Enhancing Uncertain Knowledge Graph Embedding by Learning Graph Structure-Aware Context and Fused-Triple Confidence Information with an Encoder-Decoder Framework

Yujie Song[1], Yongpan Sheng[1,2(✉)], Lirong He[3], Tao Jia[1]
1: College of Computer and Information Science, Southwest University, Chongqing 400715, China
2: School of Big Data & Software Engineering, Chongqing University, Chongqing 401331, China
3: School of Information Science and Engineering,
Chongqing Jiaotong University, Chongqing 400074, China
syujo2023@163.com, shengyp2011@gmail.com, ronghe1217@gmail.com, tjia@swu.edu.cn

Abstract

Uncertain Knowledge Graphs (UKGs) have recently garnered significant attention, primarily for their capabilities in modeling the inherent uncertainty of relation facts in a knowledge graph with a confidence score and reasoning based on uncertainty knowledge embeddings. Existing uncertain knowledge graph embedding (UKGE) methods ignore the uncertainty of neighbor entities in the modeling of graph structure-aware context, on the other hand, they have inadequately accounted for the confidence information of relational triples when learning their embeddings. To address these issues, in this paper, we propose a novel uncertainty knowledge graph embedding model, named UnKGE-GSFC, which works to learn graph structure-aware context and fused-triple confidence information in an encoder-decoder framework. To be specific, for encoding, we design an uncertainty-aware RGCN providing foundations for learning graph structure-aware context. For decoding, we model confidence for each triple by fusing its semantic information and effectively preserving sequential dependencies. Extensive experiments over publicly available UKGE benchmarks demonstrate that our approach can significantly outperform previous state-of-the-art methods in different evaluation metrics on confidence prediction and link prediction tasks, respectively.

Keywords: Uncertain knowledge graph embedding, Graph structure-aware context, Fused-triple confidence information, Encoder-decoder framework

1. Introduction

Knowledge graphs (KGs) have been demonstrated to be valuable resources for a wide range of applications [1, 2, 3]. Traditional KGs typically assume that all triples within the graph possess absolute truth and certainty during modeling. However, in complex real-world applications, this assumption is often too idealized and frequently untenable [4]. On one hand, most existing

knowledge graphs (KGs) are often derived from textual corpora in real-world scenarios. However, due to the diverse original purposes of these corpora's construction, the heterogeneity of corpora resources, and the inherent errors of automatic extraction algorithms, noise and ambiguity are inevitably introduced into the resulting KGs. Furthermore, knowledge has a temporal nature, and the validity of facts may evolve over time, giving rise to the problem of outdated facts. On the other hand, facts themselves often possess inherent randomness or probabilistic attributes that cannot be adequately captured using simple Boolean logic (true/false). Instead, they require probabilistic representations to characterize their degree of credibility. Consequently, the failure to quantify and model uncertain facts within the KG would severely undermine both their credibility and overall effectiveness in downstream reasoning and decision-making tasks.

Uncertain knowledge graph (UKG) has emerged for modeling the uncertainty in KGs. Unlike traditional deterministic KGs, the UKG introduces an additional confidence score to the standard triple, thereby extending the knowledge representation form of triples in traditional knowledge graphs into a quadruple structure, where the confidence score quantifies the degree of reliability of the fact. Several typical UKG graphs have been constructed and put into use. For example, ConceptNet [5] collects triples through manual annotation and calculates their confidence based on annotation frequency; NELL [6] integrates statistical probabilities from open-domain web extraction processes with user feedback for iterative fine-tuning. UKGs hold significant application value in domains that demand high factual accuracy, such as medical diagnosis [7, 8, 9], financial risk assessment [10, 11], autonomous driving [12, 13], and recommendation systems [14, 15].

Uncertainty knowledge graph embedding (UKGE) aims to embed uncertainty facts in the UKG into the latent, low-dimensional, and real-valued vector spaces. UKGE [16] proposed the first dedicated model for UKG representation learning. It used the DistMult [1] to measure the plausibility of a triple and mapped plausibility to confidence using a mapping function. The confidence scores were then incorporated into the objective function to preserve uncertainty information in the embedding. In addition, UKGE utilized Probabilistic Soft Logic (PSL) to reason about unseen facts and used them as training data, preserving rule constraints in the embedding representations. PASSLEAF [17] is a semi-supervised learning framework to deal with unseen facts, thus relieving the false-negative problem, and OpenUKGE [18] refine the energy function to be able to model multiple relation patterns. Additionally, GMUC [19] proposed a few-shot UKG completion method based on Gaussian Metric Learning, which can complete missing facts and confidence scores even in cases of limited data. Unlike traditional representation learning, BEUrRE [20] mappes entities and relations to a "box space", where entities are represented as hyper-rectangles and relations as transformations between rectangles. UKGs [21] treat each triple as a sequence and feed the trained embeddings into the LSTM to quickly predict confidence.

However, recent studies typically focus only on the uncertainty of individual triples, while neglecting the uncertainty information in the graph structure-aware information and the sequential dependencies within relational triplets. We shall consider an example of link prediction in the UKG to illustrate our modeling motivation, as shown in Fig. 1. If we want to reason about the confidence of (Einstein, believedIn, Unified_Field_Theory), it is insufficient to model only the uncertainty of the triple itself. When optimizing the embedding of the entity Einstein, other neighboring triples can also provide additional information, and the higher the confidence, the more information is provided. For (Einstein, believedIn, Unified_Field_Theory), the sequential dependency between the head entity, relation, and tail entity aids in inferring its confidence—"Einstein attempted to propose the Unified Field Theory" is a fact with high confidence, whereas reversing the head and tail entities in the triple leads to a completely incorrect statement.

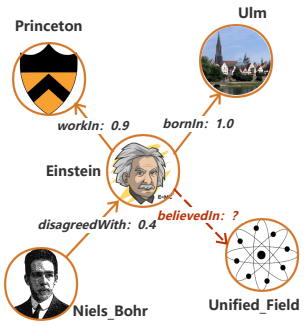


Figure 1: An example of illustrating link prediction in the UKG

Motivated by the above, we propose a novel uncertainty knowledge graph embedding model, which jointly learns graph structure-aware context and fused-triple confidence information in an encoder-decoder framework (UnKGE-GSFC). In the encoder, we use an Uncertainty-Aware Relational Graph Convolutional Network (UA-RGCN) to propagate the confidence information of the neighbor triples and control the amount of information propagated by the neighbors through the confidence to better optimize the entity embedding. In the decoder, we design two components to capture the semantic and sequential dependency information within the triples: a semantic-aware scoring component and a seq-dependency-aware scoring component. The semantic-aware scoring component calculates the confidence score of triples using the DistMult method. The seq-dependency aware scoring component treats triples as sequences and feeds them into a uncertainty-aware Gated Recurrent Unit (UA-GRU), processing the input step by step to capture the sequential dependency between entities and relations. Finally, we perform a weighted fusion of the semantic and sequential dependency information to generate the confidence score for each triple.

It is highlighted in the following four aspects:

- To the best of our knowledge, this is the first attempt to study uncertainty knowledge graph embedding from the perspective of graph structure-aware context and fused-triple confidence information.
- We propose an encoder-decoder framework, named UnKGE-GSFC. In the encoder, we aggregate the uncertainty of node neighbors on the graph structure to optimize entity embedding. In the decoder, we model the triple confidence more comprehensively from the perspective of both the semantic information and the sequential dependencies of the relational triple.
- Extensive experiments demonstrate our model significantly outperforms various state-of-the-art baselines on three real-world datasets across different settings. We also present the ablation study, and parameter sensitivity analysis to validate the effectiveness of the model.
- To illustrate the evaluation of our framework and facilitate further research on this topic, we have made the experimental details and source code of the framework publicly available at `https://github.com/shengyp/UnKGE-GSFC/`.

The remainder of this paper is organized as follows. Section 2 reviews the related work on uncertain knowledge graphs, covering representative research works associated with our work.

Section 3 introduces the preliminaries, essential notations, and definitions utilized throughout the study. Section 4 presents the technical details of our proposed uncertainty-aware embedding model, which integrates graph structure-aware context and triple confidence information; this section focuses on key components such as structural uncertainty propagation and confidence modeling. In Section 5, we conduct extensive experiments, including comparative studies and ablation analyses on multiple benchmark datasets, to demonstrate the superior performance of our model. Finally, Section 6 concludes the paper and discusses future research directions.

2. Related Work

In this section, we comprehensively surveyed the literature on two topics closely related to our work: Relational Graph Convolutional Network and Triple Confidence Modeling for Uncertain Knowledge Graph Embedding.

2.1. Relational Graph Convolutional Network

Graph Convolutional Network (GCN) [22] is a neural network model for processing graph data, which learns relations between nodes through convolutional operations. RGCN [23] is an extension of GCN, which is specifically designed to deal with multi-relation types of knowledge graphs. RGCN treats graphs under each type of relation as homogeneous graphs, and converts multi-relation graphs into a problem of aggregation of features from multiple homogeneous graphs. Based on RGCN, our work considers the confidence score of the neighbor node information whenever it is aggregated, aiming to filter low-quality information. The uncertainty-aware RGCN model is able to learn and predict more robustly, especially in the face of noisy data or unbalanced relational distributions. In addition, it can also better capture and utilize structural information in the graph, improving the model's performance on tasks such as confidence prediction.

2.2. Triple Confidence Modeling for Uncertain Knowledge Graph Embedding

The work [4] summarizes core advancements in AI knowledge processing, focusing on some strategies enabling AI to operate efficiently in environments with incomplete and ambiguous information. Currently, there has been a lot of work providing methods for modeling triple confidence. UKGE [16] is the first model specialized for UKG embedding, which uses the score function of DistMult [1] to represent the triple plausibility, and chooses the mapping function to map the plausibility to the confidence. PASSLEAF [17] uses a pool-based semi-supervised learning framework to solve the false-negative problem. UKGsE [21] is a framework for fast predicting triple confidence that treats relational facts as sequences and uses embeddings to predict confidence. BEUrRE [20] models entities as hyper-rectangles and relations as transformations on the head and tail entity rectangles, and ultimately uses the geometry to calculate the confidence. UPGAT [24] introduces an uncertainty-aware graph attention mechanism to capture probabilistic subgraph features, and pseudo-neighbor augmentation to optimize the embedding of entities and relations. UKRM [25] employs a Transformer-based architecture to mine rules for link prediction, while concurrently leveraging pre-trained language models (PLMs) to estimate triple confidence scores. To address the imbalanced distribution of confidence scores in UKGs, ssCDL [26] introduces a semi-supervised confidence distribution learning framework that leverages meta-learning to generate pseudo-labels for unlabeled triples, thereby enriching the supervision signals for embedding learning. In addition, GMUC [19] proposes for the first time a

few-shot UKG completion method based on Gaussian Metric Learning. GMUC+ [27] quantify the uncertainty of different entities and relations to capture both semantic and uncertainty information of entities and relations in the few-shot scenario. unKR [28] is a UKG embedding learning and completion tool that re-implements some works for UKG.

However, these methods usually focus only on the semantic scores of a single triple when modeling confidence. Inspired by the above methods, when calculating the confidence, the decoder component of our work not only considers the semantic information of the triple, but also treats the triple as a sequence, considering the information of its contextual sequence, to jointly model confidence.

3. Preliminary

This section provides the formal problem definitions and essential notations for UKG to establish a theoretical foundation for the proposed model.

3.1. Problem Definition

Uncertain Knowledge Graph. An Uncertain Knowledge Graph (UKG) assigns a confidence score to each relational fact, reflecting the degree of its validity. Formally, a UKG is defined over a set of entities $\mathcal{E}$ and a set of relations $\mathcal{R}$, represented as a collection of weighted triples $\mathcal{G} = \{(l, s_l)\}$. Within each tuple (l, s_l), $l = (h, r, t)$ denotes the relational triple where $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$. The value $s_l \in [0, 1]$ represents the confidence score, quantifying the probability or belief that the relational fact l holds true.

Uncertain Knowledge Graph Embedding. In alignment with traditional Knowledge Graph Embedding (KGE), the objective of an Uncertain Knowledge Graph Embedding (UKGE) model is to map each entity and relation in the UKG into a continuous low-dimensional vector space. This embedding process aims to preserve not only the intrinsic structural topology but also the confidence scores of relational facts, alongside other salient features that characterize uncertain knowledge. Specifically, the head entity h, relation r, and tail entity t are represented by their respective embedding vectors $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^k$, where k denotes the dimensionality of the space.

3.2. Important notations

To facilitate a clear understanding of the proposed framework, the primary notations and their corresponding definitions used throughout this paper are summarized in Table 1.

4. Methodology

This section delineates our approach pipeline. We first outline the overall framework. Subsequently, we provide a detailed introduction to three components and elaborate on the steps within each component. Finally, we design separate loss functions for confidence prediction of the task and the link prediction task, respectively, and combine them to jointly optimize the model.

Table 1: Summary of notations and explanation used in this paper.

Notation	Explanation
$\mathcal{G}$	An uncertain knowledge graph (UKG).
$\mathcal{E}, \mathcal{R}$	The set of entities and relations, respectively.
$l = (h, r, t)$	A relational triple fact consisting of head, relation, and tail.
s_l	The confidence score of triple l, where $s_l \in [0, 1]$.
$\boldsymbol{A}_{ij}^{(r)}$	A filtered weighted adjacency matrix for each relational subgraph.
$\mathbf{h}, \mathbf{r}, \mathbf{t}$	The d-dimensional embedding vectors for h, r, and t.
d	The dimensionality of the embedding space.
$\boldsymbol{w}, \boldsymbol{b}$	The hyperparameter of the linear layer
$s_{sem}(l)$	The semantic score for given triple l
$s_{seq}(l)$	The sequence dependency score for given triple l
σ	The Sigmoid function
α	A weighted hyperparameter that controls the contribution of semantic information and sequential dependency to the confidence score.
$s_{conf}(l)$	The predicted confidence for given triple l
$g(l)$	The ranking score predicted for given triple l.
γ	The margin hyperparameter in the ranking loss function.
$\mathcal{L}^+$	The set of train triples.
$\mathcal{L}^-$	The set of negatively sampled triples.

4.1. The Overview Framework

The proposed UnKGE-GSFC model is built upon an encoder-decoder framework to enhance the UKGE. First, an Uncertainty-Aware Relational Graph Convolutional Network (UA-RGCN) is employed as the encoder to aggregate global uncertainty information from neighboring entities (Section 4.2). Subsequently, the decoder captures uncertainty from two distinct perspectives: semantic representations and intra-triple sequential dependencies. A confidence score for each relational triple is then generated through a weighted fusion of the semantic and sequential dependency scores (Section 4.3). Finally, we integrate loss functions for both the confidence prediction task and the link prediction task to jointly optimize learned UKGE (Section 4.4). The overall architecture is illustrated in Fig. 2. In the following, we will provide a detailed exposition of these three components and their respective implementations.

4.2. Uncertainty-Aware Relational Graph Convolutional Network

The information about the neighbors of the entities in the UKG can help to optimize the embedding of the entities. To incorporate the uncertainty information in aggregating the neighbor information, we propose an uncertainty-aware relational graph convolutional network (UA-RGCN) based on the standard RGCN [23]. When aggregating information from neighbor nodes, UA-RGCN utilizes the relational triple confidence of each triple to control the amount of information propagation.

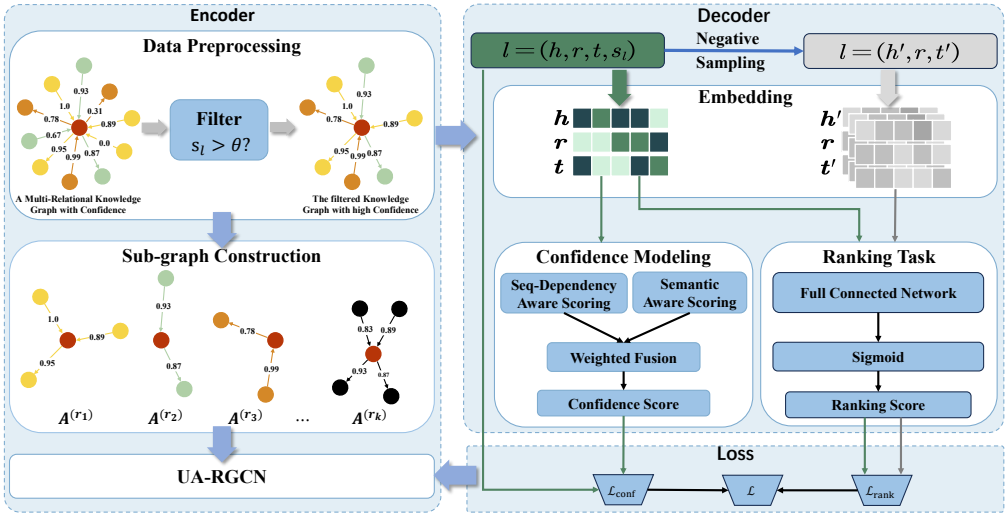


Figure 2: The overview architecture of the proposed UnKGE-GSFC model.

Given a UKG $\mathcal{G} = (l, s_l)$. For each pair, $l = (h, r, t)$ is a relational triple, where $h, t \in \mathcal{E}$ (the entity set), $r \in \mathcal{R}$ (the relation set), and $s_l \in [0, 1]$ denotes the confidence score that the relation fact is true. We first construct a weighted adjacency matrix $\boldsymbol{A}^{(r)}$ for the relational subgraph of each relation r. To avoid relational triple with low confidence scores from affecting the optimization of UA-RGCN, we set a threshold θ and filter using:

$$\boldsymbol{A}_{ij}^{(r)} = \begin{cases} s_l & if\ s_l > \theta\ in\ (h, r, t, s_l) \\ 0 & otherwise \end{cases} \tag{1}$$

For each entity i, its neighbors under relation r is denoted as $\mathcal{N}_i^r$. The weighted adjacency matrix $\boldsymbol{A}^{(r)}$ controls the amount of information propagated when a subgraph under the relation r undergoes a convolution operation. In layer l, the representation process of node $\boldsymbol{h}_i^{(l+1)}$ is formalized as:

$$\boldsymbol{h}_i^{(l+1)} = \sigma\left(\sum_{r \in R} \sum_{j \in \mathcal{N}_i^r} \boldsymbol{A}_{ij}^{(r)} \cdot \boldsymbol{W}_r^{(l)} \boldsymbol{h}_j^{(l)} + \boldsymbol{b}_r\right), \tag{2}$$

where $\boldsymbol{h}_i^{(l)} \in \mathbb{R}^d$, represents the embedding of entity i at layer l. $\boldsymbol{W}_r^{(l)} \in \mathbb{R}^{d \times d}$, represents the weight matrix for relation r at layer l. $\boldsymbol{b}_r$ is the bias term for relation r, and σ represents the LeakyReLU function, respectively. $\boldsymbol{A}_{ij}^{(r)}$ controls the amount of information passed from $h_j^{(l)}$ to $h_j^{(l+1)}$.

Through using Equ.(2), UA-RGCN can aggregate the uncertainty information of neighbor entities by layers. Weighting the contribution of each neighbor using the confidence score in the weight adjacency matrix captures the strength of the relations in the UKG. The confidence score controls the amount of information propagated by each neighbor relational triple, with a triple with higher confidence propagating more information. We refer to this graph encoder module as uncertainty-aware RGCN.

After multiple layers of UA-RGCN, we iteratively optimize the initial node embedding and ultimately obtain the embedding matrix $\boldsymbol{H} \in \mathbb{R}^{N\times d}$, which represents the embedding of all entities, where N denotes the total number of entities, and d is the dimension of the embedding.

4.3. Relational Triple Confidence Modeling

To model relational triple confidence, we design a decoder consisting of two components: a semantic-aware scoring component and a sequence-dependency-aware scoring component. These two components capture the semantic information and sequential dependency information within the triple, respectively, and the resulting scores are combined via a weighted confidence fusion component to produce the final confidence score for each triple.

4.3.1. The Semantic-Aware Scoring for Relational Triples

Inspired by the work [16], we model the semantic scores of relational triples using the DistMult [1] method. Specifically, given a triple $l = (h, r, t)$ and their embeddings $(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}) \in \mathbb{R}^d$, we generate the semantic score based on knowledge graph embeddings as follows:

$$s_{sem} = \boldsymbol{r} \cdot (\boldsymbol{h} \circ \boldsymbol{t}), \tag{3}$$

where $\circ$ is element-wise product and $\cdot$ is the inner product. Equation (3) captures the correlation between embeddings $\boldsymbol{h}$ and $\boldsymbol{t}$ under the condition of relation $\boldsymbol{r}$.

To make sure the s_{sem} scores predicted from this module are in the range $[0, 1]$, we use a Logistic mapping function that has been proven effective [16]. Formally, given a triple l with a s_{sem} score, we map to $[0, 1]$ calculated by:

$$s_{sem}(l) = \frac{1}{1 + e^{-(\boldsymbol{w} s_{sem} + \boldsymbol{b})}}, \tag{4}$$

where $\boldsymbol{w}$ is a weight parameter and $\boldsymbol{b}$ is a bias.

4.3.2. The Seq-Dependency-Aware Scoring for Relational Triples

To model the sequential dependencies within the relational triple in the UKG, we treat each relational triple as a sequence and input its concatenated embedding $\boldsymbol{x}_0$ into an uncertainty-aware Gated Recurrent Unit (UA-GRU). The representation of the UA-GRU input sequence $\boldsymbol{x}_0$ is as follows:

$$\boldsymbol{x}_0 = concat(\boldsymbol{h}, \boldsymbol{r}, \boldsymbol{t}), \tag{5}$$

where $concat(\cdot)$ is a concatenation operation that concatenates three $\mathbb{R}^d$ tensors into a $\mathbb{R}^{3\times d}$ tensor.

The GRU [29] can dynamically update the embeddings of the head entity, relation, and tail entity based on capturing the sequential dependency between entities and relations. Equation (6) is a compact form of GRU unit.

$$\begin{aligned}
\boldsymbol{z_t} &= \sigma(\boldsymbol{W}_z \cdot [\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b}_z) \\
\boldsymbol{r_t} &= \sigma(\boldsymbol{W}_r \cdot [\boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b}_r) \\
\tilde{\boldsymbol{h}}_t &= \tanh(\boldsymbol{W}_h \cdot [\boldsymbol{r_t} * \boldsymbol{h_{t-1}}, \boldsymbol{x_t}] + \boldsymbol{b}_h) \\
\boldsymbol{h_t} &= (1 - \boldsymbol{z_t}) * \boldsymbol{h_{t-1}} + \boldsymbol{z_t} * \tilde{\boldsymbol{h}}_t,
\end{aligned} \tag{6}$$

where $\boldsymbol{z_t}$ is the update gate, $\boldsymbol{r_t}$ is the reset gate, $\boldsymbol{h_t}$ is the final hidden state at time step t, $\tilde{\boldsymbol{h}}_t$ is the candidate hidden state, and $\boldsymbol{x_t} \in \mathbb{R}^{3\times d}$ is the input triple at time step t. $\boldsymbol{W}_z \in \mathbb{R}^{H\times d}$, $\boldsymbol{W_r} \in \mathbb{R}^{H\times d}$ and

$\boldsymbol{W_h} \in \mathbb{R}^{H\times d}$ are the weight matrices, where H represents the hidden layer dimension of the model. And the parameter σ is the sigmoid function, $\boldsymbol{b_z}$, $\boldsymbol{b_r}$ and $\boldsymbol{b_h}$ are the bias terms, respectively.

Multiple layers of GRU units are nested in the UA-GRU. The input triple sequence embedding undergoes updates through several GRU units. At the last time step, the hidden state $\boldsymbol{h}_n$ is input into a fully connected layer to output the triple's sequence dependency score. To ensure that the output score is in the range $[0, 1]$, we use a logistic function [16]. This process can be defined as follows:

$$s_{seq}(l) = \sigma(\boldsymbol{W_{fc}} \cdot \boldsymbol{h_n} + \boldsymbol{b}_{fc}), \tag{7}$$

where $\boldsymbol{W_{fc}} \in \mathbb{R}^{1\times H}$ is the weight matrix of the fully connected layer, $\boldsymbol{h}_n$ is the last hidden state output by the GRU, and $\boldsymbol{b}_{fc}$ is the bias term of the fully connected layer. Here, H represents the hidden layer dimension. And the parameter σ is the sigmoid function.

4.3.3. Relational Triple Confidence Fusion

We generate a confidence score for the triple by weighted fusion of the semantic score and sequential dependency score of the triple $l = (h, r, t)$. The confidence computation function is as follows:

$$s_{conf}(l) = \alpha \cdot s_{sem}(l) + (1 - \alpha) \cdot s_{seq}(l), \tag{8}$$

where α is a weighted parameter that controls the contribution of semantic information and sequential dependency to the confidence score.

4.4. Model Learning Objective

Our model is applied to both the confidence prediction task and the link prediction task. We design separate loss functions for each task and combine them to jointly optimize the model.

For the confidence prediction task, let $\mathcal{L}^+$ represent the set of positive samples, and the training objective is to minimize the mean squared error (MSE) between the true confidence score s_l and our predicted confidence score $s_{conf}(l)$ for each triple $l \in \mathcal{L}^+$, to generate the loss $\mathcal{L}_{conf}$:

$$\mathcal{L}_{conf} = \sum_{(h,r,t,s_l)\in\mathcal{L}^+} \left\| s_{conf}(h, r, t) - s_l \right\|^2. \tag{9}$$

For the link prediction task, following [26], we feed the concatenated embedding of triple $l = (h, r, t)$ containing uncertainty information into a two-layer Fully Connected Network (FCN) to compute the ranking score $g(l)$ of this triple. The output is subsequently normalized via a Sigmoid function, defined as follows:

$$g(l) = \sigma(FCN(\boldsymbol{h}\|\boldsymbol{r}\|\boldsymbol{t})), \tag{10}$$

where σ denotes a Sigmoid function, and $FCN(\cdot)$ represents the fully connected layer employed to calculate the triple ranking scores. We formulate a margin-based ranking loss, $\mathcal{L}_{rank}$, to train the model. The optimization objective is to ensure that the scores of positive samples are consistently higher than those of the negatively sampled counterparts by a predefined margin. The loss $\mathcal{L}_{rank}$ is computed by:

$$\mathcal{L}_{rank} = \sum_{(l,s_l)\in\mathcal{L}^+} \sum_{l'\in\mathcal{L}^-} s_l \cdot \left[\gamma + g(l') - g(l)\right]_+, \tag{11}$$

where $\mathcal{L}^-$ denotes the set of negative triples, which are constructed by corrupting either the head or tail entity of a valid triple with a randomly sampled entity from the entity set. The term

$g(l')$ represents the ranking score of a negative sample $l' \in \mathcal{L}^-$. Additionally, $[x]_+ = \max(0, x)$ denotes the standard hinge loss (or positive part) operator, and $\gamma > 0$ is a predefined margin hyperparameter.

We train our model by jointly minimizing the sum of the aforementioned two losses $\mathcal{L}_{conf}$ and $\mathcal{L}_{rank}$ as follows:

$$\mathcal{L} = (1 - \lambda) \cdot \mathcal{L}_{conf} + \lambda \cdot \mathcal{L}_{rank} + \|\Theta\|_2^2, \tag{12}$$

where λ is a weight that balances the importance of two losses. Θ denotes all trainable model parameters.

5. Experiments and Analysis

In this section, we present the experimental settings, introducing evaluation datasets, the evaluation protocols and metrics, evaluation baselines, along with the parameter details, and conduct a comprehensive analysis of our proposed method for the confidence prediction task and the link prediction tasks, concluding with a case study.

5.1. Experimental Settings

5.1.1. Evaluation Datasets

We conduct experiments on three benchmark datasets, including CN15k [5], NL27k [6], and PPI5k [30]. CN15k is a subgraph of ConceptNet [5], NL27k is a subgraph from the NELL database [6], and PPI5k is a protein-to-protein interaction ontology from the STRING database [30]. In addition, we divide the dataset, 85% for training, 7% for validation, and 8% for testing. The statistics of the datasets are given in Table 2.

Table 2: Statistics of the dataset extracted in this paper. Avg(s) and Std(s) are the average and standard deviation of the confidence scores.

Dataset	#Entity	#Relation	#Rel. Facts	#Train	#Validation	#Test	Avg(s)	Std(s)
CN15K	15,000	36	241,158	204,984	16,881	19,293	0.629	0.232
NL27K	27,221	417	175,412	149,100	12,278	14,034	0.797	0.242
PPI5K	4,999	7	271,666	230,929	19,017	21,720	0.415	0.213

5.1.2. Evaluation Protocols and Metrics

The confidence prediction task is constructed as a regression problem aiming at predicting the confidence score for a given triple. For each uncertain relation fact (l, s_l) in the test set, we predict the confidence score of l and use the mean squared error (MSE) and mean absolute error (MAE) as evaluation metrics. When using this evaluation approach, we use the optimal value of MSE as the early-stop value.

The link Prediction task is to complete a triple (h, r, t) when t missing, i.e., to predict t given $(h, r, ?)$. For evaluation, following the setup of research [26], we will use Mean Reciprocal Rank (MRR) and Hits@N as evaluation metrics. When using this evaluation approach, we use the optimal value of MRR as the early-stop value.

5.1.3. Evaluation Baselines

We compare our method against six state-of-the-art baselines, including UKGE [16], PASSLEAF [17], UKGsE [21], BEUrRE [20], UPGAT [24], and ssCDL [26]. UKGE is a version of research trained on data enhanced with PSL, PASSLEAF incorporates a semi-supervised mechanism based on UKGE. Both models utilize the same logistic mapping functions. UKGsE pre-processes relational facts as sequences and predicts confidence using the obtained embeddings. Unlike traditional representation learning, BEUrRE models each entity as a hyper-rectangle and relations as transformations on the rectangles of the head and tail entities, ultimately calculating confidence based on the geometric shapes of the rectangles. UPGAT utilizes an uncertainty-aware attention mechanism and neighbor augmentation to capture probabilistic features. ssCDL employs a semi-supervised meta-learning framework to generate pseudo-labels, addressing confidence imbalance in UKGs.

5.1.4. Parameters Details

We choose the hyperparameters based on the performance of the validation set. For hyper-parameter tuning, we search the best hyper-parameter as follows: learning rate $lr \in \{0.0001, 0.005, 0.001\}$, dimension $d \in \{128, 256, 512\}$, batch size $b \in \{512, 1024, 2048, 4096\}$. And for each dataset, the best combination of hyperparameters is as follows: the batch size b is set to 4096, the dimensions of entities and relations d are 512, and the learning rate lr is 0.0001. The number of negative samples is set to 30. The hyperparameter γ of the link loss is set to 0.1. During training, we use the Adam optimizer [31] for training, and set epochs to 500. To reduce computational overhead, both UA-RGCN and GRU modules are set to have 2 layers, with a Dropout rate of 0.2. We also set two important hyperparameters: θ to 0.7 and α to 0.5, respectively. The hyperparameter λ in the loss function is set to 0.7 for the CN15K and PPI5K datasets, and 0.5 for NL27K. Furthermore, we incorporate an early stopping mechanism, with a patience parameter set to 30.

Our experiments are conducted under the following hardware and software configurations: the GPU used is RTX 4090 (24GB), the CPU is Intel Xeon Platinum 8352V, and the operating system is Ubuntu 20.04. All methods are implemented using Pytorch 2.0.0 platform with Cuda 11.8[1].

5.2. Experimental Results and Analysis

5.2.1. Experiments on Confidence Prediction Task

The experimental results for the confidence prediction task are summarized in Table 3. Our proposed UnKGE-GSFC consistently outperforms all baselines in MSE and MAE across three datasets, demonstrating its superior effectiveness. Notably, on CN15K, UnKGE-GSFC reduces the MSE from 11.82 (UKGE) to 3.03, a 74.4% improvement, suggesting a stronger capacity to capture nuanced probabilistic features in UKGs. The UKGE and PASSLEAF, which focus only on the confidence of individual triples, cannot model uncertainty in the global graph structure. Although the UKGsE framework can quickly predict confidence score, it cannot characterize more uncertain information. Additionally, BEUrRE proposes an innovative triple confidence modeling approach, which can only model entities in a box space and cannot capture more complex information. In contrast, UnKGE-GSFC adeptly encodes the uncertainty inherent in the global graph structure and models confidence from both semantic and sequential dependency

[1] `https://pytorch.org/`

Table 3: Results of Confidence Prediction Task. Mean squared error (MSE) and mean absolute error (MAE) of relation fact confidence prediction ($\times 10^{-2}$). The results of the UPGAT and ssCDL are obtained from their paper.

Dataset	CN15K		NL27K		PPI5K	
Metrics	**MSE**	**MAE**	**MSE**	**MAE**	**MSE**	**MAE**
UKGE	11.82	28.86	2.81	9.89	0.56	3.92
PASSLEAF	11.46	28.90	2.32	8.54	0.49	3.50
UKGsE	12.25	29.93	11.67	26.38	1.35	6.92
BEUrRE	12.30	29.95	3.39	12.42	0.95	5.90
UPGAT	14.90	30.80	2.90	10.10	0.34	-
ssCDL	3.40	**11.60**	**0.90**	**4.20**	-	-
UnKGE-GSFC	**3.03**	12.50	1.20	5.71	**0.25**	**2.62**

perspectives, which is more effective in uncertainty reasoning. Although ssCDL is the most effective on NL27K, the lower MSE of UnKGE-GSFC on CN15K indicates better robustness against outliers and extreme errors, as MSE more heavily penalizes large deviations.

Finally, the substantially lower MSE observed across all models on PPI5K compared to CN15K and NL27K is likely attributable to its higher graph density, which facilitates the learning of more expressive embedding representations.

5.2.2. Experiments on Link Prediction Task

The experimental results for the link prediction task across three datasets are presented in Table 4. Overall, UnKGE-GSFC achieves state-of-the-art performance on several key metrics. On the CN15K dataset, UnKGE-GSFC outperforms all baselines in terms of MRR, Hits@5, and Hits@10. Similarly, for the PPI5K dataset, our model achieves the highest MRR, Hits@1, Hits@5 and Hits@10. These results suggest that UnKGE-GSFC better captures graph structure information and semantics and sequence information of triples, effectively ranking ground-truth tail entities. Unlike traditional models that struggle with high-uncertainty scenarios due to limited point-based embeddings, UnKGE-GSFC integrates semantic and sequential dependencies to provide more robust representations.

UnKGE-GSFC demonstrates strong overall performance on the most indicators, its improvements in some link prediction metrics are less pronounced compared to the confidence prediction task, particularly when contrasted with ssCDL. For example, on the CN15K and NL27K dataset, the Hits@1 metric of our model performs significantly worse. This discrepancy stems from our architectural focus on confidence modeling, and link prediction only relies on implicit optimization via margin-based ranking loss rather than a dedicated reasoning component. Despite this, UnKGE-GSFC still surpasses most baselines, underscoring its efficacy as a multi-task learning framework that leverages confidence-aware embeddings to enhance uncertain knowledge graph reasoning tasks.

5.2.3. Parameter Sensitivity Analysis

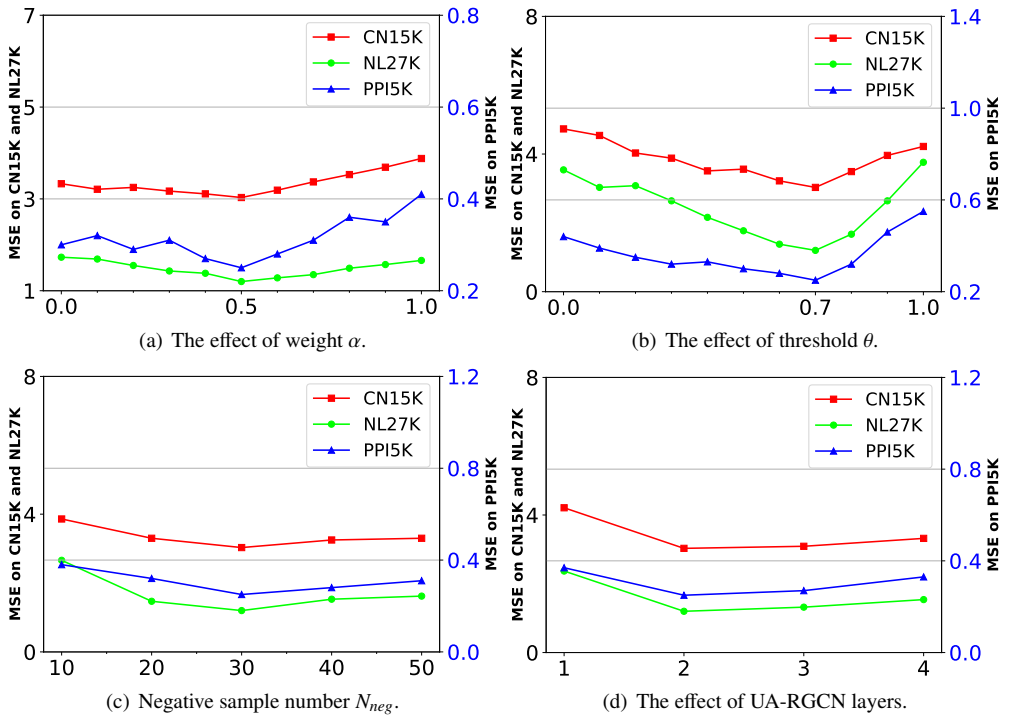
We set up experiments for confidence prediction task on three datasets to explore the effect of four important hyperparameters: weight α in triple confidence modeling, threshold θ in filtering high confidence triples, negative sample number N_{neg} in training, and the number of UA-RGCN layers. Specifically, with all other parameters fixed, we make α vary in the range $[0, 1]$ in step of

Table 4: Results of link prediction task. The three ranges of the Hits@N metric are 1, 5, and 10. The results of the UPGAT and ssCDL are obtained from their paper.

Dataset	Model	MRR	Hits@1	Hits@5	Hits@10
CN15K	UKGE	0.109	0.063	0.154	0.201
	PASSLEAF	0.110	0.064	0.156	0.202
	UKGsE	0.006	0.001	0.007	0.011
	BEUrRE	0.109	0.043	0.179	0.217
	UPGAT	-	0.078	-	-
	ssCDL	-	**0.133**	-	-
	UnKGE-GSFC	**0.119**	0.052	**0.187**	**0.251**
NL27K	UKGE	0.425	0.334	0.523	0.593
	PASSLEAF	**0.438**	0.350	0.531	0.596
	UKGsE	0.062	0.034	0.081	0.116
	BEUrRE	0.244	0.105	0.391	0.475
	UPGAT	-	0.530	-	-
	ssCDL	-	**0.636**	-	-
	UnKGE-GSFC	0.387	0.240	**0.546**	**0.631**
PPI5K	UKGE	0.761	0.627	0.924	0.957
	PASSLEAF	0.620	0.457	0.822	0.915
	UKGsE	0.361	0.243	0.490	0.615
	BEUrRE	0.427	0.001	0.916	0.955
	UPGAT	-	-	-	-
	ssCDL	-	-	-	-
	UnKGE-GSFC	**0.785**	**0.656**	**0.933**	**0.960**

0.1, make θ vary in the range $[0, 1]$ in step of 0.1, make the number of negative samples during training $N_{neg} \in \{10, 20, 30, 40, 50\}$, and make the number of UA-RGCN layers vary in $\{1, 2, 3, 4\}$.

The results about parameter α are reported in Fig. 3(a). The model performs best with α of 0.5 on all three datasets, and increasing or decreasing it weakens the performance. We can conclude that both semantic and sequential dependency information of the triple are both important for modeling triple confidence. And in Fig. 3(b), we find the best performance with parameter θ of 0.7. We conclude that if the θ is too high, too little information is aggregated; conversely, much low-quality information is aggregated, which will affect the embedding optimization. In Fig. 3(c), we find that the best performance is achieved when the number of negative samples is 30. We hypothesize that when the number of negative samples is lager or small, it is difficult for the model to fully learn the embedding of entities and relations, resulting in poor model performance. And in Fig. 3(d), we find that UA-RGCN performs best when the number of layers is 2. When the number of layers is small, the model training performance is poor; while when the number of layers is high, there may be over-smoothing, leading to worse results. At the same time, more layers lead to an increase in training time. Therefore, considering the cost of time and space, we choose 0.5 for parameter α, 0.7 for θ, 30 for N_{neg}, and 2 for layer number as the best parameter combination.



(a) The effect of weight α.

(b) The effect of threshold θ.

(c) Negative sample number N_{neg}.

(d) The effect of UA-RGCN layers.

Figure 3: The effect of four important hyperparameters in the model. The left vertical axis represents the range of CN15K and NL27K, and the right one represents the range of PPI5K.

5.3. Ablation Study

To provide a detailed analysis of the effectiveness of individual components in our model, we conduct an ablation study comparing our model and its three variant models. Specifically, w/o graph denotes the variant model that does not incorporate uncertainty-aware relational graph convolutional information, w/o sem refers to the variant model that excludes semantic information during the triple confidence modeling, and w/o seq is the variant model that disregards sequential dependency information. We compare the performance of these three variant models with our full model on the Confidence Prediction and Link Prediction tasks to explore the specific contributions of these three components.

The results of the confidence prediction task are reported in Table 5. Through analyzing the experimental data, we find that our model outperforms its variant models across all three datasets, which reveals that the three components we proposed have a positive impact on the model performance. We also notice that the w/o graph model performs most poorly on confidence prediction in all three datasets, which indicates that aggregating knowledge uncertainty through UA-RGCN and using optimized entity embedding can significantly improve the performance of our model on the confidence prediction task.

The results of link prediction in the ablation study are reported in Table 5. For the link prediction task, we observe that w/o sem and w/o seq perform notably worse than the full model in terms of Hits@k and MRR. For instance, on the NL27K dataset, removing the semantic or sequential modules leads to a substantial drop in Hits@10 (from 0.631 to 0.629 and 0.597, re-

Table 5: Results of Confidence Prediction and Link Prediction Tasks in the ablation study.

Dataset	Model	MSE	MAE	MRR	Hits@1	Hits@5	Hits@10
CN15K	w/o graph	4.22	17.38	0.098	0.041	0.151	0.204
	w/o seq	3.88	16.04	0.094	0.038	0.143	0.201
	w/o sem	3.33	14.18	0.105	0.044	0.163	0.224
	UnKGE-GSFC	**3.03**	**12.50**	**0.119**	**0.052**	**0.187**	**0.251**
NL27K	w/o graph	3.76	13.87	0.290	0.162	0.426	0.510
	w/o seq	1.66	7.47	0.364	0.230	0.508	0.597
	w/o sem	1.73	6.98	0.373	0.227	0.533	0.629
	UnKGE-GSFC	**1.20**	**5.71**	**0.387**	**0.240**	**0.546**	**0.631**
PPI5K	w/o graph	0.55	4.43	0.726	0.550	0.928	0.953
	w/o seq	0.41	4.02	0.724	0.558	0.916	0.944
	w/o sem	0.30	2.93	0.758	0.619	0.920	0.952
	UnKGE-GSFC	**0.25**	**2.62**	**0.785**	**0.656**	**0.933**	**0.960**

spectively). This indicates that while the confidence modeling module of our full model is only optimized for confidence prediction, the learned semantic and sequential uncertainties present within triples also contribute to distinguishing truth tail entities and enhancing its reasoning capabilities during the link prediction task.

5.4. Case Study

Table 6 shows some examples of confidence and entity prediction results using our model. Given the query $(h, r, t/?, ?)$, we predict the confidence of the seen triples in the dataset and also find some unseen triples that do not exist in the dataset but are reasonable in the real world. In the results, we find that these predictions are consistent with our common sense. For the triples that exist in the dataset, the confidence predictions of our model do not deviate very much from the true confidence, indicating that it can support basic uncertain judgments. It is worth noting that some quite reasonable unseen relation facts can be predicted correctly, such as hotel is a place. These examples show that our proposed approach can reasonably predict confidence scores, as well as infer new knowledge from observed knowledge, indicating that our model performs well on reasoning tasks.

6. Conclusion and Future Work

In this paper, we address the limitations of existing UKGE models in precisely capturing and returning uncertainty knowledge to reasoning for uncertain facts. In particular, we propose a novel uncertainty knowledge graph embedding method. Extensive experiments on confidence prediction and link prediction tasks demonstrate the superiority of our model. In addition, we present ablation studies, and parameter sensitivity to validate the effectiveness of the proposed method.

Despite our approach demonstrates improved performance on popular UKGE tasks, we acknowledge certain limitations. Future research will focus on two strategic directions:

Table 6: Examples of confidence prediction task using our model. N/A denotes relation facts that are not observed in KG.

Dataset	Head	Relation	Tail	Conf.	Pred.
CN15K	rush	relatedto	fast	0.968	0.927
			rapid	0.709	0.717
	hotel	usedfor	sleeping	1.0	0.966
		isa	place	N/A	0.903
NL27K	Toyota	competeswith	Honda	1.0	0.904
			Chrysler	N/A	0.922

- **Refining Graph Structure Learning for UKG**. Our model focuses on learning globally graph structure-aware representations by leveraging relational triple confidence scores to control message propagation among neighboring entities in the RGCN. To overcome the unreliable confidence scoring and potential errors in triple confidence computation, we will explore to introduce reward function-based strategies in reinforcement learning. This approach aims to enhance the efficacy of message propagation during the learning of graph structural information within the UKG.
- **Boosting Imbalanced Distributions of Triple Relation**. We observe the long-tailed relation data distributions in the UKG. We plan to leverage large language models (LLMs) to generate large-scale synthetic data to augment the imbalanced distributions of triples with few relations equipped with specific tasks.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 62202075, 62566007), the Natural Science Foundation of Chongqing, China (No. CSTB2022NSCQ-MSX1404), Fundamental Research Funds for the Central Universities (No. SWU-KR24008), Key Laboratory of Data Science and Smart Education, Hainan Normal University, Ministry of Education (No. DSIE202206), and the State Key Laboratory of Public Big Data, Guizhou University (No. PBD2024-0501).

References

[1] B. Yang, W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," *arXiv preprint arXiv:1412.6575*, 2014.

[2] D. Barbosa, H. Wang, and C. Yu, "Shallow information extraction for the knowledge web," in *Proceedings of ICDE*, 2013, pp. 1264–1267.

[3] F. Feng, X. He, X. Wang, C. Luo, Y. Liu, and T.-S. Chua, "Temporal relational ranking for stock prediction," *ACM Transactions on Information Systems (TOIS)*, vol. 37, no. 2, pp. 1–30, 2019.

[4] S. Deng, N. Zhang, and T. Wu, "From certainty to uncertainty in knowledge: Exploring modeling, extraction, representation, and applications," in *Handbook on Neurosymbolic AI and Knowledge Graphs*. IOS Press, 2025, pp. 319–362.

[5] R. Speer, J. Chin, and C. Havasi, "Conceptnet 5.5: An open multilingual graph of general knowledge," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, vol. 31, no. 1, 2017.

[6] 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.
[7] P. K. Han, W. M. Klein, and N. K. Arora, "Varieties of uncertainty in health care: a conceptual taxonomy," *Medical Decision Making*, vol. 31, no. 6, pp. 828–838, 2011.
[8] C. Leibig, V. Allken, M. S. Ayhan, P. Berens, and S. Wahl, "Leveraging uncertainty information from deep neural networks for disease detection," *Scientific reports*, vol. 7, no. 1, pp. 1–14, 2017.
[9] E. Begoli, T. Bhattacharya, and D. Kusnezov, "The need for uncertainty quantification in machine-assisted medical decision making," *Nature Machine Intelligence*, vol. 1, no. 1, pp. 20–23, 2019.
[10] E. Simangunsong, L. C. Hendry, and M. Stevenson, "Supply-chain uncertainty: a review and theoretical foundation for future research," *International journal of production research*, vol. 50, no. 16, pp. 4493–4523, 2012.
[11] W. A. Brock and C. H. Hommes, "Heterogeneous beliefs and routes to chaos in a simple asset pricing model," *Journal of Economic dynamics and Control*, vol. 22, no. 8-9, pp. 1235–1274, 1998.
[12] D. Feng, A. Harakeh, S. L. Waslander, and K. Dietmayer, "A review and comparative study on probabilistic object detection in autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 9961–9980, 2021.
[13] S. Khaitan, Q. Lin, and J. M. Dolan, "Safe planning and control under uncertainty for self-driving," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 9826–9837, 2021.
[14] C. Paliwal, A. Majumder, and S. Kaveri, "Predictive relevance uncertainty for recommendation systems," in *Proceedings of the ACM Web Conference 2024*, 2024, pp. 3900–3909.
[15] F. Zhou, T. Qian, Y. Mo, Z. Cheng, C. Xiao, J. Wu, and G. Trajcevski, "Uncertainty-aware heterogeneous representation learning in poi recommender systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 7, pp. 4522–4535, 2023.
[16] X. Chen, M. Chen, W. Shi, Y. Sun, and C. Zaniolo, "Embedding uncertain knowledge graphs," in *Proceedings of the 33th AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 3363–3370.
[17] Z.-M. Chen, M.-Y. Yeh, and T.-W. Kuo, "Passleaf: a pool-based semi-supervised learning framework for uncertain knowledge graph embedding," in *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, 2021, pp. 4019–4026.
[18] J. Li and G. Li, "Analysis of semantic-based knowledge graph embedding models in uncertainty knowledge graphs," in *2022 IEEE 24th Int Conf on High Performance Computing & Communications; 8th Int Conf on Data Science & Systems; 20th Int Conf on Smart City; 8th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*. IEEE, 2022, pp. 2084–2092.
[19] J. Zhang, T. Wu, and G. Qi, "Gaussian metric learning for few-shot uncertain knowledge graph completion," in *Proceedings of Proceedings of the 26th International Conference on Database Systems for Advanced Applications*. Springer, 2021, pp. 256–271.
[20] X. Chen, M. Boratko, M. Chen, S. S. Dasgupta, X. L. Li, and A. McCallum, "Probabilistic box embeddings for uncertain knowledge graph reasoning," *https://arxiv.org/abs/2104.04597*, 2021.
[21] S. Yang, W. Zhang, and R. Tang, "Fast confidence prediction of uncertainty based on knowledge graph embedding," in *Proceedings of the 2020 3rd International Conference on Algorithms, Computing and Artificial Intelligence*, 2020, pp. 1–6.
[22] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *https://arxiv.org/abs/1609.02907*, 2016.
[23] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *Proceedings of ESWC*. Springer, 2018, pp. 593–607.
[24] Y.-C. Tseng, Z.-M. Chen, M.-Y. Yeh, and S.-D. Lin, "Upgat: Uncertainty-aware pseudo-neighbor augmented knowledge graph attention network," in *Proceedings of PAKDD*. Springer, 2023, pp. 53–65.
[25] Y. Chen, T. Wu, Y. Liu, Y. Wang, and G. Qi, "Uncertain knowledge graph completion with rule mining," in *International Conference on Web Information Systems and Applications*. Springer, 2024, pp. 100–112.
[26] T. Wu, S. Zhu, J. Wang, N. Xu, G. Qi, and H. Wang, "Uncertain knowledge graph completion via semi-supervised confidence distribution learning," *arXiv preprint arXiv:2510.16601*, 2025.
[27] J. Wang, T. Wu, and J. Zhang, "Incorporating uncertainty of entities and relations into few-shot uncertain knowledge graph embedding," in *China conference on knowledge graph and semantic computing*. Springer, 2022, pp. 16–28.
[28] J. Wang, T. Wu, S. Chen, Y. Liu, S. Zhu, W. Li, J. Xu, and G. Qi, "unkr: A python library for uncertain knowledge graph reasoning by representation learning," in *Proceedings of SIGIR*. ACM, 2024, pp. 2822–2826.
[29] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *https://arxiv.org/abs/1406.1078*, 2014.
[30] D. Szklarczyk, J. H. Morris, H. Cook, M. Kuhn, S. Wyder, M. Simonovic, A. Santos, N. T. Doncheva, A. Roth, P. Bork *et al.*, "The string database in 2017: quality-controlled protein–protein association networks, made broadly accessible," *Nucleic acids research*, p. gkw937, 2016.

[31] D. P. Kingma, “Adam: A method for stochastic optimization,” *https://arxiv.org/abs/1412.6980*, 2014.