

Uncertain Knowledge Graph Completion via Semi-Supervised Confidence Distribution Learning

Tianxing Wu^{1,2,*} Shutong Zhu¹ Jingting Wang¹ Ning Xu^{1,2,*}
Guilin Qi^{1,2} Haofen Wang³

¹School of Computer Science and Engineering, Southeast University, China

²Key Laboratory of New Generation Artificial Intelligence Technology and its Interdisciplinary Applications (Southeast University), Ministry of Education, China

³College of Design and Innovation, Tongji University, China

{tianxingwu, shutong_zhu, xning}@seu.edu.cn

Abstract

Uncertain knowledge graphs (UKGs) associate each triple with a confidence score to provide more precise knowledge representations. Recently, since real-world UKGs suffer from the incompleteness, uncertain knowledge graph (UKG) completion attracts more attention, aiming to complete missing triples and confidences. Current studies attempt to learn UKG embeddings to solve this problem, but they neglect the extremely imbalanced distributions of triple confidences. This causes that the learnt embeddings are insufficient to high-quality UKG completion. Thus, in this paper, to address the above issue, we propose a new semi-supervised Confidence Distribution Learning (ssCDL) method for UKG completion, where each triple confidence is transformed into a confidence distribution to introduce more supervision information of different confidences to reinforce the embedding learning process. ssCDL iteratively learns UKG embedding by relational learning on labeled data (i.e., existing triples with confidences) and unlabeled data with pseudo labels (i.e., unseen triples with the generated confidences), which are predicted by meta-learning to augment the training data and rebalance the distribution of triple confidences. Experiments on two UKG datasets demonstrate that ssCDL consistently outperforms state-of-the-art baselines in different evaluation metrics.

1 Introduction

Knowledge Graphs (KGs) are usually defined as multi-relational graphs describing knowledge with deterministic triples, each of which is in the form of (*subject, predicate, object*), e.g., (*Michael Jordan, Nationality, U.S.*). Such a kind of structured knowledge has supported many applications, including question answering [14], semantic search [7], decision-making systems [27], and etc. Recently, uncertain KGs (UKGs), such as NELL [1], ConceptNet [17], and Probbase [24, 10], have received much more attention. UKGs measure the uncertainty of knowledge by associating each triple with a confidence score, which also denotes the likelihood of that triple to be true. Such a setting benefits to precise knowledge representation and reasoning in the real world.

Most KGs suffer from incompleteness [21] since new knowledge is always emerging over time, and so are UKGs. Thus, various uncertain knowledge graph (UKG) embedding methods [3, 12, 5, 2, 26, 20, 22] are proposed to perform link prediction and confidence prediction for UKG completion. UKG embedding learns the representations of entities and relations in a low-dimensional space where graph structures and triple confidences are preserved. During the learning process, the above methods neglect the fact that the distributions of triple confidences are extremely imbalanced in

*Corresponding authors.

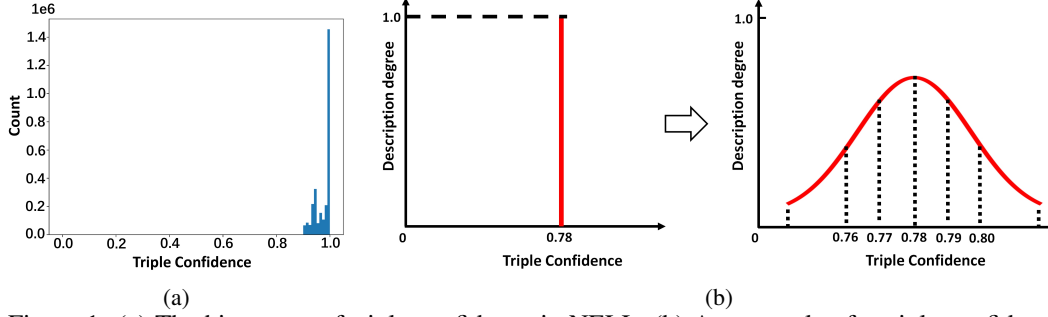


Figure 1: (a) The histogram of triple confidence in NELL; (b) An example of a triple confidence equal to 0.78, which is transformed to a confidence distribution.

most UKGs, i.e., only high-confidence triples are reserved. For example, as shown in Figure 1(a), NELL only contains the triples with confidences larger than 0.9, and this is because there is no need to store low-confidence triples which are probably erroneous. Learning on such imbalanced data will cause that the embedding-based models cannot fit on relatively lower-confidence samples, which lowers the quality of the generated embeddings for UKG completion. In this paper, we study **how to reinforce the learning process under the imbalanced confidence distribution to obtain high-quality embeddings for UKG completion**. This problem is non-trivial, and we try to solve it by the reinforcement strategies with labeled data (i.e., existing triples with confidences) and unlabeled data (i.e., sampled unseen triples without confidences), which poses two challenges as follows:

- **Challenge 1: Reinforcement with Labeled Data.** In UKG embedding learning, labeled data are triples with confidences in the training data, and such confidences are imbalanced, so the challenge is how to effectively capture the supervision signals of unseen confidences or the confidences with a small number from labeled data to reinforce the learning process.
- **Challenge 2: Reinforcement with Unlabeled Data.** Since the training data is short of low-confidence triples, we aim to apply negative sampling to add unseen triples which are often false and probably low-confidence to the training data. Thus, the challenge is how to generate reliable confidences for the unlabeled data (i.e., unseen triples) to reinforce the learning process.

To solve both challenges, we propose a new **semi-supervised Confidence Distribution Learning (ssCDL)** method for UKG completion. In ssCDL, each triple confidence is transformed into a confidence distribution. The triple confidence in UKG is a relatively fuzzy concept, e.g., there is no obvious distinction that the confidence of a triple is 0.77 or 0.78 or 0.79. This inspires us that the triples with neighboring confidences can be utilized while learning features for a particular confidence, which is similar to the usage of label distribution in facial age estimation [9]. As shown in Figure 1(b), after transforming the confidence 0.78 into a confidence distribution for a triple in the labeled data, the supervision signals of more confidences (e.g., 0.76, 0.77, 0.79, and etc.) can be introduced into the learning process, even if such confidences are few or unseen, which **can help solve challenge 1**.

ssCDL has two components: Confidence Distribution Learning based Relational Learner (CDL-RL) and Pseudo Confidence Distribution Generator (PCDG). CDL-RL iteratively learns UKG embeddings with labeled data and pseudo labeled data (i.e., negative sampled unseen triples with pseudo confidence labels) generated by PCDG for UKG completion. PCDG selects high-quality pseudo confidence labels for unlabeled data, and it is meta optimized by CDL-RL with labeled data. The whole process is actually meta self-training, which can alleviate the problem of gradual drifts [15] and introduce more reliable confidence labels for unlabeled data, thereby **overcoming the challenge 2**. Experiments on real-world datasets show the effectiveness and superiority of ssCDL compared with the state-of-the-art baselines in both UKG completion tasks of confidence prediction and link prediction.

Contributions. The main contributions of this paper are summarized as follows:

- We propose a new semi-supervised method ssCDL, which applies meta self-training to generate reliable confidences for unlabeled data in UKG embedding learning. This fully exploits unlabeled data to augment the training data so as to resolve the problem of imbalanced confidence distribution.
- We design a new confidence distribution learning strategy in UKG embedding learning, which transforms triple confidences into confidence distributions and this benefits to capture the supervision information of few or unseen confidences in the labeled data.

- We conduct comprehensive experiments on UKG datasets, which not only shows that ssCDL outperforms baselines in different evaluation metrics for different tasks, but also verifies the effectiveness of confidence distribution learning and meta self-training for UKG completion.

2 Related Work

In this section, we review the existing studies on UKG completion, which refers to confidence prediction and link prediction. Relational learning is widely used to acquire UKG embeddings for UKG completion, and the core idea is to embed entities and relations with the structure and confidence information in the UKG, which is called normal relational learning [22]. Besides, few-shot relational learning further consider the long-tail distribution of relations in modeling real-world UKGs.

Normal Relational Learning for UKG. UKGE [3] is a classic method in this field, which uses the scoring function of DistMult [25] to model triple confidences and solves the false negative problem with probabilistic soft logic. PASSLEAF [5] extends UKGE for other types of scoring functions, and also uses semi-supervised learning to alleviate the false negative problem. BEUrRE [2] applies box embedding for UKG embedding learning, in which entities are represented as boxes, relationships are modeled as affine transformations of head and tail entity boxes, and triple confidences are modeled by the intersection between transformed boxes. UPGAT [20] incorporates subgraph features and generalizes graph attention network for UKG completion. UKGsE [26] treats each triple as a short sentence and learns the confidence using LSTM. GTransE [12] and FocusE [16] associate triple confidences with margin operations in the loss functions, which only solve the task of link prediction for UKGs. UKRM [4] tries to mine rules using transformer to link prediction and leverage pre-trained language model to compute triple confidences.

Few-Shot Relational Learning for UKG. Recently, few-shot UKG completion has attracted much attention, e.g., GMUC [28] and GMUC+ [23] apply metric learning to learn UKG embeddings and achieve good performance, but the few-shot problem is not the focus of this paper. unKR [22] is a UKG embedding learning and completion tool which re-implements both works of few-shot relational learning and normal relational learning for UKG.

There also exist some works (e.g., [6]) regarding reasoning for query-answering on UKG, but the focus is not UKG completion, so we will not compare our method with such works. Existing relational learning methods for UKG completion neglect the fact that the distribution of triple confidences is extremely imbalanced in most UKGs, which causes the performance of UKG completion is still unsatisfactory. Our proposed method ssCDL aims to solve this problem by reinforcing the UKG embedding learning process using both labeled data and unlabeled data.

3 Preliminaries

3.1 Problem Definition

Definition 1. Uncertain Knowledge Graph. *An uncertain knowledge graph is a repository of factual knowledge denoted as a set of quadruples $\mathcal{G} = \{(h, r, t, s) | h, t \in \mathcal{E}, r \in \mathcal{R}, s \in [0, 1]\}$, where \mathcal{E} and \mathcal{R} are respectively the sets of entities and relations, and s is the confidence score measuring the triple uncertainty, which represents the likelihood of the triple (h, r, t) being true.*

Definition 2. Uncertain Knowledge Graph Completion. *Uncertain knowledge graph completion has two sub-tasks, which are confidence prediction and link prediction. Given a query $(h, r, t, ?)$ where (h, r, t) is a factual triple, confidence prediction is to estimate the missing triple confidence. Given another query $(h', r', ?)$ where h' is a head entity and r' is a relation, link prediction is to predict the missing tail entity.*

3.2 Confidence Distribution Learning

Confidence distribution learning (CDL) aims to learn a model which can accurately estimate the confidence distribution of each given triple. CDL is a variant of label distribution learning [8] (LDL) applied in UKG completion. LDL is a machine learning paradigm that not only predicts the labels relevant to instances, but also quantifies the degree of relevance of each label. Before CDL, triple

confidences are transformed into confidence distributions to capture the supervision information of few or unseen confidences in the labeled data.

Confidence distribution is defined as a discrete distribution in this paper, and this is also the setting of LDL [8]. In this way, since the confidence interval is $[0, 1]$, we directly set the confidence labels at a granularity of $\frac{1}{n}$, and the ordered confidence label set is $\{0, \frac{1}{n}, \frac{2}{n}, \dots, \frac{n-1}{n}, 1\}$. For a given quadruple (h, r, t, s) in a UKG, we define the confidence distribution of (h, r, t) as $s = \langle s^i \rangle \in \mathbb{R}^{n+1}$, where $\sum_{i=0}^n s^i = 1$ and $s^i \in [0, 1]$ is a description degree that the confidence label $\frac{i}{n}$ describe the triple (h, r, t) . In this paper, the confidence distribution s is generated by a Gaussian distribution $\mathcal{N}(s, \sigma^2)$, where σ is the standard deviation, and the confidence s is the mean, which causes that s has the highest description degree. Thus, in CDL, each piece of knowledge can be represented as a quadruple (h, r, t, s) , where $s \sim \mathcal{N}(s, \sigma^2)$ is the confidence distribution. Here, we empirically set n as 100, i.e., we have 101 confidence labels in total.

4 Methodology

4.1 Overview

Figure 2(a) provides the overview of our semi-supervised confidence distribution learning method ssCDL, which consists of two key components: CDL-based relational learner (CDL-RL) and pseudo confidence distribution generator (PCDG). At first, we apply the strategy mentioned in Section 3.2 to transform all triple confidences in the labeled data into confidence distributions. We utilize CDL-RL to learn UKG embeddings with labeled data and pseudo labeled data generated by PCDG. PCDG generates high-quality pseudo confidences labels for unlabeled data, and CDL-RL iteratively exploits these pseudo labeled data and further improves its performance. CDL-RL and PCDG have the same structure (Figure 2(b)), but their training processes are different. CDL-RL is optimized by minimizing the losses on confidence prediction and link prediction with labeled data and pseudo labeled data, while PCDG evaluates the performance of CDL-RL after exploiting pseudo labeled data generated by PCDG and takes it as the meta learning objective. PCDG is meta optimized by CDL-RL with labeled data. ssCDL is learned by meta self-training with iteratively training CDL-RL and PCDG.

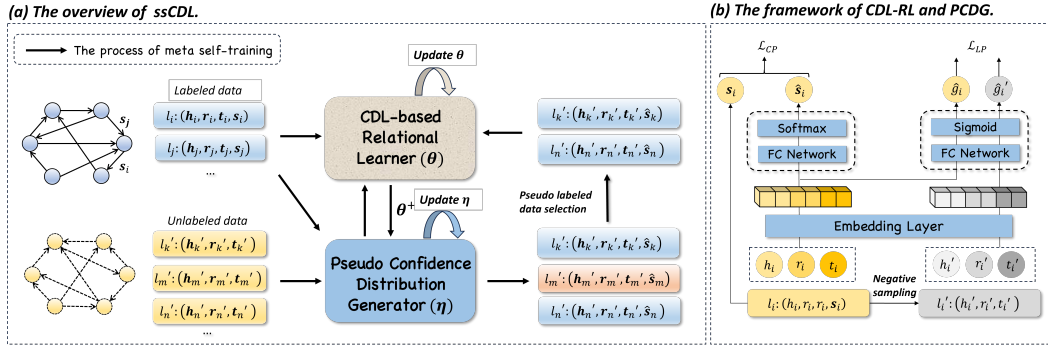


Figure 2: (a) The overview of ssCDL; (b) The framework of CDL-RL and PCDG.

4.2 CDL-based Relational Learner

The main purpose of CDL-RL is to learn the embeddings of entities and relations. Given the i -th quadruple (h_i, r_i, t_i, s_i) in the labeled data, it will be transformed into $l_i = (h_i, r_i, t_i, s_i)$ as the input of CDL-RL after mapping the confidence s_i to the confidence distribution s_i using a Gaussian distribution. As shown in Figure 2(b), CDL-RL is trained by minimizing the losses on confidence prediction and link prediction, i.e., \mathcal{L}_{CP} and \mathcal{L}_{LP} , and we explain the details as follows.

To compute the loss \mathcal{L}_{CP} on confidence prediction, we first concatenate the embeddings of h_i, r_i , and t_i and feed it into a two-layer fully connected network (FCN), which produces an $(n+1)$ -dimensional vector. We then apply the Softmax function as the activation function, and the predicted confidence distribution \hat{s}_i of the triple (h_i, r_i, t_i) can be computed as:

$$\hat{s}_i = \text{Softmax}(\text{FCN}_1(\mathbf{h}_i || \mathbf{r}_i || \mathbf{t}_i)) \quad (1)$$

where $||$ represents the concatenation between embeddings, and FCN_1 is a function that the FCN transforms the concatenated embedding into an $(n+1)$ -dimensional vector. Thus, the Kullback-

Leibler (KL) divergence can be utilized to measure the similarity between the predicted confidence distribution and the ground confidence distribution. Besides, we apply the loss function of the mean squared error (MSE) between the expectation $\mathbb{E}[\hat{s}_i]$ of predicted confidence distribution and the ground confidence s_i . Based on these, we formulate the learning objective of CDL-RL on confidence prediction as minimizing the KL divergence and MSE together, and define \mathcal{L}_{CP} as follows:

$$\mathcal{L}_{CP} = \underbrace{\sum_{\mathcal{D}} \sum_{a=0}^n s_i^a \ln \frac{s_i^a}{\hat{s}_i^a}}_{\text{KL divergence}} + \beta \underbrace{\sum_{\mathcal{D}} (\mathbb{E}[\hat{s}_i] - s_i)^2}_{\text{MSE}} \quad (2)$$

where s_i^a and \hat{s}_i^a are the a -th elements in s_i and \hat{s}_i respectively, \mathcal{D} denotes the training set, and $\beta \in [0, 1]$ is a hyper-parameter that controls the influence of the MSE.

To compute the loss \mathcal{L}_{LP} on link prediction, we first feed the concatenated embedding of h_i, r_i , and t_i into another two-layer FCN to compute the rank score of the triple (h_i, r_i, t_i) . This FCN outputs a single scalar normalized by the Sigmoid function as the rank score, which is computed as:

$$\hat{g}_i = \text{Sigmoid}(\text{FCN}_2(\mathbf{h}_i \parallel \mathbf{r}_i \parallel \mathbf{t}_i)) \quad (3)$$

Here, FCN_2 is a function that the FCN transforms the concatenated embedding into a scalar. We specifically design a margin-based ranking loss function to optimize CDL-RL on link prediction, and define \mathcal{L}_{LP} as follows:

$$\mathcal{L}_{LP} = \sum_{\mathcal{D}} \sum_{\mathcal{D}'} s_i [\gamma + \hat{g}'_i - \hat{g}_i]_+ \quad (4)$$

where \hat{g}'_i is the rank score of a negative sample which is generated by replacing either the head or tail entity of the given positive sample with a randomly chosen entity, the number of negative samples for each positive sample is empirically set as 50, \mathcal{D}' is the set of all negative samples, $[x]_+ = \max[0, x]$ denotes the positive part of x , and $\gamma > 0$ is a margin hyper-parameter.

To balance \mathcal{L}_{CP} and \mathcal{L}_{LP} in the training process, we use the idea of uncertainty weights [11] to dynamically adjust the proportion of loss for each task (we only have two tasks: confidence prediction and link prediction) during training. The final loss function of optimizing CDL-RL is defined as:

$$\mathcal{L}(\mathcal{D}, \theta) = \frac{1}{2\lambda_{CP}^2} \mathcal{L}_{CP} + \frac{\phi}{2\lambda_{LP}^2} \mathcal{L}_{LP} + \log(\lambda_{CP} \cdot \lambda_{LP}) \quad (5)$$

where λ_{CP} and λ_{LP} are observation noise parameters of confidence prediction and link prediction respectively, $\phi \in [0, 1]$ (empirically set as 0.1) is a weight limiting the influence of a relatively larger number of negative samples, which may cause \mathcal{L}_{LP} becoming too large, and θ represents the parameters of CDL-RL.

Besides the labeled data used for training CDL-RL, we apply PCDG (details will be given in Section 4.3) to generate pseudo labeled data, which are also utilized to reinforce the training of CDL-RL. Since most pseudo labeled data are the triples with low confidences, which has almost no impact on minimizing \mathcal{L}_{LP} , we only use such pseudo labeled data to minimize \mathcal{L}_{CP} to avoid ineffective computations. Thus, we re-define \mathcal{L}_{CP} as follows:

$$\mathcal{L}_{CP} = \sum_{\mathcal{D}} \sum_{a=0}^n s_i^a \ln \frac{s_i^a}{\hat{s}_i^a} + \beta \sum_{\mathcal{D}} (\mathbb{E}[\hat{s}_i] - s_i)^2 + w_p \left(\sum_{\mathcal{D}_p} \sum_{b=0}^n s_j^b \ln \frac{s_j^b}{\hat{s}_j^b} + \beta \sum_{\mathcal{D}_p} (\mathbb{E}[\hat{s}_j] - s_j)^2 \right) \quad (6)$$

where \mathcal{D}_p denotes the set of pseudo labeled data for training CDL-RL, s_j and \hat{s}_j respectively represent the pseudo confidence distribution and the predicted confidence distribution of the j -th pseudo labeled quadruple l_j generated by PCDG, and w_p is the weight of pseudo labeled data.

4.3 Pseudo Confidence Distribution Generator

Since we aim to make full use of unlabeled data to improve the quality of UKG embedding learning in CDL-RL, we propose PCDG, which is used for generating high-quality pseudo confidence distributions for unlabeled data. PCDG is based on the idea of meta-learning, which solves the gradual shift [15] of traditional self-training. At first, we perform negative sampling to get unlabeled

data (usually low-confidence triples) and feed them to PCDG. Given a positive sample in the labeled data, we generate one negative sample by replacing either the head or tail entity with a randomly chosen entity. Then, PCDG uses the most updated CDL-RL as the meta learning objective to evaluate the quality of pseudo confidence distributions by checking whether such data generated by PCDG can effectively improve CDL-RL.

In Figure 2(b), PCDG has the same structure as the CDL-RL, and we denote the parameters of the PCDG as η . In the training process, PCDG first generates pseudo confidence distributions for unlabeled data, and such pseudo labeled data contains gradient information of PCDG, which enables that it can be optimized. We denote these pseudo labeled data as \mathcal{D}_{tmp} . Note that although \mathcal{D}_{tmp} and \mathcal{D}_p are both generated by PCDG, they are different and so do their roles. \mathcal{D}_{tmp} is generated during the phase of optimizing PCDG, while \mathcal{D}_p is generated in the phase of training CDL-RL. Then, we minimize the loss on the labeled data (i.e., \mathcal{D}) after CDL-RL updates once (i.e., CDL-RL performs one gradient descent step on both \mathcal{D} and \mathcal{D}_{tmp}). The loss function of PCDG can be expressed as:

$$\mathcal{L}(\eta) = \mathcal{L}(\mathcal{D}, \theta^+) \quad (7)$$

where $\mathcal{L}(\mathcal{D}, \theta^+)$ (computed by Equation (5)) is the loss of CDL-RL on \mathcal{D} with θ^+ . Here, θ^+ represents the parameters of the CDL-RL after one gradient update as:

$$\theta^+ = \theta - \alpha \nabla_{\theta}(\mathcal{L}(\mathcal{D} \cup \mathcal{D}_{tmp}, \theta)) \quad (8)$$

where α is the learning rate, and $\mathcal{L}(\mathcal{D} \cup \mathcal{D}_{tmp}, \theta)$ is the loss of CDL-RL on \mathcal{D} and \mathcal{D}_{tmp} with θ .

4.4 Meta Self-training

In this subsection, we introduce the overall meta self-training process of ssCDL, which iteratively trains CDL-RL and PCDG. In the process of selecting pseudo labeled data from PCDG and inputting them into CDL-RL, we apply a simple yet effective strategy. In the labeled data, the original confidence of each triple should take the lead in the transformed confidence distribution. Thus, in the pseudo labeled data, if the highest description degree of a pseudo confidence label is larger than a fixed threshold, the corresponding pseudo confidence distribution is treated as high-quality and the pseudo labeled data will be selected for training CDL-RL, while others will be removed.

Algorithm 1 gives the details on meta self-training of ssCDL. We define two important time points, i.e., the epoch of starting training PCDG T_{PCDG} and the epoch of starting using pseudo labeled data for CDL-RL T_{CDLRL} . We first initialize the parameters θ and η of CDL-RL and PCDG respectively, and take the labeled data \mathcal{D} and sampled unlabeled data \mathcal{D}_u as the input of ssCDL. When the number of the current epoch N_{cur} is less than T_{PCDG} , we only need to optimize CDL-RL with \mathcal{D} (line 2-4). This setting tries to make UKG embeddings become stable in this period, which will help us train PCDG soon. When N_{cur} is larger than or equal to T_{PCDG} but less than T_{CDLRL} , it indicates that UKG embeddings have stabilized, and we start to train PCDG. PCDG first generates \mathcal{D}_{tmp} for unlabeled data, and feeds \mathcal{D}_{tmp} and \mathcal{D} together into the most updated CDL-RL, so we get the parameters θ^+ of CDL-RL updated for one more time (line 6-7). PCDG will utilize θ^+ to update itself with \mathcal{D} (line 8). In this period, we do not directly use PCDG to generate \mathcal{D}_p and feed it to CDL-RL because in the early stages of PCDG training, the quality of the generated labels cannot be guaranteed. Therefore, the training process of CDL-RL is the same as before (line 9). If N_{cur} is larger than or equal to T_{CDLRL} , we will exploit pseudo labeled data \mathcal{D}_p to help train CDL-RL (line 11-18). We use PCDG to generate \mathcal{D}_p and apply the pseudo label selection strategy, and the selected \mathcal{D}_p will be used together with \mathcal{D} to optimize CDL-RL, while the optimization of PCDG is the same as before (line 12-14). Afterwards, PCDG and CDL-RL will continuously repeat the above training process and optimize themselves until the maximum epoch is reached, i.e., PCDG generates pseudo labeled data for CDL-RL, and CDL-RL provides the meta learning objective for PCDG. This iterative training process is our complete meta self-training. During training, all parameters including embeddings, are updated using stochastic gradient descent (SGD) in each minibatch.

5 Experiments

In this section, we present experiments to show the effectiveness and superiority of ssCDL on the UKG completion tasks of confidence prediction and link prediction. We also analyze the effects of

Algorithm 1: Meta Self-Training of ssCDL

Input: Labeled data \mathcal{D} , unlabeled data \mathcal{D}_u , the parameters of CDL-RL θ , the parameters of PCDG η , the number of current epoch $N_{cur} = 1$, the number of maximum epoch T_{max} , the epoch of starting training PCDG T_{PCDG} , the epoch of starting using pseudo labeled data for CDL-RL T_{CDLRL} , the learning rate α .

```
1 while  $N_{cur} \leq T_{max}$  do
2   if  $N_{cur} < T_{PCDG}$  then
3      $\theta \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}(\mathcal{D}, \theta);$   $\triangleright$  Update CDL-RL with labeled data.
4   end
5   if  $T_{PCDG} \leq N_{cur} < T_{CDLRL}$  then
6      $\mathcal{D}_{tmp} = f_{\eta}(\mathcal{D}_u);$ 
7      $\theta^+ \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}(\mathcal{D} \cup \mathcal{D}_{tmp}, \theta);$ 
8      $\eta \leftarrow \eta - \alpha \nabla_{\eta} \mathcal{L}(\mathcal{D}, \theta^+);$   $\triangleright$  Meta update PCDG with labeled data.
9      $\theta \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}(\mathcal{D}, \theta);$   $\triangleright$  Update CDL-RL with labeled data.
10  end
11  if  $N_{cur} \geq T_{CDLRL}$  then
12     $\mathcal{D}_{tmp} = f_{\eta}(\mathcal{D}_u);$ 
13     $\theta^+ \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}(\mathcal{D} \cup \mathcal{D}_{tmp}, \theta);$ 
14     $\eta \leftarrow \eta - \alpha \nabla_{\eta} \mathcal{L}(\mathcal{D}, \theta^+);$   $\triangleright$  Meta update PCDG with labeled data.
15     $\mathcal{D}_p = f_{\eta}(\mathcal{D}_u);$ 
16     $\mathcal{D}_p = \text{Select}(\mathcal{D}_p);$   $\triangleright$  Pseudo labeled data selection.
17     $\theta \leftarrow \theta - \alpha \nabla_{\theta} \mathcal{L}(\mathcal{D} \cup \mathcal{D}_p, \theta);$   $\triangleright$  Update CDL-RL with labeled data and pseudo labeled data.
18  end
19   $N_{cur} ++;$ 
20 end
```

CDL and meta self-training in ssCDL with ablation experiments, and further explore the performance of ssCDL on predicting the confidences of low-confidence triples. We also investigate the sensitivity of three key hyper-parameters of ssCDL, which is shown in Appendix C. The source code of ssCDL is publicly available at: https://github.com/seucoin/unKR/tree/main/unKR_ssCDL.

Datasets. We conducted experiments on two widely used UKG datasets, i.e., NL27k extracted from NELL and CN15k extracted from ConceptNet (more details are in Table 1). We followed the setting of UKGE [3] to partition the dataset into 85% for training, 7% for validation, and 8% for testing.

Baselines. We compared ssCDL with the state-of-art embedding-based UKG completion methods, including UKGE [3], PASSLEAF [5], UKGsE [26], BEURRE [2], and UPGAT [20], in confidence prediction and link prediction. More details of such baselines are given in Appendix A.

5.1 Experimental Setup

Implementation Details. ssCDL was implemented by Pytorch-lightning, and all the experiments were conducted on an RTX3090 GPU card. The optimal hyper-parameters of ssCDL on NL27k are as follows: the standard deviation of Gaussian distribution $\sigma = 0.6$, the weight of pseudo labeled data $w_p = 0.7$, the MSE weight $\beta = 1$, the learning rate $\alpha = 0.001$, the margin $\gamma = 0.1$, the threshold for pseudo labeled data selection: 0.03, the batch size: 4096, and the embedding size: 128. The optimal hyper-parameters of ssCDL on CN15k are as follows: the standard deviation of Gaussian distribution $\sigma = 0.6$, the weight of pseudo labeled data $w_p = 0.3$, the MSE weight $\beta = 1$, the learning rate $\alpha = 0.001$, the margin $\gamma = 0.1$, the threshold for pseudo labeled data selection: 0.015, the batch size: 4096, and the embedding size: 512. We conducted sensitivity analysis on σ , w_p , and the threshold for

Table 1: The statistics of UKG Datasets.

Dataset	#Entities	#Relations	#Quadruples
NL27k	27,221	404	175,412
CN15k	15,000	36	241,158

pseudo labeled data selection, and more details are given in Appendix C. The experimental results of all baselines refer to the implementation of unKR [22], and more details are given in Appendix B.

Evaluation Protocol. We evaluated ssCDL and baselines on the tasks of confidence prediction and link prediction with the following evaluation metrics. For confidence prediction, we selected **Mean Squared Error (MSE)**: the mean squared error of predicted confidences and ground confidences, and **Mean Absolute Error (MAE)**: the mean absolute error of predicted confidences and ground confidences. For link prediction, we chose **Hits@1**: the proportion of ranks equal to one for all tail entities, and **Weighted Mean Reciprocal Rank (WMRR)**: the weighted average multiplicative inverse of the ranks for all tail entities.

5.2 Confidence Prediction

We compared ssCDL with all baselines on CN15k and NL27k in the task of confidence prediction. As shown in Table 2, ssCDL outperforms all baselines in both MSE and MAE. Compared with the best baseline on NL27k, ssCDL reduces MSE and MAE by 52.6% and 17.6%, respectively. Compared with the best baseline on CN15k, ssCDL reduces MSE and MAE by 63.8% and 53.2%, respectively. It demonstrates that ssCDL can effectively capture the semantics, structure, and confidence information in UKGs, enabling more accurate prediction of triple confidences.

We noticed that all methods’ performance on NL27k are better than that on CN15k. This can be attributed to the triple confidences in ConceptNet are determined solely by its data sources and the frequency with which they are mentioned. While different data sources may be of different quality, most triples in ConceptNet are generally correct. As a result, there is no significant distinction in reliability between high-confidence and low-confidence triples in ConceptNet. This situation may cause that the performance of all methods on CN15K is not that good, so it is necessary to build better UKG completion benchmark datasets.

Table 2: The comparison results between ssCDL and baselines on NL27k and CN15k for confidence prediction and link prediction. The best results are indicated by bold numbers, while the runner-up results are indicated by underlined numbers (CP: confidence prediction, LP: link prediction).

Dataset	NL27k				CN15k			
Task	CP		LP		CP		LP	
Metric	MSE	MAE	WMRR	Hits@1	MSE	MAE	WMRR	Hits@1
UKGE _{logi}	0.029	0.060	0.593	0.462	0.246	0.409	0.118	0.072
UKGE _{rect}	0.033	0.071	0.580	0.452	0.202	0.364	0.127	0.060
BEUrRE	0.089	0.222	0.272	0.117	0.117	0.283	0.138	0.039
PASSLEAF _{DistMult}	0.023	<u>0.051</u>	0.676	0.553	0.216	0.379	0.170	0.078
PASSLEAF _{CompLex}	0.024	<u>0.052</u>	0.708	<u>0.586</u>	0.231	0.400	<u>0.196</u>	<u>0.086</u>
PASSLEAF _{RotatE}	0.019	0.063	<u>0.715</u>	<u>0.580</u>	<u>0.094</u>	<u>0.248</u>	<u>0.137</u>	<u>0.037</u>
UKGsE	0.122	0.271	0.064	0.031	0.103	0.256	0.012	0.002
UPGAT	0.029	0.101	0.658	0.530	0.149	0.308	0.165	0.078
ssCDL	0.009	0.042	0.727	0.636	0.034	0.116	0.207	0.133

5.3 Link Prediction

Table 2 also presents the comparison results on link prediction, and ssCDL achieves the best Hits@1 and WMRR, which demonstrates that ssCDL is capable of predicting more accurate tail entities for different queries. Compared with confidence prediction, ssCDL does not have a quite significant improvement on link prediction compared with baselines. This is because our designed CDL and meta self-training are used to optimize confidence prediction to get better UKG embedding, and link prediction only benefits from such embeddings besides minimizing the margin-based ranking loss. This relatively implicit optimization may cause the improvement in link prediction is not as significant as that in confidence prediction. However, ssCDL still outperforms all baselines on link prediction, indicating that the training of ssCDL is an effective multi-task learning process.

5.4 Ablation Study

To investigate the contributions of CDL and meta self-training in ssCDL, we conducted ablation study on both datasets, and the results are given in Table 3.

Ablation on Confidence Distribution Learning. To evaluate the effectiveness of our CDL strategy, we did not use confidence distributions of triples, but only utilized their ground confidences for training (denoted as w/o cdl). This variant shows a significant performance decline on both datasets, which confirms the effectiveness of CDL. This also indicates that confidence distributions enable ssCDL to better utilize the supervision information of different confidences in the training data and reinforce the embedding learning process.

Ablation on Meta Self-Training. To verify the effectiveness of meta self-training, we removed PCDG, and only applied labeled data to train CDL-RL (denoted as w/o mst). This variant also exhibits performance decline, demonstrating that the pseudo labeled data generated by PCDG do assist ssCDL in learning better UKG embeddings. Besides, w/o mst outperforms w/o cdl, illustrating that the supervision information of the labeled data has a more important influence on the entire learning process compared to the potential information of the unlabeled data.

Table 3: The ablation study of ssCDL on confidence prediction and link prediction. The best results are highlighted in bold. w/o cdl refers to removing confidence distribution learning, and w/o mst refers to removing meta self-training (CP: confidence prediction, LP: link prediction).

Dataset	NL27k				CN15k			
Task	CP		LP		CP		LP	
Metric	MSE	MAE	WMRR	Hits@1	MSE	MAE	WMRR	Hits@1
w/o cdl	0.015	0.057	0.586	0.482	0.044	0.141	0.149	0.090
w/o mst	0.010	0.045	0.718	0.619	0.035	0.118	0.200	0.128
ssCDL	0.009	0.042	0.727	0.636	0.034	0.116	0.207	0.133

5.5 Low-Confidence Triples Analysis

As mentioned before, ssCDL is designed to reinforce the UKG embedding learning process under the imbalanced confidence distribution. Since nearly 80% of the triples in CN15k and NL27k have the confidence higher than 0.5, we conducted the experiments on predicting the confidences of actual low-confidence triples to demonstrate that ssCDL can alleviate the problem of imbalanced confidence distribution in confidence prediction. Since low-confidence triples are less reliable, it is unnecessary to perform link prediction. We randomly selected the triples with the confidences lower than 0.5 from the test set of each dataset, and used MAE to evaluate the performance on confidence prediction. The comparison results, as illustrated in Figure 3, reveal that most baselines are not good at handling confidence prediction on low-confidence triples, while ssCDL still demonstrates the robustness and outperforms all baselines on both datasets. This fully reflects the effectiveness of ssCDL in predicting the confidences of low-confidence triples.

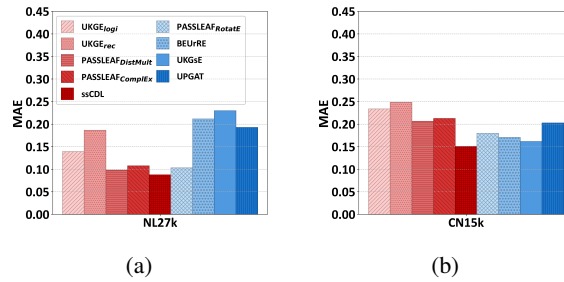


Figure 3: The comparison results of confidence prediction on low-confidence triples in NL27k and CN15k.

6 Conclusion

In this paper, we propose a new semi-supervised confidence distribution learning method ssCDL for UKG completion. ssCDL is composed of CDL-based relational learner and pseudo confidence

distribution generator, which are iteratively trained by meta self-training. Such a semi-supervised learning framework and the introduction of triple confidence distributions benefit to solve the problem of extremely imbalanced distributions of triple confidences in UKG embedding learning for UKG completion. Experimental results demonstrate that ssCDL has the best performance on real-world UKG datasets compared with the state-of-the-art baselines. The effectiveness of different strategies used in ssCDL has been verified in the ablation study.

As for the future work, we plan to study rule learning and reasoning on UKGs based on confidence distribution learning. We will also explore to apply large language model to UKG completion, and use UKGs for reliable retrieval-augmented generation.

7 Acknowledgements

This work is supported by the NSFC (Grant No. 62376058, 52378009, 62576093, U23B2057, 62176185, 62476058), the Southeast University Interdisciplinary Research Program for Young Scholars, and the Big Data Computing Center of Southeast University.

References

- [1] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam Hruschka, and Tom Mitchell. Toward an Architecture for Never-Ending Language Learning. In *Proc. of AAAI*, volume 24, pages 1306–1313, 2010.
- [2] Xuelu Chen, Michael Boratko, Muhao Chen, Shib Sankar Dasgupta, Xiang Lorraine Li, and Andrew McCallum. Probabilistic Box Embeddings for Uncertain Knowledge Graph Reasoning. In *Proc. of NAACL*, pages 882–893, 2021.
- [3] Xuelu Chen, Muhao Chen, Weijia Shi, Yizhou Sun, and Carlo Zaniolo. Embedding Uncertain Knowledge Graphs. In *Proc. of AAAI*, volume 33, pages 3363–3370, 2019.
- [4] Yilin Chen, Tianxing Wu, Yunchang Liu, Yuxiang Wang, and Guilin Qi. Uncertain Knowledge Graph Completion with Rule Mining. In *Proc. of WISA*, pages 100–112, 2024.
- [5] Zhu-Mu Chen, Mi-Yen Yeh, and Tei-Wei Kuo. PASSLEAF: A Pool-bAsed Semi-Supervised LEARNING Framework for Uncertain Knowledge Graph Embedding. In *Proc. of AAAI*, volume 35, pages 4019–4026, 2021.
- [6] Weizhi Fei, Zihao Wang, Hang Yin, Yang Duan, Hanghang Tong, and Yangqiu Song. Soft reasoning on uncertain knowledge graphs. *arXiv preprint arXiv:2403.01508*, 2024.
- [7] Sainyam Galhotra and Udayan Khurana. Semantic Search over Structured Data. In *Proc. of CIKM*, pages 3381–3384, 2020.
- [8] Xin Geng. Label Distribution Learning. *IEEE Transactions on Knowledge and Data Engineering*, 28:1734–1748, 2016.
- [9] Xin Geng, Chao Yin, and Zhi-Hua Zhou. Facial Age Estimation by Learning from Label Distributions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35:2401–2412, 2013.
- [10] Lei Ji, Yujing Wang, Botian Shi, Dawei Zhang, Zhongyuan Wang, and Jun Yan. Microsoft concept graph: Mining semantic concepts for short text understanding. *Data Intelligence*, 1(3):238–270, 2019.
- [11] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proc. of CVPR*, pages 7482–7491, 2018.
- [12] Natthawut Kertkeidkachorn, Xin Liu, and Ryutaro Ichise. GTransE: Generalizing Translation-Based Model on Uncertain Knowledge Graph Embedding. In *Proc. of JSAT*, pages 170–178, 2020.
- [13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. of ICLR*, 2015.

- [14] Yunshi Lan, Gaole He, Jinhao Jiang, Jing Jiang, Wayne Xin Zhao, and Ji-Rong Wen. A Survey on Complex Knowledge Base Question Answering: Methods, Challenges and Solutions. In *Proc. of IJCAI*, pages 4483–4491, 2021.
- [15] Shikun Liu, Andrew Davison, and Edward Johns. Self-Supervised Generalisation with Meta Auxiliary Learning. In *Proc. of Neurips*, 2019.
- [16] Sumit Pai and Luca Costabello. Learning embeddings from knowledge graphs with numeric edge attributes. In *Proc. of IJCAI*, pages 2869–2875, 2021.
- [17] Robyn Speer, Joshua Chin, and Catherine Havasi. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. In *Proc. of AAAI*, volume 31, 2017.
- [18] Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *Proc. of ICLR*, 2019.
- [19] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex Embeddings for Simple Link Prediction. In *Proc. of ICML*, pages 2071–2080, 2016.
- [20] Yen-Ching Tseng, Zu-Mu Chen, Mi-Yen Yeh, and Shou-De Lin. UPGAT: Uncertainty-Aware Pseudo-neighbor Augmented Knowledge Graph Attention Network. In *Proc. of PAKDD*, pages 53–65, 2023.
- [21] Jing Wang, Shuo Zhang, and Runzhi Li. Gate feature interaction network for relation prediction in knowledge graph. *Data Intell.*, 6(3):749–770, 2024.
- [22] Jingting Wang, Tianxing Wu, Shilin Chen, Yunchang Liu, Shutong Zhu, Wei Li, Jingyi Xu, and Guilin Qi. unKR: A Python Library for Uncertain Knowledge Graph Reasoning by Representation Learning. In *Proc. of SIGIR*, pages 2822–2826, 2024.
- [23] Jingting Wang, Tianxing Wu, and Jiatao Zhang. Incorporating Uncertainty of Entities and Relations into Few-Shot Uncertain Knowledge Graph Embedding. In *Proc. of CCKS*, pages 16–28, 2022.
- [24] Wentao Wu, Hongsong Li, Haixun Wang, and Kenny Q Zhu. Probase: A Probabilistic Taxonomy for Text Understanding. In *Proc. of SIGMOD*, pages 481–492, 2012.
- [25] Bishan Yang, Scott Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases. In *Proc. of ICLR*, 2015.
- [26] Shihan Yang, Weiya Zhang, Rui Tang, Mingkai Zhang, and Zhensheng Huang. Approximate inferring with confidence predicting based on uncertain knowledge graph embedding. *Information Sciences*, 609:679–690, 2022.
- [27] Mingshan You, Jiao Yin, Hua Wang, Jinli Cao, Kate Wang, Yuan Miao, and Elisa Bertino. A knowledge graph empowered online learning framework for access control decision-making. *World Wide Web*, 26(2):827–848, 2023.
- [28] Jiatao Zhang, Tianxing Wu, and Guilin Qi. Gaussian Metric Learning for Few-Shot Uncertain Knowledge Graph Completion. In *Proc. of DASFAA*, pages 256–271, 2021.

A Baselines

We compared ssCDL with the state-of-art embedding-based UKG completion methods published in recent years, including:

- **UKGE** [3] is a classic UKG completion method, which is the first work on relational learning on UKG and it has two variants: UKGE_{logi} using the logistic function as the mapping function, and UKGE_{rect} which takes a bounded rectifier as the mapping function.
- **PASSLEAF** [5] improves the generalization ability of UKGE and applies semi-supervised learning for the first time in UKG completion to solve the false negative problem. Now it is the best UKG completion method. PASSLEAF_{DistMult}, PASSLEAF_{Complex}, and PASSLEAF_{RotatE} are three variants of PASSLEAF, which use the scoring functions of DistMult [25], ComplEx [19] and RotatE [18], respectively.
- **UKGsE** [26] treats each knowledge fact as a short sentence, and is a typical model of UKG completion leveraging a pre-trained language model.
- **BEUrRE** [2] is a representative UKG completion model based on box embedding. The geometry of boxes endows the model with calibrated probabilistic semantics and facilitates the incorporation of relational constraints.
- **UPGAT** [20] generalizes the graph attention network and uses it to capture the local structural information in UKG completion. It is a typical UKG completion method modeling structural contexts in UKG.

B Implementation Details.

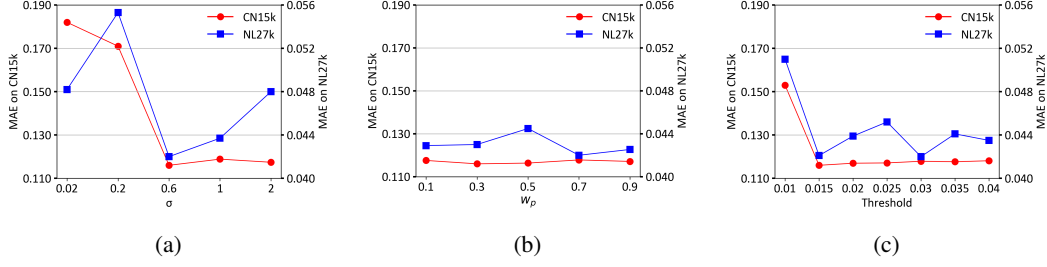
We used Adam optimizer [13] for SGD training. For hyper-parameter tuning, we searched the best hyper-parameter from the following settings: the batch size $\in \{512, 1024, 2048, 4096\}$, the embedding size $\in \{128, 256, 512\}$, the margin $\gamma \in \{0.01, 0.05, 0.1, 0.5, 1\}$, and the MSE weight $\beta \in \{0.6, 0.8, 1\}$. We conducted detailed sensitivity analysis on the following parameters, including the standard deviation of Gaussian distribution $\sigma \in \{0.02, 0.2, 0.6, 1, 2\}$, the weight of pseudo labeled data $w_p \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$, and the threshold for pseudo labeled data selection $\in \{0.01, 0.015, 0.02, 0.025, 0.03, 0.035, 0.04\}$.

The search results indicate that the optimal hyper-parameters of ssCDL on NI27k are as follows: the standard deviation of Gaussian distribution $\sigma = 0.6$, the weight of pseudo labeled data $w_p = 0.7$, the MSE weight $\beta = 1$, the learning rate $\alpha = 0.001$, the margin $\gamma = 0.1$, the threshold for pseudo labeled data selection: 0.03, the batch size: 4096, and the embedding size: 128. The optimal hyper-parameters of ssCDL on CN15k are as follows: the standard deviation of Gaussian distribution $\sigma = 0.6$, the weight of pseudo labeled data $w_p = 0.3$, the MSE weight $\beta = 1$, the learning rate $\alpha = 0.001$, the margin $\gamma = 0.1$, the threshold for pseudo labeled data selection: 0.015, the batch size: 4096, and the embedding size: 512. We evaluated ssCDL on the validation set every ten epochs. The maximum epochs on NI27k and CN15k were empirically set to 500 and 300, respectively.

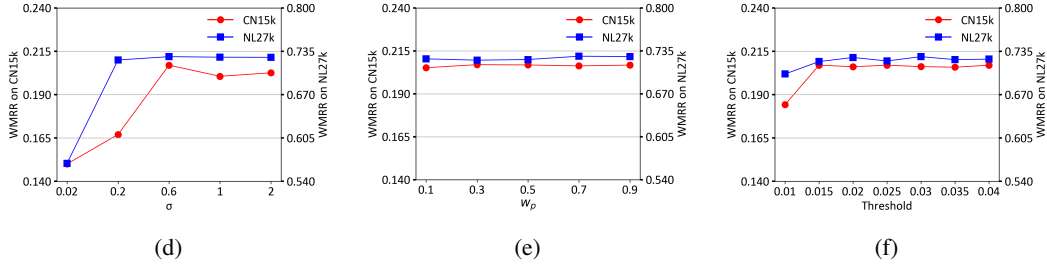
C Parameter Sensitivity

We conducted sensitivity analysis on three hyper-parameters, including the standard deviation of Gaussian distribution σ , the weight of pseudo labeled data w_p , and the threshold for pseudo labeled data selection. We utilized MAE and WMRR to evaluate the performance on confidence prediction and link prediction, respectively. Figure 4 shows the impacts of different hyper-parameters of ssCDL for confidence prediction and link prediction on both datasets.

Impact of σ . As shown in Figure 4(a) and 4(d), ssCDL achieves the optimal performance, then the effect deteriorates, as the value of σ increases on both datasets. When the value of σ is relatively small, the high description degree is mainly distributed in the confidence near the ground confidence, preventing the method from capturing the global features of UKGs, which leads to an inaccurate estimation of the confidence distribution. However, as σ increases, the differences of confidence distributions generated by different ground confidences become small, which gradually weakens the ability of distinguishing between different confidences. ssCDL achieves the best results on both datasets when $\sigma = 0.6$.



The impacts of hyper-parameters on confidence prediction. (a) (b) (c) show the impacts of σ , w_p , and the threshold for pseudo labeled data selection on NL27k and CN15k respectively.



The impacts of hyper-parameters on link prediction. (d) (e) (f) show the impacts of σ , w_p , and the threshold for pseudo labeled data selection on NL27k and CN15k respectively.

Figure 4: Sensitivity experiments on NL27k and CN15k.

Impact of w_p . For the hyper-parameter w_p , Figure 4(b) and 4(e) show that the performance of ssCDL improves at first and then decreases with the growing of w_p , which indicates that a reasonable balance between labeled data and pseudo labeled data is necessary to achieve the optimal performance of ssCDL. Our experimental findings indicate that 0.7 is the optimal w_p on NL27k and 0.3 is the optimal w_p on CN15k.

Impact of the threshold for pseudo labeled data selection. As shown in Figure 4(c) and 4(f), with the increase of the threshold, the performance of ssCDL also exhibits a trend of first increasing and then decreasing. A quite low threshold will cause that the pseudo label selection strategy to select noisy data, while a quite high threshold will limit the use of high-quality pseudo labeled data. ssCDL achieves the best results when the threshold is set as 0.03 on NL27k and 0.015 on CN15k, respectively.

D Limitation

Our current study has conducted experiments on CN15k. Although CN15k is a classic benchmark for UKG completion, since the most triples in ConceptNet are generally correct, there is no significant distinction in reliability between high-confidence and low-confidence triples. We will build better UKG completion benchmark datasets in the future.