Integrating Entity Attributes for Error-Aware Knowledge Graph Embedding

Qinggang Zhang[1], Junnan Dong[1], Qiaoyu Tan[1], and Xiao Huang[1]
[1]Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR
qinggangg.zhang@connect.polyu.hk, hanson.dong@connect.polyu.hk, qiaoytan@polyu.edu.hk
Corresponding Author: Xiao Huang Email: xiaohuang@comp.polyu.edu.hk

Abstract—Knowledge graphs (KGs) can structurally organize large-scale information in the form of triples and significantly support many real-world applications. While most KG embedding algorithms hold the assumption that all triples are correct, considerable errors were inevitably injected during the construction process. It is urgent to develop effective error-aware KG embedding, since errors in KGs would lead to significant performance degradation in downstream applications. To this end, we propose a novel framework named Attributed Error-aware Knowledge Embedding (AEKE). It leverages the semantics contained in entity attributes to guide the KG embedding model learning against the impact of erroneous triples. We design two triple-level hypergraphs to model the topological structures of the KG and its attributes, respectively. The confidence score of each triple is jointly calculated based on self-contradictory within the triple, consistency between local and global structures, and homogeneity between structures and attributes. We leverage confidence scores to adaptively update the weighted aggregation in the multi-view graph learning framework and margin loss in KG embedding, such that potential errors will contribute little to KG learning. Experiments on three real-world KGs demonstrate that AEKE outperforms state-of-the-art KG embedding and error detection algorithms.

Index Terms—Knowledge graph, graph neural network, anomaly detection, node representation learning

1 Introduction

Knowledge graphs (KGs) can aggregate millions of relational facts in the form of triples [1], i.e, (head entity, relation, tail entity). Examples include general-purpose KGs, e.g., YAGO [2] and DBpedia [3], and domain-specific KGs, e.g., biomedical KGs [4] and agricultural KGs [5]. KGs are essential supporters for many knowledge-driven artificial intelligent systems, such as KG-enhanced recommender systems [6] and KG-based conversational agents [7]. Meanwhile, KG embedding has been intensively studied, which can improve the generalization and adaptability of KGs in downstream tasks. By representing entities as continuous vectors and each relationship as an operation in the same space, such as translation and projection, we can perform KG inference in continuous spaces with simple numerical computations.

While many KG embedding algorithms have been explored [8], [9], [10], the impact of erroneous triples has often been ignored. Manual construction is impractical due to the massive scale of KGs. Most real-world KGs are extracted from web corpora using heuristic algorithms [3], [11], [12]. A considerable number of noisy triples were inevitably introduced into KGs due to the noises in the original sources and the imperfect extraction algorithms [13]. For example, NELL [14] is a frequently used KG with 2.4 million triples, with an accuracy of 74%, corresponding to roughly 0.6 million erroneous triples [15]. Errors in KGs would lead to significant performance degradation in downstream applications. Thus, it is urgent to develop effective error-aware KG embedding.

It is nontrivial to enable error-aware learning in KGs, given that the patterns of KG errors are unknown and diverse [16], [17], [18]. Recently, a few studies explore guiding the KG embedding model learning against the impact of erroneous triples [19], [20], [21]. Particularly, Vault [22] is the first work that performs graph representation learning while considering erroneous triples during the learning phase. It estimates a probability score of reliability to determine the quality of a triple via several prior models fitted with existing KGs. The similar concept of judgments for each triple is also applied in CKRL [23] and NoiGAN [24]. The former model generates confidence scores for triples via internal structure information and utilizes them in representation learning to produce robust representations, while the latter improves it in the aspect of sample selection. The key idea of these methods is to guide the embedding model to focus on more convincing triples by exploiting the internal graph structures. However, merely KG topological structures are often not effective to support the detection of nontrivial errors [25]. For instance, given a triple $\{London, is\_larger\_than, Washington\}$ with sparse graph structure, it is hard to know which $London$ it refers to, since there are many cities called $London$. Thus, efforts have been devoted to employing extra information sources, such as related webpages [26], external knowledge bases [25] and annotation information from crowdsourcing websites [27]. But these supplemental sources are often prohibitive to acquire which impedes its success in practice.

KGs are often associated with fertile and valuable attribute data, describing the property of entities. Fig. 1 illustrates a toy KG with six entities. Each entity contains a set of attributes, e.g., the entity $MadameCurie$ has an attribute list of $\{birth\_date, gender, nationality, language\}$ which indicates its role as a human being, while $\{latitude, longitude, population, area\}$ shows $MariaSalomea$'s role as a location. Considering that most entities in KGs are not typed or are

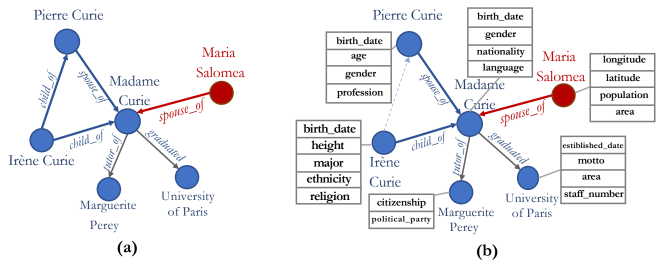


Fig. 1: Running examples of complex errors in real-world KGs. (a) presents a KG error with mismatched head/tail entities and relations. This error can be detected by reasoning over neighboring triples. But, real-world KGs are often incomplete and noisy. (b) demonstrates a further difficult case with some key links missing. In this case, the rich semantics in entity attribute types can facilitate the detection of errors.

very loosely typed, being able to learn about the semantic portrait of the entity from the attribute set is valuable.

Modeling and measuring the correlation between entity attributes and KG structure can guide the KG embedding model learning against the impact of erroneous triples. In real-world KGs, entity attributes often show high dependency with the graph structure, i.e., entities with relevant semantic meanings are usually linked by specific relations. For example, entities with attribute set $\{latitude, longitude, population, area\}$ are generally connected to $live\_in$ or $born\_in$ while entities with $\{pen\_name, writing\_style\}$ are more likely connected to $author\_of$. It means that triples whose entities have inconsistent attributes with its neighboring relations are more likely to contain noisy facts. Taking the triple $\{MadameCurie, spouse\_of, MariaSalomea\}$ in Fig. 1-(b) as example, analyzing the attribute list of $MadameCurie$ and $MariaSalomea$ and comparing the learned semantics with the corresponding relation, it can be easily deduced that this triple is erroneous since it is impossible to has the relation of $spouse\_of$ between a person and a location. Hence, we propose to enrich the entity semantics with its attribute list, and measure the degree of consistency between attributes and graph structure as anomaly signal to guide the learning process of KGs to enable effective error-aware KG embeddings.

However, it is still a challenging task to integrate entity attributes into error-aware KG embedding, with two major key challenges. (i) The heterogeneity of entity attribute makes it hard to utilize. In general, attributes are not uniformly distributed over all entities. As shown in Fig. 1-(b), different entities usually have distinct types or numbers of attributes. Furthermore, the same entity can represent different roles in different triples. When an entity describes its different roles, it tends to associate with different attribute sets to represent the certain semantics. For example, an entity of both scientist and amateur writer prefers an attribute list of $\{research\_area, citation, h\_index\}$ when describes its role as $scientist$ and $\{pen\_name, writing\_style, notable\_book\}$ when it means the status of $writer$. Therefore, a tailored and uniform encoder is desired to fuse such heterogeneous attribute information to depict the various levels of semantics for entities. (ii) It is hard to leverage the semantics learned from attributes to help the KG embedding model learn error-aware embeddings. Entity attributes can be used to implicitly portray the semantics of entities, but it is hard to integrate different KG components, i.e. entities, relations and attributes into a suitable vector space since they always exhibit rather distinct characteristics. Some existing KG embedding models attempt to integrate attributes into the learning framework by directly fusing the learned attribute semantics into the entity embedding [28], [29], [30], [31]. However, in this fusing way, it is hard to measure the complex correlation and dependency between entity attributes and knowledge graph structure, which is crucial to guide the embedding model to filter out noisy information from hidden erroneous triples.

Through this paper, we aim to answer the following research questions. ❶ How to jointly embed different kinds of KG components, i.e, entities, relations and attributes, into suitable vector spaces? ❷ How to construct an effective detector that calculates the confidence score for each triple based on the learned features? ❸ How to make the best of confidence information learned from the detection module to get the error-aware KG embeddings? To solve these problems, in this paper, we propose a novel framework named Attributed Error-aware Knowledge Embedding, i.e, AEKE, that leverages the semantics contained in entity attributes to guide the KG embedding model learning against the impact of erroneous triples. The key idea is to guide the embedding model to focus on more convincing triples by exploiting the correlation between knowledge graph structures and entity attributes. Concretely, we designed two triple-level hypergraphs, i.e, *relational hypergraph*, and *attribute hypergraph*, to model the topological structure of the original KG and its attributes, respectively. A contrastive learning framework is then used to learn the representation of each instance from these two different views. Analyzing and mining the unique structure of KGs and the correlation between graph structure and attributes, the confidence score of each triple can be calculated by considering three anomaly signals, i.e., self-contradictory within the triple, global acknowledgment across triples, and conformity between attribute and graph structure. We leverage confidence scores to adaptively update the weighted aggregation in contrastive learning and margin loss in KG embedding, such that potential errors would contribute little to KG learning. The main contributions of this work are concluded as follows:

- In this research, we propose a novel knowledge graph embedding framework, i.e. AEKE, which leverages the semantics contained in entity attributes to guide the KG embedding model learning against the impact of erroneous triples.
- We design a tailored multi-view contrastive learning framework to use attribute information as a congruent view to guide the learning process of KGs.
- Based on the KG structure characteristics and the previously learned multi-view features, we propose to measure the confidence score of each triple by considering anomaly signals of three levels.
- We leverage confidence scores to adaptively update weighted aggregation in the multi-view graph learning framework and margin loss in KG embedding, such that potential errors will contribute little to KG modeling.

A preliminary version of this paper was published in the proceedings of the 31st ACM International Conference on

Information & Knowledge Management (CIKM 2022) [32]. In the conference version, we introduced a contrastive graph learning framework for Knowledge Graph Error Detection, named CAGED. The key idea of CAGED is to detect the potential errors from noisy KG by exploiting the internal graph structures. Despite its superiority over existing error detection models, we found that there are still two significant limitations that need to be addressed before its successful application in real-world scenarios. 1) CAGED performs error detection merely relying on KG topological structures. However, real-world KGs are often incomplete and sparse. Simply relying on KG topological structures is often not effective to support the detection of nontrivial errors with sparse graph structure. 2) Given the detected errors, it is still hard to get effective KG embeddings for downstream tasks. A straightforward solution is to retrain the KG by filtering out all the possible noisy triples. However, we found that it is infeasible in practice since in real-world KGs, it is difficult to directly conclude whether a triple is true or not without being tested in practice or strictly and mathematically proven. As shown in previous study [32], even the SOTA error detection method could only achieve 60% accuracy. Consequently, directly filtering out all the possible erroneous triples will lead to severe information loss, especially for these long-tail entities which only have few links in the original KG.

In this paper, instead of performing KG modeling via a two-step way, i.e., (i) conduct error detection first, and then (ii) retrain the KG by filtering out all the possible noisy triples, we propose an end-to-end graph learning framework, named AEKE. While CAGED serves as a KG error detection model seeking anomaly labels for each triple, our newly proposed AEKE focuses on error-aware KG embedding learning that aims to encode entities and relations into a low-dimensional vector space while considering erroneous triples during the learning phase. In particular, we extend the preliminary work as follows: 1) We formally define the task of error-aware KG embedding and highlight its difference with error detection. 2) Rather than relying solely on internal topological structures, we leverage additional information, i.e., entity attributes, to guide AEKE learning against the impact of nontrivial errors. 3) We introduce a novel training objective function, where the confidence score is learnable and dynamically updated during the representation learning phase so that the noisy information from erroneous triples can be minimized while the valuable information from correct triple will be mostly preserved during the training process. 4) More experiments are conducted to verify the effectiveness of our proposed model on both detection performance and downstream task of KG completion.

2 Problem Statement

Given a knowledge graph $\mathcal{G}$, we define $\mathcal{G} = (\mathcal{E}, \mathcal{R})$, indicating an aggregation of an entity set $\mathcal{E}$ and a relation set $\mathcal{R}$. We use $(h, r, t) \in \mathcal{T}$ to represent a triple fact inside one KG, where there is a relation $r \in \mathcal{R}$ linking a head entity h and a tail entity t. Learning from the KG representation, we use uppercase bold letters to denote matrices (e.g., $\mathbf{W}$), lowercase bold letters to represent vectors (e.g., z).

When taking entity attributes into consideration, we use $\mathcal{A}$ to denote the attribute set and define the attributes of entity h in the form of $\{a_{h,1}, a_{h,2}, \ldots, a_{h,|A_h|}\}$ where $a_{h,i} \in \mathcal{A}$ is the i^{th} attribute type and $|A_h|$ is the total number of attribute types with regard to the entity h. Here, we only utilize attribute types to quantitatively depict the entity semantics without considering accurate values due to its low quality. For example, given the entity $MariaSalomea$ in Fig. 1, we reconstruct its semantics with the attribute set $\{longitude, latitude, population\}$ instead of $\{longitude: 53.74W, latitude: 20.77N, population: 1.5M\}$. We adopt $\mathbf{E}$ and $\mathbf{R}$ to represent the feature matrices of entity and relation, respectively. Similarly, we adopt $\mathbf{A}$ to denote the learned attribute matrix of entities in $\mathcal{G}$, where i-th row of $\mathbf{A} \in \mathbb{R}^{D \times F_A}$ represents the feature indicators to the attributes $a_{h,i} \in \mathcal{A}$ of entity $h \in \mathcal{E}$. D is the max value across the attribute number of each entity. In our model, these embeddings will be used in two ways: (i) for representation learning, and (ii) for confidence learning.

TABLE 1: Notations summary.

Notation	Description
$\mathcal{G}$	The original knowledge graph
$\mathcal{A}$	Entity attribute set
$\mathcal{G}_r$	*relational hypergraph* constructed from $\mathcal{G}$
$\mathcal{G}_a$	*attribute hypergraph* constructed from $\mathcal{A}$
$\{a_{hi} \mid i = 1, \ldots, n\}$	A list of attribute types for entity h
(h, r, t)	A triple of (head, relation type, tail)
p_i	Basic representation of i^{th} triple
x_i	Representation of i^{th} triple learned from $\mathcal{G}_r$
z_i	Representation of i^{th} triple learned from $\mathcal{G}_a$
$C(h, r, t)$	Confidence score of triple (h, r, t)

Definition 1. ***Errors in Knowledge Graph.*** *Given a triple in a KG, denoted as (h, r, t), if there is a mismatch among its head entity h, relation r, and tail entity t, then this triple (h, r, t) is an error. There are two types of mismatching. First, relevant entities might be connected by wrong relations. E.g., $\{ElonMusk, founder\_of, Tesla\}$, where $ElonMusk$ is connected with $Tesla$ indeed, but he is the CEO, not the founder. Second, irrelevant entities might be connected. E.g., $\{BruceLee, place\_of\_birth, England\}$, in which $BruceLee$ was born in Los Angeles.*

In a KG, it is difficult to directly conclude whether a triple is true or not without being tested in practice or mathematically proven. Thus, following previous studies [15], [19], [23], [32], [33], we introduce the concept of triple confidence and try to validate the whole KG from the perspective of triples, where the confidence value indicates the degree of a triple being true.

Definition 2. ***Triple Confidence.*** *In this paper, we introduce the concept of confidence to measure the correctness of each triple. Its value is set to be in the range $[0, 1]$. The closer the value gets to 0, the more likely the triple is incorrect.*

Definition 3. ***Error-aware KG Embedding.*** *Given a KG $\mathcal{G}$, with noisy triples, we aim to learn low-dimensional representations for all entities $\mathbf{E}$ and relations $\mathbf{R}$, i.e., a mapping $\{\mathcal{G}\} \rightarrow \{\mathbf{E}, \mathbf{R}\}$, such that all information in correct triples can be preserved, while the impact of false triples on the embedding is minimized. The performance of error-aware embedding is evaluated by applying $\mathbf{E}$ and $\mathbf{R}$ to downstream tasks.*

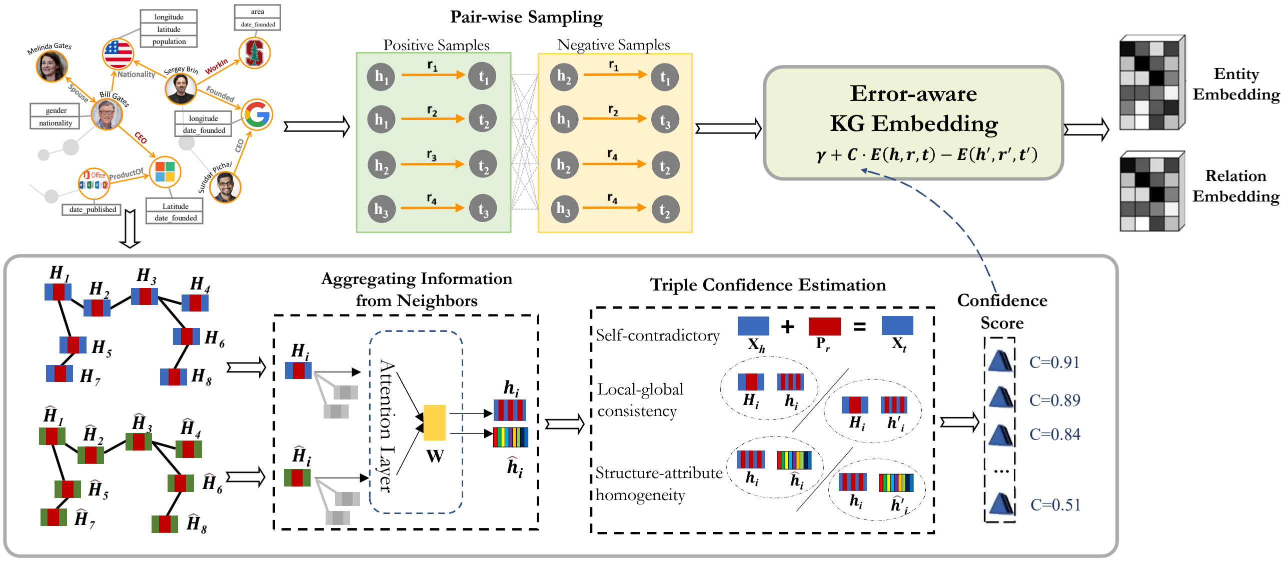


Fig. 2: We perform a relation-induced construction process to build the *relational hypergraph*, and construct the *attribute hypergraph* based on entity attributes. These two hypergraphs can be regarded as congruent views of the target KG. A contrastive learning framework is used to learn the representation of each instance from these two different views. Meanwhile, a triple confidence estimation module is designed to calculate the confidence score of each triple by considering: self-contradictory within the triple; local-global consistency in graph structure; and structure-attribute homogeneity. Under the joint adaptive training scheme, we leverage confidence scores to adaptively update the weighted aggregation in contrastive learning and margin loss in KG embedding, such that potential errors would contribute little to KG learning.

3 Methodology

We propose AEKE to learn error-aware KG embedding by integrating entity attributes. As shown in Fig. 2, AEKE consists of three key components. ❶ KG representation learning model. It incorporates a confidence score $C(h,r,t)$ into the traditional KG embedding model to isolate the impact of noise over embedding vectors. ❷ Triple confidence learning module. It aims to learn a confidence score $C(h,r,t)$ to measure the correctness and significance of each triple with the favor of both internal structural information and external heterogeneous attribute information. ❸ Joint adaptive training scheme. A tailored adaptive mechanism is applied to optimize the KG representation learning and confidence learning model under an end-to-end framework.

3.1 Knowledge Graph Representation Learning

Knowledge graphs can effectively store and handle structured information of real-world entities and facts, and have been widely utilized in various knowledge-driven applications. In fact, real-world KGs always suffer from serious quality problem since amounts of errors were introduced into KGs in construction phase due to the original noises in sources and the imperfect extraction methods. However, most existing knowledge graph embedding methods assume that all triple facts hold true to the knowledge graph. Therefore, the noisy information is overfitted into KG embeddings, which leads to significant performance degradation in downstream tasks.

To eliminate the impact of noisy triples over embedding vectors, following a previous study [23], [33], we introduce the concept of confidence to describe whether a triple fact is noisy or not. Its value is set to be in the range $[0,1]$. The closer the value gets to 0, the more likely the triple is incorrect. With the introduction of confidence, we can define the new error-aware objective function to eliminate the noisy data from the learning process of KG embedding model.

$$L_{emb} = \sum_{(h,r,t)\in\mathcal{G}} \sum_{(h',r',t')\in\mathcal{G}'} \max(0, \gamma + C(h,r,t)\cdot E(h,r,t) - E(h',r',t')), \tag{1}$$

where $E(h,r,t) = \|e_h + e_r - e_t\|_2$ is the traditional energy score for translational embedding models following translation assumption. $\gamma > 0$ is the hyperparameter of margin, and $\mathcal{G}$ represents the sampled positive triple set. Here, the triple confidence $C(h,r,t)$ forces our model to pay more attention to those more convincing facts. For pair-wise training, since there are no explicit negative triples in KGs, we sample negative triples complying with the following rules:

$$\begin{aligned}\mathcal{G}' = &\left\{(h',r,t) \mid h' \in \mathcal{E}\right\} \cup \left\{(h,r,t') \mid t' \in \mathcal{E}\right\} \\ &\cup \left\{(h,r',t) \mid r' \in \mathcal{R}\right\}, \quad (h,r,t) \in \mathcal{G}.\end{aligned} \tag{2}$$

It means that one entity or relation in a positive triple is randomly replaced by another entity or relation in the overall set. Note that different from TransE, we also add relation replacements for better performances. The corruption of entity (relation) might lead to false negative samples. To this end, we discard all triples already in $\mathcal{G}$ from $\mathcal{G}'$ to make sure our generated negative triples are truly negative.

3.2 Triple Confidence Learning with Entity Attributes

To enhance the robustness of our error-aware KG embedding model, we learn a confidence score $C(h,r,t)$ for each triple. As illustrated in Fig. 2, our key idea is to measure the correctness and significance of each triple by exploiting the

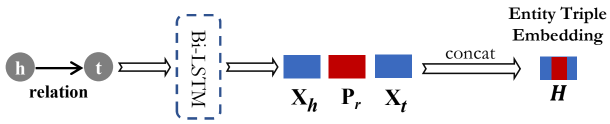


Fig. 3: Triple feature extraction for local relational structure.

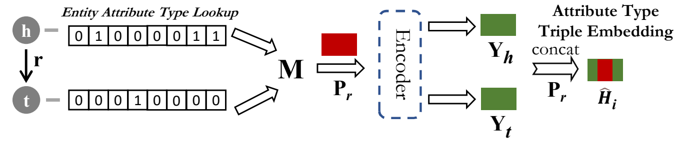


Fig. 4: Attribute triple Embedding initialization by capturing attribute information.

correlation between knowledge graph structures and entity attributes. In the real world, KGs are often accompanied by entity attributes, which have three paradigms as follows.

- In triple level, relations can be interpreted as translations operating on the low-dimensional embeddings of the entities. So, the more a triple fits the translation assumption, i.e. $\mathbf{h}+\mathbf{r} \simeq \mathbf{t}$, the more convincing this triple should be considered.
- In graph level, the connected triples that share the same entity are always semantically relevant. Intuitively, a KG can be regarded as a social group, where each triple is an individual. The degree of acknowledgements from neighboring triples to the target triple reflects whether the target triple can be properly integrated into the society.
- Entity attributes often show high dependency with the graph structure, i.e., entities whose attributes show relevant semantic meanings are usually linked by specific relations. It means triples whose entities have inconsistent attributes with its neighbors are more likely to contain noisy facts.

Based on these observations, we propose a novel confidence learning model to answer the research problems mentioned at the beginning of this paper. To model the different components in KGs, i.e., entity, relation and attribute, into suitable vector spaces, we proposed a multi-view knowledge graph learning framework in which a dual-view structure information encoder and a relation-specific attribute encoder are designed to model the topological KG structure and entity attributes, respectively. Specifically, we first design two triple-level hypergraphs, i.e., *relational hypergraph* and *attribute hypergraph* to model the topological structure of the original KG and its attribute, respectively. And then we employ a unified contrastive learning framework to model their dependency. To construct an effective detector that calculates the confidence score for each triple based on the learned features, we propose to measure the confidence score of each triple by considering three anomaly signals, i.e., self-contradictory within relational triple structure, global consistency across triples, and attribute-structure dependency.

3.2.1 Multi-view construction

Knowledge graph embedding projects entities and relations into low dimensional vector space, which has been successfully applied in KG-based tasks. However, the existing embedding approaches only model entities and their relations within triples, ignoring the global correlation among triples in KGs. To model the complex topological graph structure, in this paper, we propose a novel view towards knowledge graph modeling that augments the target KG into hyper-views (triple-level), by regarding each relational triple as nodes. Concretely, we perform a relation-induced construction process to build the *relational hypergraph*. In essence, we follow one strategical criterion: we regard each relational triple as nodes and link them in the *relational hypergraph* if they share the same entity in the original KG. Such triple-level transformation could filter out low-level information without changing the structure of possible errors in original KGs since a KG error usually occurs inside a triple as a mismatch of the head entity, tail entity, and their corresponding relation.

Definition 4. ***Relational Hypergraph.*** *Given a knowledge graph $\mathcal{G} = \{(h,r,t)|h,t \in \mathcal{E}, r \in \mathcal{R}\}$, the corresponding relational hypergraph is the triple-level graph $\mathcal{G}_r = (\widetilde{\mathcal{V}}, \widetilde{\mathcal{A}_r}, \widetilde{\mathcal{X}_r})$, where $\widetilde{\mathcal{V}}$ and $\widetilde{\mathcal{A}_r}$ are the set of nodes and adjacency metric respectively. $\widetilde{\mathcal{X}_r} = \{\Gamma(h,r,t) \mid (h,r,t) \in \mathcal{G}\}$ represents the feature matrix of $\widetilde{\mathcal{V}}$, where $\Gamma(\cdot)$ is a concatenation function. $\widetilde{\mathcal{A}_r}(v,u|u,v \in \widetilde{\mathcal{V}}) = 1$, if u and v share the same entity in the original KG, i.e., $\mathcal{G}$.*

When taking entity attributes into consideration, we build another triple-level graph, i.e. *attribute hypergraph*, to reconstruct the knowledge graph from the perspective of attributes. Specifically, we treat each triple in the original KGs as an individual instance and reconstruct its semantics with the feature learned from the attribute sets of its head and tail entity.

Definition 5. ***Attribute Hypergraph.*** *Given the same knowledge graph $\mathcal{G} = \{(h,r,t)|h,t \in \mathcal{E}, r \in \mathcal{R}\}$, with entity attribute set $\mathcal{A} = \{\{a_{h,1}, a_{h,2}, \ldots, a_{h,|A_h|}\}|h \in \mathcal{E}\}$ where $a_{h,i}$ is the i^{th} attribute type for entity h, the attribute hypergraph can be denoted as $\mathcal{G}_a = (\mathcal{V}_a, \mathcal{A}_a, \mathcal{X}_a)$, where $\mathcal{V}_a$ and $\mathcal{A}_a$ are the set of nodes and adjacency metric respectively, which have the same structure with $\boldsymbol{G}_r$. And X_a represent the feature matrices of each node $v_i \in \mathcal{V}_a$ learned from corresponding entity attributes.*

These two hypergraphs can be regarded as congruent views of the target KG. The *relational hypergraph* models the correlation between relational triples, while the *attribute hypergraph* represents the distribution of entity attributes. Considering that entity attributes always show high dependency with graph structure, i.e., entity attributes are always linked with certain relevant relations, for normal triples in KGs, we can easily find enough relevant attributes in *attribute hypergraph* to reconstruct its semantics learned from *relational hypergraph*. Thus, it helps us assess the trustworthiness of each triple in the original KG by measuring the consistency between its representations learned from these two views.

3.2.2 Learning from view I: relational hypergraph

Existing KG embedding approaches only model entities and their relations within triples, ignoring the global correlation among triples in KGs. In this section, we propose a new graph encoder in a dual-view which learns both the local structure within relational triples (local view), and the contextual information buried among their neighboring community simultaneously (global view).

Local view of relational structure modeling. Constructing *relational hypergraph* from the original knowledge graph, in some way, may lose some local structure information, which means the translational or sequential structure inside a triple (head $\rightarrow$ relation $\rightarrow$ tail). Since every instance in the *relational hypergraph* is transformed from a corresponding triple (h, r, t) in the original knowledge graph, we first randomly initialize the embedding of entities and relations in the original knowledge graph, and then adopt a local information modeling layer, i.e, a set of BiLSTM units, to to integrate the embedding of head entity, relation and tail entity (e_h, e_r, e_t) into triple-level representation p. Taking the i^{th} triple (h, r, t) as an example, our local information modeling layer is formulated as follows.

$$p_i = \mathcal{G}_{local}(h, r, t) = f_{concat}(f_{BiLSTM}\left(e_h, e_r, e_t\right)). \tag{3}$$

The output triple embedding p_i is supposed to well capture the local relational structure of the input triple. Thus, we use them as the initial embedding of each node in *relational hypergraph*.

Global view of neighbor information aggregation. Except the local structure inside triples, abundant contextual information among a community of triples are useful in detecting anonymous triples. To model such global contextual information, an intuitive way is to use graph neural network (GNN) or its variants. However, existing graph neural networks are not optimal for KG embedding task since they ignore the existence of anomalous instance in the graph. In other words, potential noisy triples may draw relatively equal attention as normal ones, which leads to a more challenging triple representation learning. To reduce impacts from potential anomalies, we adopt an attention-based architecture, i.e. $\mathcal{G}_{global}$, to selectively aggregate messages from the neighboring triples.

Given an anchor triple $p_i \in \mathbb{R}^d$ in *relational hypergraph*, we update its embedding by attending over its neighborhoods' features, e.g. $\{p_1, p_2, \ldots, p_m\}, p_m \in \mathbb{R}^d$ through a two-layer graph attention network. A single graph attentional layers is given by:

$$att_{ij} = f_{att}\left(\mathbf{W}p_i, \mathbf{W}p_j\right). \tag{4}$$

Here, att_{ij} is a raw attention coefficient which indicates the importance of triple j to triple i. $\mathbf{W} \in \mathbb{R}^{n \times d}$ is a learnable linear transformation matrix to project the initialized triple representations into the same high-level vector space. f_{att} is the attentional function. To make attention coefficients easily comparable across different triples, We normalize the attention values by applying a softmax function:

$$\alpha_{ij} = \frac{\exp\left(att_{ij}\right)}{\sum_{k=1}^{m}\exp\left(att_{ik}\right)}. \tag{5}$$

Then, the final triple embedding in *relational hypergraph* can be calculated with a nonlinearity sigmoid function as follows.

$$x_i = \sigma\left(\sum_{j=1}^{m}\alpha_{ij}\mathbf{W}p_j\right). \tag{6}$$

3.2.3 Learning from view II: attribute hypergraph

Most of the current knowledge graph embedding frameworks focus on learning the inner structure of relational triples. However, entity attributes contain amounts of accurate and targeted semantic information that can describe entities quantitatively, which is equally valuable to enable effective KG embedding. To facilitate information transfer between entity attributes and the target KG for error-aware embedding, we propose a tailored a relation-specific encoder, i.e. g_{attr} learn the attribute-based triple representation from *attribute hypergraph*.

Attribute hypergraph embedding.

Attributes are not uniformly distributed over all entities. In general, different entities always have different types or numbers of attributes, even the same entity may represent different roles in different triples. When an entity describes its different roles in different triples, it tends to associate with different attribute sets to represent the certain semantics. Thus, it is necessary to choose which attribute corresponds to the semantics the current entity represents when we want to reconstruct triple-level semantics based on the attributes of its head/tail entities. To capture which level of information the current entity mainly focuses on, we therefore adopt the relation-specific mechanism, which uses the relation type as an indicator to lead the selection of entity attributes in different triples.

Given a triple (h, r, t), in order to extract the rich semantics information from more valuable attributes of entity h and t, we first splice the embeddings of attribute type into an embedding matrix, denoted by $\mathbf{M}_h = \left\{a_{h,1}, a_{h,2}, \ldots, a_{h,|A_h|}\right\}$ and $\mathbf{M}_t = \left\{a_{t,1}, a_{t,2}, \ldots, a_{t,|A_t|}\right\}$, then feed them to an attention module with relation embedding e_r to get the integrated representation $\hat{e}_h$ and $\hat{e}_t$ for target entity h and t. Take entity h as an example:

$$att_{h,i} = f_{att}\left(f_{emb}\left(e_r\right), f_{emb}\left(a_{h,i}\right)\right), \tag{7}$$

$$\alpha_{h,i} = \frac{\exp\left(att_{h,i}\right)}{\sum_{j=1}^{|A_h|}\exp\left(att_{h,j}\right)}, \tag{8}$$

where f_{emb} and f_{att} are all single-layer feed-forward neural networks. The $\alpha_{h,i}$ is the normalized attention weight of attribute $a_{h,i}$. Now, we can compute the aggregated attribute-based representation $\hat{\boldsymbol{e}}_h$ of A_h, which is the weighted sum of all the transformed representation of attributes in it:

$$\hat{e}_t = \sum_{i=1}^{|A_t|}\alpha_{t,i} * f_{emb}\left(a_{t,i}\right). \tag{9}$$

Then given a triple (h, r, t) in *attribute hypergraph*, we can get the attribute-based entity representations, i.e., $\hat{e}_h$ and $\hat{e}_t$, by using this attribute encoder. The final attribute-based triple embedding q_i can be calculated as:

$$q_i = f_{concat}(\hat{e}_h; e_r; \hat{e}_t). \tag{10}$$

Aggregating neighbor attribute information. Homogeneously, given an anchor triple $q_i \in \mathbb{R}^d$ in *attribute hypergraph*, we follow the global view of *relational hypergraph* to update the embedding by attending over neighbors' features of q_i, e.g., $\{q_1, q_2, \ldots, q_m\}, q_m \in \mathbb{R}^d$ through the same two-layer graph attention network:

$$z_i = \sigma\left(\sum_{j=1}^{m} \alpha_{ij} \mathbf{W} q_j\right). \tag{11}$$

3.2.4 Model learning

To learn discriminative features from both *relational hypergraph* and *attribute hypergraph*, we adopt a tailored contrastive loss to optimize the proposed multi-view knowledge graph neural network.

Contrasting between structure and attribute views. The *relational hypergraph* models the global correlation between relational triples, while the *attribute hypergraph* represent the distribution of entity attributes. These two hypergraphs can be regarded as congruent views of the original KG since entity attributes always show high dependency with graph structure. In this paper, we adopt the normalized temperature-scaled cross entropy loss as our contrastive objective and train the encoders by maximizing the mutual information between triple embeddings, i.e. x_i and z_i, learned from these two views.

$$\mathcal{L}_{\text{sa}}(x_i, z_i) = -\log \frac{\exp\left(\text{sim}\left(x_i, z_i\right)/\tau\right)}{\sum_{j \in \{1,2,\ldots,N\}\setminus\{i\}} \exp\left(\text{sim}\left(x_i, z_j\right)/\tau\right)}, \tag{12}$$

where τ denotes the temperature parameter. $sim(x_i, z_i)$ denotes the cosine similarity of triple embedding x_i from *relational hypergraph* and z_i learned from *attribute hypergraph*.

Contrasting between local and global views. As mentioned at the beginning of § 3.2, the connected triples that share the same entity are always semantically relevant, and thus the degree of acknowledgements from neighboring triples to the target triple reflects whether the target triple can properly integrate into the community. To model such paradigm between individual triple and its community, we propose another contrastive training objective by maximizing the mutual information between p_i and x_i. Here, p_i is the embedding of i^{th} triple learned from local relational structure (Eq. (3)), while x_i is the embedding of the same triple that learned by aggregating information from its neighboring community (Eq. (6)).

$$\mathcal{L}_{\text{lg}}(p_i, x_i) = -\log \frac{\exp\left(\text{sim}\left(p_i, x_i\right)/\tau\right)}{\sum_{j \in \{1,2,\ldots,N\}\setminus\{i\}} \exp\left(\text{sim}\left(p_i, x_j\right)/\tau\right)}. \tag{13}$$

Finally, by combining the above two losses, we have our final contrastive objective function, i.e $\mathcal{L}_{con}$ defined below:

$$\mathcal{L}_{con} = \frac{1}{2N} \sum_{i=1}^{N} \left(\mathcal{L}_{sa}\left(x_i, z_i\right) + \mathcal{L}_{lg}\left(p_i, x_i\right)\right). \tag{14}$$

3.2.5 Triple confidence estimation

Based on the KG paradigms and the previously learned multi-view features, we propose to measure the triple confidence by considering three anomaly signals: self-contradictory within local relational structure, global consistency across triples, and attribute-structure dependency.

Self-contradictory measurement. In triple level, relations can be interpreted as translations operating on the low-dimensional embeddings of the entities. So, the more a triple fits the translation assumption, i.e. $\mathbf{h} + \mathbf{r} \simeq \mathbf{t}$, the more convincing this triple should be considered. Existing KG embedding algorithms have developed various energy functions to model the translational structure for better learning embeddings. In this paper, we take a simple squared euclidean distance to measure the unconformity of each triple with translation assumption, and define the local triple confidence $LT(h, r, t)$ as follows.

$$LT(h, r, t) = \frac{1}{1 + e^{-\|e_h + e_r - e_t\|_2}}. \tag{15}$$

Global acknowledgement estimation. The connected triples that share the same entity are always semantically relevant, and thus the degree of acknowledgements from neighboring triples to the target triple reflects whether the target triple can properly integrate into the KG. Inspired by social identity theory [34], [35], the KG can be regarded as a social group, where each triple is an individual. The degree of acknowledgements from other individuals to the targeted individual (target triple) reflects whether the targeted individual can properly integrate into the society, i.e., the KG. We believe that only a true triple can achieve popular recognition from its neighboring triples. In other words, if a triple is well accepted, we tend to believe that it is trustworthy. Hence, we define another confidence function to measure the global acknowledgement of target triple (h, r, t).

$$GT(h, r, t) = sim(p_i, x_i), \tag{16}$$

where $p_i = \mathcal{G}_{local}(h, r, t)$ is the local embedding of triple (h, r, t), and $x_i = \mathcal{G}_{global}(h, r, t)$ represent the feature learned from its global context.

Structure-attribute dependency estimation. The *relational hypergraph* models the correlation between relations, while the *attribute hypergraph* represent the distribution of entity attributes. Considering that entity attributes always show high dependency with graph structure, i.e. entity attributes are always linked with certain relevant relations, for normal triples in KGs, we can easily find enough relevant attributes in *attribute hypergraph* to reconstruct its semantics learned from *relational hypergraph*. So, the consistency between its representations learned from these two views can be regarded as an effective anomaly signal to assess the trustworthiness of each triple in the original KG.

$$AT(h, r, t) = sim(x_i, z_i), \tag{17}$$

where $x_i = \mathcal{G}_{local}(h, r, t)$, $z_i = g_{attri}(h, r, t)$ are the representations of triple (h, r, t), learned from *relational hypergraph* and *attribute hypergraph*, respectively.

Triple confidence. Finally, we measure the confidence score of each triple based on the previously learned features and define final score functions for detecting potential errors as follows:

$$\begin{aligned} C(h, r, t) = \sigma(&LT(h, r, t) + \lambda_1 \cdot GT(h, r, t) \\ &+ \lambda_2 \cdot AT(h, r, t)), \end{aligned} \tag{18}$$

where $LT(h, r, t)$ reflects the degree of self-conformity within the local relational triple, $GT(h, r, t)$ denotes the consistency between local-global structure, $AT(h, r, t)$ measures the similarity of the same sample between *relational hypergraph* and *attribute hypergraph*. When the confidence value is larger, the triple (h, r, t) is more likely to be a normal one.

Algorithm 1: Error-aware KG Embedding Learning

Input: A real-world Knowledge graph $\mathcal{G} = (\mathcal{E}, \mathcal{R})$ with entity attributes $\mathcal{A}$.
Output: Error-aware KG Embeddings, i.e, $\mathbf{E}$ and $\mathbf{R}$.

```
Initialize network parameters, learnable embedding and
  confidence metric Construct two subgraphs G_r, i.e.,
  relational hypergraph and G_a, i.e., attribute hypergraph
while not converged do
  for (h, r, t) ∈ G do
    p_i ← modelling local relational triple structure,
      defined in Eq. (3);
    x_i ← modelling triple embedding from relational
      hypergraph G_r defined in Eq. (6);
    z_i ← modelling triple embedding from attribute
      hypergraph G_a, defined in Eq. (11);
    Update the confidence score of each triple (h, r, t)
      as defined in Eq.(18);
    Incorporate the learned confidence score into the
      KG embedding loss and jointly optimize the
      model with contrastive loss, define in Eq.(19).
```

3.3 Joint Adaptive Training Scheme

In this section, we introduce the details of the training scheme to answer the third research question mentioned at the beginning of this paper, i.e. how to make the best of confidence information learned from the detection module to get the error-aware KG embeddings. Concretely, we define a comprehensive objective function to tightly integrates the KG embedding model and confidence learning model that work jointly to get the error-aware embeddings. The learning process of AEKE is summarized in Algorithm 1.

$$\mathcal{L} = \mathcal{L}_{con} + \beta \mathcal{L}_{emb}. \tag{19}$$

Under this scheme, KG representation learning and confidence learning mechanism could be mutually beneficial for each other in each iteration that the former provides a promising embedding in latent space for the latter, while the latter singles out less credible triples to improve the performance of the former. Through iteratively joint estimation of triple confidence, we aim at reducing impacts from potential anomalies and learn optimal representation of the target KG.

4 Experiments

In this section, we conduct comprehensive experiments on a variety of real-world KGs to verify the effectiveness of the proposed framework AEKE. Specifically, we aim to answer the following questions.

- ***Q1*** (§ 4.2): Is our proposed AEKE valid and effective for distinguishing noisy triples?
- ***Q2*** (§ 4.3): How effective is the KG embeddings learned by AEKE in comparison with the state-of-the-art KG embedding models?
- ***Q3*** (§ 4.4): How does each component of AEKE contribute to its prominent performance?
- ***Q4*** (§ 4.5): How do the hyperparameters influence the performance of AEKE?
- ***Q5*** (§ 4.6): Is our proposed AEKE an efficient method compared to baselines?
- ***Q6*** (§ 4.7): How does our proposed AEKE perform error-aware embedding in real-world scenarios?

4.1 Datasets

We evaluate our proposed AEKE on three real-world benchmark datasets, including FB15K-237, YAGO15K, and DB15K. Due to the human curation, these benchmark datasets contain highly reliable facts. Following prior work [23], [24], we amplify each of three datasets by incorporating 5% and 15% noisy triples to imitate real-world errors, respectively. Since most errors in real-world KGs derive from the misunderstanding between similar entities, e.g., the error $(Newton, Nationality, England)$ is more likely to occur in real-world KGs than $(Newton, Nationality, Google)$, in this paper, the noisy triples are appropriately generated in the following way. Given a positive triple (h, r, t), the head or tail entity is replaced to form a harder and more confusing negative triple (h', r, t) or (h, r, t'), where h' (or t') should have appeared in the head (or tail) position with the same relation in the dataset.

FB15K-237 [36] stands as the widely-applied subset with 114 relations and 10054 entities under Freebase, which is known as the huge knowledge base with over 1 billion triples. FB15K-237 proves to be better than FB15K-237 by getting over the challenges FB15K-237 faced for inverse relations and keeping symmetrical, asymmetrical and combinatorial relationships, as well as the attributes.

DB15K [37] is a subset of Wikidata, making up for the weakness of Wikipedia. To avoid test leakage, it also excludes inverse relations using the same procedure as the derivation of FB15K-237.

YAGO15K [38] augmented WordNet with over one million entities, transforming WordNet from the plane data to a knowledge graph. YAGO15K gets formed based on YAGO aligned with the entities in Freebase.

4.2 Capability of AEKE in Distinguishing KG Errors (Q1)

In this section, we conduct experiment on KGs in terms of KG error detection task to verify the capability of AEKE in distinguishing errors. Specifically, we rank all the triples in the target KG according to their confidence scores in ascending order. The top ranked triples are treated as potential errors. The experiments are conducted on three benchmark datasets, including FB15K-237, DB15K and YAGO15K with noisy triples to be different ratios of 5% and 15% of KGs. In the following part, we elaborate the experiment settings and prerequisites of baseline models and evaluation metrics.

4.2.1 Baselines

From three aspects we single out models as the baseline to evaluate the error detection performance, among which **CKRL** [23] and **NoiGAN** [24] represent the error-aware embedding methods; **TTMF** [15], **CAGED** [32], and **CrossVal** [25] represent the state-of-the-art error detection alternatives; **TransE** [39], **DistMult** [40], **ComplEx** [41], **SimplE** [42] **TuckER** [43] and **EARL** [44] are picked for being representative knowledge graph embedding methods. The details of the baseline methods are elaborated as follows:

TransE [39] assumes that entities and relations are embedded in the same space, allowing for the approximation of the tail entity related to a given head entity and relation after training. As a result, the original triples are transformed into word vectors using either L1 or L2 norm.

TABLE 2: Error detection results of Precision@K and Recall@K based on the three datasets with anomaly ratio $= 5\%$.

		FB15K-237					DB15K					YAGO15K				
		K=1%	K=2%	K=3%	K=4%	K=5%[a]	K=1%	K=2%	K=3%	K=4%	K=5%[a]	K=1%	K=2%	K=3%	K=4%	K=5%[a]
$Precision@K$	TransE	0.756	0.674	0.605	0.546	0.488	0.671	0.566	0.535	0.472	0.443	0.534	0.455	0.362	0.319	0.279
	ComplEx	0.718	0.651	0.590	0.534	0.485	0.679	0.612	0.532	0.469	0.424	0.503	0.426	0.369	0.310	0.273
	DistMult	0.709	0.646	0.582	0.529	0.483	0.638	0.587	0.523	0.499	0.445	0.513	0.423	0.347	0.347	0.302
	SimplE	0.744	0.667	0.611	0.556	0.515	0.709	0.616	0.554	0.504	0.455	0.560	0.459	0.373	0.319	0.281
	TuckER	0.742	0.680	0.614	0.552	0.514	0.711	0.630	0.550	0.509	0.452	0.549	0.454	0.375	0.331	0.276
	EARL	0.762	0.692	0.639	0.582	0.531	0.729	0.652	0.573	0.530	0.483	0.578	0.487	0.394	0.338	0.303
	TTMF	0.815	0.767	0.713	0.612	0.579	0.744	0.685	0.623	0.557	0.510	0.701	0.569	0.454	0.413	0.346
	CAGED	0.852	0.796	0.735	0.665	0.595	0.802	0.722	0.658	0.596	0.554	0.724	0.599	0.489	0.429	0.373
	CKRL	0.789	0.736	0.684	0.630	0.574	0.787	0.723	0.634	0.599	0.526	0.662	0.531	0.438	0.382	0.327
	NoiGAN	0.837	0.788	0.727	0.649	0.585	0.823	0.718	0.676	0.606	0.553	0.737	0.592	0.477	0.403	0.379
	CrossVal	0.874	0.814	0.742	0.667	0.596	0.819	0.707	0.645	0.572	0.548	0.754	0.647	0.562	0.500	0.424
	AEKE	**0.892**	**0.822**	**0.753**	**0.691**	**0.614**	**0.853**	**0.756**	**0.705**	**0.625**	**0.582**	**0.778**	**0.676**	**0.590**	**0.519**	**0.440**
$Recall@K$	TransE	0.151	0.269	0.363	0.437	0.488	0.134	0.226	0.321	0.377	0.443	0.106	0.182	0.217	0.255	0.279
	ComplEx	0.144	0.260	0.354	0.427	0.485	0.135	0.244	0.319	0.375	0.424	0.100	0.170	0.221	0.248	0.273
	DistMult	0.142	0.258	0.349	0.423	0.483	0.127	0.234	0.313	0.399	0.445	0.102	0.169	0.208	0.277	0.302
	SimplE	0.149	0.267	0.366	0.445	0.515	0.142	0.246	0.332	0.403	0.455	0.112	0.184	0.224	0.255	0.281
	TuckER	0.148	0.272	0.368	0.442	0.514	0.142	0.252	0.330	0.407	0.452	0.110	0.182	0.225	0.265	0.276
	EARL	0.152	0.277	0.383	0.466	0.531	0.146	0.261	0.344	0.424	0.483	0.116	0.195	0.236	0.270	0.303
	TTMF	0.163	0.306	0.427	0.489	0.579	0.148	0.274	0.373	0.445	0.510	0.140	0.227	0.272	0.330	0.346
	CAGED	0.171	0.318	0.441	0.532	0.595	0.160	0.288	0.395	0.477	0.554	0.145	0.240	0.293	0.343	0.374
	CKRL	0.158	0.294	0.410	0.504	0.574	0.157	0.289	0.380	0.479	0.526	0.132	0.212	0.262	0.305	0.327
	NoiGAN	0.167	0.315	0.436	0.519	0.585	0.164	0.287	0.405	0.484	0.553	0.147	0.236	0.286	0.322	0.379
	CrossVal	0.175	0.325	0.445	0.533	0.596	0.163	0.282	0.387	0.457	0.548	0.150	0.258	0.337	0.400	0.424
	AEKE	**0.178**	**0.328**	**0.452**	**0.552**	**0.614**	**0.170**	**0.302**	**0.423**	**0.500**	**0.582**	**0.155**	**0.270**	**0.354**	**0.415**	**0.440**

[a] Please note that according to Eqs.(20)-(21), when K equals 5%, $Precision@K$ and $Recall@K$ are equal.

DistMult [40] is a bi-linear model that calculates the confidence of potential semantics for entities and relations in the vector space. It simplifies the RESCAL model by representing the relational matrix as a diagonal matrix, removing the limitation. However, DistMult can only handle symmetric relations in a knowledge graph.

ComplEx [41] is another notable bi-linear model that builds upon DistMult. It improves upon DistMult by introducing complex numbers and expanding the model into the complex number space. This enhancement enables ComplEx to handle both symmetric and asymmetric relations in a knowledge graph.

SimplE [42] SimplE embeddings offer interpretability and allow the integration of specific background knowledge by employing weight tying. It provides evidence of its complete expressiveness and establishes a bound on the size of its embeddings to ensure full expressivity.

TuckER [43] is a linear model that offers a relatively simple approach to link prediction in knowledge graphs. It leverages the Tucker decomposition of a binary tensor containing known facts.

EARL [44] focuses on learning embeddings for a specific group of entities referred to as reserved entities. To obtain embeddings for the entire set of entities, it encodes their unique characteristics by considering their connected relations, k-nearest reserved entities, and multi-hop neighbors.

TTMF [15] utilizes semantic information to calculate the trustworthiness of triples in order to distinguish between normal instances and anomalies. This differentiation is achieved through corresponding confidence values.

CAGED [32] By joint training with KG embedding and contrastive learning loss, CAGED assesses the trustworthiness of each triple based on two learning signals, i.e., the consistency of triple representations across multi-views and the self-consistency within the triple.

NoiGAN [24] aims to learn noise-aware knowledge graph embeddings. It combines error detection and completion tasks using a unified Generative Adversarial Networks (GAN) framework.

CKRL [23] is an advanced confidence-aware knowledge representation learning method. It aims to overcome the assumption made in conventional KRL, where all triples in the original graph are considered correct. CKRL focuses more on triples with lower confidence, as they may indicate potential noise.

CrossVal [25] suggests using an external human-curated knowledge graph as an auxiliary information source to aid in error detection within a target knowledge graph. The external knowledge graph is constructed based on human-curated knowledge repositories and tends to have high precision.

4.2.2 Evaluation protocol

The implementation of both the baselines and our proposed framework is carried out using PyTorch. For the baseline methods, we utilize the publicly available codes for conducting our experiments. Training of our proposed framework and the baselines is performed on a Nvidia 3090 GPU server. Specifically, we employ the Adam optimizer with a fixed batch size of 256 to optimize all models. The model parameters are initialized using the default Xavier initializer, and the initial learning rate is set to 0.01. Additionally, we maintain a fixed embedding size of 100 for all models.

In order to assess the performance of the baseline approaches, we utilize ranking measures. These measures involve calculating ranks based on the scores assigned by the models to the triples in the knowledge graph (KG). A lower score indicates a lower likelihood of a triple being correct. The triples in the target KG are ranked in ascending order according to their scores, where the top-ranked triples have a higher probability of being incorrect. To provide a fair

TABLE 3: Results of knowledge graph completion.

Dataset		FB15K-237				DB15K				YAGO15K			
Metrics		MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10	MRR	Hit@1	Hit@3	Hit@10
0%	TransE	0.302	0.211	0.344	0.468	0.420	0.385	0.431	0.509	0.311	0.199	0.369	0.523
	SimplE	0.288	0.202	0.314	0.455	0.437	0.392	0.445	0.505	0.296	0.188	0.344	0.487
	EARL	0.308	0.209	0.338	0.474	0.447	0.389	0.459	0.508	0.317	0.199	0.372	0.509
	RGCN	0.363	0.306	0.387	0.512	0.474	0.410	0.487	0.576	0.381	0.269	0.424	0.588
	RGHAT	0.422	0.362	0.446	**0.535**	0.483	0.422	0.499	0.588	0.412	0.295	0.467	0.618
	NoiGAN	0.424	0.371	0.445	0.526	0.480	0.421	0.483	0.581	0.413	0.310	0.465	0.605
	CKRL	0.410	0.362	0.438	0.536	0.482	0.416	0.490	0.586	0.408	0.298	0.461	0.610
	Ours	**0.427**	**0.375**	**0.453**	0.534	**0.486**	**0.426**	**0.503**	**0.587**	**0.418**	**0.316**	**0.474**	**0.622**
5%	TransE	0.294	0.201	0.335	0.465	0.407	0.358	0.417	0.481	0.288	0.161	0.345	0.493
	SimplE	0.266	0.178	0.281	0.416	0.407	0.354	0.412	0.466	0.277	0.166	0.313	0.455
	EARL	0.284	0.194	0.308	0.437	0.407	0.366	0.416	0.466	0.283	0.180	0.348	0.468
	RGCN	0.348	0.261	0.375	0.493	0.456	0.408	0.472	0.551	0.365	0.250	0.397	0.562
	RGHAT	0.401	0.342	0.442	0.507	0.460	0.411	0.472	0.565	0.392	0.275	0.442	0.580
	NoiGAN	0.418	0.360	0.440	0.515	0.474	0.415	0.479	0.578	0.404	0.305	0.457	0.598
	CKRL	0.400	0.340	0.426	0.512	0.471	0.410	0.483	0.572	0.397	0.292	0.449	0.594
	Ours	**0.424**	**0.369**	**0.447**	**0.519**	**0.482**	**0.424**	**0.499**	**0.583**	**0.412**	**0.308**	**0.465**	**0.612**
15%	TransE	0.267	0.185	0.309	0.451	0.386	0.340	0.401	0.469	0.276	0.149	0.334	0.473
	SimplE	0.242	0.154	0.241	0.361	0.368	0.314	0.358	0.416	0.231	0.134	0.288	0.419
	EARL	0.259	0.169	0.266	0.381	0.368	0.326	0.362	0.416	0.237	0.147	0.32	0.431
	RGCN	0.325	0.237	0.356	0.476	0.437	0.398	0.455	0.534	0.343	0.238	0.375	0.549
	RGHAT	0.395	0.332	0.416	0.498	0.442	0.398	0.454	0.540	0.371	0.258	0.427	0.565
	NoiGAN	0.409	0.348	0.427	0.506	0.463	0.401	0.462	0.566	0.385	0.295	0.448	0.589
	CKRL	0.396	0.334	0.422	0.501	0.458	0.400	0.467	0.559	0.375	0.275	0.437	0.579
	Ours	**0.417**	**0.363**	**0.440**	**0.512**	**0.475**	**0.411**	**0.492**	**0.579**	**0.401**	**0.299**	**0.451**	**0.600**

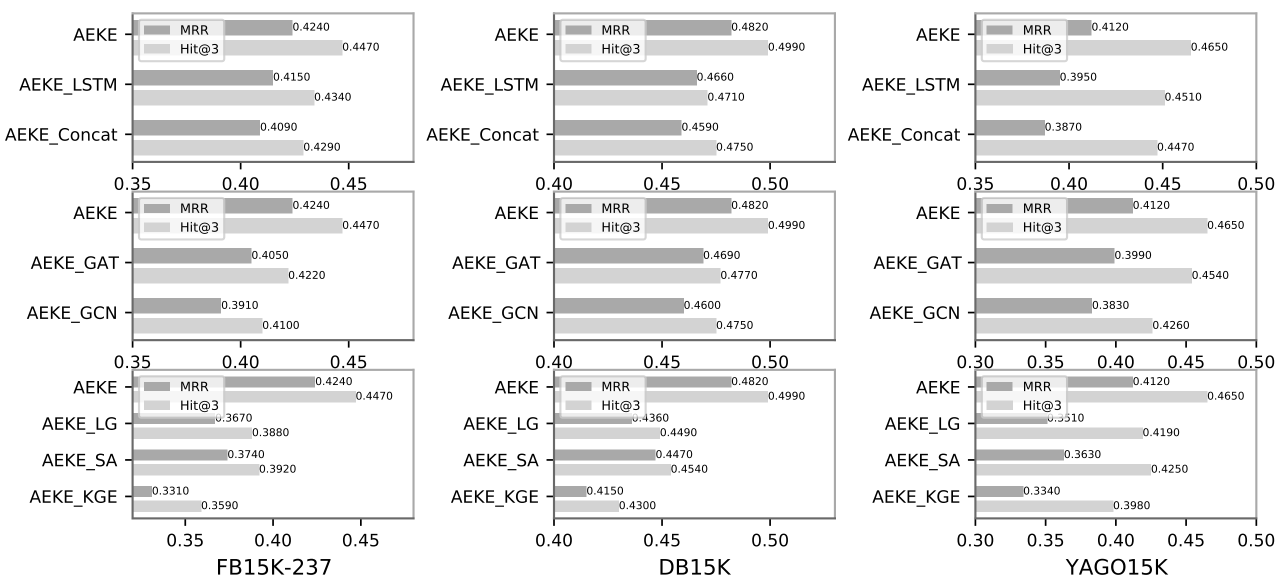


Fig. 5: KG completion results of AEKE variants based on the three datasets with anomaly ratio = 5%.

assessment of KG validation performance, we employ the following two evaluation measures.

Precision@K assesses the percentage of real anomalies found in the first K queries.

$$\text{Precision@K} = \frac{|\text{ Errors Discovered in Top K Ranking }|}{\text{K}}. \quad (20)$$

Recall@K measures reflects the percentage of real anomalies found in the overall number of ground truth anomalies.

$$\text{Recall@K} = \frac{|\text{ Errors Discovered in Top K Ranking }|}{|\text{ Total number of Errors in the KG }|}. \quad (21)$$

4.2.3 Experimental results

We now answer the first question, i.e., **Q1** evaluating the effectiveness of AEKE. The experimental results are summarized in Table 2. We have three observations as follows.

Obs. 1.AEKE demonstrates superior performance compared to both embedding methods and state-of-the-art error detection baselines. Our proposed model, AEKE, outperforms all baselines in terms of recall and precision evaluation metrics. Specifically, at a 5% anomaly ratio and k value of 5%, AEKE achieves a 1.8% improvement over the second-best method.

Obs. 2. Knowledge graph embedding baselines, such as TransE, ComplEx, and DistMult, generally yield satisfactory results. However, when compared to tailored error detection and error-aware methods, these knowledge graph representation baselines exhibit inferior performance. The reason behind this is that KG embedding frameworks do not account for errors in the KG, resulting in the inability

TABLE 4: Results of Precision@K% and Recall@K% based on FB15K-237 with anomaly ratio = 5%.

			$Precision@K$		
Top@K	1%	2%	3%	4%	5%[a]
TransE(none)	0.7565	0.6749	0.6058	0.5466	0.4884
TransE(attr)	0.7668	0.6808	0.6214	0.5493	0.5011
ComplEx(none)	0.7185	0.6508	0.5900	0.5342	0.4856
ComplEx(attr)	0.7250	0.6589	0.5924	0.5453	0.4873
DistMult(none)	0.7088	0.6461	0.5828	0.5298	0.4839
DistMult(attr)	0.7263	0.6501	0.6000	0.5306	0.4862
AEKE(none)	0.8522	0.7959	0.7352	0.6658	0.5957
AEKE	**0.8919**	**0.8223**	**0.7529**	**0.6916**	**0.6141**
			$Recall@K$		
Top@K	1%	2%	3%	4%	5%[a]
TransE(none)	0.1513	0.2700	0.3635	0.4373	0.4884
TransE(attr)	0.1534	0.2723	0.3728	0.4394	0.5011
ComplEx(none)	0.1437	0.2603	0.3540	0.4273	0.4856
ComplEx(attr)	0.1450	0.2636	0.3554	0.4362	0.4873
DistMult(none)	0.1418	0.2584	0.3497	0.4239	0.4839
DistMult(attr)	0.1452	0.2600	0.3600	0.4245	0.4862
AEKE(none)	0.1706	0.3180	0.4411	0.5326	0.5957
AEKE	**0.1783**	**0.3289**	**0.4517**	**0.5532**	**0.6141**

[a] Please note that according to Eqs.(20)-(21), when K equals 5%, $Precision@K$ and $Recall@K$ are equal.

to learn discriminative representations for normal and noisy triples. This outcome emphasizes the need for an error-aware representation learning framework to achieve robust KG embeddings.

Obs. 3. The inclusion of auxiliary human-curated information proves to be useful in detecting errors in KGs. For instance, CrossVal, which utilizes an external human-curated KG as auxiliary information, demonstrates better detection performance than other baselines. However, our model surpasses it due to its ability to leverage the relational structure within triples, the global contextual structure across triples, and the rich semantics derived from entity attributes. This comprehensive approach allows our model to excel in error detection.

4.3 Quality of KG Embeddings leaned by AEKE (Q2)

The goal of our model is to learn effective knowledge graph embeddings that could facilitate various applications. To verify the quality and effectiveness of learned embeddings, we evaluate AEKE in terms of KG completion task. Knowledge graph completion is a traditional evaluation task that aims to complete the incomplete triples that lack a head entity, tail entity or relation. In the following part, we elaborate the experiment settings and prerequisites of baseline models and evaluation measurements.

4.3.1 Baseline methods

To validate the quality of KG embeddings leaned by our proposed AEKE, we compare it with the *strongest* baselines according to our best knowledge. Following the taxonomy aforementioned in § 5, we divide all baselines into three categories: (i) *embedding-based*: **TransE** [39], **SimplE** [42], **EARL** [44]; (ii) state-of-art completion models: **RGCN** [45] and **RGHAT** [46]; (iii) *Error-aware* embedding method: **NoiGAN** [24] and **CKRL** [23].

4.3.2 Evaluation protocol

In this research paper, our primary focus revolves around entity prediction. To provide a more precise description, we establish the KG completion as a task that involves predicting either the head entity in a given query $(?, r, t)$ or the tail entity in a given query $(h, r, ?)$. To be specific, we mask the head or tail entity of each triple in the test dataset and require each method to predict the missing entities. To maintain consistency with the previous study [23], [24], we utilize two evaluation metrics: 1) Mean Rank of correct entities, denoted as MRR, and 2) Hits@K, which denotes the proportion of correct answers ranked within the top K positions.

4.3.3 Experiment results

We conduct the experiments on three benchmark datasets, including FB15K-237, DB15K and YAGO15K with noisy triples to be different ratios of 5% and 15% of KGs. Evaluation results are shown in Table 3. In general, we have the following observations.

Obs. 1. AEKE consistently outperforms the embedding-based models and other tailored KG completion competitors over different anomaly ratios, which verifies the quality and effectiveness of KG embeddings learned by AEKE.

Obs. 2. Error-aware embedding methods, including NoiGAN, CKRL and our proposed AEKE, always show better performance than other models. And the embedding-based methods, i.e. TransE, SimplE, and EARL, are always surpassed by both KG completion models, i.e. RGCN and RGHAT. It is because embedding-based methods that are trained in a pair-wise mode and only model the local relational structure of triples, are more likely to over-fit noisy information.

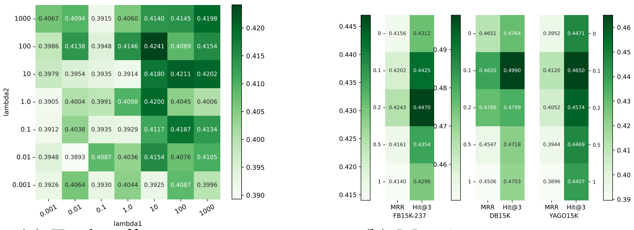
Obs. 3. As the anomaly rate increases, the performance gap between the baseline models and our AEKE become more significant. Specifically, comparing with second-best method, our AEKE just gets the improvements of 0.3%, 0.6% and 0.5% on FB15k-237, DB15K and YAGO15K respectively. As the anomaly rate increases to 15%, our AEKE gets more significant improvements of 0.8%, 1.2% and 1.6% on FB15k-237, DB15K and YAGO15K, respectively. It indicates that our proposed framework is more robust, especially for KGs with larger scale of noises.

4.4 Ablation Study (Q3)

We now investigate the third question. In this part, four pairs of variants of AEKE are used for this ablation study.

4.4.1 The role of attribute hypergraph encoder

Entity attributes can be used to implicitly portray the semantics of entities, but it is hard to integrate different KG components, i.e. entities, relations and entity attributes into a suitable vector space since they always exhibit rather distinct characteristics. To leverage the semantics contained in entity attributes to guide the KG embedding model learning against the impact of erroneous triples, in this paper, we first build an attribute-based hypergraph and then propose a novel attribute encoder to model the attribute information with heterogeneous structure. To test the effect of our proposed attribute encoder, we remove the attribute view from AEKE, and denote the variant as AEKE(none). In this part, we compare our model and its variant with three KG embedding



(a) Trade-off parameters. (b) Margin parameter γ

Fig. 6: Margin parameter and trade-off parameters analysis on the three datasets.

models in terms of the capability in distinguishing KG errors. Specifically, we use TransE, ComplEx and DistMult to model the KGs without attribute information and denote them as TransE(none), ComplEx(none) and DistMult(none), respectively. In the same time, we directly integrate attributes into KG embedding frameworks by initializing the entity feature with the concatenated attribute semantics, and denote them as TransE(attr), ComplEx(attr) and DistMult(attr).

As shown in Table 4, after introducing the attributes, all baselines as well as our model, present better performance, which verifies the value and importance of attribute information in KG representation learning. But the improvement of the baselines is quite marginal compared to our model. Particularly, there is only an increase of 0.99%, 0.65%, 1.75% for TranE, ComplEx and DistMult, respectively. And our model shows a significant improvement of 3.97% at K equals 1%. It is because these baseline models attempt to integrate attributes into KG embedding framework by directly initializing the entity feature with the learned attribute semantics and train the model in pair-wise mode. However, in this fusing way, it is hard to measure the complex correlation and dependency between entity attributes and knowledge graph structure, which is crucial to guide the embedding model to filter out noisy information from hidden erroneous triples.

4.4.2 The role of relational hypergraph encoder

To assess the effectiveness of capturing translational structure within each triple using a local information modeling layer, we introduce two variations: AEKE_LSTM and AEKE_Concat. AEKE_LSTM replaces the Bi-LSTM units in our local information modeling layer with LSTM, while AEKE_Concat removes the Bi-LSTM units and directly concatenates randomly initialized embeddings of the head entity, relation, and tail entity to form the local representation of each triple. Our results indicate a significant drop in performance for both variants, with AEKE_LSTM outperforming AEKE_Concat. This outcome is expected because a simple concatenation approach can lead to the loss of local structural information.

Next, to validate the error-aware functionality of our global encoder, we substitute our tailored graph encoder with RGCN [45] and RGHAT [46], creating AEKE_GCN and AEKE_GAT, respectively. As observed from Fig. 5, AEKE_GAT outperforms AEKE_GCN, but there remains a noticeable gap compared to our model. This gap arises due to RGCN assuming that all observed triples in the KG are correct, which can result in overfitting on noisy facts and the failure to detect errors. On the other hand, RGHAT employs an attention mechanism to learn KG representations, which has the potential to filter out some noisy information. However, AEKE_GAT still does not exhibit excellent performance in the error detection task. This can be attributed to the fact that RGHAT applies the attention mechanism from the perspective of entities and relations, whereas KG errors often occur at the triple level, where mismatches between the head entity, tail entity, and the corresponding relation are common. In contrast, our graph encoder incorporates a customized error-aware attention layer that takes triple-level embeddings as input. This tailored approach enables us to effectively filter out noisy facts.

TABLE 5: Complexity analysis (space complexity, time complexity, the overall training time T_{total} and the average training time in each epoch T_{epoch}).

Method	Space Complexity	Time Complexity	on FB15K-237 T_{epoch}	T_{total}
TransE	$\mathcal{O}(N_e m + N_r n)$	$\mathcal{O}(N_t m)$	2.5 s	0.4 h
SimplE	$\mathcal{O}(N_e m + 2N_r n)$	$\mathcal{O}(N_t mn)$	3.9 s	0.6 h
EARL	$\mathcal{O}(N_e m + N_r n)$	$\mathcal{O}(N_e{}^2 m + N_e m^2)$	6 s	0.7 h
NoiGAN	$\mathcal{O}(N_e m + N_r n + N_t t)$	$\mathcal{O}(N_t m^3)$	96 s	2.9 h
CAGED	$\mathcal{O}(N_e m + N_r n + N_t t)$	$\mathcal{O}(N_e{}^2 m + N_e m^2)$	131 s	1.7 h
CrossVal	$\mathcal{O}(N_e m + N_r n + N_k d)$	$\mathcal{O}(N_e m^2 + N_k d^2)$	590 s	4.5 h
AEKE	$\mathcal{O}(N_e m + N_r n + N_a m)$	$\mathcal{O}((N_e + N_a) m^2)$	375 s	3.3 h

4.4.3 The role of joint adaptive training scheme

To evaluate the effectiveness of our joint optimization approach, we conduct experiments using different training losses, resulting in three variants: AEKE_SA, AEKE_LG, and AEKE_KGE. From the results depicted in Fig. 5, it is evident that all three variants are inferior to our proposed model. Notably, AEKE_KGE performs particularly poorly, even when compared to straightforward baseline methods. This discrepancy arises due to the fact that the negative sampling employed in the KG embedding framework primarily assists our model in learning rich structural and semantic information within triples. However, such local features alone are insufficient to enable effective error detection across the entire KG. Contrastive learning, on the other hand, complements the negative samples by facilitating the learning of distinguishable across-view representations. With the distinguishable across-view representations, denoising noisy triples on KGs will be much more effective.

4.5 Parameter Analysis (Q4)

In this section, we investigate the impact of three key parameters in AEKE and report the results in Figs. 6(a) and 6(b).

4.5.1 Trade-off parameters in confidence score function

As shown in Eq.(18), λ_1 and λ_2 are trade-off coefficients that balance the contribution of three learning signals for error detection, i.e., the self-contradictory within the triple embedding, global alignment among triples, and the conformity between attributes and the graph structure. The larger λ_1 and λ_2 indicate that effective error detection relies more on the signals of global acknowledgement and cross-view conformity, respectively. To search for suitable trade-off parameters, we vary them from 10^{-3} to 10^{3}. We perform completion on FB15K-237 and the performance in terms of

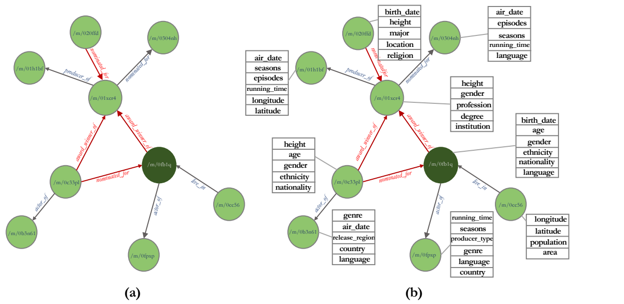


Fig. 7: Case study on FB15K-237. (a) presents a 2-hop subgraph extracted from FB15K-237 centering at the target entity $/m/0fb1q$. While (b) demonstrates the corresponding subgraph with attribute information attached to each entity.

MRR is shown in Fig. 6(b). From the results, we observe that the completion performance is much better when the value of λ_1 and λ_2 is larger than 1, with an optimal result at $\lambda_1 = 10$ and $\lambda_2 = 100$. We omit the results on the other two datasets since they show similar trends.

4.5.2 Margin parameter

The translation-based KG embedding loss function utilizes γ as a margin to regulate the distance between the representations of positive and negative pairs, as indicated in Eq.(1). In order to examine the influence of γ, we vary its value from 0 to 1.0 and present the results in Fig.6(a). The DB15K dataset achieves optimal outcomes when λ_1 is approximately 0.1. From a global perspective, the variations in the parameter λ do not appear to have a significant impact, as its value remains stable within a specific range, as observed in YAGO15K. However, for FB15K-237, the performance notably improves when λ is set below 0.5, with the best result obtained at 0.2. The overall performance changes exhibit a stable trend with minimal fluctuation as γ varies. This is primarily attributed to the joint training approach employed, which prevents the model from becoming suboptimal.

4.6 Complexity Analysis (Q5)

In this section, we conduct a comprehensive complexity analysis to investigate the efficiency of our proposed AEKE. N_e, N_r and N_t denote the number of entities, relations and triples, respectively. m, n and t are the dimensions of entity, relation and triple embeddings. N_k is the number of entities in external KG used in CrassVal, and d is the dimension of entity embedding in external KG. N_a is the number of attributes in AEKE. As shown in Table 5, we have the following observations: 1) Embedding-based methods including TransE, SimplE, and EARL are time and space efficient and run faster than error detection and error-aware methods in general since they only need to calculate the mean square losses. In the same time, EARL costs more time than TransE and SimplE since it introduces multi-layer convolution network in knowledge graph embedding. 2) In terms of computation time per epoch, NoiGAN and CAGED outperform AEKE and CrossVal. This is attributed to NoiGAN and CAGED leveraging fewer data sources for KG modeling. Nonetheless, the overall training time of NoiGAN is considerably larger than CAGED due to the challenges in convergence that are commonly encountered with GAN-based models. 3) Although both our proposed AEKE and CrossVal use extra data sources for KG modeling, our proposed AEKE shows better efficiency than CrossVal. It is because entity attributes used in AEKE are naturally attached to entities, which can be directly leveraged for KG learning, while CrossVal need extra time to align the knowledge from the original KG and external KG since a entity may have different IDs or names in different KGs.

TABLE 6: Confidence score of each triple in the sugraph.

Triple: {head entity, relation, tail entity}	Triple Confidence		
	Grounded	NoiGAN	AEKE
$\{/m/01xcr4, nominated\_for, /m/0304nh\}$	Normal	0.7553	0.8921
$\{/m/01xcr4, producer\_of, /m/01h1bf\}$	Normal	0.7017	0.8427
$\{/m/0fb1q, live\_in, /m/0cc56\}$	Normal	0.8257	0.9254
$\{/m/0fb1q, actor\_of, /m/0fpxp\}$	Normal	0.7682	0.8710
$\{/m/0c33pl, actor\_of, /m/0b3n61\}$	Normal	0.7944	0.8549
$\{/m/0fb1q, award\_winner\_of, /m/01xcr4\}$	Erroneous	0.6836	0.1684
$\{/m/0c33pl, nominated\_for, /m/0fb1q\}$	Erroneous	0.2673	0.1086
$\{/m/020ffd, nominated\_for, /m/01xcr4\}$	Erroneous	0.7864	0.4309
$\{/m/0c33pl, award\_winner\_of, /m/01xcr4\}$	Erroneous	0.6781	0.1146

4.7 Case Study on Interpretability (Q6)

We conduct a case study to investigate how AEKE can perform error-aware learning in real-world scenarios. We select the entity $/m/0fb1q$ as the target entity and visualize its 2-hop subgraph and the corresponding attribute information in Fig. 7. There are nine triples in the subgraph, five of which are normal ones and the other four are erroneous (in red). To verify the ability of AEKE in distinguishing noisy triples, we show the confidence score of each triple in Table 6. To make the effect measurable and comparable, apart from the ground truth labels, we also include the confidence score that are learned from the SOTA error-aware embedding method, i.e., NoiGAN, as counterparts.

Based on the results shown in Table 6, we have the following observations: (i) compared to NoiGAN, our proposed AEKE shows its superiority in distinguishing erroneous triples. To be clear, NoiGAN learn a confidence set of $\{0.6836, 0.2673, 0.7864, 0.6781\}$ for the four real-world errors, which means three erroneous triples have escaped the detection with the help of the conflicting and complicated structure. Our method, in contrast, assigns a confidence set of $\{0.1684, 0.1086, 0.4309, 0.1146\}$ to four triples, which indicates that it accurately identifies three errors, i.e., only one erroneous triple is oddly assigned a slightly higher score of 0.4309. (ii). When measuring the correctness of normal triples, AEKE shows better stability and robustness than NoiGAN in most cases. Specifically, the normal triples get a confidence set of $\{0.8921, 0.8427, 0.9254, 0.8710, 0.8549\}$ using AEKE, which are generally closer to 1 (the ground truth value), compared with the confidence set $\{0.7553, 0.7017, 0.8257, 0.7682, 0.7944\}$ learned from NoiGAN. Interestingly, the triple with more erroneous neighbors tends to have a lower confidence score in NoiGAN, while this phenomenon is not that obvious in our proposed AEKE. It is mainly because AEKE measures the triple confidence by considering three different anomaly signals, i.e, self-contradictory within relational triple structure, global consistency across triples, and attribute-structure dependency, while NoiGAN only relies on pure graph structure to estimate triple confidence, which makes it more vulnerable to neighboring erroneous instances.

5 Related Work

5.1 Knowledge Graph Embedding

A lot of efforts have been laid on the KG embedding to learn the representation of triples within KGs after establishment, contributing an essential base for downstream tasks [47], [48], [49], [50], [51]. Translational distance models represented by TransE [39] seize greater attention than before, more applications prove its reliability for embedding based on the design to embed entities and relations into a continuous vector space as the projection of the original triples. As to three methods which sharing roots with each other, Rescal [46] represents the bi-relational data into three-dimensional tensors, in the form of an entity matrix and a relation matrix. DistMult [40] simplifies Rescal with a bi-linear formulation, taking entities as low dimensional vectors and relations from a bi-linear mapping function.However, this kind of simplification brings the restriction of a symmetry problem, ComplEx [41] improved DistMult and fixed the problem by introducing complex value which enables the model to handle both symmetric and asymmetric problems.

KG embedding methods with external information. While the traditional ones are showing their advantages, more and more researches start to focus on attaching importance to external information of the original KG aiming to support better embedding methods. DKRL [52] introduces entitiy description into the KG embedding for better semantic understanding . It trains the energy function with mutual promotion of structural embedding and description embedding. KDCoE [53], similarly, also takes the advantage of abundant semantics as a semi-supervised algorithm, which embeds multilingual entity descriptions through a co-learning method for knowledge alignment across languages.

5.2 Knowledge Graph Error Detection

Error detection, being one of the most challenging problems, has been the subject of extensive research for many years. Initially, studies on error detection primarily focused on traditional methods such as rule-based approaches [54], classification-based methods [55], clustering-based methods [56], [57], distance-based methods [58], distribution-based methods [59], and others [60], [61]. Among these approaches, several notable methods have been developed specifically for error detection in knowledge graphs (KGs). One such method is KGClean [62], a classification-based framework that employs a novel approach called AL-detect to determine the truthfulness of triples and subsequently detect and repair erroneous data. Another method, SDValidate [59], utilizes statistical distributions to identify error relations in noisy KGs, without relying on any external knowledge. Inspired by SDValidate and PRA [63], PaTyBRED [55] incorporates type and path features into local relation classifiers to detect relation assertion errors in KGs. However, in many cases, it is challenging to acquire a sufficiently labeled dataset with explicit error labels, which limits the broader application of unsupervised learning methods in error detection tasks.

The advancement of graph embedding techniques, such as TransE [39], KBGAN [64], ConvE [65], and DistMult [34], has led to a significant focus on embedding-based error detection methods. Numerous models have been developed, demonstrating the effectiveness of embedding approaches on benchmark datasets. One notable model is TransT [66], which is a translating embedding model based on TransE [39] and incorporates triple trustworthiness to handle noisy knowledge graphs. TransT introduces two sub-models that consider entity types and entity descriptions, respectively, in order to calculate triple trustworthiness values. In the Correction Tower framework proposed by Abedini et al. [17], three distinct embedding-based solutions are introduced to address different types of errors, including outliers, inconsistent triples, and erroneous relations. These solutions are then combined to form a filter-like structure that detects and corrects errors within the knowledge graph.

5.3 Error-aware Knowledge Graph Embedding

Different from KG error detection that aims to get the anomaly labels of each triple, error-aware KG embedding is an end-to-end KG representation learning framework that encodes entities and relations into low-dimensional vector space with the consideration of erroneous triples during the learning phase. Vault [22] is the first work that aims to detect possible errors in KGs while learning knowledge representations. It estimates a probability score of reliability to determine the quality of a triple via several prior models fitted with existing KGs. A similar concept of judgments for each triple is also applied in CKRL [23] and NoiGAN [24]. The former model generates confidence scores for triples via internal structure information and utilizes them in representation learning to produce robust representations, while the latter improves it in the aspect of sample selection.

6 Conclusions and Future Work

Learning effective and robust embeddings of entities and relations in KGs can facilitate various downstream applications. Most existing efforts take advantage of the internal structure within KGs to help train a discriminative detector. But, they are confronted with the bottleneck that the internal topological information implied in pure KGs is far from abundant to support the validation task in real-world scenarios. In this paper, we propose a novel KG representation learning framework, i.e., AEKE, to incorporate the semantic information contained in entity attributes to automatically validate triples in KGs. We treat the original KG without attributes information as *relational hypergraph*, and build an *attribute hypergraph* based on these side-information and treat it as a congruent view of target KG. The confidence score of each triple is calculated by considering: self-contradictory within the triple; local-global consistency in graph structure; and structure-attribute homogeneity between triple-level views. Experiment results demonstrate that AEKE outperforms state-of-the-art KG error detection algorithms. Since real-world KGs are always evolving and introducing new knowledge, extending our method to temporal KG representation learning will be a very valuable and promising future work. Besides, in the future, we will also explore using the error-aware knowledge graph representation learning method in AEKE for downstream applications such as question answering and recommender systems.

7 ACKNOWLEDGEMENT

The work was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU 25208322).

REFERENCES

[1] Q. Wang, Z. Mao, B. Wang, and L. Guo, "Knowledge graph embedding: A survey of approaches and applications," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 12, pp. 2724–2743, 2017.

[2] F. Mahdisoltani, J. Biega, and F. M. Suchanek, "YAGO3: A knowledge base from multilingual wikipedias," in *Conference on Innovative Data Systems Research*, 2015.

[3] J. Lehmann, R. Isele, M. Jakob, A. Jentzsch, D. Kontokostas, P. N. Mendes, S. Hellmann, M. Morsey, P. Van Kleef, S. Auer *et al.*, "DBpedia – a large-scale, multilingual knowledge base extracted from wikipedia," *Semantic Web journal*, vol. 6, no. 2, pp. 167–195, 2015.

[4] P. Ernst, A. Siu, and G. Weikum, "Knowlife: A versatile approach for constructing a large knowledge graph for biomedical sciences," *BMC bioinformatics*, vol. 16, p. 157, 05 2015.

[5] Y. Chen, J. Kuang, D. Cheng, J. Zheng, M. Gao, and A. Zhou, "Agrikg: an agricultural knowledge graph and its applications," in *International conference on database systems for advanced applications*. Springer, 2019, pp. 533–537.

[6] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *The world wide web conference*, 2019, pp. 3307–3313.

[7] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge graph embedding based question answering," in *Proceedings of the twelfth ACM international conference on web search and data mining*, 2019, pp. 105–113.

[8] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, "Jointly embedding knowledge graphs and logical rules," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 192–202.

[9] J. Feng, M. Huang, M. Wang, M. Zhou, Y. Hao, and X. Zhu, "Knowledge graph embedding by flexible translation," in *Fifteenth International Conference on the Principles of Knowledge Representation and Reasoning*, 2016.

[10] S. Guo, Q. Wang, L. Wang, B. Wang, and L. Guo, "Knowledge graph embedding with iterative guidance from soft rules," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[11] K. Bollacker, R. Cook, and P. Tufts, "Freebase: A shared database of structured general human knowledge," in *AAAI*, vol. 7, 2007, pp. 1962–1963.

[12] S. Heindorf, M. Potthast, B. Stein, and G. Engels, "Vandalism detection in wikidata," in *CIKM*, 2016, pp. 327–336.

[13] R. Fasoulis, K. Bougiatiotis, F. Aisopos, A. Nentidis, and G. Paliouras, "Error detection in knowledge graphs: Path ranking, embeddings or both," *arXiv preprint arXiv:2002.08762*, 2020.

[14] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, B. Yang, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel *et al.*, "Never-ending learning," *Communications of the ACM*, vol. 61, no. 5, pp. 103–115, 2018.

[15] S. Jia, Y. Xiang, X. Chen, and K. Wang, "Triple trustworthiness measurement for knowledge graph," in *The World Wide Web Conference*, 2019, pp. 2865–2871.

[16] A. Melo and H. Paulheim, "Detection of relation assertion errors in knowledge graphs," in *Proceedings of the Knowledge Capture Conference*, 2017, pp. 1–8.

[17] F. Abedini, M. R. Keyvanpour, and M. B. Menhaj, "Correction tower: A general embedding method of the error recognition for the knowledge graph correction," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 34, no. 10, p. 2059034, 2020.

[18] Z. Zhang, F. Zhuang, H. Zhu, C. Li, H. Xiong, Q. He, and Y. Xu, "Towards robust knowledge graph embedding via multi-task reinforcement learning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[19] Y. Shan, C. Bu, X. Liu, S. Ji, and L. Li, "Confidence-aware negative sampling method for noisy knowledge graph embedding," in *2018 IEEE International Conference on Big Knowledge (ICBK)*. IEEE, 2018, pp. 33–40.

[20] J. Pujara, E. Augustine, and L. Getoor, "Sparsity and noise: Where knowledge graph embeddings fall short," in *Proceedings of the 2017 conference on empirical methods in natural language processing*, 2017, pp. 1751–1756.

[21] M. Nayyeri, S. Vahdati, E. Sallinger, M. M. Alam, H. S. Yazdi, and J. Lehmann, "Pattern-aware and noise-resilient embedding models," in *European Conference on Information Retrieval*. Springer, 2021, pp. 483–496.

[22] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 601–610.

[23] R. Xie, Z. Liu, F. Lin, and L. Lin, "Does william shakespeare really write hamlet? knowledge representation learning with confidence," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[24] K. Cheng, Y. Zhu, M. Zhang, and Y. Sun, "Noigan: Noise aware knowledge graph embedding with gan," 2019.

[25] Y. Wang, F. Ma, and J. Gao, "Efficient knowledge graph validation via cross-graph representation learning," in *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020, pp. 1595–1604.

[26] J. Lehmann, D. Gerber, M. Morsey, and A.-C. Ngonga Ngomo, "Defacto-deep fact validation," in *International semantic web conference*. Springer, 2012, pp. 312–327.

[27] M. N. Jeyaraj, S. Perera, M. Jayasinghe, and N. Jihan, "Probabilistic error detection model for knowledge graph refinement," in *Proceedings of the International Conference on Computational Linguistics and Intelligent Text Processing (CiCLing'19)*, 2019.

[28] M. Li, N. Gao, C. Tu, J. Peng, and M. Li, "Incorporating attributes semantics into knowledge graph embeddings," in *2021 IEEE 24th International Conference on Computer Supported Cooperative Work in Design (CSCWD)*. IEEE, 2021, pp. 620–625.

[29] Z. Liu, Y. Cao, L. Pan, J. Li, and T.-S. Chua, "Exploring and evaluating attributes, values, and structures for entity alignment," *arXiv preprint arXiv:2010.03249*, 2020.

[30] B. D. Trisedya, J. Qi, and R. Zhang, "Entity alignment between knowledge graphs using attribute embeddings," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 297–304.

[31] Y. Lin, Z. Liu, and M. Sun, "Knowledge representation learning with entities, attributes and relations," *ethnicity*, vol. 1, pp. 41–52, 2016.

[32] Q. Zhang, J. Dong, K. Duan, X. Huang, Y. Liu, and L. Xu, "Contrastive knowledge graph error detection," in *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, 2022, pp. 2590–2599.

[33] S. Wang, X. Huang, C. Chen, L. Wu, and J. Li, "Reform: Error-aware few-shot knowledge graph completion," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 1979–1988.

[34] T. Trouillon, C. R. Dance, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Knowledge graph completion via complex tensor factorization," *arXiv preprint arXiv:1702.06879*, 2017.

[35] J. C. Turner and P. J. Oakes, "The significance of the social identity concept for social psychology with reference to individualism, interactionism and social influence," *British Journal of Social Psychology*, vol. 25, pp. 237–252, 1986.

[36] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, and M. Gamon, "Representing text for joint embedding of text and knowledge bases," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1499–1509.

[37] Y. Liu, H. Li, A. Garcia-Duran, M. Niepert, D. Onoro-Rubio, and D. S. Rosenblum, "Mmkg: multi-modal knowledge graphs," in *The Semantic Web: 16th International Conference, ESWC 2019, Portorož, Slovenia, June 2–6, 2019, Proceedings 16*. Springer, 2019, pp. 459–474.

[38] F. Mahdisoltani, J. Biega, and F. Suchanek, "Yago3: A knowledge base from multilingual wikipedias," in *7th biennial conference on innovative data systems research*. CIDR Conference, 2014.

[39] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, and O. Yakhnenko, "Translating embeddings for modeling multi-relational data," *Advances in neural information processing systems*, vol. 26, 2013.

[40] B. Yang, W. tau Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," 2015.

[41] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, “Complex embeddings for simple link prediction,” in *International conference on machine learning*. PMLR, 2016, pp. 2071–2080.
[42] S. M. Kazemi and D. Poole, “Simple embedding for link prediction in knowledge graphs,” *Advances in neural information processing systems*, vol. 31, 2018.
[43] I. Balažević, C. Allen, and T. M. Hospedales, “Tucker: Tensor factorization for knowledge graph completion,” *arXiv preprint arXiv:1901.09590*, 2019.
[44] M. Chen, W. Zhang, Z. Yao, Y. Zhu, Y. Gao, J. Z. Pan, and H. Chen, “Entity-agnostic representation learning for parameter-efficient knowledge graph embedding,” *arXiv preprint arXiv:2302.01849*, 2023.
[45] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. v. d. Berg, I. Titov, and M. Welling, “Modeling relational data with graph convolutional networks,” in *European semantic web conference*. Springer, 2018, pp. 593–607.
[46] D. Nathani, J. Chauhan, C. Sharma, and M. Kaul, “Learning attention-based embeddings for relation prediction in knowledge graphs,” *arXiv preprint arXiv:1906.01195*, 2019.
[47] J. Ashish, B. Ashwin Ramesh, Z. Mohammad Zaki, B. Debapriya, and M. Fillia, “A survey on contrastive self-supervised learning,” *Technologies (Basel)*, vol. 9, no. 2, p. 2, 2021.
[48] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’06)*, vol. 2. IEEE, 2006, pp. 1735–1742.
[49] O. Henaff, “Data-efficient image recognition with contrastive predictive coding,” in *International Conference on Machine Learning*. PMLR, 2020, pp. 4182–4192.
[50] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.
[51] S. Monka, L. Halilaj, S. Schmid, and A. Rettinger, “Contrakg: Contrastive-based transfer learning for visual object recognition using knowledge graphs.” *CoRR*, 2021.
[52] R. Xie, Z. Liu, J. Jia, H. Luan, and M. Sun, “Representation learning of knowledge graphs with entity descriptions,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.
[53] M. Chen, Y. Tian, K.-W. Chang, S. Skiena, and C. Zaniolo, “Co-training embeddings of knowledge graphs and entity descriptions for cross-lingual entity alignment,” *arXiv preprint arXiv:1806.06478*, 2018.
[54] Y. Ma, H. Gao, T. Wu, and G. Qi, “Learning disjointness axioms with association rule mining and its application to inconsistency detection of linked data,” in *CSWS*. Springer, 2014, pp. 29–41.
[55] A. Melo and H. Paulheim, “Detection of relation assertion errors in knowledge graphs,” in *Proceedings of the Knowledge Capture Conference*, 2017, pp. 1–8.
[56] H. Paulheim and A. Gangemi, “Serving dbpedia with dolce–more than just adding a cherry on top,” in *ISWC*. Springer, 2015, pp. 180–196.
[57] X. Wang, X. L. Wang, and D. M. Wilkes, “A minimum spanning tree-inspired clustering-based outlier detection technique,” in *Industrial Conference on Data Mining*. Springer, 2012, pp. 209–223.
[58] J. Debattista, C. Lange, and S. Auer, “A preliminary investigation towards improving linked data quality using distance-based outlier detection,” in *Joint international semantic technology conference*. Springer, 2016, pp. 116–124.
[59] H. Paulheim and C. Bizer, “Improving the quality of linked data using statistical distributions,” *IJSWIS*, vol. 10, no. 2, pp. 63–86, 2014.
[60] J. Lehmann, D. Gerber, M. Morsey, and A.-C. N. Ngomo, “Defacto-deep fact validation,” in *International semantic web conference*. Springer, 2012, pp. 312–327.
[61] B. Shi and T. Weninger, “Discriminative predicate path mining for fact checking in knowledge graphs,” *Knowledge-based systems*, vol. 104, pp. 123–133, 2016.
[62] C. Ge, Y. Gao, H. Weng, C. Zhang, X. Miao, and B. Zheng, “Kgclean: An embedding powered knowledge graph cleaning framework,” *arXiv preprint arXiv:2004.14478*, 2020.
[63] N. Lao and W. W. Cohen, “Relational retrieval using a combination of path-constrained random walks,” *Machine learning*, vol. 81, no. 1, pp. 53–67, 2010.
[64] Z. Wang, J. Zhang, J. Feng, and Z. Chen, “Knowledge graph embedding by translating on hyperplanes,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, 2014.
[65] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu, “Learning entity and relation embeddings for knowledge graph completion,” in *Twenty-ninth AAAI conference on artificial intelligence*, 2015.
[66] Y. Zhao, H. Feng, and P. Gallinari, “Embedding learning with triple trustiness on noisy knowledge graph,” *Entropy*, vol. 21, no. 11, p. 1083, 2019.

Qinggang Zhang is a Ph.D. student at the Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR. He received the bachelor degree of Engineering in Computer Science from the Northwestern Polytechnical University, China. He is currently a group member in DEEP Lab supervised by Dr. Xiao Huang at the Hong Kong Polytechnic University. His research interests include graph neural networks, knowledge graphs, network anomaly detection.

Junnan Dong received his M.S. degree from the Hong Kong Polytechnic University, supervised by Dr. Xiao Huang. He is currently a Ph.D. student in DEEP Lab, Department of Computing at the Hong Kong Polytechnic University. His research interests mainly include graph neural networks and knowledge-enhanced reasoning. He has published papers on WSDM and WWW related to learning with knowledge graphs.

Qiaoyu Tan received the BEng degree from the College of Computer Science and Technology, Southwest University, China, in 2017. He is currently working toward the PhD degree in the Department of Computer Science and Engineering, Texas A&M University, College station, Texas. His research interests include graph neural networks, knowledge graphs, network anomaly detection and recommendation systems. He has published more than 20 peer-reviewed research papers and serves as a reviewer and PC member for several premier conferences and journals.

Xiao Huang is an assistant professor in the Department of Computing at The Hong Kong Polytechnic University. He received Ph.D. in Computer Engineering from Texas A&M University in 2020, M.S. in Electrical Engineering from Illinois Institute of Technology in 2015, and B.S. in Engineering from Shanghai Jiao Tong University in 2012. He is a program committee member of AAAI 2021-2023, ICLR 2022-2023, NeurIPS 2021-2022, KDD 2019-2022, TheWebConf 2022-2023, ICML 2021-2022, IJCAI 2020-2022, CIKM 2019-2022, WSDM 2021-2023, SDM 2022, ICKG 2020-2021.