

Article

CosUKG: A Representation Learning Framework for Uncertain Knowledge Graphs

Qiuhui Shen ^{1,*}  and Aiyun Qu ²¹ School of Computer Science and Technology, Zhoukou Normal University, Zhoukou 466001, China² College of Network Security, Jinling Institute of Technology, Nanjing 211199, China; quaiyan@jit.edu.cn

* Correspondence: maple_dancing@zknk.edu.cn

Abstract: Knowledge graphs have been extensively studied and applied, but most of these studies assume that the relationship facts in the knowledge graph are correct and deterministic. However, in the objective world, there inevitably exist uncertain relationship facts. The existing research lacks effective representation of such uncertain information. In this regard, we propose a novel representation learning framework called CosUKG, which is specifically designed for uncertain knowledge graphs. This framework models uncertain information by measuring the cosine similarity between transformed vectors and actual target vectors, effectively integrating uncertainty into the embedding process of the knowledge graph while preserving its structural information. Through multiple experiments on three public datasets, the superiority of the CosUKG framework in representing uncertain knowledge graphs is demonstrated. It achieves improved representation accuracy of uncertain information without increasing model complexity or weakening structural information.

Keywords: knowledge graph; uncertain knowledge graph; knowledge representation learning; uncertainty information; confidence

MSC: 68T07; 68T30; 68T50



Citation: Shen, Q.; Qu, A. CosUKG: A Representation Learning Framework for Uncertain Knowledge Graphs. *Mathematics* **2024**, *12*, 1419. <https://doi.org/10.3390/math12101419>

Academic Editor: Francesco Calimeri

Received: 11 March 2024

Revised: 22 April 2024

Accepted: 29 April 2024

Published: 7 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

A knowledge graph is a structured representation of facts in the form of a directed graph with multiple relationships. Nodes represent entities, and edges represent relationships between entities. Currently, knowledge graphs have been extensively studied and applied. However, these studies and applications are based on the assumption that the facts in the knowledge graph are correct, which does not align with the objective world. In the process of building a knowledge graph, it is inevitable to include some erroneous or incomplete information from the objective world. Therefore, in order to better express knowledge of the objective world, knowledge graphs should consider describing these uncertain facts with errors. To differentiate, a knowledge graph that only considers certain facts is referred to as a deterministic knowledge graph, while a knowledge graph that includes uncertain information is referred to as an uncertain knowledge graph.

Knowledge graph embedding refers to encoding the entities in a knowledge graph into spatial vectors using a certain encoding method, aiming to capture both semantic and structural information. Researchers have conducted extensive studies on this topic and proposed various embedding models, such as distance-based models like TransE [1], TransH [2], TransR [3]; semantic matching models like DistMult [4], SME [5], HolE [6]; complex geometric models like TorusE [7], QuatE [8], Poincaré [9]; and neural network models like ConvKB [10], ConvE [11], SCAN [12], etc. [13]. However, most of these models are designed for traditional deterministic knowledge graphs, and there is relatively limited research on embedding uncertain information. Currently, research in this area is still in the developmental stage. Therefore, after realizing the importance of uncertain knowledge, it

is crucial to study the embedding of uncertain knowledge graphs for various applications and better representation of the objective world within knowledge graphs.

Currently, existing models for embedding uncertain knowledge graphs mostly transform the scoring functions of deterministic knowledge graph embedding models into uncertain information in the form of function transformations. This approach does not truly incorporate uncertainty into the embedding process and weakens the embedding of structural information within the knowledge graph. Additionally, these models often have high computational complexity.

In light of this, this paper proposes a novel framework for uncertain knowledge graph embedding called CosUKG. This framework not only effectively incorporates uncertainty into the representation learning process of knowledge graphs but also captures the structural information of the knowledge graph. Furthermore, a training strategy involving negative samples is designed to further enhance the embedding accuracy.

The specific contributions of this paper are as follows:

- A novel framework for uncertain knowledge graph embedding, CosUKG, is proposed.
- A new method for generating negative samples specifically for uncertain knowledge graphs is introduced.
- A training strategy involving negative samples designed for uncertain knowledge graphs is presented.
- A confidence inference method for unobserved relationship facts is proposed.
- Multiple experiments on three general datasets demonstrate the superior performance of CosUKG in uncertain knowledge graph representation learning.

The remaining sections of this paper are organized as follows. In Section 2, we review relevant literature on deterministic and uncertain knowledge graphs. Section 3 provides a formal definition of uncertain knowledge graphs, uncertain knowledge graph representation learning models, and link prediction tasks. Section 4 elaborates on the proposed CosUKG framework. In Section 5, we present the experimental results. Finally, we conclude our research with a summary and discussion in the concluding section.

2. Related Work

In recent years, a considerable amount of work has been devoted to studying models for knowledge graph representation learning. We have collected existing models and conducted in-depth research and analysis. A comprehensive summary of knowledge graph representation learning models from both deterministic and uncertain perspectives is presented in Figure 1. Our proposed CosUKG framework leverages the research findings of some existing deterministic knowledge graph representation learning models while addressing the challenges present in uncertain knowledge graph representation. Therefore, we first provide an overview of deterministic knowledge graph models, followed by a summary of uncertain knowledge graph representation learning models and an analysis of the issues they face.

2.1. Deterministic Knowledge Graph Representation Learning Models

The scoring function is a crucial component in deterministic knowledge graph representation learning models. Based on the scoring function, these models can be categorized into five types, distance models, semantic matching models, geometric models, neural network models, and auxiliary information models [14], as shown in Figure 1.

In distance models, a distance-based scoring function is utilized. These functions consider the distance between the head and tail entities and transform the relationship into a translation vector or projection matrix. The transformed distance between the head and tail entities is expected to be small. Examples of distance models include TransE, TransH, TransR, TransD [15], KG2E [16], TransA [17], TransF [18], TransM [19], etc.

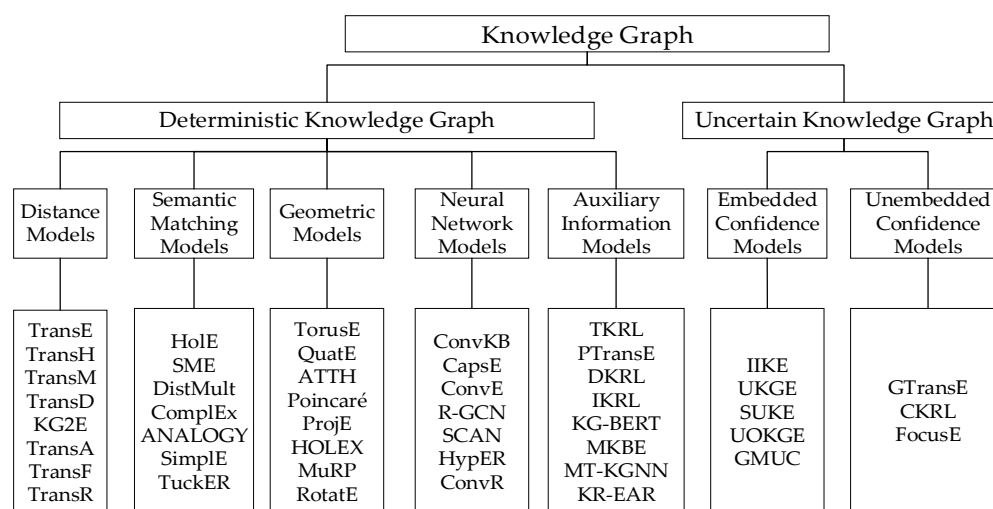


Figure 1. Classification of knowledge graph representation learning models.

Geometric models refer to representing triples in non-Euclidean spaces such as complex space, Gaussian space, toroidal space, hyperbolic space, etc. The relationship is treated as a complex geometric transformation in the semantic space, and the scoring function measures the distance between the transformed head entity and tail entity. Examples of geometric models include TorusE, QuatE, Poincaré, ATTH [20], ProjE [21], HOLEX [22], MuRP [23], RotatE [24], etc.

Semantic matching models employ similarity-based scoring functions to measure the plausibility of triples by matching the latent semantics of entities and projecting them into a vector space. Examples of semantic matching models include HoIE, SME, DistMult, ComplEx [25], ANALOGY [26], Simple [27], TuckER [28], RESCAL [29], etc.

Neural network models refer to treating triples as inputs and utilizing the powerful feature capturing ability of neural networks to represent knowledge graphs in a feature space and learn feature representations of entities and relations. Examples of this type of model include ConvKB, ConvE, SCAN, CapsE [30], R-GCN [31], Hyper [32], ConvR [33], etc.

Auxiliary information models refer to incorporating additional auxiliary information such as text, images, entity types, or relation paths during knowledge graph embedding. Examples of auxiliary information models include TKRL [34], PTransE [35], DKRL [36], IKRL [37], KG-BERT [38], MKBE [39], MT-KGNN [40], KR-EAR [41], etc.

2.2. Uncertain Knowledge Graph Representation Learning Models

Current research on the representation of uncertain knowledge in knowledge graphs can be categorized into two types: those that explicitly incorporate uncertainty into their embeddings and those that do not.

The first type uses uncertainty information but does not preserve it in the embedding space. For example, CKRL [42] multiplies the confidence level used to express uncertainty information directly by the corresponding triple's scoring function. They then optimize using a margin-based ranking loss function such that triples with higher confidence levels receive greater optimization weights. Similarly, the GTransE [43] model utilizes the TransE scoring function but adjusts the margin between positive and negative triples dynamically based on their confidence levels. This allows triples with higher confidence levels to have a larger margin, while those with lower confidence levels have a smaller margin, making the model more focused on learning from high-confidence triples. The FocusE [44] model uses the same scoring function as GTransE but directly adds larger weights to triples with higher confidence levels during optimization, thus increasing their impact.

Although these three models increase the weight of high-confidence triples during optimization in different ways, the numerical value of confidence is not preserved in

the embedding space. This can lead to poor results for certain tasks related to uncertain knowledge graphs, such as confidence prediction and link prediction.

The second type incorporates uncertainty information and preserves it in the embedding space. For example, IIKE [45] assumes that each node in the knowledge graph is independent and models confidence using the TransE scoring function. The confidence of each triple is defined as the joint probability, and the prediction task is defined as the conditional probability. Optimization is performed using a mean squared error loss function. Although uncertainty information is preserved in the embedding space, the model does not consider the correlation information in the knowledge graph. Both UKGE [46] and SUKE [47] use the scoring function of the Distmult model and map the scoring function to confidence through a function. They both optimize using a mean squared error loss function. The difference is that UKGE only focuses on confidence learning and does not fully utilize structural information, while SUKE considers both confidence and structural information. However, SUKE needs to learn two vector representations simultaneously for each entity and relation, making the model complex and training difficult. BEUrRE [48] models entities as axis-aligned hyperrectangles and relations as affine transformations between the head and tail entity rectangles. The overlapped region after transformation represents uncertainty information. It models the relationship information at two levels and can represent hierarchical structure well, but the learned vectors are complex and computationally expensive.

To address the issues with uncertain knowledge graph representation learning mentioned above, this paper proposes an embedding model framework that can incorporate uncertainty information while fully capturing structural and semantic information. Multiple experiments were conducted on three datasets, and the proposed model showed excellent performance in both confidence prediction and relation fact classification tasks.

3. Problem Definition

In this section, we provide definitions for uncertain knowledge graphs as well as uncertain knowledge graph representation learning and prediction.

Definition 1. *Uncertain Knowledge Graph.* An Uncertain Knowledge Graph is commonly defined as $UKG = \{E, R, Q, C\}$, where E represents the set of entities, R represents the set of relations, C represents the set of uncertainty information, and $Q \subseteq E \times R \times E \times C$ represents the set of quadruples. The quadruple $(h, r, t, c) \in Q$ represents all existing quadruples with uncertainty information in the UKG. h is the head entity, t is the tail entity, and both h and t belong to the entity set E . $r \in R$ represents the directed relation between h and t . $c \in C$ represents uncertainty information, which is the probability or confidence that the triple (h, r, t) holds true. It is also known as the confidence, and its value ranges from 0 to 1. The larger the value of c , the higher the likelihood that (h, r, t) is true. However, a low value of c does not imply that (h, r, t) is false; it simply indicates a lower probability of its truthfulness.

Definition 2. *Representation learning for Uncertain Knowledge Graphs.* Uncertain Knowledge Graph representation learning refers to the process of encoding relationships and entities in a given UKG while preserving structural, semantic, and confidence information. Uncertain Knowledge Graph representation learning is also known as Uncertain Knowledge Graph embedding, where h, r, t , respectively, represent the embedding vectors for h, r, t .

Definition 3. *Uncertain Knowledge Graph prediction.* Uncertain Knowledge Graph prediction is the task of predicting the probability of a relationship fact (h, r, t) being true, i.e., $(h, r, t, ?)$. It involves predicting the confidence c for a given triple (h, r, t) .

4. The CosUKG Framework

To genuinely incorporate confidence into the encoding of entities and relations, while fully capturing both structural and semantic information, the CosUKG framework is proposed. Firstly, an overview of the CosUKG framework is provided, followed by the motivation behind its proposal in Section 4.2. Subsequently, in Section 4.3, the principles of embedding confidence in CosUKG and reflecting structural and semantic information are elucidated. Section 4.4 describes the training process of the framework, concluding with the method for inferring confidence of unobserved triples.

4.1. Overview of the CosUKG Framework

The CosUKG framework consists of three modules, initialization, training, and evaluation, as shown in Figure 2. The initialization module is responsible for reading positive samples, generating negative samples, selecting models, and generating embedding vectors. Model selection aims to choose an appropriate model, which includes pre-defined scoring functions suitable for UKG. The embedding vector generation is responsible for initializing the embedding vectors and updating and storing intermediate vectors.

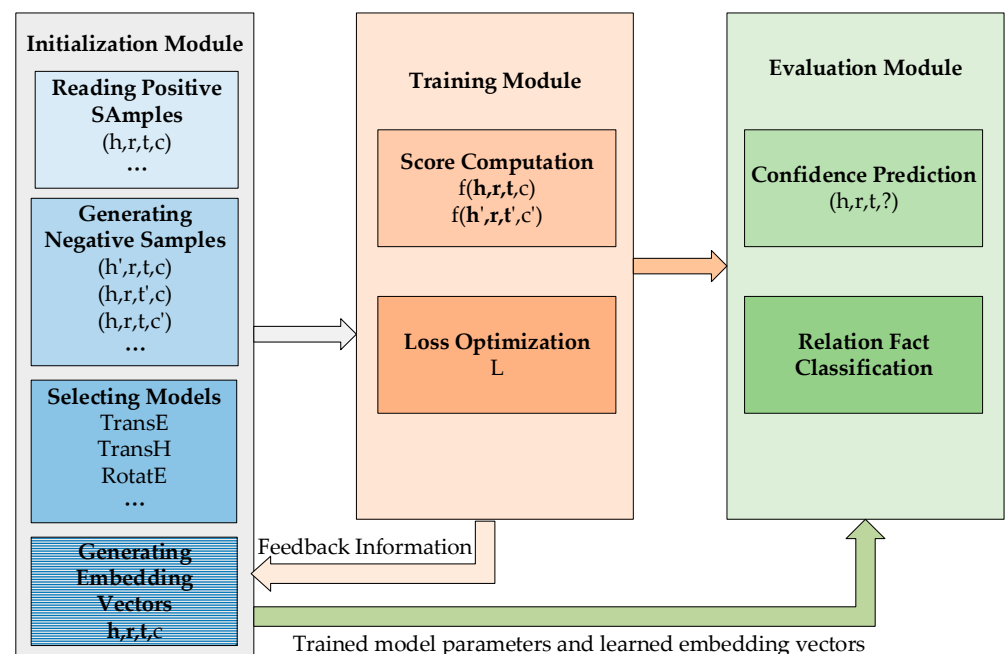


Figure 2. The overall structure of the CosUKG framework.

The training module consists of two functionalities: score computation and loss optimization. Score computation involves using the model's scoring function to calculate the scores of positive and negative samples. Then, through loss optimization, an appropriate loss function is selected to optimize the embedding vectors. The results are passed to the embedding vector generation module, and this process iterates until reaching the maximum number of iterations. The choice of loss function should correspond to the selected model.

Finally, the evaluation module assesses the trained model parameters and learned embedding vectors. It analyzes the embedding performance of CosUKG through two tasks: confidence prediction and relation fact classification.

4.2. Motivation behind the CosUKG

From the analysis of Section 2, it can be seen that a large amount of research work proposes deterministic knowledge graph representation learning models, all aiming to bring the head entity as close as possible to the tail entity through transformation. In deterministic knowledge graph representation learning, each relational fact is deterministic,

and, during representation learning, for each deterministic relational fact, the smaller the value of its scoring function, the better, ideally being 0.

As shown in Figure 3, taking TransE as an example, it is desirable that $\mathbf{h} + \mathbf{r} - \mathbf{t} = \mathbf{0}$, that is, $\mathbf{h} + \mathbf{r} = \mathbf{t}$, which is the ideal result. However, in reality, it is often difficult to achieve; that is, there is usually a gap between the transformed vector $\hat{\mathbf{t}} = \mathbf{h} + \mathbf{r}$ and the true \mathbf{t} . Assuming an offset angle θ represents this distance, the smaller the offset angle, the closer $\hat{\mathbf{t}}$ is to the target vector \mathbf{t} . When $\theta = 0^\circ$, $\hat{\mathbf{t}} = \mathbf{t}$, which means $\cos \theta = (\hat{\mathbf{t}} \cdot \mathbf{t}) / (|\hat{\mathbf{t}}| * |\mathbf{t}|) = 1$. In other words, for a deterministic relational fact (h, r, t) , the cosine similarity between $\hat{\mathbf{t}}$ and \mathbf{t} is 1. If the distance between $\hat{\mathbf{t}}$ and \mathbf{t} increases, the offset angle θ will also increase, and the cosine similarity will decrease. Furthermore, according to Definition 1, uncertain knowledge graphs assign a confidence to each relational fact (h, r, t) to represent uncertainty, with values ranging from 0 to 1. A higher confidence indicates a higher likelihood that the relational fact (h, r, t) is true, which is consistent with the change in cosine similarity values. Therefore, by modeling confidence using cosine similarity and incorporating them into the scoring function formula, uncertainty can be represented in the embedding vector space.

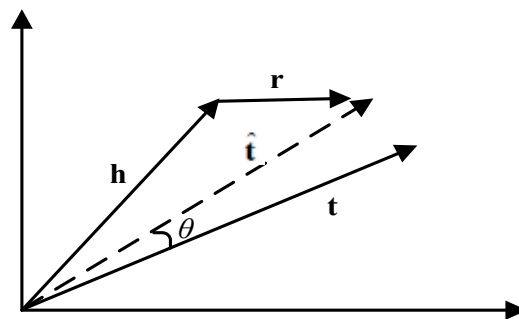


Figure 3. Simple illustration of offset angle θ .

4.3. Embedding Confidence, Structural, and Semantic Information

It is natural to model confidence information using the offset angle θ between vectors. When the confidence is 1, $\hat{\mathbf{t}} = \mathbf{t}$, meaning the offset angle from the transformed vector $\hat{\mathbf{t}}$ to the target vector \mathbf{t} is 0, which results in $\cos 0^\circ = 1$. When the confidence is 0, $\hat{\mathbf{t}}$ and \mathbf{t} are significantly different, indicating an offset angle of greater than or equal to 90 degrees from $\hat{\mathbf{t}}$ to \mathbf{t} . Therefore, the confidence $c = \cos \theta = (\hat{\mathbf{t}} \cdot \mathbf{t}) / (|\hat{\mathbf{t}}| * |\mathbf{t}|)$, which is based on the well-known cosine of the angle equation and is the reason why the proposed model framework is named CosUKG. The general scoring function is designed as Equation (1):

$$f = (\hat{\mathbf{t}} \cdot \mathbf{t}) - c(|\hat{\mathbf{t}}| * |\mathbf{t}|) \quad (1)$$

The embedding of structural and semantic information is primarily reflected in the transformation process from the head entity to the tail entity, which determines how $\hat{\mathbf{t}}$ is derived. There are multiple choices for this process, such as the simple TransE model where the transformed vector $\hat{\mathbf{t}} = \mathbf{h} + \mathbf{r}$, and the scoring function is defined as:

$$f = ((\mathbf{h} + \mathbf{r}) \cdot \mathbf{t}) - c(|\mathbf{h} + \mathbf{r}| * |\mathbf{t}|) \quad (2)$$

The TransE model performs well only in handling one-to-one relationships. It tends to make errors when dealing with one-to-many, many-to-one, and many-to-many relationships. This limitation also applies to CosUKG, restricting it to handling only one-to-one uncertain relational facts. To improve upon this, one may choose the TransH, TransR, or TransD models, which all utilize the relation r to project or transform h, t . In summary, $\varphi_1^r(\hat{\mathbf{t}}) = \varphi_2^r(\mathbf{h}) + \mathbf{r}$, where $\varphi_i^r(\cdot)$ represents matrix multiplication with respect to r . The scoring function can be represented by Equation (3):

$$f = ((\varphi_2^r(\mathbf{h}) + \mathbf{r}) \cdot \varphi_1^r(\mathbf{t})) - c(|\varphi_2^r(\mathbf{h}) + \mathbf{r}| * |\varphi_1^r(\mathbf{t})|) \quad (3)$$

The TransX series models excel in modeling one-to-many, many-to-one, and many-to-many relationships. If we select Equation (3) as the scoring function, CosUKG can handle uncertain relational facts of types including one-to-many, many-to-one, and many-to-many.

Compared to the TransX series models, the RotatE model can better embed structural and semantic information, where $\hat{\mathbf{t}} = \mathbf{h} \odot \mathbf{r}$, with \odot representing Hadamard (or element-wise) product. The scoring function is defined by Equation (4):

$$f = ((\mathbf{h} \odot \mathbf{r}) \cdot \mathbf{t}) - c(|\mathbf{h} \odot \mathbf{r}| * |\mathbf{t}|) \quad (4)$$

The RotatE model accurately handles relationships of one-to-one, one-to-many, many-to-one, and many-to-many types, and this capability extends to CosUKG. Therefore, when using Equation (4), CosUKG can model uncertain relational facts of types including one-to-one, one-to-many, many-to-one, and many-to-many.

Similarly, this framework can fully leverage existing research achievements in deterministic knowledge graphs, particularly distance models. This ensures the quality of the embeddings with minimal increase in model complexity.

4.4. Framework Training

The choice of loss function and negative sample generation method are two crucial aspects in model training. The selection of the loss function depends on the generation method. For example, TransE utilizes a margin-based ranking loss function, represented by Equation (5) in the CosUKG framework. On the other hand, RotatE employs a negative sampling loss function, represented by Equation (6) in the CosUKG framework. In the equations, $p(\mathbf{h}'_i, \mathbf{r}, \mathbf{t}'_i, c'_i)$ denotes the sampling probability of negative samples, which is used as the weight for negative samples. Q^- represents the set of negative samples, which includes quadruples that do not exist in Q and is defined as $Q^- = \{(h', r, t, c) | h' \in E \wedge h' \neq h \wedge (h, r, t, c) \in Q\} \cup \{(h, r, t', c) | t' \in E \wedge t' \neq t \wedge (h, r, t, c) \in Q\} \cup \{(h, r, t, c') | c' \in C \wedge c' \neq c \wedge (h, r, t, c) \in Q\}$.

$$L = \sum_{(h,r,t,c) \in Q} \sum_{(h',r,t',c') \in Q^-} [\gamma - f(\mathbf{h}, \mathbf{r}, \mathbf{t}, c) + f(\mathbf{h}', \mathbf{r}, \mathbf{t}', c')]_+ \quad (5)$$

$$L = -\log \sigma(\gamma - f(\mathbf{h}, \mathbf{r}, \mathbf{t}, c)) - \sum_{i=1}^n p(\mathbf{h}'_i, \mathbf{r}, \mathbf{t}'_i, c'_i) \log \sigma(f(\mathbf{h}'_i, \mathbf{r}, \mathbf{t}'_i, c'_i) - \gamma) \quad (6)$$

A simple uniform distribution of negative samples with zero confidences cannot be used during the training process. Therefore, a new negative sample generation strategy has been designed for CosUKG. Specifically, for a quadruple (h, r, t, c) , if $c > x_1$, n negative samples (h, r, t, c') are generated, where the value range of c' is $(0, 1 - c]$; if $c < x_2$, n negative samples (h, r, t, c') are generated, where the value range of c' is $[1 - c, 1)$; and, if $c < x_1$, n negative samples (h', r, t', c) are generated. x_1 and x_2 are hyperparameters, where $x_1 > x_2$, and they need to be adjusted during the training process. The negative sample generation process is described in Algorithm 1. Compared to using a uniform distribution, this negative sampling strategy is more nuanced, capable of generating a greater number of genuine negative samples. By enhancing the contribution of negative sample scores to the loss function, it aids the model in learning more about the structure and semantic features of the knowledge graph.

The computational complexity of Algorithm 1 mainly lies in detecting whether the generated negative samples are true negative samples, i.e., filtering out the samples that already exist in Q . Assuming that the time required for one detection operation is T_{filter} , the time complexity of generating n negative samples is $O(nT_{\text{filter}})$.

Algorithm 1 The Process of Generating Negative Samples

Input: a quadruple (h, r, t, c) , negative sampling size n , threshold parameter x_1 and x_2 , where $x_1 > x_2$.

- 1: **if** $c > x_1$:
- 2: Generate n quadruples (h, r, t, c') , where $c' \in \{c'_i | 0 < c'_i < (1 - c), i = 1, 2, 3, \dots, n\}$, $(h, r, t, c') \in Q^-$
- 3: **else if** $c < x_2$:
- 4: Generate n quadruples (h, r, t, c') , where $c' \in \{c'_i | (1 - c) < c'_i < 1, i = 1, 2, 3, \dots, n\}$, $(h, r, t, c') \in Q^-$
- 5: **else:**
- 6: Generate n quadruples (h', r, t, c) or (h, r, t', c) , $(h', r, t, c) \in Q^-, (h, r, t', c) \in Q^-$
- 7: **End if**

Output: n quadruples (h', r, t', c')

The model training process is illustrated in Algorithm 2. During each iteration, B quadruples are sampled from the training set. n negative samples are generated for each quadruple, and the scores of both positive and negative samples are calculated. Then, the loss is computed using either Equation (5) or Equation (6), and the embedding vectors of entities and relations are updated based on the gradients. Assuming that the time required to compute the scores of positive and negative samples is T_f , and the time to compute the loss value is T_{loss} , the time consumption for processing one batch of samples is $B(nT_{filter} + T_f + T_{loss})$. The time complexity for S iterations is $O(SB(nT_{filter} + T_f + T_{loss}))$.

Algorithm 2 The Training Process of CosUKG

Input: training set $Q_{test} = \{(h, r, t, c)\}$, embedding dimension d , scoring function f , batch size B , negative sampling size n , margin γ , iterations **step**, etc.

- 1: Initialize entity embedding vectors and relation embedding vectors.
- 2: Sample B quadruples from the training set, $S_B \in Q_{test}$
- 3: **for** each quadruple $(h, r, t, c) \in S_B$ **do**
- 4: Generate n true negative samples (h', r, t', c') using Algorithm 1.
- 5: Calculate $f(h, r, t, c)$ and $f(h', r, t', c')$
- 6: Update embedding parameters based on gradients using Equation (5) or Equation (6).
- 7: **End for**
- 8: Repeat steps 2 to 7 until the maximum number of iterations **step** is reached.

Output: Learned relation and entity embedding vectors.

4.5. Inference of Confidence for Unobserved Relation Facts

In the UKGE, SUKE, and BEUrRE models, probability soft logic is introduced to calculate the confidence of unobserved facts. Although these models have achieved good results, the logical rules and weights are manually determined without considering the uncertainty of these rules and weights. This approach is somewhat inconsistent with objective facts and has certain limitations. In this paper, we propose to model uncertainty information using the cosine similarity of vector angles and leverage the transitivity of angles to make reasonable inferences about the confidence of unobserved facts.

For example, consider a logical rule defined as $(e_0, r_0, e_1, c_1) \wedge (e_1, r_1, e_2, c_2) \wedge (e_2, r_2, e_3, c_3) \rightarrow (e_0, r_3, e_3, c_0)$ with a length of 3, where $e_0, e_1, e_2, e_3 \in E$, $r_0, r_1, r_2, r_3 \in R$, and $c_1, c_2, c_3 \in C$ represent entities and relations, respectively. Taking the TransE model as an example, in the CosUKG framework, the geometric interpretation of this rule is shown in Figure 4.

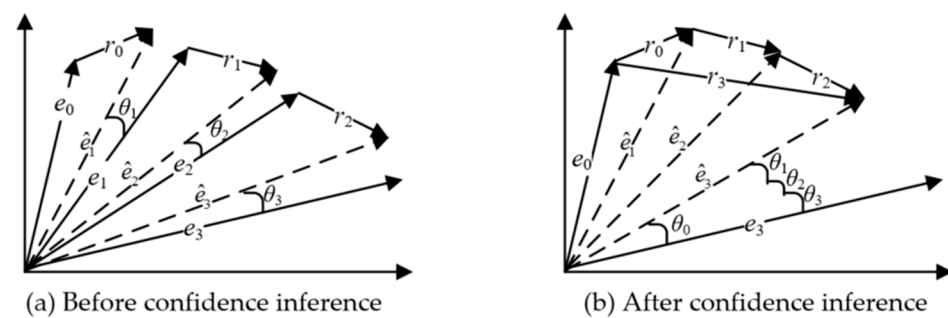


Figure 4. Example of confidence inference.

In this case, we have $c_1 = \cos \theta_1$, $c_2 = \cos \theta_2$, $c_3 = \cos \theta_3$. According to Equation (2), we can calculate $\hat{e}_1 = e_0 + r_0$, $\hat{e}_2 = e_1 + r_1$, and $\hat{e}_3 = e_2 + r_2$, as shown in Figure 4a. Then, based on the transitivity of angles and vector arithmetic rules, we obtain Figure 4b, which represents the confidence of the unobserved fact (e_0, r_3, e_3) as $c_0 = \cos \theta_0 = \cos(\theta_1 + \theta_2 + \theta_3)$.

5. Experimental and Results Analysis

To evaluate the effectiveness of the proposed CosUKG framework, we conducted confidence prediction and relation fact classification tasks on three public datasets. In addition, we also performed ablation studies and parameter sensitivity analysis.

5.1. Datasets

The three public datasets used in the experiments are CN15k, NL27k, and PPI5k [46], which are the most widely used datasets in uncertain knowledge graph research. CN15k is a subset of ConceptNet, containing 15,000 entities, 36 relations, and 241,158 facts with confidence information. NL27k [49] is a subset of NELL, with 27,221 entities, 404 relations, and 175,412 facts with confidence information. It is larger and more complex than CN15k. PPI5k is a subset of STRING [50], containing 5000 entities, seven relations, and 271,666 facts with confidence information. The specific statistics are shown in Table 1.

Table 1. Statistics of the datasets.

Dataset	#Entity	#Relation	#Rel.facts	Avg(s)	Std(s)	#Training	#Validation	#Test	#Density
CN15k	15,000	36	241,158	0.629	0.232	204,984	16,881	19,293	16.1
NL27k	27,221	404	175,412	0.797	0.242	149,100	12,278	14,034	6.4
PPI5k	5000	7	271,666	0.415	0.213	230,929	19,017	21,720	54.3

"#Entity" represents the number of entities, "#Relation" represents the number of relations, "#Rel.facts" represents the number of uncertain relation facts, "Avg(s)" and "Std(s)" are the average and standard deviation of confidence, "#Training" represents the number of relation facts in the training set, "#Validation" represents the number of relation facts in the validation set, "#Test" represents the number of relation facts in the test set, and "#Density" represents the average number of quads per entity.

Among the datasets, PPI5k has the fewest entities but the highest number of quads, while NL27k has the most entities but the fewest number of quads. Calculating the density using the average number of quads per entity in each dataset, PPI5k has the highest density at 54.3, followed by CN15k with a density of 16.1 and NL27k with the lowest density at 6.4. The datasets contain a large number of quadruples (head, relation, tail, confidence), with confidence ranging from 0 to 1. For example, (twitter, competitionswith, facebook, 0.859) is a fact in the NL27k dataset.

To ensure the fairness of the experimental results, the datasets were divided according to convention, with 85% used for training, 7% for validation, and 8% for testing.

5.2. Experimental Setup

During the embedding learning phase, the optimizer used is Adam, and a grid search approach is employed to fine-tune the hyperparameters. The dimensions of entity and relation

embedding vectors are set as follows: embedding dimension $\mathbf{d} \in \{128, 256, 500, 1000\}$, learning rate $\mathbf{lr} \in \{0.002, 0.001, 0.0001, 0.00005\}$, batch size $\mathbf{B} \in \{512, 1024\}$, negative sampling size $\mathbf{n} \in \{32, 64, 128, 256, 512\}$. The confidence threshold values \mathbf{x}_1 and \mathbf{x}_2 are set as follows: $\mathbf{x}_1 \in \{0.7, 0.75, 0.8, 0.85, 0.9\}$, $\mathbf{x}_2 \in \{0.3, 0.25, 0.2, 0.15, 0.1\}$.

In the comparative experiments, the scoring functions using Equations (2)–(4) are referred to as CosUKG_{TransE}, CosUKG_{TransH}, and CosUKG_{RotatE}, respectively. The uncertain knowledge graph models UKGE, URGE, SUKE, and BEUrRE are used for comparison, and their results are obtained from the original papers. Missing data indicate that the corresponding results are not available in the original papers.

5.3. Analysis of Experimental Results

5.3.1. Confidence Prediction

The confidence prediction task refers to predicting the confidence value c , i.e., $(h, r, t, ?)$, of an unobserved quadruple. In this paper, for each quadruple in the test set, the predicted value of confidence is $\hat{c} = (\hat{\mathbf{t}} \cdot \mathbf{t}) / (|\hat{\mathbf{t}}| * |\mathbf{t}|)$.

Evaluation metrics: The mean squared error and mean absolute error between the predicted and true values are calculated as evaluation metrics for confidence prediction. A smaller value of these metrics indicates better performance of the model.

Experimental results: To ensure fairness in comparison with baseline models, we use the same threshold values as the baselines for displaying the results. Specifically, the threshold values \mathbf{x}_1 for CN15k and NL27k are 0.85, and the threshold value \mathbf{x}_1 for PPI5k is 0.7. The prediction results of CosUKG are shown in Table 2.

Table 2. Confidence prediction ($\times 10^{-2}$).

Model	CN15k		NL27k		PPI5k	
	MSE	MAE	MSE	MAE	MSE	MAE
URGE	10.32	22.72	7.48	11.35	1.44	6.00
UKGE _{rect}	8.61	19.90	2.36	6.90	0.95	3.79
UKGE _{logi}	9.86	20.74	3.43	7.93	0.96	4.07
SUKE	5.12	17.82	0.77	3.19	0.29	1.95
BEUrRE	7.49	19.88	2.01	6.89	—	—
CosUKG _{TransE}	3.86	15.96	1.99	5.05	0.33	2.21
CosUKG _{TransH}	<u>3.83</u>	<u>14.9</u>	1.67	4.19	0.31	2.02
CosUKG _{RotatE}	3.70	14.70	<u>1.41</u>	3.08	<u>0.30</u>	<u>1.99</u>

The optimal results are indicated in **bold**, while the second-best results are annotated with an underline. A dash (—) signifies that the experiment was not conducted on this dataset, indicating missing data for that particular item.

From Table 2, it can be observed that the proposed CosUKG outperforms all baseline models on CN15k, with a significant reduction in MSE and MAE compared to the baseline models. This clearly demonstrates the notable effectiveness of CosUKG in incorporating uncertain information into embeddings. It is also worth noting that the MSE result is second-best on NL27k, while the MAE result is the best. On PPI5k, the results are consistently second-best. There are two reasons for these observations. First, it can be attributed to the characteristics of the datasets themselves. PPI5k has the highest density, followed by NL27k. On these two datasets, both MSE and MAE are smaller than CN15k, especially PPI5k, where MSE is exceptionally small. This indicates that high-density datasets are advantageous for embedding learning, but they provide limited room for improvement in evaluation metrics for researchers. Second, the scoring function of CosUKG is derived from the scoring function of a translation model, which is not precise enough when dealing with embedding learning on high-density datasets. These datasets contain a significant number of one-to-many relationship facts, making it relatively challenging for the translation model to learn such relationships accurately. Despite these two factors, CosUKG still achieves

impressive prediction results, once again highlighting its advantages in learning confidence representations.

To provide a more comprehensive illustration of the disparity between predicted confidences and actual confidences, we have plotted scatter plots using CN15k as an example, as shown in Figures 5–7. The x-axis represents the index of quadruples in the test set, while the y-axis represents the confidences. The red dots represent the actual confidences, while the black dots represent the predicted confidences.

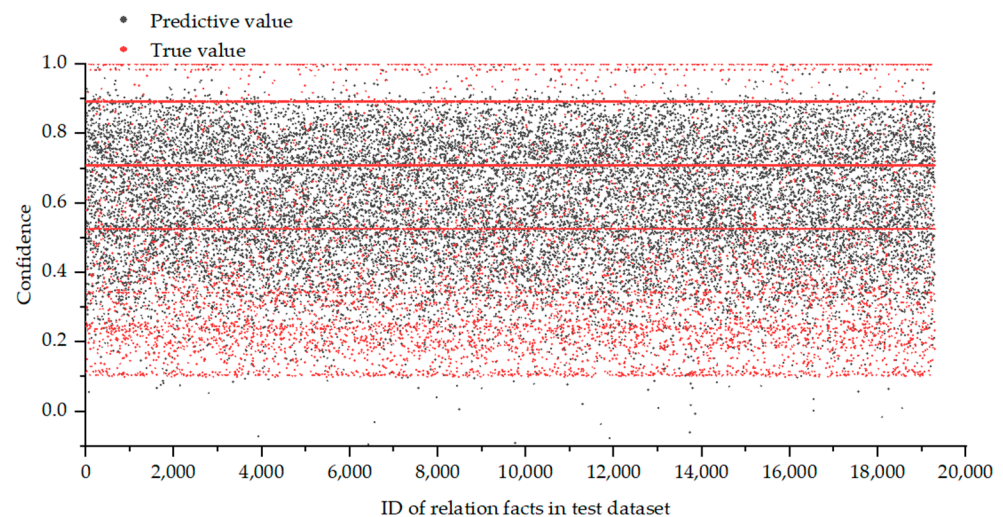


Figure 5. Confidence prediction results on CN15k.

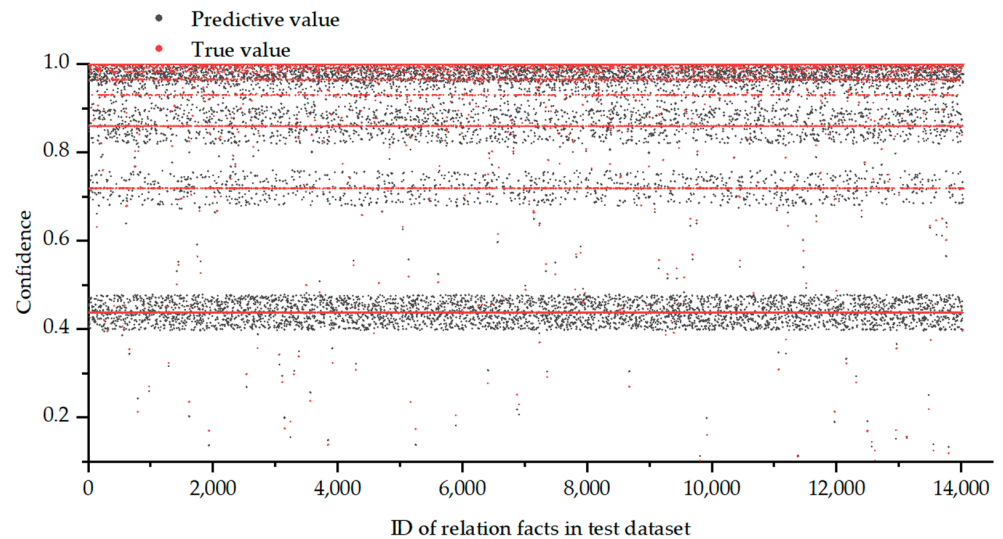


Figure 6. Confidence prediction results on NL27k.

The scatter plots reveal that on the CN15k dataset, the actual confidences are mostly distributed within the range of $[0.1, 1.0]$, while the predicted confidences are mainly concentrated in the range of $[0.2, 0.9]$. Overall, the model demonstrates good prediction performance. On NL27k, the distribution of actual confidences is relatively concentrated, and the predicted confidences are also close to the actual values, indicating a relatively better prediction performance of the model. On PPI5k, the actual confidences are mostly distributed within the range of $[0.1, 0.6]$ and around 0.9, and the distribution of the predicted confidences closely aligns with that of the actual confidences, indicating excellent predictive results by the model. This observation is consistent with the results presented in Table 2.

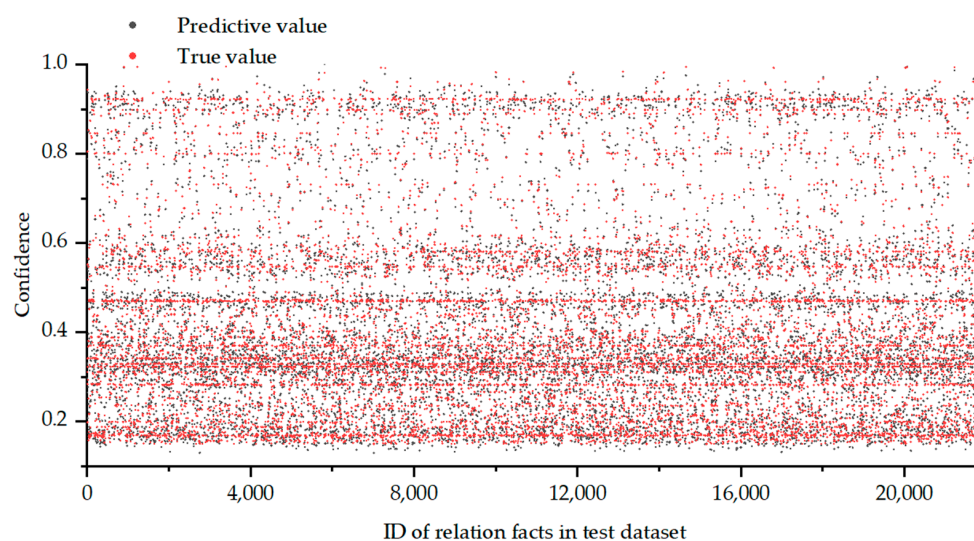


Figure 7. Confidence prediction results on PPI5k.

5.3.2. Relation Fact Classification

Relation fact classification is a binary classification task used to determine whether an unobserved relation fact is a strong relation fact, and, if so, the quadruple is considered reliable. Specifically, for each quadruple in the test set, a confidence value \hat{c} is calculated based on $\hat{c} = (\hat{\mathbf{t}} \cdot \mathbf{t}) / (|\hat{\mathbf{t}}| * |\mathbf{t}|)$, and if $\hat{c} > x_1$, it is classified as a strong relation fact; otherwise, it is considered a weak relation fact. However, errors are inevitable during the classification process. For example, for a quadruple (h, r, t, c) , and $c < x_1$, it belongs to the weak relation fact category, but if $\hat{c} > x_1$ after calculation, it is misclassified as a strong relation fact, resulting in false positives (FP), true positives (TP), false negatives (FN), and true negatives (TN). The specific judgment criteria are shown in Table 3.

Table 3. Method for relation fact classification judgment.

True Confidence	Real Classification	Predictive Confidence	Predictive Classification	Classification Results
$c > x_1$	strong relation fact	$\hat{c} > x_1$	strong relation fact	TP
$c > x_1$	strong relation fact	$\hat{c} < x_1$	weak relation fact	FN
$c < x_1$	weak relation fact	$\hat{c} > x_1$	strong relation fact	FP
$c < x_1$	weak relation fact	$\hat{c} < x_1$	weak relation fact	TN

Evaluation metrics: The evaluation metrics for assessing the quality of classification are accuracy (Accu.) and F1-score. A higher value indicates better classification performance. Accuracy is the ratio of correctly classified instances (true positives + true negatives) to the total number of examples (true positives + true negatives + false positives + false negatives). The F1-score is composed of precision (Precision = true positives / (true positives + false positives)) and recall (Recall = true positives / (true positives + false negatives)), specifically calculated as $F1\text{-score} = 2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$.

Experimental results: The results of relation fact classification are presented in Table 4. Based on Table 4, it can be observed that the CosUKG model achieved the optimal F1-score and accuracy on the CN15k dataset. The results of CosUKG_{TransE}, CosUKG_{TransH}, and CosUKG_{RotatE} were all higher than the baseline model. Particularly, CosUKG_{RotatE} showed a significant improvement of 26.67% in F1-score compared to UKG_{rect}, demonstrating the superior performance of the proposed CosUKG framework in embedding confidence information. Additionally, the model achieved the highest accuracy and the second-highest F1-score on the NL27k dataset. On the PPI5k dataset, it obtained the second-best F1-score. The F1-score and accuracy are closely related to the threshold value x_1 for strong relation facts. By adjusting the value of x_1 , better results in terms of F1-score and accuracy can be

achieved. However, for fair comparison, the results presented in Table 4 used the same threshold for strong relation facts as the UKGE model.

Table 4. Results of relation fact classification.

Model	CN15k		NL27k		PPI5k	
	F1-Score	Accu.	F1-Score	Accu.	F1-Score	Accu.
URGE	21.2	86.0	83.6	88.7	85.2	98.6
UKGE _{rect}	28.8	90.4	92.3	<u>95.2</u>	95.1	99.4
UKGE _{logi}	25.9	90.1	88.4	93.0	94.5	99.5
SUKE	26.2	81.0	95.0	93.6	97.8	99.7
CosUKG _{TransE}	34.10	90.57	88.52	92.93	93.57	98.77
CosUKG _{TransH}	<u>35.26</u>	<u>91.71</u>	91.43	94.10	95.7	99.19
CosUKG _{RotatE}	36.48	93.00	<u>93.38</u>	95.65	<u>96.30</u>	99.35

The optimal results are indicated in **bold**, while the secondary results are underscored.

The performance of CosUKG on the NL27k and PPI5k datasets aligns with the results of confidence prediction experiments. Specifically, CosUKG shows limited improvement in embedding performance on dense datasets but exhibits prominent advantages on sparse datasets.

5.3.3. Case Study

Taking the CN15k dataset as an example, confidence scores for predicting unobserved facts were estimated based on the inference rules shown in Table 5.

Table 5. Inference rules.

Inference Rules
$(A, \text{relatedto}, B) \wedge (B, \text{relatedto}, C) \rightarrow (A, \text{relatedto}, C)$
$(A, \text{hassubevent}, B) \wedge (B, \text{causes}, C) \rightarrow (A, \text{causes}, C)$
$(A, \text{notdesires}, B) \wedge (B, \text{madeof}, C) \rightarrow (A, \text{notdesires}, C)$
$(A, \text{partof}, B) \wedge (B, \text{isa}, C) \wedge (C, \text{synonym}, D) \rightarrow (A, \text{partof}, D)$

According to the inference rules, previously unobserved relation facts were discovered in the CN15k dataset. The predicted confidence closely aligned with the calculated confidence through inference, as shown in Table 6. For example, the unobserved relation fact (fish, notdesires, metal) was derived from the rule $(A, \text{notdesires}, B) \wedge (B, \text{madeof}, C) \rightarrow (A, \text{notdesires}, C)$. The confidence for $(A, \text{notdesires}, B)$ was $c_{AB} = \cos(\theta_{AB}) = 0.709$, and the confidence for (B, madeof, C) was $c_{BC} = \cos(\theta_{BC}) = 0.999$. According to the calculation method for confidence of unobserved relation facts described in Section 4.5, the theoretical value of the confidence for (fish, notdesires, metal) should be $c_{\text{fish}} = \cos(\theta_{AB} + \theta_{BC}) = \cos(\theta_{AB})\cos(\theta_{BC}) - \sin(\theta_{AB})\sin(\theta_{BC}) = c_{AB}c_{BC} - \sqrt{1 - c_{AB}^2}\sqrt{1 - c_{BC}^2} \approx 0.707$. The model's predicted confidence of 0.710 is very close to the theoretical value, differing only by 0.003. Moreover, the predicted confidence aligns with human common sense, further demonstrating the excellent performance of the proposed CosUKG framework for reasoning with unobserved knowledge.

Table 6. Partial unobserved relation facts and their confidence.

Unobserved Relation Facts	Predictive Confidence	Theoretical Confidence
(abbey, relatedto, building)	0.616	0.619
(baghdad, partof, nation)	0.746	0.741
(fish, notdesires, metal)	0.710	0.707
(going to party, causes, being happy)	0.792	0.793

5.4. Ablation Research

To demonstrate the effectiveness of the proposed negative sample generation method, ablation studies were conducted by removing the negative sample generation module from CosUKG while keeping the hyperparameters consistent. Subsequently, ten experiments were performed on each of the three datasets, and the results were averaged. In order to distinguish them, the models without the negative sample generation module for CosUKG_{TransE}, CosUKG_{TransH}, and CosUKG_{RotatE} were labeled as CosUKG^{-Neg}_{TransE}, CosUKG^{-Neg}_{TransH}, and CosUKG^{-Neg}_{RotatE}, respectively.

Table 7 displays the confidence prediction results. Compared with Table 2, it can be observed that after removing the negative sample generation module, the values of MSE and MAE have increased. Although the model still outperforms the baseline model, the degree of performance improvement is significantly reduced. This not only demonstrates the contribution of the negative sample generation module to CosUKG but also further confirms the scientific validity of using vector cosine similarity to model uncertainty information.

Table 7. Confidence prediction ($\times 10^{-2}$).

Model	CN15k		NL27k		PPI5k	
	MSE	MAE	MSE	MAE	MSE	MAE
CosUKG ^{-Neg} _{TransE}	4.59	17.53	2.14	5.61	0.36	2.36
CosUKG ^{-Neg} _{TransH}	<u>4.32</u>	<u>16.94</u>	1.84	4.59	0.32	2.13
CosUKG ^{-Neg} _{RotatE}	3.91	15.49	<u>1.43</u>	3.16	<u>0.31</u>	2.03

The optimal results are indicated in **bold**, while the second-best results are annotated with an underline.

To further visually demonstrate the impact of the negative sample generation module, a comprehensive comparison was made between the modified CosUKG without the module and the complete CosUKG. Figure 8 shows the changes in MSE and MAE before and after removing the negative sample generation module. Overall, after removing the module, both MSE and MAE increased, with the specific increase rates shown in Table 8. Particularly on the CN15k dataset, the average increases in MSE and MAE were 15.46% and 9.63%, respectively, once again highlighting the beneficial role of the negative sample generation method in enhancing the performance of CosUKG.

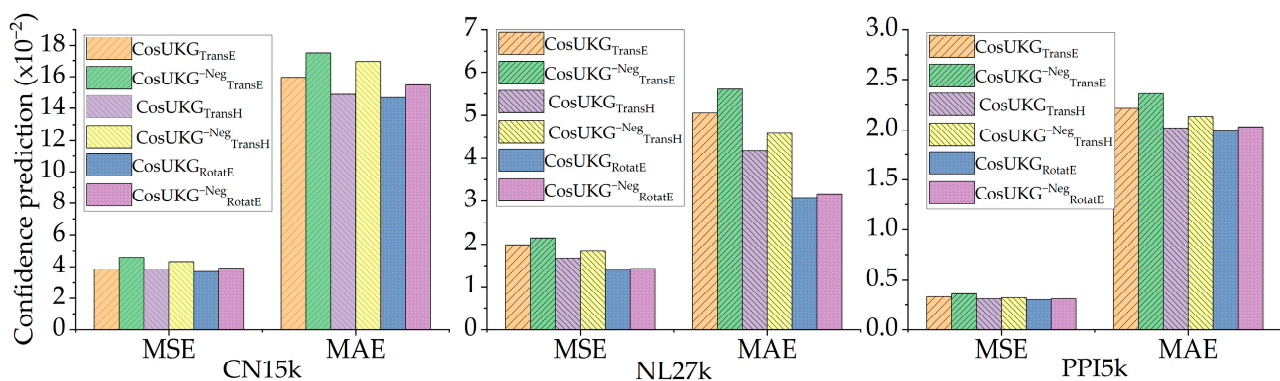


Figure 8. MSE and MAE before and after removing the negative sample generation module.

Table 8. The increase in MSE and MAE.

Model	CN15k		NL27k		PPI5k	
	MSE	MAE	MSE	MAE	MSE	MAE
CosUKG ^{-Neg} _{TransE}	18.91%	9.84%	7.54%	11.09%	9.09%	6.79%
CosUKG ^{-Neg} _{TransH}	12.79%	13.69%	10.18%	9.55%	3.23%	5.45%
CosUKG ^{-Neg} _{RotatE}	5.68%	5.37%	1.42%	2.60%	3.33%	2.01%
Average	15.46%	9.63%	6.38%	7.74%	5.22%	4.75%

5.5. Sensitivity Analysis

In the previous experiments, it was mentioned that the threshold for strong relationship facts is closely related to F1-score and accuracy. Here, by varying the value of the threshold x_1 for strong relationship facts, the changes in F1-score and accuracy on the three datasets are observed. Taking CosUKG_{RotatE} as an example, the results are shown in Figure 9.

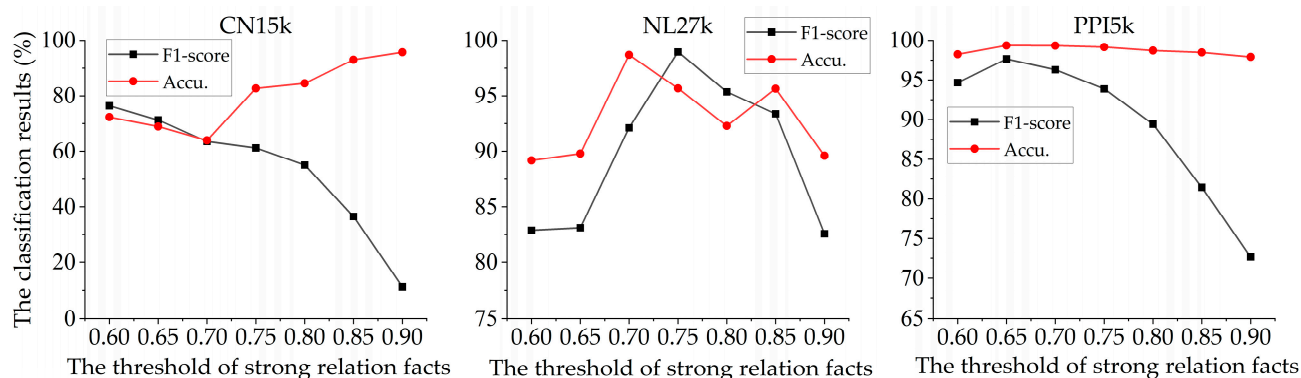


Figure 9. F1-score and accuracies for different values of the threshold for strong relationship facts.

It can be observed that on the CN15k dataset, as the threshold for strong relationship facts increases, the F1-score decreases while the accuracy increases. On the NL27k dataset, the highest accuracy is achieved when the threshold for strong relationship facts is set to 0.7, and the maximum F1-score is obtained when the threshold is set to 0.75. On the PPI5k dataset, the maximum accuracy and F1-score are achieved when the threshold is set to 0.65, and they decrease as the threshold increases.

To investigate this phenomenon of different trends on the three datasets, it was found that it is related to the distribution of confidences in the datasets. The distribution plots of confidences for the test sets of the three datasets are shown in Figures 10–12. From Figure 10, it can be seen that the confidences of the CN15k dataset are mostly distributed below 0.75. When the threshold for strong relationship facts is set to 0.8, both the accuracy and F1-score are relatively high because it can effectively classify confidences below 0.75 and those above 0.85. However, if the threshold is set to 0.85, which is closer to 0.9, not all confidences around 0.9 can be properly classified as above 0.85, resulting in a decrease in the F1-score.

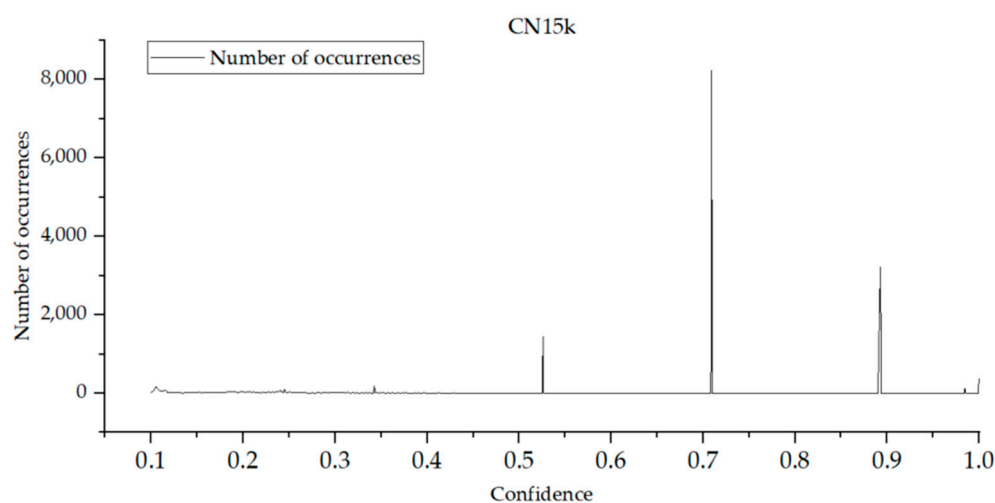


Figure 10. Distribution plot of confidences for the test set of CN15k.

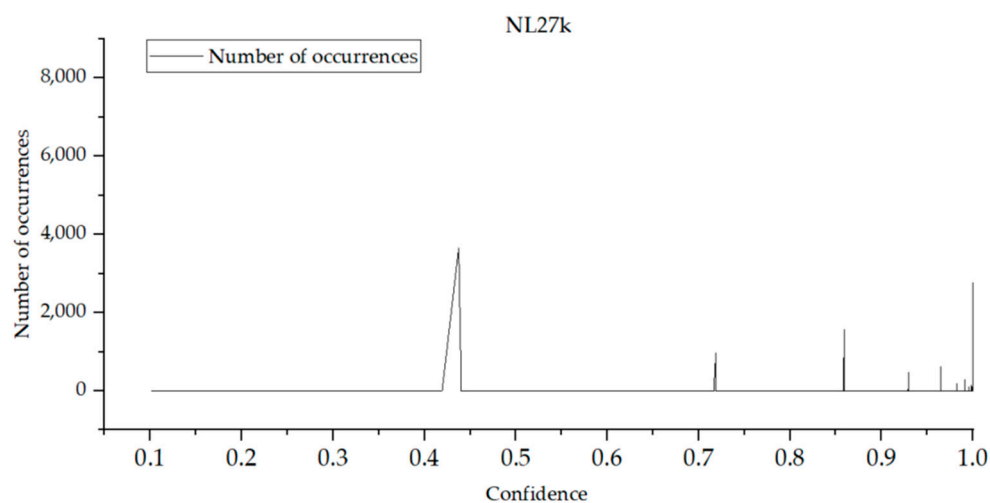


Figure 11. Distribution plot of confidences for the test set of NL27k.

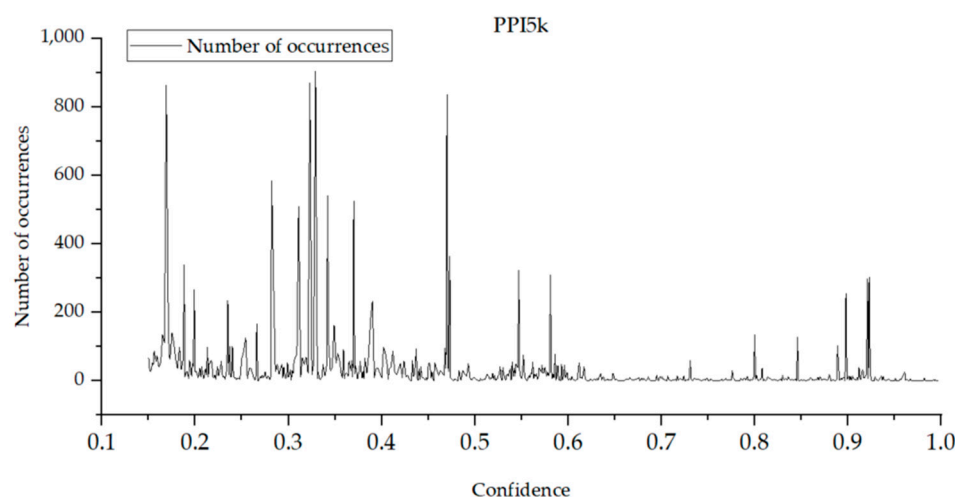


Figure 12. Distribution plot of confidences for the test set of PPI5k.

From Figure 11, it can be seen that the NL27k dataset has a significant number of confidences around 0.85. Therefore, when the threshold for strong relationship facts is set to 0.85, the obtained accuracy and F1-score are not the best, as there are many confidences around 0.85 that cannot be properly classified. However, a decent F1-score can be achieved when the threshold is set to 0.75 and 0.8.

From Figure 12, it can be observed that the PPI5k dataset has fewer confidences distributed around 0.7. Therefore, when the threshold for strong relationship facts is set to 0.7, both the accuracy and F1-score obtained are relatively good.

In conclusion, the accuracy and F1-score are sensitive to the choice of threshold x_1 for strong relationship facts. When setting the threshold, it is important to consider the distribution of confidences in the dataset itself.

Furthermore, the impact of different parameter values x_2 used in generating negative samples on confidence prediction results was also investigated. It was found that the results on the CN15k and NL27k datasets were not affected by the choice of x_2 . This is because there are very few relationship facts with confidences less than 0.3 in these two datasets, which have minimal impact on the training process. On the PPI5k dataset, different values of x_2 slightly affect the prediction results. Taking CosUKGTransE as an example, Figure 13 shows the changes in MSE and MAE with different values of x_2 . The variation is minimal because there are many relationship facts in PPI5k with confidences between 0.15 and 0.3, and the contribution of x_2 from 0.1 to 0.3 in generating negative samples is limited.

Therefore, while x_2 does have some influence on confidence prediction results, its impact is very slight according to Algorithm 1.

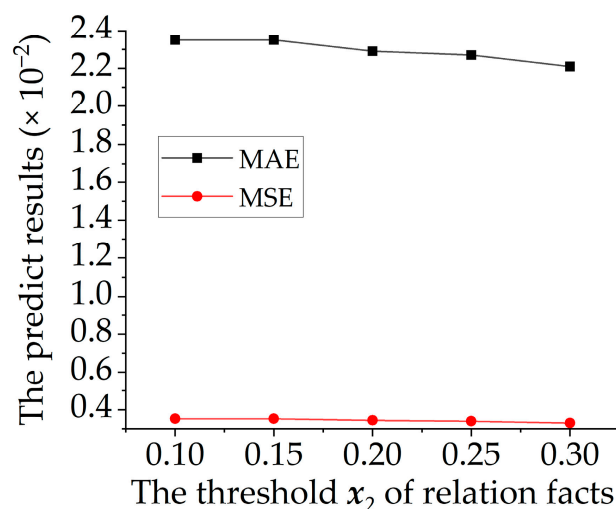


Figure 13. Changes in MSE and MAE with different values of x_2 on PPI5k.

6. Conclusions

A representation learning framework called CosUKG is proposed for embedding uncertainty information in uncertain knowledge graphs. CosUKG not only integrates confidences into the encoding of entities and relationships but also effectively captures structural information. Multiple sets of experiments were conducted on three datasets, and the results demonstrate that the proposed CosUKG exhibits outstanding performance in confidence prediction and relation fact classification. The effectiveness of the proposed negative sample generation strategy was validated through ablation experiments. Sensitivity analysis of the relation fact threshold parameter further confirms the superiority of CosUKG. The introduction of CosUKG provides a scientifically sound foundation for downstream applications of uncertain knowledge graphs. For instance, in the field of traditional Chinese medicine (TCM), where numerous vague and uncertain pieces of knowledge exist, the CosUKG framework can represent such uncertainty. This capability effectively mitigates the problem of insufficient representation of uncertain knowledge in TCM, thus providing data support for knowledge applications in this domain. Future plans include integrating additional models into the CosUKG framework, such as KG2E, TorusE, QuatE, Poincaré, ATTH, ProjE, HOLEX, and MuRP, and evaluating them on more datasets. Furthermore, there is an intention to explore embedding multidimensional information and comprehensive methods for representing knowledge graphs in greater detail.

Author Contributions: Conceptualization, Q.S.; methodology, Q.S.; software, A.Q.; validation, Q.S., A.Q.; writing—original draft preparation, Q.S.; writing—review and editing, Q.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data are unavailable due to privacy or ethical restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; Yakhnenko, O. Translating embeddings for modeling multi-relational data. *Adv. Neural Inf. Process. Syst.* **2013**, *26*, 2787–2795.
2. Wang, Z.; Zhang, J.; Feng, J.; Chen, Z. Knowledge graph embedding by translating on hyperplanes. In Proceedings of the AAAI Conference on Artificial Intelligence, Quebec City, QC, Canada, 27–31 July 2014; pp. 1112–1119.
3. Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; Zhu, X. Learning entity and relation embeddings for knowledge graph completion. In Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, Austin, TX, USA, 25–30 January 2015; pp. 2181–2187.

4. Yang, B.; Yih, W.-t.; He, X.; Gao, J.; Deng, L. Embedding entities and relations for learning and inference in knowledge bases. *arXiv* **2014**, arXiv:1412.6575.
5. Bordes, A.; Glorot, X.; Weston, J.; Bengio, Y. A semantic matching energy function for learning with multi-relational data. *Mach. Learn.* **2014**, *94*, 233–259. [[CrossRef](#)]
6. Nickel, M.; Rosasco, L.; Poggio, T. Holographic embeddings of knowledge graphs. In Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 1955–1961.
7. Ebisu, T.; Ichise, R. TorusE: Knowledge Graph Embedding on a Lie Group. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 1819–1826.
8. Zhang, S.; Tay, Y.; Yao, L.; Liu, Q. Quaternion knowledge graph embeddings. *arXiv* **2019**, arXiv:1904.10281.
9. Nickel, M.; Kiela, D. Poincaré embeddings for learning hierarchical representations. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; pp. 6341–6350.
10. Nguyen, T.D.; Nguyen, D.Q.; Phung, D. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), New Orleans, LA, USA, 1–6 June 2018; pp. 327–333.
11. Dettmers, T.; Minervini, P.; Stenetorp, P.; Riedel, S. Convolutional 2d knowledge graph embeddings. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 1811–1818.
12. Shang, C.; Tang, Y.; Huang, J.; Bi, J.; He, X.; Zhou, B. End-to-end structure-aware convolutional networks for knowledge base completion. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 3060–3067.
13. Xu, Y.; Zhang, H.; Cheng, K.; Liao, X.; Zhang, Z.; Li, L. Comprehensive Survey on Knowledge Graph Embedding. *Comput. Eng. Appl.* **2022**, *58*, 30–50.
14. Shen, Q.; Zhang, H.; Xu, Y.; Wang, H.; Cheng, K. Comprehensive Survey of Loss Functions in Knowledge Graph Embedding Models. *Comput. Sci.* **2023**, *50*, 149–158.
15. Ji, G.; He, S.; Xu, L.; Liu, K.; Zhao, J. Knowledge graph embedding via dynamic mapping matrix. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Beijing, China, 26–31 July 2015; pp. 687–696.
16. He, S.; Liu, K.; Ji, G.; Zhao, J. Learning to represent knowledge graphs with gaussian embedding. In Proceedings of the 24th ACM International on Conference on Information and Knowledge Management, Melbourne, Australia, 18–23 October 2015; pp. 623–632.
17. Xiao, H.; Huang, M.; Hao, Y.; Zhu, X. TransA: An adaptive approach for knowledge graph embedding. *arXiv* **2015**, arXiv:1509.05490.
18. Feng, J.; Huang, M.; Wang, M.; Zhou, M.; Hao, Y.; Zhu, X. Knowledge graph embedding by flexible translation. In Proceedings of the Fifteenth International Conference on Principles of Knowledge Representation and Reasoning, Cape Town, South Africa, 25–29 April 2016; pp. 557–560.
19. Fan, M.; Zhou, Q.; Chang, E.; Zheng, T.F. Transition-based knowledge graph embedding with relational mapping properties. In Proceedings of the 28th Pacific Asia Conference on Language, Information and Computing, Phuket, Thailand, 12–14 December 2014; pp. 328–337.
20. Chami, I.; Wolf, A.; Juan, D.C.; Sala, F.; Ravi, S.; Ré, C. Low-Dimensional Hyperbolic Knowledge Graph Embeddings. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 6901–6914.
21. Shi, B.; Weninger, T. ProjE: Embedding projection for knowledge graph completion. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 1236–1242.
22. Xue, Y.; Yuan, Y.; Xu, Z.; Sabharwal, A. Expanding holographic embeddings for knowledge completion. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; pp. 4496–4506.
23. Balazevic, I.; Allen, C.; Hospedales, T. Multi-relational poincaré graph embeddings. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 4463–4473.
24. Sun, Z.; Deng, Z.H.; Nie, J.Y.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In International Conference on Learning Representations. *arXiv* **2019**, arXiv:1902.10197.
25. Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; Bouchard, G. Complex embeddings for simple link prediction. In Proceedings of the 33rd International Conference on International Conference on Machine Learning—Volume 48, New York, NY, USA, 19–24 June 2016; pp. 2071–2080.
26. Liu, H.; Wu, Y.; Yang, Y. Analogical inference for multi-relational embeddings. In Proceedings of the 34th International Conference on Machine Learning—Volume 70, Sydney, NSW, Australia, 6–11 August 2017; pp. 2168–2178.
27. Kazemi, S.M.; Poole, D. Simple embedding for link prediction in knowledge graphs. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, Montréal, QC, Canada, 3–8 December 2018; pp. 4289–4300.
28. Balazevic, I.; Allen, C.; Hospedales, T. TuckER: Tensor Factorization for Knowledge Graph Completion. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 5188–5197.

29. Nickel, M.; Tresp, V.; Kriegel, H.P. A three-way model for collective learning on multi-relational data. In Proceedings of the 28th International Conference on Machine Learning, Bellevue, WA, USA, 28 June–2 July 2011; pp. 809–816.
30. Vu, T.; Nguyen, T.D.; Nguyen, D.Q.; Phung, D. A capsule network-based embedding model for knowledge graph completion and search personalization. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 2180–2189.
31. Schlichtkrull, M.; Kipf, T.N.; Bloem, P.; Van Den Berg, R.; Titov, I.; Welling, M. Modeling relational data with graph convolutional networks. In Proceedings of the European Semantic Web Conference, Heraklion, Greece, 3 June 2018; pp. 593–607.
32. Balažević, I.; Allen, C.; Hospedales, T.M. Hypernetwork knowledge graph embeddings. In Proceedings of the International Conference on Artificial Neural Networks, Munich, Germany, 17 September 2019; pp. 553–565.
33. Jiang, X.; Wang, Q.; Wang, B. Adaptive convolution for multi-relational learning. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019; pp. 978–987.
34. Xie, R.; Liu, Z.; Sun, M. Representation learning of knowledge graphs with hierarchical types. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 2965–2971.
35. Lin, Y.; Liu, Z.; Luan, H.; Sun, M.; Rao, S.; Liu, S. Modeling Relation Paths for Representation Learning of Knowledge Bases. In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Lisbon, Portugal, 17–21 September 2015; pp. 705–714.
36. Xie, R.; Liu, Z.; Jia, J.; Luan, H.; Sun, M. Representation learning of knowledge graphs with entity descriptions. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; pp. 2659–2665.
37. Xie, R.; Liu, Z.; Luan, H.; Sun, M. Image-embodied knowledge representation learning. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 3140–3146.
38. Yao, L.; Mao, C.; Luo, Y. KG-BERT: BERT for knowledge graph completion. *arXiv* **2019**, arXiv:1909.03193.
39. Pezeshkpour, P.; Chen, L.; Singh, S. Embedding Multimodal Relational Data for Knowledge Base Completion. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 3208–3218.
40. Tay, Y.; Tuan, L.A.; Phan, M.C.; Hui, S.C. Multi-task neural network for non-discrete attribute prediction in knowledge graphs. In Proceedings of the 2017 ACM Conference on Information and Knowledge Management, Singapore, 6–10 November 2017; pp. 1029–1038.
41. Lin, Y.; Liu, Z.; Sun, M. Knowledge representation learning with entities, attributes and relations. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016; pp. 2866–2872.
42. Xie, R.; Liu, Z.; Lin, F.; Lin, L. Does william shakespeare really write hamlet? Knowledge representation learning with confidence. *arXiv* **2017**, arXiv:1705.03202.
43. Kertkeidkachorn, N.; Liu, X.; Ichise, R. GTransE: Generalizing Translation-Based Model on Uncertain Knowledge Graph Embedding. *Adv. Intell. Syst. Comput.* **2019**, *1128*, 170–178.
44. Pai, S.; Costabello, L. Learning Embeddings from Knowledge Graphs with Numeric Edge Attributes. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, Online, 19–26 August 2021; pp. 2869–2875.
45. Fan, M.; Zhou, Q.; Zheng, T.F. Learning embedding representations for knowledge inference on imperfect and incomplete repositories. In Proceedings of the 2016 IEEE/WIC/ACM International Conference on Web Intelligence (WI), Omaha, NE, USA, 16 January 2016; pp. 42–48.
46. Chen, X.; Chen, M.; Shi, W.; Sun, Y.; Zaniolo, C. Embedding uncertain knowledge graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 3363–3370.
47. Wang, J.; Nie, K.; Chen, X.; Lei, J. SUKE: Embedding Model for Prediction in Uncertain Knowledge Graph. *IEEE Access* **2020**, *9*, 3871–3879. [[CrossRef](#)]
48. Chen, X.; Boratko, M.; Chen, M.; Dasgupta, S.S.; Li, X.L.; McCallum, A. Probabilistic Box Embeddings for Uncertain Knowledge Graph Reasoning. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Online, 6–11 June 2021; pp. 882–893.
49. Mitchell, T.; Cohen, W.; Hruschka, E.; Talukdar, P.; Yang, B.; Betteridge, J.; Carlson, A.; Dalvi, B.; Gardner, M.; Kisiel, B.; et al. Never-ending learning. *Commun. ACM* **2018**, *61*, 103–115. [[CrossRef](#)]
50. Szklarczyk, D.; Morris, J.H.; Cook, H.; Kuhn, M.; Wyder, S.; Simonovic, M.; Santos, A.; Doncheva, N.T.; Roth, A.; Bork, P.; et al. The STRING database in 2017: Quality-controlled protein–protein association networks, made broadly accessible. *Nucleic Acids Res.* **2017**, *45*, 362–368. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.