i, s_i\right), \tag{13}$$

where $R(r_i, s_i)$ is the reward of selecting relation r_i at state s_i and T is the maximum number of explorations of each episode. The optimization is done using the Monte-Carlo Policy Gradient (REINFORCE) algorithm, and the gradients of the model parameters are calculated using the following equation:

$$\nabla_\theta J(\theta) \approx \nabla_\theta \sum_{i=1}^{T} R\left(r_i, s_i\right) \log \pi_\theta\left(r_i|s_i\right) \tag{14}$$

where $\pi_\theta(r_i|s_i)$ denotes the probability of the chosen action.

For the task of collaborative KG reasoning to fuse knowledge of multiple distributed KGs, the training suffers from the increasing number of relations and heterogeneous feature spaces. To optimize the policy network for finding both intra-KG and inter-KG paths, we propose two mechanisms for the training of HackRL with the algorithm of reinforcement learning, i.e., action mask (AM) and retrain with sampled paths (RT). We directly train the model based on the path-finding process without pre-training or fine-tuning unless the agent fails to find a successful path.

Action mask: At each step, the agent can only locate at a KG. Thus, it can only choose relations of the current KG or the cross-KG action. Therefore, we design an action mask mechanism to filter out half of the relations in the action space. As shown in Fig. 5, we set a mask for the agent at each step whose length equals the size of the action space. According to the KG that the current entity belongs to, the corresponding values of the current KG's relations are filled with 1, and other values are set to 0. Note that we assume that the *Identical* relation belongs to $\mathcal{G}_1$ and its inverse, $Identical^{-1}$, belongs to $\mathcal{G}_2$. At each step, the output probability of the policy network will be multiplied by the action mask first and then be renormalized as the output probabilities of possible actions. With this mechanism, the impossible actions are filtered out at each step to improve the probability of finding successful paths.

Retrain with sampled paths: When the agent fails to choose a successful path to the target entity, we not only punish it but also sample a feasible path with a biased random walk for a retraining demonstration to guide the agent to optimize its policy. Because we want the agent to find not only diverse intra-KG paths but also inter-KG paths, we generate both intra-KG and inter-KG teaching paths. The intra-KG paths are generated following

Algorithm 1: Agent's learning algorithm

Input: Query relation r_q, aligned knowledge graphs $\mathcal{G}_1$, $\mathcal{G}_2$, training set $\mathcal{T}_{r_q}$
Output: Parameters of the HackRL θ, successful paths for relation reasoning

Construct the training environment $\mathcal{K}$ by removing the triples containing r_q and r_q^{-1};
Select a triple $\left(e_o, r_q, e_t\right)$ from $\mathcal{T}_{r_q}$ for training;
Initialize the hyperparameters, including the dropout rate, the policy network weights, and the reward weights;
for *epoch* $\leftarrow 1$ *to* N **do**
  Initialize the LSTM's hidden state $\mathbf{h}_0$ to $\mathbf{0}$, and the step i to 0;
  while $i < max\_steps$ **do**
    Compute the state vector $\mathbf{s}_i$;
    Compute $\pi_\theta\left(r_i \mid s_i\right)$;
    Action Mask;
    Action Dropout and recompute the normalized probability distribution of different relations $\hat{\pi}_\theta\left(r_i \mid s_i\right)$;
    Randomly sample a relation according to the $\hat{\pi}_\theta\left(r_i \mid s_i\right)$;
    if r_i *is invalid* **then**
      Add $\langle s_i, r_i\rangle$ to the negative action set $\mathcal{M}_{neg}$;
      Force Forward;
      $i \leftarrow i+1$;
    else
      Add $\langle s_i, r_i\rangle$ to the positive action set $\mathcal{M}_{pos}$;
      Update $\mathbf{s}_i$ to $\mathbf{s}_{i+1}$;
      $i \leftarrow i+1$;
    if *reach* e_t *or* $i = max\_steps$ **then**
      break;
  Update θ with the gradient of the invalid actions using $\sum_{\mathcal{M}_{neg}} log\pi_\theta\left(r_i \mid s_i\right)(-1)$;
  if *reach* e_t **then**
    Calculate R_{total};
    Update θ with the gradient of the valid actions using $\sum_{\mathcal{M}_{pos}} log\pi_\theta\left(r_i \mid s_i\right) R_{total}$;
  else
    Reward shaping;
    Update θ with the gradient of the valid actions using $\sum_{\mathcal{M}_{pos}} log\pi_\theta\left(r_i \mid s_i\right) R_{shaping}$;
    Sampling intra-KG and inter-KG paths to retrain and update θ;

the path sampling method of the supervised policy learning in DeepPath [22]. Specifically, for the failed sample (e_o, e_t), we randomly select an intermediate entity e_{inter} and then carry out two breadth-first searches (BFS) between (e_o, e_{inter}) and (e_{inter}, e_t) to sample a concatenated path between e_o and e_t. The purpose of conducting such path sampling instead of direct BFS between e_o and e_t is to encourage diverse sample paths by preventing preferred shortest paths. For the inter-KG paths, the space of the candidate intermediate entities is greatly enlarged. We analyze the normalized correlation coefficients between the existence of links and the length of cross-KG paths founded by AttnPath over aligned KGs, as shown in Fig. 6. We can see that the short and direct cross-KG paths are more relevant to the existence of corresponding links. Therefore, we want the inter-KG paths to be short and directly connect the queried entities without many

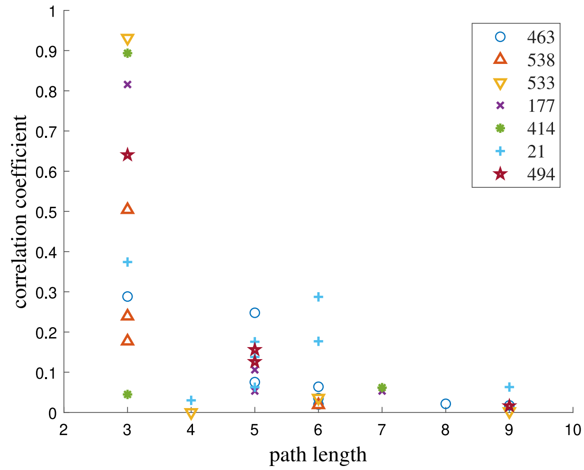


Fig. 6. Visualization of the length–correlation coefficient of cross-KG paths for a few relations.

other appended relations. To achieve this, we adopt a simple trick to sample short and direct inter-KG paths. Instead of randomly picking an intermediate entity from the entire entity set, we randomly pick an intermediate entity e_{inter} from the aligned entities of e_o, e_t, and their one-hop neighbors. We then conduct two BFS between (e_o, e_{inter}) and (e_{inter}, e_t) to obtain concatenated inter-KG sample paths.

The sampled paths are then utilized to train the policy network in a supervised way (the same as DeepPath [22]). For supervised learning, the rewards for the successful sampled paths is +1, and the gradient to optimize the policy network is as follows:

$$\nabla_\theta J(\theta) = \sum_i log\pi_\theta\left(a = r_{si} | s_i\right), \tag{15}$$

where r_{si} is the i-th relation in the sampled path. In this way, the parameters are optimized to maximize the correct relations being selected at each step of the sampled paths. With the retrain with sampled paths mechanism, the agent learns quickly from the ground truth demonstration paths for finding successful paths.

The learning process of the agent is summarized in Algorithm 1. At each step, the agent first calculates the probability of each relation being selected and utilizes the action mask and action dropout mechanism to filter out some irrational relations. Then, a relation is selected to forward the relation path. Depending on whether the action is valid, the agent will receive different rewards and observe different new states, and the actions taken and the states observed while making actions will be recorded to corresponding positive and negative action sets. This process will continue until the agent has arrived at the correct entity or reaches the maximum exploration number. The policy network will be trained based on the state and action sets with their respective rewards.

The overall process of the collaborative knowledge reasoning is shown in Fig. 7, which consists of five main steps: first, we identify equivalent anchor links based on the BERT-INT model; based on the anchor links, we learn embeddings of entities and relations in a unified vector space with the objective of minimizing the embedding distance between equivalent entities; we then construct the environment based on the KGs and their anchor links, and we employ the agent to explore the indicative cross-KG relation paths with our designed policy network and training mechanism; finally, we use the inferred relation paths to train a prediction model to judge the existence of corresponding query relation between query entities. Specifically, we construct a

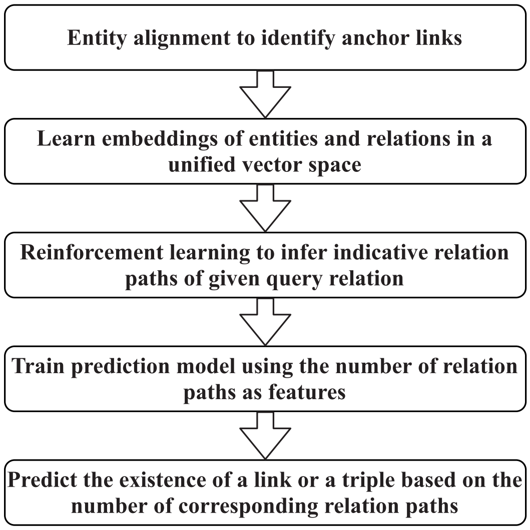


Fig. 7. The flowchart of agent's path-finding process.

Table 1
Statistics of BDP15K dataset.

Datasets		Entities	Relations	Triples
ZH-EN	Chinese	66,469	2,830	153,929
	English	98,125	2,317	237,674
JA-EN	Japanese	65,744	2,043	164,373
	English	95,680	2,096	233,319
FR-EN	French	66,858	1,379	192,191
	English	105,889	2,209	278,590

three-layer feedforward neural network as the prediction model, with its input dimension being the number of relation path types for the query relation. Each dimension of the input vector represents the number of relation paths of corresponding type between given entity pairs. The prediction model is trained with positive samples of the query relation and negative samples generated by randomly corrupting the tail entity based on the binary cross entropy loss.

4. Experiments and discussion

In this section, we describe the experiments in detail to verify the effects of fusing knowledge from multiple KGs on knowledge reasoning and the effectiveness and efficiency of our model. We first describe the datasets and parameter settings we used in the experiments and then carry out a series of experiments. We prove that our model's overall performance is better than traditional embedding-based methods, random walk methods, and other baseline methods.

4.1. Datasets and settings

The experiments are based on three large-scale cross-lingual datasets from DBP15K. The statistics of these three datasets are shown in Table 1 [36]. All of them are subsets of larger datasets. In these real-world KGs, only a few entities and relations are densely connected, and hundreds or thousands of relations have no more than ten triples in the datasets. We refer to these relations as long-tail relations and observe that they destroy the performance of collaborative reasoning. Therefore, we only select the triples with the top-200 relations. For each triple (h, r, t), we add the inverse triple $\left(t, r^{-1}, h\right)$ to the KGs to facilitate path-finding, allowing the agent to step back.

We conduct reasoning for a relation of one KG at a time. For the task r_q, we remove all the triples with r_q or r_q^{-1} from the KG. The removed triples are split into two sets, with 30% being used as training samples and 70% as testing samples. For each sample, we generate 100 negative samples by changing the tail entity with its n-hop nearest neighbors. The following is a summary of the hyperparameters in our model. For the embedding model, we follow the paramter settings in KECG [29] except that the embedding dimension of entities and relations is set as 100. The hidden dimension of LSTM is set to 200. The dimension of HGA is 200. Thus, the dimension of the state vector is 600 with three kinds of embeddings. The dimension of the first hidden layer of the feedforward neural network in the policy network is 512, while that of the second hidden layer is 1024. The output dimension of the policy network is 802. For the reward function, the weights of different reward functions are set by grid search as follows: λ_1 is 0.1, λ_2 is 0.7, λ_3 is 0.1, and λ_4 is 0.1. The action dropout rate is set to 0.3. We use Adam [61] to optimize the parameters of the policy network. In each epoch, the agent is allowed to explore a path for no more than $max\_steps = 50$ steps. We conduct experiments on a personal station with an Intel (R) Xeon(R) CPU E5-2630 v4 @ 2.2 GHZ and GPU TITAN RTX (32G).

Following previous studies, we compare with the commonly used methods based on embeddings or paths. For the embedding-based knowledge reasoning methods, we select two state-of-the-art methods designed for graph completion, TransE [19] and TransR [20]. The implementations of these embedding-based methods are based on the OpenKE toolkit released by [62]. For path-based methods, we select PRA, DeepPath, and AttnPath to compare with our model. The implementations of the path-based methods are based on the code released by their authors. To evaluate the effects of integrating knowledge of aligned KGs and the models' ability to explore useful complementary knowledge, we apply them to both single KGs and aligned KGs. The TransE, TransR, PRA, DeepPath, and AttnPath implemented on aligned KGs are named MTransE, MTransR, MPRA, MDeepPath, and MAttnPath for simplicity.

4.2. Results

In accordance with previous works [22,24], the metrics used to evaluate the ability of the proposed model are the mean average precision (MAP) and the mean success rate of finding paths (MSR). The MSR indicates the average success rate of finding paths for different relations on a dataset. For the link prediction task, each test sample is considered a query like $(h, r, ?)$, and the candidate target entities are ranked according to their predicted confidence. For the fact prediction task, the positive and negative triples are directly ranked.

4.2.1. Results of link prediction

For the positive and negative samples with the head entity h and relation r, we use all the inferred paths of r to train a neural network classifier with one hidden layer, and rank all the tail entities according to their prediction scores. Tables 2–4 show the MAP results of different models on specific prediction tasks of the three datasets. It can be seen that HackRL benefits a lot from the knowledge of aligned KGs and outperforms other methods on most link prediction tasks. From these cases, we can see that embedding-based methods perform worse than path-based methods and fail to gain much from the aligned KGs on most query relations. This is because embedding-based methods can only carry out single-hop reasoning, which can hardly utilize the multi-hop information in aligned KGs, and the increase of entities and relations reduces their learning efficiencies. Path-based methods are much easier to perform better on collaborative

Table 2
Link prediction results (MAP) on the ZH-EN dataset.

Tasks	TransE	MTransE	TransR	MTransR	PRA	MPRA	DeepPath	MDeepPath	AttnPath	MAttnPath	HackRL
463	0.4250	0.4115	0.4384	0.5056	0.3315	0.8540	0.5136	0.5942	0.4602	0.6597	**0.9249**
343	0.2910	0.2353	0.2072	0.1914	0.5968	0.6267	0.8070	0.8609	0.7177	0.7074	**0.8907**
538	0.2501	0.2631	0.2424	0.2654	0.4030	0.4042	0.6562	0.5761	0.3869	0.5570	**0.7249**
533	0.2180	0.1524	0.1408	0.1635	0.8265	0.8923	0.9239	0.8878	0.8310	0.8986	**0.9224**
102	0.3121	0.3089	0.2948	0.3113	0.4789	0.4742	0.8542	0.6310	0.5941	0.8666	**0.9327**
177	0.9060	0.9156	0.8708	0.8922	0.9627	0.9842	0.9942	**0.9986**	0.9914	0.9950	**0.9986**
414	0.4031	0.4005	0.3763	0.3374	0.6313	0.5612	0.8583	0.8132	0.7707	0.9527	**0.9652**
21	0.7990	0.8574	0.8203	0.8121	0.8217	0.8422	0.8536	**0.9882**	0.8529	0.9681	0.9708
75	0.4104	0.4074	0.4022	0.3572	0.7838	0.7844	0.9382	0.9436	0.9497	0.9446	**0.9774**
494	0.5245	0.4474	0.4402	0.3890	0.6905	0.6400	0.8817	0.8513	0.7866	0.8857	**0.8971**

Table 3
Link prediction results (MAP) on the JA-EN dataset.

Tasks	TransE	MTransE	TransR	MTransR	PRA	MPRA	DeepPath	MDeepPath	AttnPath	MAttnPath	HackRL
496	0.5118	0.5402	0.5424	0.5031	0.6897	0.6674	0.8986	0.8934	0.7411	0.8642	**0.9046**
87	0.1792	0.1744	0.1639	0.1704	0.3806	0.3534	0.8988	0.8768	0.6256	0.7438	**0.8992**
441	0.0969	0.0996	0.0902	0.0942	0.8634	0.8975	0.8668	0.8654	0.8912	0.8936	**0.9126**
1173	0.7959	0.7986	0.7998	0.7711	0.8516	0.8767	0.8305	0.9139	0.8163	0.8568	**0.9321**
179	0.1241	0.1231	0.0904	0.0993	0.8701	0.9082	0.9422	0.8856	0.9012	0.9396	**0.9481**
824	0.6931	0.7079	0.6634	0.6972	0.8649	0.9151	0.9474	0.9518	0.9040	0.9314	**0.9544**
506	0.1022	0.0945	0.0986	0.1084	0.8676	0.8986	0.8954	**0.9827**	0.9075	0.9365	0.9430
351	0.1041	0.0874	0.0924	0.0904	0.8941	**0.9516**	0.8835	0.9443	0.9357	0.9337	0.9483
322	0.1042	0.0991	0.1034	0.1002	0.8824	**0.9789**	0.9362	0.9718	0.9065	0.9374	0.9427
119	0.1021	0.0966	0.0968	0.0885	0.8975	0.8263	0.8955	0.8575	0.8879	0.9145	**0.9189**

Table 4
Link prediction results (MAP) on the FR-EN dataset.

Tasks	TransE	MTransE	TransR	MTransR	PRA	MPRA	DeepPath	MDeepPath	AttnPath	MAttnPath	HackRL
844	0.1953	0.1406	0.1781	0.1627	0.8459	0.8983	0.8546	0.8790	0.8351	0.8303	**0.8886**
41	0.5006	0.5328	0.4566	0.3582	0.7481	0.8818	0.8858	0.9021	0.8857	0.9799	**0.9905**
110	0.1382	0.1137	0.1462	0.1495	0.7593	**0.8777**	0.8027	0.8209	0.8254	0.8159	0.8319
552	0.1079	0.1180	0.1093	0.1021	0.2605	0.2624	0.2678	0.2519	0.2719	0.4765	**0.5464**
573	0.2704	0.2293	0.2802	0.2947	0.4627	0.4897	0.2808	0.4741	0.4413	0.4665	**0.5804**
900	0.1130	0.1522	0.1459	0.1456	0.7817	0.8193	0.7897	0.8089	0.8032	0.8211	**0.8448**
150	0.1485	0.1474	0.1398	0.1373	0.3623	0.4803	0.4271	0.5012	0.4278	0.6914	**0.8091**
61	0.4643	0.4506	0.3688	0.4046	0.6113	0.7559	0.7918	0.8975	0.5154	0.9002	**0.9152**
833	0.6878	0.6902	0.6651	0.6600	0.7464	**0.9901**	0.9714	0.9638	0.9565	0.9147	0.9728
228	0.1614	0.1425	0.1551	0.1447	0.7447	0.7388	0.8087	**0.8474**	0.7813	0.7745	0.8389

knowledge reasoning because they are able to utilize the supplementary knowledge contained in the cross-KG paths. However, in some cases, the path-based methods also perform worse on aligned KGs than on single KGs, indicating that they cannot effectively learn useful paths in the heterogeneous environment. We also notice that HackRL does not achieve the best results on some query relations, where PRA or DeepPath achieves better performances. However, DeepPath requires the pre-training process, which consumes a great deal of training time and is inefficient.

Table 5 shows the overall link prediction results for the three datasets. It can be seen that the proposed HackRL obtains state-of-the-art performances on the three datasets when compared with other models. The embedding-based methods perform much worse than the path-based ones, especially on the JA-EN and FR-EN datasets, which are denser KGs. This indicates that path-based multi-hop features are more useful for the relation prediction of dense KGs, since the increase of corresponding relation triples increases the difficulty of learning their embeddings and decreases the vectors' expressiveness. The performances of the path-based reasoning methods on aligned KGs are better than their performances on single KGs, indicating that the cross-KG paths are helpful for the prediction of relations. In particular, compared with AttnPath, the results also show that HackRL benefits from the hierarchical graph attention mechanism and the designed training mechanisms.

Table 6 shows the MSR to compare agents' abilities in different models to learn equivalent relation paths. Fig. 8 demonstrates the results of MSR for the relation 463 (*president*) of the ZH-EN dataset. The experimental results show that the DRL methods on aligned KGs are significantly worse than those on single KGs on finding successful paths, which also indicates that the heterogeneous environment and the increase of action space make the

Table 5
Link prediction results (MAP).

Models	ZH-EN	JA-EN	FR-EN
TransE	0.4665	0.2514	0.2333
MTransE	0.4492	0.2520	0.2081
TransR	0.4324	0.2459	0.2217
MTransR	0.4306	0.2423	0.2130
PRA	0.6580	0.7890	0.7212
MPRA	0.7095	0.8030	0.7927
DeepPath	0.8286	0.9006	0.7570
MDeepPath	0.8200	0.9114	0.7935
AttnPath	0.7353	0.8423	0.7456
MAttnPath	0.8493	0.8906	0.7883
HackRL	**0.9207**	**0.9279**	**0.8441**

Table 6
Mean success rate (MSR) of different models.

Models	ZH-EN	JA-EN	FR-EN
DeepPath	0.1060	0.1982	0.1005
MDeepPath	0.0899	0.1393	0.0530
AttnPath	0.1572	0.3280	0.2150
MAttnPath	0.1083	0.2425	0.1581
HackRL	0.1160	0.2579	0.1652

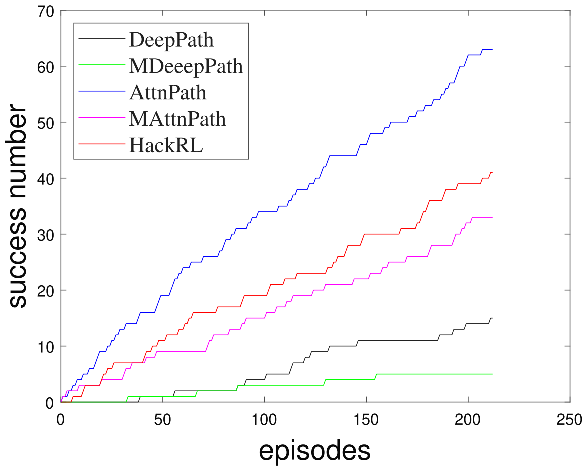


Fig. 8. Path success number on 463 of ZH-EN dataset.

Table 7
Fact prediction results (MAP).

Models	ZH-EN	JA-EN	FR-EN
TransE	0.1644	0.0712	0.0470
MTransE	0.1609	0.0738	0.0410
TransR	0.1544	0.0758	0.0438
MTransR	0.1495	0.0743	0.0445
PRA	0.2502	0.4697	0.4524
MPRA	0.2620	0.4963	0.4535
DeepPath	0.1717	0.4366	0.4413
MDeepPath	0.3280	0.4924	0.4534
AttnPath	0.2137	0.5424	0.4401
MAttnPath	0.4252	0.5661	0.4610
HackRL	**0.5029**	**0.5912**	**0.4994**

learning more difficult and decrease the probability of finding successful paths. Despite pre-training, DeepPath has much lower success rates on the three datasets. In contrast, AttnPath performs much better than DeepPath because of the force forward mechanism, which resamples a feasible relation to extend the relation path while the agent chooses an invalid action. It can also be seen that our proposed HackRL performs better than DeepPath and AttnPath on aligned KGs. With the proposed hierarchical graph attention and the training mechanism, our model is more suitable for exploring cross-KG relation paths in the heterogeneous environment and can learn more successful paths by filtering out infeasible actions and reducing the probability of selecting irrelevant actions.

4.2.2. Results of fact prediction

As an important downstream task of knowledge reasoning, the fact prediction task is intended to predict whether a given triple (h, r, t) is correct, which is widely used to evaluate models' ability to judge true and false triples. Similarly, we use ZH-EN, JA-EN, and FR-EN as the benchmark datasets for this task. Note that the classification assessment requires negative triples, which have already been provided in the generated datasets. For each relation, we also use all the inferred paths as features to train a neural network classifier with one hidden layer and use the prediction scores to predict the correctness of given triples. Unlike the link prediction task, fact prediction sorts all positive and negative samples for a particular relation in the test set instead of just ranking the target entities. Table 7 shows the overall results of all the methods. We can see that path-based methods outperform all the embedding methods, which indicates that embedding-based single-hop reasoning methods cannot effectively distinguish similar triples because the embeddings of neighboring entities are always similar. The path-based methods perform significantly better on aligned KGs than on single KGs, which indicates that the use of cross-KG path features improves these models' ability to distinguish similar triples. We can also see that HackRL significantly outperforms other baseline models. Meanwhile, the comparisons between HackRL and MAttnPath suggest that the proposed policy component and the training and optimization mechanism improves our model's performance on the task of collaborative fact prediction. The overall framework of HackRL enhances its probability of finding successful and promising cross-KG relation paths, which enriches the features for the learning of downstream models.

Table 8
Model setting ablations on the ZH-EN dataset.

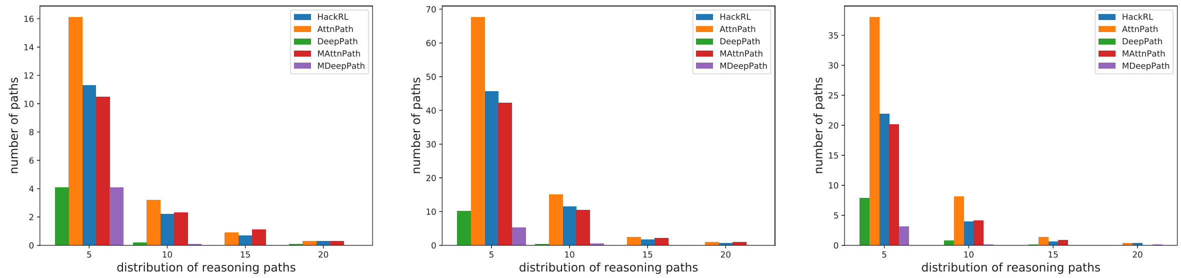
Tasks	Link prediction	Fact prediction
HackRL	0.9297	0.5029
-HGA	0.8903	0.4662
-AM	0.8400	0.4352
-RT	0.8913	0.4811

4.2.3. Ablation study

The ablation study is carried out to demonstrate the effectiveness of different components of the proposed model. Table 8 shows the results of link prediction and fact prediction on the ZH-EN dataset under different model settings. The first observation is that all three components are essential for improving the model's performance. When we remove the hierarchical graph attention module to perform the cross-KG path-finding for the downstream link and fact prediction tasks, the original results drop by 3.94% and 3.67%, respectively. It can be seen that the action space reduction based on the action mask mechanism plays an important role in inferring feasible relation paths. When we remove the action mask mechanism, in the link prediction task, the original result decreases by 8.97%, while in the fact prediction task, the MAP drops by 6.77%. The results also suggest that the retrain with our sampled cross-KG paths plays a part in improving the model's performance. Our model eliminates the problem of inferring relation paths in heterogeneous aligned KGs and is more suitable for large-scale KGs.

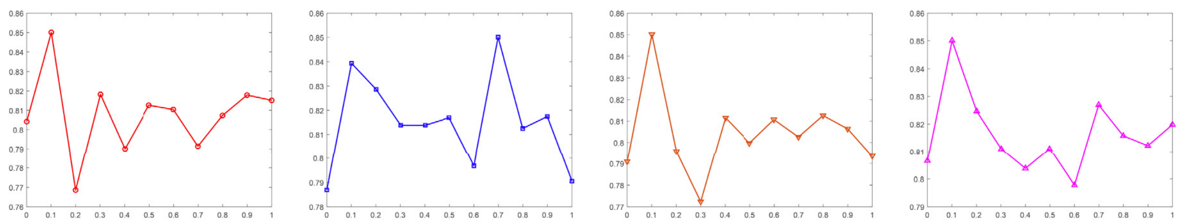
4.2.4. Parameter sensitivity

In this section, we analyze the parameter sensitivity of four weight parameters of different rewards. Fig. 10 shows the link prediction MAP of HackRL on the FR-EN dataset with the change of reward weights. Each subgraph shows how the MAP varies with the weight of a kind of reward, while the hyperparameters of other rewards are unchanged. It can be seen from the results that the parameter settings of the rewards greatly impact the link prediction results of HackRL. However, there is almost no linear law between the MAP and a single reward parameter. Therefore, it is difficult to find the optimal set of hyperparameters of rewards. In contrast, the determination of the efficiency reward weight is more important, since its change impacts the link prediction result in a greater deal. The reason is that the lengths of the inferred cross-KG relation paths have great impacts on the prediction of corresponding relations. However, the optimal lengths of cross-KG paths for different relations are not the same. As a result, the overall MAP on the whole dataset fluctuates with the efficiency weight. Similarly, the overall link prediction results of HackRL fluctuate with the settings of other reward weights as the optimal parameter settings that impact the inferred cross-KG relation paths are different with different query relations. In conclusion, the optimal hyperparameter settings of the reward weights should be carefully selected after multiple attempts, which can be resolved by grid search in a more efficient way. Besides, the optimal parameter settings of different datasets may be different.



(a) ZH-EN (b) JA-EN (c) FR-EN

Fig. 9. The distribution of path lengths of the DRL-based methods.



(a) λ_1 (b) λ_2 (c) λ_3 (d) λ_4

Fig. 10. Link prediction MAP of HackRL on the FR-EN dataset with different weights of rewards.

4.2.5. Qualitative analysis

In these experiments, we conduct qualitative research on the datasets, which is intended to show the searching and reasoning process of HackRL on the cross-KG paths in reality and evaluate the quality of the inferred paths. Since the facts described in the three datasets are largely the same, and the names of many relations in the other two datasets are missing, we only show an instance of the ZH-EN dataset here. Table 9 demonstrates the details of the reasoning paths for *president* from ZH-EN. The paths are inferred by the DRL-based methods; the top 8 paths are analyzed. It can be seen that DeepPath and MDeepPath cannot learn more than eight successful relation paths for this case. The results indicate that DeepPath can hardly learn various successful paths even though it has the pre-training process. With the LSTM and graph attention network-enabled memory component and the force forward mechanism, AttnPath can learn more successful paths. However, on aligned KGs, AttnPath cannot learn a lot of inter-KG paths, and its learned inter-KG paths are less indicative semantically. As can be seen from the results, HackRL can learn more inter-KG relation paths with the assistance of the hierarchical graph attention and the action mask mechanism. In addition, the inter-KG relation paths learned by HackRL are shorter and much more semantically related to the query relation. In particular, the equivalent relation is found by HackRL, which is quite indicative to the query relation's existence. Therefore, HackRL effectively fuses the knowledge from different KGs by concatenating relations from different KGs to form equivalent relation paths to indicate the existence of certain links between a pair of entities. For example, HackRL finds the equivalent cross-KG relation path identical$\rightarrow$president^e $\rightarrow$identical^{-1} of the relation *president*, which strongly indicates that someone is the president based on the existence of such a relation between the equivalent entities of the query entities in the aligned KG.

To analyze the quality of the paths found by different models, we illustrate the path distributions in Fig. 9. From the results, we can see that AttnPath finds the maximum number of successful paths. However, in the heterogeneous aligned KGs, HackRL finds more cross-KG relation paths than MAttnPath and MDeepPath. With the efficiency reward applied in the above DRL-based models, all of them are prone to find short and direct paths as it can be seen that the lengths of most successful paths are less than 5.

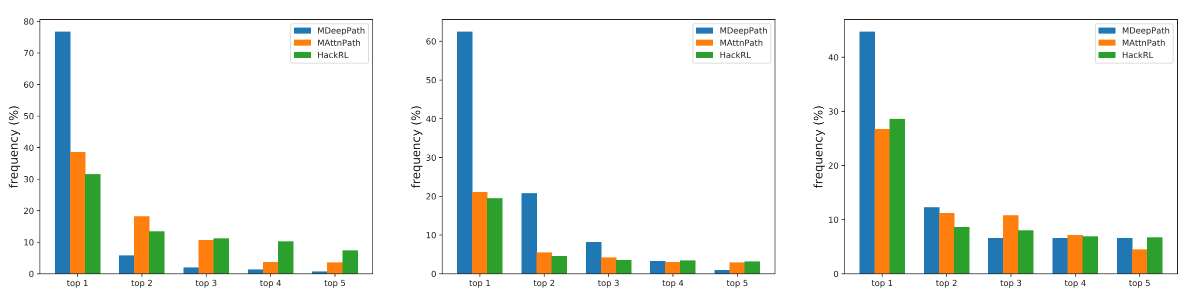
4.3. Discussion

The proposed HackRL model is a relation inference model based on reinforcement learning for the knowledge fusion and collaborative reasoning of multiple aligned KGs, where a novel policy network is designed to enable the agent to make better inferences in the heterogeneous environment, and several mechanisms are designed and adopted to optimize the training of the model. With these designs, HackRL achieves state-of-the-art performances on the link prediction and fact prediction tasks. In particular, the LSTM module enables the agent to retain the experience contained in its history trajectory and the HGA module improves the agent's ability to perceive the heterogeneous environment. The expansion of the action space improves the probability of selecting an invalid action, which greatly increases the difficulty of learning successful paths. To deal with this, we propose a simple yet powerful action mask mechanism to filter out infeasible actions, which effectively improves the probability of finding successful paths together with the force forward mechanism. The retrain with sampled paths mechanism prompts the agent to learn shorter and more direct cross-KG paths, which are more indicative to the corresponding relations.

However, the collaborative reasoning of multiple KGs is a difficult task and our proposed model has some remaining defects to be further investigated. We count the frequencies of the top five paths found by MDeepPath, MAttnPath, and HackRL for each reasoning task to evaluate their ability of focusing on the most indicative relation patterns. Fig. 11 shows the average of the frequencies from top one to top five on the three datasets. From the results, we find that the top one and top two paths found by MDeepPath are more focused than those of MAttnPath and HackRL. This indicates that, because of the pre-training process, MDeepPath is more likely to concentrate on a small number of relation path patterns, which in turn make it cannot discover more path patterns. Compared with MAttnPath, HackRL is less focused on the top two relation paths, indicating that it can learn more types of cross-KG relation paths. However, it may also introduce more less-indicative noisy path features, which may destroy its performance in some cases. The reason may be

Table 9
Top 8 frequent paths of *president* (ZH-EN) inferred by the DRL-based models;$^{-1}$ indicates inverse relation and ^{e} indicates equivalent relation in the aligned KG.

Method	Reasoning path
DeepPath	Successor^{-1}; Predecessor; Successor^{-1} →predecessor; Successor^{-1} →party→title^{-1}
MDeepPath	Successor^{-1}; AlmaMater→almaMater^{-1}; Prime minister; Deputy^{-1}; Predecessor; Successor^{-1} →predecessor^{-1} →successor^{-1}
AttnPath	Predecessor; Successor^{-1}; Premier^{-1}; Predecessor→prime minister^{-1}; Successor→after→prime minister^{-1} →successor^{-1}; Predecessor^{-1} →placeofbirth→country^{-1}→allegiance^{-1} →predecessor^{-1} Successor→predecessor→prime minister^{-1} →predecessor; Predecessor→successor^{-1} →party→title^{-1}
MAttnPath	Prime minister^{-1}; Successor^{-1}; Predecessor; Successor^{-1} →after→title→appointer→title^{-1}; Incumbent^{-1} →inaugural→appointer^{-1}→leader^{-1}; Premier; Successor→identical→after^{-e} →president^{e}→identical^{-1} →successor; Predecessor→premier→party→party^{-1}
HackRL	Prime minister^{-1}; Successor^{-1}; Identical→president^{e} →identical^{-1}; Party^{-1}; Vice president^{-1}; Identical→president^{e} →after^{e} →identical^{-1}→predecessor; Predecessor; Prime minister→predecessor^{-1}



(a) ZH-EN (b) JA-EN (c) FR-EN

Fig. 11. Frequencies of the top 5 paths for all reasoning tasks.

that the LSTM-based memory module cannot learn experience from various heterogeneous history trajectories very well due to the enrichment of relation path patterns. Moreover, KGs contain many other types of information, such as the entity description and attributes. Integrating different types of information into the DRL-based model may help it to learn more accurate and deep semantic relation patterns.

5. Conclusion and future work

In this paper, we propose the exploration and utilization of indicative cross-KG relation paths over multiple aligned KGs to integrate and fuse the complementary knowledge for collaborative reasoning and decision-making. We propose a novel DRL-based model named HackRL to explore the most informative paths. In order to eliminate the problems caused by feature space heterogeneity of different KGs, we embed different KGs into a unified vector space by minimizing the embedding distance between the equivalent entities identified by entity alignment. We also incorporate the LSTM and HGA mechanisms into the model to enable the agent to learn feasible paths from the heterogeneous environment. To eliminate the impact of the increase of action space, we propose an action mask mechanism to filter out unreasonable actions before selecting a relation to proceed. Additionally, we find that short and direct cross-KG relation paths are much more useful; therefore, we propose sampling such paths to retrain failed episodes to guide the agent to learn. Three famous cross-lingual knowledge graph datasets are utilized to validate the proposed model on two downstream tasks, link prediction and fact prediction. Experimental results indicate that the cross-KG paths with the anchor links as intermediate relations improve the performance of the path-based reasoning. Qualitative analysis also suggests that our proposed HGA module, and the action mask and retrain with sampled paths mechanisms are helpful for finding indicative cross-KG relation paths.

In terms of future work, we are interested in combining integrated information, such as structural information, description information, and attribute information, with the reinforcement learning framework to capture deeper semantic knowledge for knowledge fusion and collaborative reasoning. We would also like to refine the LSTM-based memory component and capture the semantic correlation between a relation and the found paths by using their embeddings to improve the prediction performance.

CRediT authorship contribution statement

Linyao Yang: Methodology, Software, Investigation, Writing – original draft. **Xiao Wang:** Framework design, Writing review, Funding acquisition. **Yuxin Dai:** Resources. **Kejun Xin:** Conceptualization, Writing review. **Xiaolong Zheng:** Writing review. **Weiping Ding:** Writing review. **Jun Zhang:** Writing review, Funding acquisition. **Fei-Yue Wang:** Supervision, Funding acquisition.

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.

Acknowledgments

This work was supported by the National Key R&D Program of China under Grant 2018AAA0101502 and the Science and technology project of SGCC (State Grid Corporation of China): fundamental theory of human-in-the-loop hybrid-augmented intelligence for power grid dispatch and control.

References

[1] Xiao Wang, Ke Zeng, Xue-Liang Zhao, Fei-Yue Wang, Using web data to enhance traffic situation awareness, in: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), 2014, pp. 195–199.
[2] Fei-Yue Wang, Weishan Zhang, Yonglin Tian, Rui Qin, Xiao Wang, Bin Hu, Federated data: Toward new generation of credible and trustable artificial intelligence, IEEE Trans. Comput. Soc. Syst. 8 (3) (2021) 538–545.
[3] Fabian M. Suchanek, Gjergji Kasneci, Gerhard Weikum, Yago: a core of semantic knowledge, in: Proceedings of the 16th International Conference on World Wide Web, 2007, pp. 697–706.
[4] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al., Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia, Semantic Web 6 (2) (2015) 167–195.
[5] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, Jamie Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, 2008, pp. 1247–1250.
[6] Yujia Huo, Derek F. Wong, Lionel M. Ni, Lidia S. Chao, Jing Zhang, Hetropy: Explainable learning diagnostics via heterogeneous maximum-entropy and multi-spatial knowledge representation, Knowl.-Based Syst. 207 (2020) 106389.
[7] Marc Franco-Salvador, Paolo Rosso, Roberto Navigli, A knowledge-based representation for cross-language document retrieval and categorization, in: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, 2014, pp. 414–423.
[8] Yongrui Chen, Huiying Li, Dam: Transformer-based relation detection for question answering over knowledge base, Knowl.-Based Syst. 201–202 (2020) 106077.
[9] Dongxiang Zhang, Rui Cao, Sai Wu, Information fusion in visual question answering: A survey, Inf. Fusion 52 (2019) 268–280.
[10] Yifan Zhu, Qika Lin, Hao Lu, Kaize Shi, Ping Qiu, Zhendong Niu, Recommending scientific paper via heterogeneous knowledge embedding based attentive recurrent neural networks, Knowl.-Based Syst. 215 (2021) 106744.
[11] Yifei Zhao, Hang Gao, Shuai Wang, Fei-Yue Wang, A novel approach for traffic signal control: A recommendation perspective, IEEE Intell. Transp. Syst. Mag. 9 (3) (2017) 127–135.
[12] Junchen Jin, Haifeng Guo, Jia Xu, Xiao Wang, Fei-Yue Wang, An end-to-end recommendation system for urban traffic controls and management under a parallel learning framework, IEEE Trans. Intell. Transp. Syst. 22 (3) (2021) 1616–1626.
[13] Shuangyin Li, Heng Wang, Rong Pan, Mingzhi Mao, Memorypath: A deep reinforcement learning framework for incorporating memory component into knowledge graph reasoning, Neurocomputing 419 (2021) 273–286.
[14] Jianjiang Lu, Yanhui Li, Bo Zhou, Dazhou Kang, Reasoning within extended fuzzy description logic, Knowl.-Based Syst. 22 (1) (2009) 28–37.
[15] Qi Wang, Yuede Ji, Yongsheng Hao, Jie Cao, Grl: Knowledge graph completion with GAN-based reinforcement learning, Knowl.-Based Syst. 209 (2020) 106421.
[16] Chen Li, Xutan Peng, Shanghang Zhang, Hao Peng, Philip S. Yu, Min He, Linfeng Du, Lihong Wang, Modeling relation paths for knowledge base completion via joint adversarial training, Knowl.-Based Syst. 201–202 (2020) 105865.
[17] Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, Li Deng, Embedding entities and relations for learning and inference in knowledge bases, in: Proceedings of the International Conference on Learning Representations (ICLR), 2015.
[18] Qianjin Zhang, Ronggui Wang, Juan Yang, Lixia Xue, Knowledge graph embedding by translating in time domain space for link prediction, Knowl.-Based Syst. 212 (2021) 106564.
[19] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Durán, Jason Weston, Oksana Yakhnenko, Translating embeddings for modeling multi-relational data, in: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2, 2013, pp. 2787–2795.
[20] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, Xuan Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2181–2187.
[21] Z. Li, X. Jin, S. Guan, Y. Wang, X. Cheng, Path reasoning over knowledge graph: A multi-agent and reinforcement learning based method, in: 2018 IEEE International Conference on Data Mining Workshops (ICDMW), 2018, pp. 929–936.
[22] Wenhan Xiong, Thien Hoang, William Yang Wang, DeepPath: A reinforcement learning method for knowledge graph reasoning, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2017, pp. 564–573.
[23] Rajarshi Das, Shehzaad Dhuliawala, Manzil Zaheer, Luke Vilnis, Ishan Durugkar, Akshay Krishnamurthy, Alex Smola, Andrew McCallum, Go for a walk and arrive at the answer: Reasoning over paths in knowledge bases using reinforcement learning, in: International Conference on Learning Representations, 2018, pp. 1–15.
[24] Heng Wang, Shuangyin Li, Rong Pan, Mingzhi Mao, Incorporating graph attention mechanism into knowledge graph reasoning based on deep reinforcement learning, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 2623–2631.
[25] Hoang Long Nguyen, Dang Thinh Vu, Jason J. Jung, Knowledge graph fusion for smart systems: A survey, Inf. Fusion 61 (2020) 56–70.
[26] Alexander Smirnov, Tatiana Levashova, Knowledge fusion patterns: A survey, Inf. Fusion 52 (2019) 31–40.
[27] Xuelu Chen, Muhao Chen, Changjun Fan, Ankith Uppunda, Yizhou Sun, Carlo Zaniolo, Multilingual knowledge graph completion via ensemble knowledge transfer, in: Findings of the Association for Computational Linguistics: EMNLP 2020, 2020, pp. 3227–3238.
[28] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[29] Chengjiang Li, Yixin Cao, Lei Hou, Jiaxin Shi, Juanzi Li, Tat-Seng Chua, Semi-supervised entity alignment via joint knowledge embedding model and cross-graph model, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 2723–2732.
[30] Kangzhi Zhao, Xiting Wang, Yuren Zhang, Li Zhao, Zheng Liu, Chunxiao Xing, Xing Xie, Leveraging demonstrations for reinforcement recommendation reasoning over knowledge graphs, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 239–248.
[31] Yutao Zhang, Jie Tang, Zhilin Yang, Jian Pei, Philip S. Yu, COSNET: Connecting heterogeneous social networks with local and global consistency, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015, pp. 1485–1494.
[32] Huynh Thanh Trung, Nguyen Thanh Toan, Tong Van Vinh, Hoang Thanh Dat, Duong Chi Thang, Nguyen Quoc Viet Hung, Abdul Sattar, A comparative study on network alignment techniques, Expert Syst. Appl. 140 (2020) 1–17.

[33] Zequn Sun, Qingheng Zhang, Wei Hu, Chengming Wang, Muhao Chen, Farahnaz Akrami, Chengkai Li, A benchmarking study of embedding-based entity alignment for knowledge graphs, Proc. VLDB Endow. 13 (12) (2020) 2326–2340.
[34] Muhao Chen, Yingtao Tian, Mohan Yang, Carlo Zaniolo, Multilingual knowledge graph embeddings for cross-lingual knowledge alignment, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 1511–1517.
[35] Bayu Distiawan Trisedya, Jianzhong Qi, Rui Zhang, Entity alignment between knowledge graphs using attribute embeddings, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 297–304.
[36] Zhichun Wang, Qingsong Lv, Xiaohan Lan, Yu Zhang, Cross-lingual knowledge graph alignment via graph convolutional networks, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 349–357.
[37] Zequn Sun, Chengming Wang, Wei Hu, Muhao Chen, Jian Dai, Wei Zhang, Yuzhong Qu, Knowledge graph alignment network with gated multi-hop neighborhood aggregation, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34(01), 2020, pp. 222–229.
[38] W. Zeng, X. Zhao, J. Tang, X. Lin, Collective entity alignment via adaptive features, in: 2020 IEEE 36th International Conference on Data Engineering (ICDE), 2020, pp. 1870–1873.
[39] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Rui Yan, Dongyan Zhao, Relation-aware entity alignment for heterogeneous knowledge graphs, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019, pp. 5278–5284.
[40] Weixin Zeng, Xiang Zhao, Wei Wang, Jiuyang Tang, Zhen Tan, Degree-aware alignment for entities in tail, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR), 2020, pp. 811–820.
[41] Zequn Sun, Wei Hu, Qingheng Zhang, Yuzhong Qu, Bootstrapping entity alignment with knowledge graph embedding, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 2018, pp. 4396–4402.
[42] Yuting Wu, Xiao Liu, Yansong Feng, Zheng Wang, Dongyan Zhao, Neighborhood matching network for entity alignment, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 6477–6487.
[43] Xiaobin Tang, Jing Zhang, Bo Chen, Yang Yang, Hong Chen, Cuiping Li, BERT-INT: A BERT-based interaction model for knowledge graph alignment, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 2020, pp. 3174–3180.
[44] Xiaojun Chen, Shengbin Jia, Yang Xiang, A review: Knowledge reasoning over knowledge graph, Expert Syst. Appl. 141 (2020) 1–21.
[45] Stefan Schoenmackers, Oren Etzioni, Daniel S. Weld, Jesse Davis, Learning first-order horn clauses from web text, in: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, 2010, pp. 1088–1098.
[46] H. Chen, W. Ku, H. Wang, L. Tang, M. Sun, Scaling up Markov logic probabilistic inference for social graphs, IEEE Trans. Knowl. Data Eng. 29 (2) (2017) 433–445.
[47] Fan Yang, Zhilin Yang, William W. Cohen, Differentiable learning of logical rules for knowledge base reasoning, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 2316–2325.
[48] Zhen Wang, Jianwen Zhang, Jianlin Feng, Zheng Chen, Knowledge graph embedding by translating on hyperplanes, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 1112–1119.
[49] Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, Jun Zhao, 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, 2015, pp. 687–696.
[50] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Eric Gaussier, Guillaume Bouchard, Complex embeddings for simple link prediction, in: Proceedings of the 33rd International Conference on Machine Learning, vol. 48, 2016, pp. 2071–2080.
[51] Seyed Mehran Kazemi, David Poole, SimplE embedding for link prediction in knowledge graphs, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 4289–4300.
[52] Tim Dettmers, Minervini Pasquale, Stenetorp Pontus, Sebastian Riedel, Convolutional 2D knowledge graph embeddings, in: Proceedings of the 32th AAAI Conference on Artificial Intelligence, 2018, pp. 1811–1818.
[53] Ni Lao, William W. Cohen, Relational retrieval using a combination of path-constrained random walks, Mach. Learn. 81 (1) (2010) 53–67.
[54] Ni Lao, Tom Mitchell, William W. Cohen, Random walk inference and learning in a large scale knowledge base, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Edinburgh, Scotland, UK., 2011, pp. 529–539.
[55] Qi Wang, Yongsheng Hao, Jie Cao, ADRL: An attention-based deep reinforcement learning framework for knowledge graph reasoning, Knowl.-Based Syst. 197 (2020) 105910.
[56] Xi Victoria Lin, Richard Socher, Caiming Xiong, Multi-hop knowledge graph reasoning with reward shaping, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3243–3253.
[57] Liwei Huang, Mingsheng Fu, Fan Li, Hong Qu, Yangjun Liu, Wenyu Chen, A deep reinforcement learning based long-term recommender system, Knowl.-Based Syst. 213 (2021) 106706.
[58] Teng Liu, Bin Tian, Yunfeng Ai, Fei-Yue Wang, Parallel reinforcement learning-based energy efficiency improvement for a cyber-physical system, IEEE/CAA J. Autom. Sin. 7 (2) (2020) 617–626.
[59] Teng Liu, Bin Tian, Yunfeng Ai, Li Li, Dongpu Cao, Fei-Yue Wang, Parallel reinforcement learning: a framework and case study, IEEE/CAA J. Autom. Sin. 5 (4) (2018) 827–835.
[60] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, Yoshua Bengio, Graph attention networks, in: International Conference on Learning Representations, 2018.
[61] Diederik P. Kingma, Jimmy Ba, Adam: A method for stochastic optimization, in: Yoshua Bengio, Yann LeCun (Eds.), in: 3rd International Conference on Learning Representations, 2015.
[62] Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, Juanzi Li, OpenKE: An open toolkit for knowledge embedding, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2018, pp. 139–144.