Supervised Relational Learning with Selective Neighbor Entities for Few-Shot Knowledge Graph Completion

Jiewen Hou[1], Tianxing Wu[1,2(✉)], Jingting Wang[1], Shuang Wang[1,2], and Guilin Qi[1,2]

[1] Southeast University, Nanjing, China
{jiewenhou,jtwang,shuangwang,gqi}@seu.edu.cn
[2] Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications, Southeast University, Ministry of Education, Nanjing, China
tianxingwu@seu.edu.cn

Abstract. Knowledge graphs have powerful reasoning capabilities, but they suffer from incompleteness and long-tail distributions of relations. Few-shot knowledge graph (KG) completion aims to address these issues by completing missing triplets for few-shot relations which only have limited existing triplets. Existing methods attempt to learn few-shot relation embeddings by utilizing the head and tail entity embeddings within the same triplets. Such entity embeddings are enhanced from their respective neighborhoods, but these methods fail to select crucial neighbor entities relevant to the relations. In this paper, to solve this problem, we propose a new Supervised Relational Learning (SuperRL) model with selective neighbor entities for few-shot KG completion. In SuperRL, we first enhance head and tail entity embeddings based on a cascaded embedding enhancement network with different neighbor entity encoders, which can select crucial neighbor entities for few-shot relations from different perspectives. We then jointly perform dual contrastive learning and metric learning to provide different supervision signals for relational learning. Extensive experiments on benchmark datasets have substantiated the superiority of SuperRL in different evaluation metrics over the state-of-the art baselines. The source code is publicly available at: https://github.com/seucoin/SuperRL.

1 Introduction

Knowledge graph (KG) consists of a set of factual triplets, where each one denoted as (h, r, t) expresses the relation r between a head entity h and a tail entity t. Large-scale knowledge graphs (KGs) [18,23,24] offer potent reasoning capabilities for various intelligent applications [5,20], but KGs are incomplete since new knowledge is always emerging over time. This inspires the exploration of KG completion, which seeks to predict the absent elements within incomplete triplets. Thus, various relational learning models [19] have been proposed to learn embeddings of entities and relations to perform KG completion. However, real-world KGs suffer from the long-tail problem, where a significant proportion of relations only have a limited number of triplets. Consequently, the performance of traditional embedding based KG completion methods

G. Demartini et al. (Eds.): ISWC 2024, LNCS 15231, pp. 144–161, 2025.
https://doi.org/10.1007/978-3-031-77844-5_8

degrades significantly when completing missing triplets for few-shot relations which only have limited existing triplets.

To address this challenge, several relational learning methods for few-shot KG completion have been proposed, including metric-based methods (e.g., GMatching [25] and FAAN [15]) and meta-learner-based methods (e.g., MetaR [2], GANA [14], and HiRe [22]). These methods focus on completing missing triplets for few-shot relations, and the core task is to predict the tail entity t for each query triplet $(h, r, ?)$ only given K reference triplets about the relation r. Metric-based methods complete missing triplets by calculating similarities with few-shot relation embeddings generated from the reference triplets, while meta-learner-based methods achieve this by transferring meta-information from the derived few-shot relation embeddings. As the embeddings of few-shot relations are initially unknown, most existing methods use the head and tail entity embeddings enhanced from their respective neighborhoods to compose entity pair embeddings, which are further utilized to generate few-shot relation embeddings. However, they overlook two types of neighbor entities crucial for representing few-shot relations, resulting in the embeddings results insufficient to high-quality few-shot KG completion.

In this study, we propose a new **Super**vised **R**elational **L**earning (**SuperRL**) model, which can select crucial neighbor entities for few-shot KG completion. We categorize such crucial neighbor entities into two groups: directly relevant entities and indirectly relevant entities. **Directly relevant entities** consist of neighbor entities from two different levels:

- **Triplet-level relevant entities** are the neighbor entities of the head entity (or the tail entity) relevant to the tail entity (or the head entity). As illustrated in Fig. 1.i, the neighbor entity `United States` is geographically relevant to the tail entity `Washington DC` and the neighbor entity `White House` holds geographical relevance to the head entity `Joe Biden`. These triplet-level relevant entities are beneficial to represent the current relation `Geo-political location` between `Joe Biden` and `Washington DC`.
- **Context-level relevant entities** are the neighbor entities of the head entity (or the tail entity) relevant to the neighbor entities of the tail entity (or the head entity). As shown in Fig. 1.ii, the neighbor entity `Southeast Asia` of the head entity `Coconut` is geographically relevant to the neighbor entity `Southeast Asian countries` of the tail entity `Malaysia`. The context-level relevant entities are important to represent the relation `Agriculture from country`.

Indirectly relevant entities are introduced based on their N-order relevance to the directly relevant entities:

- **One-order indirectly relevant entities** are the neighbor entities relevant to directly relevant entities. As shown in Fig. 1.iii, the neighbor entity `Democratic Party`, is politically relevant to the `United States` which is a directly relevant entity to the tail entity `Washington DC`. This one-order indirectly relevant entity emphasizes the political attribute of the `Geo-political location` relation, which is crucial for representing the `Geo-political location` relation.

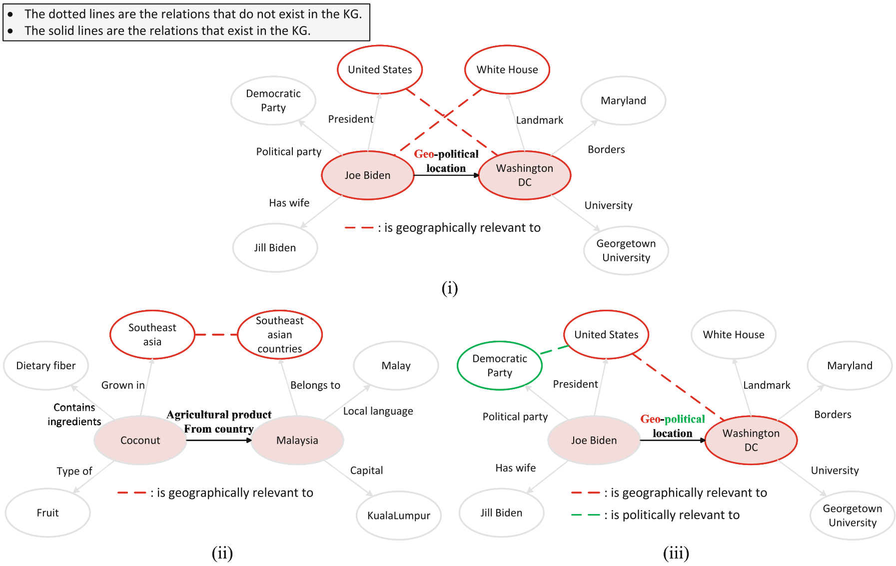

Fig. 1. An illustration of directly relevant entities and indirectly relevant entities in a KG. (i) **Triplet-level relevant entities**: the neighbor entity `United States` is geographically relevant to the tail entity `Washington DC`, and the neighbor entity `White House` is geographically relevant to the head entity `Joe Biden`. (ii) **Context-level relevant entities**: the neighbor entity `Southeast Asia` is geographically relevant to the neighbor entity `Southeast Asian countries`. (iii) **One-order indirectly relevant entities**: The neighbor entity `Democratic Party` is politically relevant to the directly relevant entity `United States`.

– ***N*-order indirectly relevant entities** are the neighbor entities relevant to $(N-1)$-order indirectly relevant entities.

In our proposed SuperRL, a cascaded embedding enhancement network is firstly established to generate entity pair embeddings enriched with the information of both directly and indirectly relevant entities for few-shot relations. To achieve this, we design two neighbor entity encoders within the network to enhance embeddings of the head and tail entities leveraging different-level directly relevant entities. Besides, such neighbor entity encoders are utilized to construct multiple enhancement layers, which are connected in a cascading manner to successively incorporate the information of indirectly relevant entities. We then apply dual contrastive learning and metric learning with entity pair embeddings to provide different supervision signals for relational learning. In dual contrastive learning, we aim to improve the consistency of entity pair embeddings enhanced by different neighbor entity encoders. In metric learning, supervision signals are introduced by calculating similarities between entity pair embeddings of the given queries and few-shot relation embeddings derived from existing triplets. Extensive experiments on two benchmark datasets demonstrate that SuperRL significantly outperforms the state-of-the-art baselines in different evaluation metrics.

In summary, this paper has the following contributions:

- We identify two types of neighbor entities (i.e., directly relevant entities and indirectly relevant entities) crucial for representing few-shot relations. They play a vital role in relational learning for few-shot KG completion.
- We propose a new supervised relational learning model (i.e., SuperRL) for few-shot KG completion. SuperRL applies a cascaded embedding enhancement network to enhance head and tail entity embeddings by encoding directly and indirectly relevant entities for few-shot relations. Dual contrastive learning and metric learning are jointly utilized to provide different supervision signals for relational learning.
- We conduct extensive experiments on two benchmark datasets for few-shot KG completion and the results show the effectiveness and superiority of SuperRL compared with baseline methods.

2 Related Work

2.1 Relational Learning for KG Completion

The key of traditional relational learning methods [19] for KG completion is to learn embeddings of entities and relations. KG embeddings preserve structural information, which can be used for KG completion. The pioneering model TransE [1] introduces the concept of using relation r as a translation for learning embeddings, expressed as $\boldsymbol{h} + \boldsymbol{r} \approx \boldsymbol{t}$ for the triplet (h, r, t). A scoring function is then employed to assess the translation quality and facilitate learning in a unified embedding space. Subsequent models, such as TransH [21] and TransR [12], incorporate relation-specific information to enhance embedding learning. ComplEx [17] learns KG embeddings in a complex space, using bilinear transformations to model relations between entities. RotatE [16] interprets the relation as a rotation between the head entity and the tail entity in a complex space. ConvE [4] and ConvKB [13] utilize convolution operators to augment entity/relation embedding learning. However, these methods require a substantial number of triplets for each relation in the training process, and their performance significantly degrades in few-shot settings where only a limited number of triplets are available for each relation. Therefore, a series of few-shot KG completion methods have been proposed to solve this problem (see Sect. 2.2 for details).

2.2 Few-Shot KG Completion

Existing few-shot KG completion methods can be categorized into two primary groups: (1) Metric-based methods: The first work in formulating few-shot KG completion is GMatching [25]. GMatching comprises a neighbor encoder that enhances entity pair embeddings from their one-hop neighbors to represent few-shot relations, and a matching processor comparing similarities between the relation representations acquired from the query and the reference set, respectively. FSRL [27] extends the setting to more shots and investigates how to integrate relation representations acquired from multiple reference triplets. FAAN [15] proposes a one-hop neighbor aggregator with dynamic attention mechanism to enhance entity pair embeddings. CIAN [11] further incorporates a cross-attention module to facilitate interactions between neighbor entities. (2) Meta-learner-based methods: MetaR [2] learns to transfer meta information of few-shot relations with the MAML training strategy [6], and generates few-shot relation embeddings by averaging the embeddings of all reference triplets. GANA [14] places increased emphasis on neighboring information and introduces a gated and attentive neighbor aggregator accordingly. HiRe [22] attempts to learn relation embeddings by jointly capturing three-level relational information.

Since few-shot KG completion lacks sufficient triplet instances, existing methods have utilized entity neighborhoods to introduce more contextual information, thereby improving the completion performance. However, the major limitation of these methods is that they neglect relevant neighbor entities crucial for learning generalizable few-shot relation embeddings, which may lower the quality of the learnt embeddings for few-shot KG completion, and we aim to solve this problem in this paper.

3 Problem Formulation

In this section, we formally define the problem of few-shot KG completion and its relevant concepts.

Definition 1 (Knowledge Graph). *A knowledge graph is a repository of factual knowledge denoted as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ represents the set of entities, $\mathcal{R}$ stands for the set of relations (including high-frequency relations and few-shot relations), and $\mathcal{T}$ denotes the set of relational triplets defined as $\mathcal{T} = \{(h, r, t) \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}\}$.*

Definition 2 (Neighbor Entity). *Given a triplet (h, r, t) of a few-shot relation r, the set of one-hop neighbor entities of the h and t is defined as $\mathcal{N}_h$ and $\mathcal{N}_t$, respectively. Each neighbor entity in $\mathcal{N}_h$ (or $\mathcal{N}_t$) is directly linked to the head entity h (or the tail entity t).*

Definition 3 (Few-Shot Knowledge Graph Completion). *Few-shot knowledge graph completion is to predict the missing tail entity for each query $(h, r, ?)$ of a relation $r \in \mathcal{R}$ only given its reference set $\mathcal{S}_r = \{(h_i, r, t_i)|(h_i, r, t_i) \in \mathcal{G}\}$, and $|\mathcal{S}_r|$ is a small number.*

For few-shot KG completion, we divide few-shot relations into three disjoint sets: $\mathcal{R}_{train}$, $\mathcal{R}_{valid}$, and $\mathcal{R}_{test}$ for training, validating, and testing, respectively. The training process is to train our model to accomplish K-shot KG completion for each few-shot relation $r \in \mathcal{R}_{train}$, where the tasks can be denoted as $\mathcal{T}_{train} = \{(\mathcal{S}_r, \mathcal{Q}_r)|r \in \mathcal{R}_{train}\}$. $\mathcal{S}_r = \{(h_i, r, t_i)\}_{i=1}^{K}$ and $\mathcal{Q}_r = \{(h_i, r, t_i)\}_{i=1}^{|\mathcal{Q}_r|}$ are the reference set and query set of r, respectively. In the process of validating and testing, our model performance is evaluated on $\mathcal{T}_{valid} = \{(\mathcal{S}_{r'}, \mathcal{Q}_{r'})|r' \in \mathcal{R}_{valid}\}$ and $\mathcal{T}_{test} = \{(\mathcal{S}_{r''}, \mathcal{Q}_{r''})|r'' \in \mathcal{R}_{test}\}$, respectively.

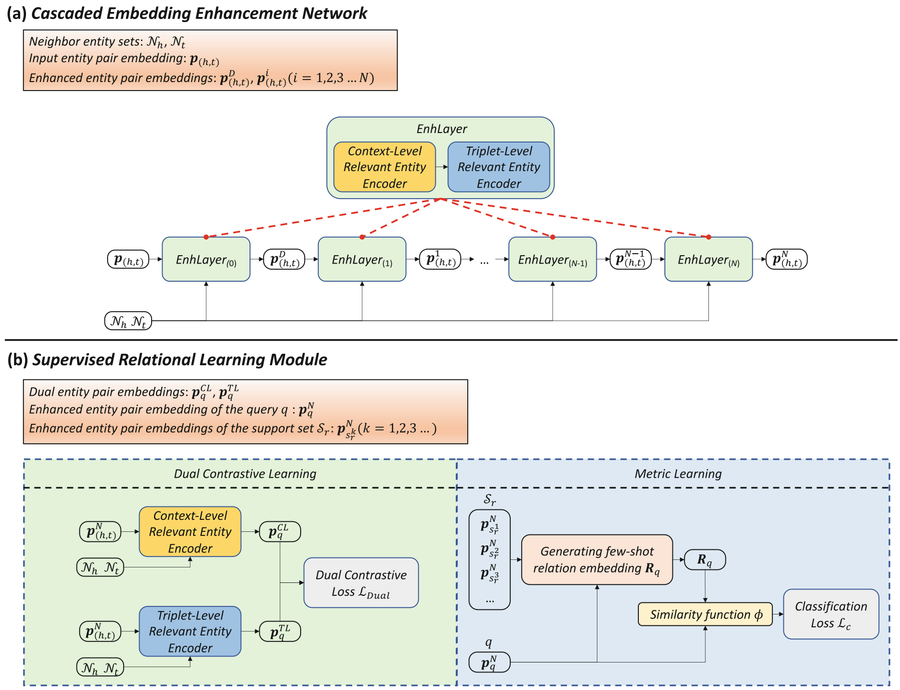

Fig. 2. The overview of SuperRL. **(a) The cascaded embedding enhancement network**, which uses neighbor entity encoders to capture the information of directly relevant entities and utilizes multiple enhancement layers to further incorporate the information of indirectly relevant entities. **(b) The supervised relational learning module**, which applies dual contrastive learning and metric learning to provide different supervision signals.

4 Methodology

In this section, we introduce SuperRL, our supervised relational learning model with selective neighbor entities for few-shot KG completion. Figure 2 provides an overview of SuperRL, which consists of two key components: (1) **Cascaded Embedding Enhancement Network**. Multiple enhancement layers consisting of different neighbor entity encoders are used for generating entity pair embeddings enriched with the information of both directly relevant entities and indirectly relevant entities for few-shot relations. (2) **Supervised Relational Learning Module**. Dual contrastive learning and metric learning provide different supervision signals with entity pair embeddings for relational learning.

4.1 Cascaded Embedding Enhancement Network

In this subsection, we introduce our designed neighbor entity encoders and the enhancement layers composed of such encoders in the proposed cascaded embedding enhancement network.

Neighbor Entity Encoders. We design two different neighbor entity encoders, which are context-level relevant entity encoder and triplet-level relevant entity encoder, and they use a cross-attention mechanism to identify context-level relevant entities and triplet-level relevant entities for head and tail entities, respectively. Specifically, given a triplet (h, r, t), the above neighbor entity encoders take the entity pair embedding $\boldsymbol{p}_{(h,t)}$ (the concatenation of $\boldsymbol{h}$ and $\boldsymbol{t}$), and the corresponding neighbor entity sets $\mathcal{N}_h = \{e_{h_i} | (h, r_{h_i}, e_{h_i}) \in \mathcal{G}\}$ and $\mathcal{N}_t = \{e_{t_j} | (t, r_{t_j}, e_{t_j}) \in \mathcal{G}\}$ as the input. Context-level relevant entity encoder and triplet-level relevant entity encoder generate entity pair embeddings $\boldsymbol{p}_{(h,t)}^{CL}$ and $\boldsymbol{p}_{(h,t)}^{TL}$ as the output, respectively. Such generated entity pair embeddings are actually enriched with the information of context-level relevant entities and triplet-level relevant entities, respectively.

We first generate neighbor embeddings for the neighbor entities by introducing relation information. Taking h as the target, we generate the neighbor embedding $\boldsymbol{n}_{h_i}$ for the neighbor entity e_{h_i} by integrating its embedding $\boldsymbol{e}_{h_i}$ with the relation embedding $\boldsymbol{r}_{h_i}$ as follows:

$$\boldsymbol{n}_{h_i} = ReLU([\boldsymbol{r}_{h_i} \| \boldsymbol{e}_{h_i}] W_n + b_n), \tag{1}$$

where $\|$ means concatenation, $W_n \in \mathbb{R}^{2d \times d}$ and $b_n \in \mathbb{R}^d$ are learnable parameters shared by all neighbor entities, and d is the embedding dimension. Similarly, we perform the above procedures on the neighbor entity e_{t_j} of the tail entity t and obtain its neighbor embedding $\boldsymbol{n}_{t_j}$.

(1) Context-Level Relevant Entity Encoder. The context-level relevant entity encoder utilizes context-level relevant entities to enhance the embeddings of the head entity h and the tail entity t. We apply a cross-attention mechanism to identify context-level relevant entities based on embedding similarities, and generate context-level neighbor representation for enhancement. Taking the head entity h as the target, the cross-attention mechanism takes $\mathcal{N}_h$ and $\mathcal{N}_t$ as the input and outputs the context-level neighbor representation $\overline{\boldsymbol{h}}_{CL}$ for h. Specifically, for each $e_{h_i} \in \mathcal{N}_h$, we transform the embedding of each $e_{t_j} \in \mathcal{N}_t$ into the query $\boldsymbol{q}_{t_j}^{CL}$, the neighbor entity embedding $\boldsymbol{e}_{h_i}$ into the key $\boldsymbol{k}_{h_i}^{CL}$, and the neighbor embedding $\boldsymbol{n}_{h_i}$ into the value $\boldsymbol{v}_{h_i}^{CL}$ with three linear transformations, $W_{CL}^Q, W_{CL}^K, W_{CL}^V \in \mathbb{R}^{d \times d}$, respectively, as follows:

$$\boldsymbol{q}_{t_j}^{CL} = \boldsymbol{e}_{t_j} W_{CL}^Q, \;\; \boldsymbol{k}_{h_i}^{CL} = \boldsymbol{e}_{h_i} W_{CL}^K, \;\; \boldsymbol{v}_{h_i}^{CL} = \boldsymbol{n}_{h_i} W_{CL}^V. \tag{2}$$

These linear transformations increase the learnable parameter space, enabling the encoder to learn optimal weights to generate few-shot relation embeddings. Then, we compute the embedding similarity $\mathrm{Att}_{e_{h_i}}^{CL}$ of e_{h_i} with respect to $\mathcal{N}_t$ as:

$$\mathrm{Att}_{e_{h_i}}^{CL} = \frac{\sum_{e_{t_j} \in \mathcal{N}_t} \boldsymbol{q}_{t_j}^{CL} W_{CL} {\boldsymbol{k}_{h_i}^{CL}}^T}{|\mathcal{N}_t|}, \tag{3}$$

where $W_{CL} \in \mathbb{R}^{d \times d}$ is a trainable weight matrix interpreting the relation between the neighbor entity e_{t_j} and the neighbor entity e_{h_i}, measuring context-level entity relevance. Afterwards, we aggregate neighbor embeddings of context-level relevant entities, generating the context-level neighbor representation $\overline{\boldsymbol{h}}_{CL}$ for h as follows:

$$\overline{\boldsymbol{h}}_{CL} = \sum_{e_{h_i} \in \mathcal{N}_h} \beta_{h_i} \boldsymbol{v}_{h_i}^{CL}, \quad \beta_{h_i} = \frac{exp(\text{Att}_{e_{h_i}}^{CL})}{\sum_{e_{h_k} \in \mathcal{N}_h} exp(\text{Att}_{e_{h_k}}^{CL})}, \tag{4}$$

where β_{h_i} denotes the attention weight of the neighbor entity e_{h_i}. A higher value of β_{h_i} indicates a stronger relevance between e_{h_i} and $\mathcal{N}_t$, thus capturing a more valuable context-level relevant entity. Correspondingly, for the tail entity t, the cross-attention mechanism takes $\mathcal{N}_t$ and $\mathcal{N}_h$ as the input, then generates the context-level neighbor representation $\overline{\boldsymbol{t}}_{CL}$ for t. Finally, we use the context-level neighbor representations $\overline{\boldsymbol{h}}_{CL}$ and $\overline{\boldsymbol{t}}_{CL}$ as well as the original input entity pair embedding $\boldsymbol{p}_{(h,t)}$ to compose the enhanced entity pair embedding $\boldsymbol{p}_{(h,t)}^{CL}$ as follows:

$$\boldsymbol{p}_{(h,t)}^{CL} = ReLU([\overline{\boldsymbol{h}}_{CL} \| \overline{\boldsymbol{t}}_{CL}] W_{E_1} + b_{E_1}) W_{E_2} + \boldsymbol{p}_{(h,t)}. \tag{5}$$

where $W_{E_1}, W_{E_2} \in \mathbb{R}^{2d \times 2d}$, $b_{E_1} \in \mathbb{R}^{2d}$ are trainable matrices.

(2) Triplet-Level Relevant Entity Encoder. The triplet-level relevant entity encoder utilizes triplet-level relevant entities to enhance the embeddings of the head entity h and tail entity t. Similarly, we apply a cross-attention mechanism to identify the triplet-level relevant entities based on embedding similarities, and generate triplet-level neighbor representation for enhancement. Taking the head entity h as the target, the cross-attention mechanism takes $\mathcal{N}_h$ and $\boldsymbol{t}$ as the input, and outputs the triplet-level neighbor representation $\overline{\boldsymbol{h}}_{TL}$ for h. Specifically, for each $e_{h_i} \in \mathcal{N}_h$, we transform the tail entity embedding $\boldsymbol{t}$ into the query $\boldsymbol{q}_t^{TL}$, the neighbor entity embedding $\boldsymbol{e}_{h_i}$ into the key $\boldsymbol{k}_{h_i}^{TL}$, and its corresponding neighbor embedding $\boldsymbol{n}_{h_i}$ into the value $\boldsymbol{v}_{h_i}^{TL}$ with three linear transformations, $W_{TL}^Q, W_{TL}^K, W_{TL}^V \in \mathbb{R}^{d \times d}$, respectively, as follows:

$$\boldsymbol{q}_t^{TL} = \boldsymbol{e}_t W_{TL}^Q, \quad \boldsymbol{k}_{h_i}^{TL} = \boldsymbol{e}_{h_i} W_{TL}^K, \quad \boldsymbol{v}_{h_i}^{TL} = \boldsymbol{n}_{h_i} W_{TL}^V. \tag{6}$$

These linear transformations increase the parameter space in the same way used in the context-level neighbor entity encoder. Then we compute the embedding similarity $\text{Att}_{e_{h_i}}^{TL}$ of e_{h_i} with respect to t as:

$$\text{Att}_{e_{h_i}}^{TL} = \boldsymbol{q}_t^{TL} W_{TL} \boldsymbol{k}_{h_i}^{TL^T}, \tag{7}$$

where $W_{TL} \in \mathbb{R}^{d \times d}$ is a trainable weight matrix that measures the triplet-level entity relevance between e_{h_i} and t. Then we compute triplet-level neighbor representation $\overline{\boldsymbol{h}}_{TL}$ as follows:

$$\overline{\boldsymbol{h}}_{TL} = \sum_{e_{h_i} \in \mathcal{N}_h} \alpha_{h_i} \boldsymbol{v}_{h_i}^{TL}, \quad \alpha_{h_i} = \frac{exp(\text{Att}_{e_{h_i}}^{TL})}{\sum_{e_{h_j} \in \mathcal{N}_h} exp(\text{Att}_{e_{h_j}}^{TL})}, \tag{8}$$

where α_{h_i} denotes the attention weight of the neighbor entity e_{h_i}. A higher value of α_{h_i} means a stronger relevance between the i-th neighbor entity of h and the tail entity t, thus capturing a more valuable triplet-level relevant entity. Correspondingly, for the tail entity t, the cross-attention mechanism takes $\mathcal{N}_t$ and $\boldsymbol{h}$ as the input and computes the triplet-level neighbor representation $\bar{\boldsymbol{t}}_{TL}$ for t. Finally, we use the triplet-level neighbor representations $\bar{\boldsymbol{h}}_{TL}$ and $\bar{\boldsymbol{t}}_{TL}$ as well as the original input entity pair embedding $\boldsymbol{p}_{(h,t)}$ to compose the enhanced entity pair embedding $\boldsymbol{p}_{(h,t)}^{TL}$ as follows:

$$\boldsymbol{p}_{(h,t)}^{TL} = ReLU([\bar{\boldsymbol{h}}_{TL} \| \bar{\boldsymbol{t}}_{TL}]W_{E_3} + b_{E_2})W_{E_4} + \boldsymbol{p}_{(h,t)}. \tag{9}$$

where $W_{E_3}, W_{E_4} \in \mathbb{R}^{2d \times 2d}$ and $b_{E_2} \in \mathbb{R}^{2d}$ are all trainable matrices.

Enhancement Layers. The aforementioned neighbor entity encoders actually utilize directly relevant entities to enrich the embeddings of the head entity h and the tail entity t in each given triplet (h, r, t). To fully extract the information of directly relevant entities, we choose both the triplet-level and context-level relevant entity encoders to build the enhancement layer. Since indirectly relevant entities can only be introduced on the basis of directly relevant entities, we connect multiple enhancement layers in a cascading manner to further enhance $\boldsymbol{h}$ and $\boldsymbol{t}$ with the information of indirectly relevant entities, from one order to N orders successively.

The i-th enhancement layer, denoted as $EnhLayer_{(i)}$, where a context-level relevant entity encoder and a triplet-level relevant entity encoder are connected sequentially. This is the optimal connection order of the two encoders according to the experimental results, and more details are discussed in Sect. 5.4. Generally, $EnhLayer_{(0)}$ enhances $\boldsymbol{h}$ and $\boldsymbol{t}$ with the information of directly relevant entities, and it takes $\boldsymbol{p}_{(h,t)}$, $\mathcal{N}_h$, and $\mathcal{N}_t$ as the input. The output is the enhanced entity pair embedding $\boldsymbol{p}_{(h,t)}^{D}$. Furthermore, the output of $EnhLayer_{(0)}$ also serves as the input of $EnhLayer_{(1)}$, which takes $\boldsymbol{p}_{(h,t)}^{D}$, $\mathcal{N}_h$, and $\mathcal{N}_t$ as the input and generates the entity pair embedding $\boldsymbol{p}_{(h,t)}^{1}$ further enriched with the information of one-order indirectly relevant entities. Similarly, the input of the N-th enhancement layer includes the output of $(N-1)$-th enhancement layer $\boldsymbol{p}_{(h,t)}^{N-1}$, $\mathcal{N}_h$, and $\mathcal{N}_t$, and the output is the enhanced entity pair embedding $\boldsymbol{p}_{(h,t)}^{N}$ denoted as:

$$\boldsymbol{p}_{(h,t)}^{N} = EnhLayer_{(N)}(\boldsymbol{p}_{(h,t)}^{N-1}, \mathcal{N}_h, \mathcal{N}_t). \tag{10}$$

In summary, with an $N+1$ layers cascaded embedding enhancement network, we generate the entity pair embedding $\boldsymbol{p}_{(h,t)}^{N}$ enriched with the information of both directly relevant entities and indirectly relevant entities from one to N orders for the subsequent supervised relational learning.

4.2 Supervised Relational Learning Module

To perform supervised relational learning, we first construct the reference set $\mathcal{S}_r$ and the query set $\mathcal{Q}_r$ as follows. Given a few-shot relation $r \in \mathcal{T}_{train}$, we randomly sample K triplets to construct the K-shot reference set $\mathcal{S}_r = \{s_r^k\}_{k=1}^{K}$ where $s_r^k \equiv (h_k, r, t_k) \in \mathcal{S}_r$, and a batch of triplets to construct the query set $\mathcal{Q}_r = \{(h_i, r, t_i)\}_{i=1}^{|\mathcal{Q}_r|}$. For each triplet in $\mathcal{Q}_r$, we replace its tail entity to construct a set of negative query triplets $\mathcal{Q}_r^- = \{(h_q, r, t_q^-) \mid (h_q, r, t_q) \in \mathcal{Q}_r\}$, where $t_q^- \in \{\mathcal{C}_{h_q,r} \setminus t_q\}$. $\mathcal{C}_{h_q,r}$ is the candidate entity set generated from the KG $\mathcal{G}$. We then generate entity pair embeddings for all triplets in $\mathcal{S}_r$, $\mathcal{Q}_r$ and $\mathcal{Q}_r^-$, using the cascaded embedding enhancement network. Finally, we jointly perform dual contrastive learning and metric learning in the supervised relational learning module with these entity pair embeddings for relational learning. Dual contrastive learning is designed to maximize the consistency of entity pair embeddings enhanced with different encoders, since contrastive learning [3] can achieve high-quality embeddings in the encoding process by maximizing the consistency among positive samples, and is often used in the design of various encoders. Besides, we choose metric learning to optimize entity and relation embeddings, because it is widely used to solve the few-shot problem in machine learning, which classifies unseen instances according to their distances to few seen instances in an embedding space learned by deep learning, also achieving the state-of-the-art performance [10].

Dual Contrastive Learning. We introduce dual contrastive learning to improve the consistency of entity pair embeddings using different neighbor entity encoders. This process inherently maximizes the agreement between context-level neighbor representations and triplet-level neighbor representations for the same triplet after multi-layer enhancement. More specifically, for each query $q = (h, r, t) \in \mathcal{Q}_r$, we first enhance the output of cascaded embedding enhancement network $\boldsymbol{p}_{(h,t)}^N$, utilizing a triplet-level relevant entity encoder and a context-level relevant entity encoder, respectively. Then we obtain dual entity pair embeddings $\boldsymbol{p}_q^{TL}$ and $\boldsymbol{p}_q^{CL}$, which are enriched with triplet-level and context-level neighbor representations, respectively. Similarly, we generate dual entity pair embeddings $\boldsymbol{p}_{q^-}^{TL}$ and $\boldsymbol{p}_{q^-}^{CL}$ for each negative query q^-. Based on these, we define two contrastive loss terms for dual contrastive learning, aiming to maximize the consistency of dual entity pair embeddings for positive queries and encourage differences in dual entity pair embeddings between the positive and negative queries as:

$$\begin{aligned} \mathcal{L}_{TL} &= \frac{1}{|Q_r|} \sum_{q \in \mathcal{Q}_r} -log \frac{exp(sim(\boldsymbol{p}_q^{TL}, \boldsymbol{p}_q^{CL})/\tau)}{\sum_i exp(sim(\boldsymbol{p}_q^{TL}, \boldsymbol{p}_{q_i^-}^{CL})/\tau)}, \\ \mathcal{L}_{CL} &= \frac{1}{|Q_r|} \sum_{q \in \mathcal{Q}_r} -log \frac{exp(sim(\boldsymbol{p}_q^{CL}, \boldsymbol{p}_q^{TL})/\tau)}{\sum_i exp(sim(\boldsymbol{p}_q^{CL}, \boldsymbol{p}_{q_i^-}^{TL})/\tau)}, \end{aligned} \tag{11}$$

where τ denotes the temperature parameter, and the dual contrastive loss is a combination of the above two contrastive loss terms as follows:

$$\mathcal{L}_{Dual} = \mathcal{L}_{TL} + \mathcal{L}_{CL}. \tag{12}$$

Metric Learning. Inspired by [15], we adopt metric learning to complete missing triplets by calculating similarities between entity pair embeddings of the given queries and few-shot relation embeddings derived from the reference set. Specifically, given a query $q = (h, r, t) \in \mathcal{Q}_r$ and the reference set $\mathcal{S}_r = \{s_r^k\}_{k=1}^K$, we first use the cascaded embedding enhancement network to generate entity pair embeddings $\boldsymbol{p}_q^N$ (i.e., $\boldsymbol{p}_{(h,t)}^N$) and $\boldsymbol{p}_{s_r^k}^N$ for q and each $s_r^k \in \mathcal{S}_r$, respectively. We then compute the embedding similarity γ_k between $\boldsymbol{p}_q^N$ and $\boldsymbol{p}_{s_r^k}^N$ as:

$$\gamma_k = \frac{exp(\boldsymbol{p}_q^N \odot \boldsymbol{p}_{s_r^k}^N)}{\sum_{s_r^k \in \mathcal{S}_r} exp(\boldsymbol{p}_q^N \odot \boldsymbol{p}_{s_r^k}^N)}, \tag{13}$$

where $\odot$ denotes the element-wise production. The few-shot relation embedding $\boldsymbol{R}_q$ for q is generated by aggregating all $\boldsymbol{p}_{s_r^k}^N$ based on γ_k as $\boldsymbol{R}_q = \sum_{s_r^k \in \mathcal{S}_r} \gamma_k \boldsymbol{p}_{s_r^k}^N$, which emphasizes the importance of queries during relational learning. For each negative query $q^- \in \mathcal{Q}_r^-$, the generation of the few-shot relation embedding $\boldsymbol{R}_{q^-}$ is the same as above. To predict the query triplet q based on the learned few-shot relation embedding, we propose the similarity score function ϕ between $\boldsymbol{p}_q^N$ and the corresponding few-shot relation embedding $\boldsymbol{R}_q$ as $\phi(q, \mathcal{S}_r) = \boldsymbol{p}_q^N \odot \boldsymbol{R}_q$.

Here, we define a classification loss with a margin to ensure that a positive query in $\mathcal{Q}_r$ and the reference set have a higher similarity score, as well as a negative query in $\mathcal{Q}_r^-$ and the reference set should have a lower similarity score as follows:

$$\mathcal{L}_{cls} = \sum_{q \in \mathcal{Q}_r} \sum_{q^- \in \mathcal{Q}_r^-} [\gamma - \phi(q, \mathcal{S}_r) + \phi(q^-, \mathcal{S}_r)]_+, \tag{14}$$

where $[x]_+$ is the hinge loss defined as $[x]_+ = \max(0, x)$, and $\gamma > 0$ is the margin.

Training Process. We optimize SuperRL by jointly considering the classification loss and the dual contrastive loss. Our optimization $\mathcal{L}$ is defined as follows:

$$\mathcal{L} = \mathcal{L}_{cls} + \lambda \mathcal{L}_{Dual}, \tag{15}$$

where λ is a hyperparameter that controls the influence of the dual contrastive loss term. We adopt a batch sampling based meta-training procedure [27] to minimize $\mathcal{L}$ for model optimization.

5 Experiments

In this section, we present comprehensive experiments to show the performance of SuperRL and the effectiveness of its essential components.

5.1 Datasets and Evaluation Metrics

We conducted experiments on two widely used few-shot KG completion datasets, Nell-One and Wiki-One, which are constructed by [25]. For fair comparison, we followed the experimental setup of GMatching [25], where relations associated with more than 50 but less than 500 triplets are chosen as few-shot KG completion tasks. For each target relation, we used the candidate set provided by Gmatching. The statistics of both datasets are provided in Table 1. We used 51/5/11 and 133/16/34 tasks for training/validation/testing on Nell-One and Wiki-One, respectively, following the common setting in the literature. We reported MRR (mean reciprocal rank) and Hits@n (n = 1,

5) on both datasets for the evaluation of performance. MRR is the mean reciprocal rank of the correct entities, and Hits@n is the ratio of correct entities that rank in top n. We compared our proposed model against other baseline methods under 3-shot and 5-shot settings, which are the most common settings in the literature.

5.2 Baselines

We compared our model with two categories of baselines: (1) The traditional relational learning models: including TransE [1], TransH [21], DisMult [26], ComplEx [17] and ComplEx-N3 [9]. We used OpenKE [7] to reproduce the results of these models with hyperparameters reported in the original papers. The models are trained using all triplets from relations in the background knowledge graph [25] and training relations, as well as relations from all reference sets. (2) The state-of-the-art few-shot KG completion models: GMatching [25], MetaR [2], FAAN [15], CIAN [11], HiRe [22] and FSRL [27]. For HiRe and FAAN, we directly reported results obtained from the original papers. For GMatching and MetaR, we reported the results provided by [15] and [11], respectively. As FSRL [27] was initially reported in different settings, where the candidate set is much smaller, we reported the results re-implemented by [15] under the same setting with other methods. As CIAN [11] used different training data with different experimental settings, we reproduced it on the same setting as other baselines. As reported in HiRe [22], the reproduced results of GANA [14] is less competitive, we left GANA out in our comparison. All reported results are produced on the same setting and compared using the same training, validation, and test datasets.

Table 1. Statistics of the datasets.

Dataset	#Entities	#Relations	#Triples	#Tasks
Nell-One	68,545	358	181,109	67
Wiki-One	4,838,244	822	5,859,240	183

5.3 Experimental Setup

For fair comparison, we used the entity and relation embeddings pretrained by TransE [1] on both datasets, released by GMatching [25], for the initialization of SuperRL. On both datasets, we set the maximum number of neighbors as 100, and the embedding dimension is set to 100 and 50 on Nell-One and Wiki-One, respectively. We set the temperature parameter τ as 0.1, and the margin of the classification loss γ is set to 5.0. We used Adam optimizer [8] as the optimizer with an initial learning rate of $8e^{-5}$ for Nell-One and $2e^{-4}$ for Wiki-One, respectively. For all experiments except the sensitivity test, the number of enhancement layers L is set to 2 and 8 on Nell-One and Wiki-One, respectively, while the hyperparameter λ is set to 0.06 and 0.09 for 3-shot and 5-shot on both datasets. We applied mini-batch gradient descent to train our model with a batch size of 512 on both datasets. We evaluated on validation set every 10k steps and chose the best model within 300k steps based on MRR. All models are implemented by PyTorch and trained on one V100 GPU card.

5.4 Comparison with Baselines

Table 2. Comparison against state-of-the-art methods on Nell-One and Wiki-One. The best results are indicated by bold numbers, while the runner-up results are indicated by underlined numbers for each metric. The table lacks certain metric results, which are not provided in original papers.

Methods	Nell-One MRR 3-shot	Nell-One MRR 5-shot	Nell-One Hits@5 3-shot	Nell-One Hits@5 5-shot	Nell-One Hits@1 3-shot	Nell-One Hits@1 5-shot	Wiki-One MRR 3-shot	Wiki-One MRR 5-shot	Wiki-One Hits@5 3-shot	Wiki-One Hits@5 5-shot	Wiki-One Hits@1 3-shot	Wiki-One Hits@1 5-shot
TransE	0.162	0.168	0.180	0.186	0.085	0.082	0.040	0.052	0.027	0.057	0.013	0.042
TransH	0.266	0.279	0.299	0.317	0.145	0.162	0.092	0.095	0.091	0.092	0.045	0.047
DisMult	0.201	0.214	0.223	0.246	0.146	0.140	0.052	0.077	0.041	0.078	0.020	0.035
ComplEx	0.228	0.239	0.252	0.253	0.165	0.176	0.064	0.070	0.053	0.063	0.029	0.030
ComplEx-N3	-	0.305	-	0.399	-	0.205	-	-	-	-	-	-
Gmatching	-	0.176	-	0.233	-	0.113	-	0.263	-	0.337	-	0.197
MetaR	0.210	0.209	0.311	0.280	0.119	0.141	0.317	0.323	0.379	0.385	0.261	0.270
FSRL	0.219	0.195	0.296	0.279	0.139	0.108	0.102	0.113	0.131	0.135	0.050	0.056
FAAN	0.247	0.279	0.309	0.364	0.183	0.200	0.298	0.341	0.368	0.395	0.228	0.281
CIAN	0.281	0.289	0.349	0.362	0.211	0.215	0.328	0.329	0.398	0.404	0.258	0.253
HiRe	-	0.306	-	0.439	-	0.207	-	0.371	-	0.419	-	0.319
SuperRL	**0.312**	**0.330**	**0.404**	**0.441**	**0.223**	**0.234**	**0.369**	**0.388**	**0.446**	**0.458**	**0.297**	**0.320**
SuperRL*	0.310	0.317	0.402	0.393	0.223	0.227	0.355	0.362	0.414	0.442	0.286	0.285

Table 2 compares SuperRL (the context-level relevant entity encoder and the triplet-level relevant entity encoder are connected sequentially in enhancement layers) and its variant SuperRL* (the triplet-level relevant entity encoder and the context-level relevant entity encoder are connected sequentially in enhancement layers) against the baselines on Nell-One and Wiki-One under 3-shot and 5-shot settings. In general, traditional KG completion methods are inferior to few-shot KG completion methods. This is because they are designed for the scenarios with sufficient training data. Overall, our SuperRL model outperforms all baseline methods under two settings on both datasets, validating its capability for few-shot KG completion, while SuperRL* is also competitive in comparisons, achieving runner-up results on most metrics. Specifically, SuperRL outperforms other methods significantly in terms of the Hits@1 metric on Nell-One, indicating that SuperRL is more inclined to rank correct triplets at the top position. This is because SuperRL introduces more precise information for few-shot KG completion by identifying directly and indirectly relevant entities. In real-world applications, Hits@1 is more crucial in KG completion, highlighting the superiority of SuperRL in practical scenarios. In addition, on the Wiki-One dataset, where the long-tail distribution of relations is more pronounced, SuperRL significantly improves Hits@5 compared with other baselines while maintaining the optimal Hits@1. This indicates that introducing directly and indirectly relevant entities can effectively tackle few-shot KG completion tasks in more challenging scenarios.

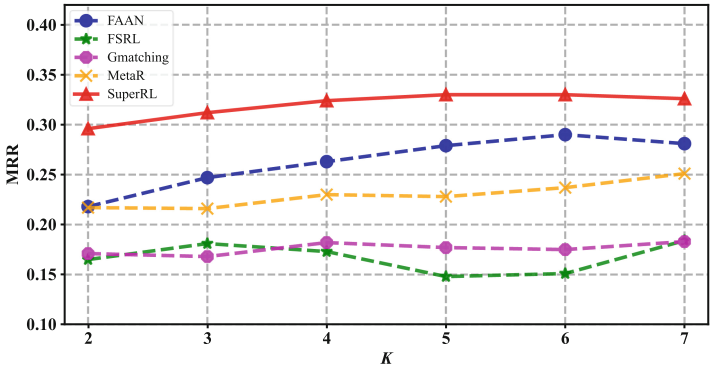

Fig. 3. Impact of few-shot size K on Nell-One.

We conducted experiments on SuperRL when varying the few-shot size K on Nell-One and Wiki-One. As other baselines lack experimental results on Wiki-One in original papers, we only present the comparison results on Nell-One in Fig. 3. Specifically, when the value of K is small, SuperRL significantly outperforms other baseline methods in terms of MRR. This is because directly and indirectly relevant entities are more qualified for few-shot KG completion when only a small number triplets are available. Furthermore, as the value of K increases, SuperRL gets relatively stable improvements compared to other baselines. This is attributed to our metric learning, which aggregates more crucial information for few-shot KG completion as the number of reference triplets increases.

5.5 Ablation Study

To validate the effectiveness of key components in SuperRL, we conducted ablation experiments on the cascaded embedding enhancement network and the supervised relational learning module on both datasets under 5-shot setting. The experimental results are shown in Table 3.

Ablation on Cascaded Embedding Enhancement Network. To verify the effectiveness of the cascaded embedding enhancement network and its different neighbor entity encoders, we removed the triplet-level relevant entity encoders, context-level relevant entity encoders, and the whole cascaded embedding enhancement network in SuperRL, respectively. After removing the context-level or triplet-level relevant entity encoders (i.e., w/o Con-Encoders or w/o Tri-Encoders), we observed a certain degree of performance decline on both datasets, revealing the effectiveness of different neighbor entity encoders. Specifically, on Wiki-One, the performance declines more significantly when removing context-level relevant entity encoders. This indicates that context-level relevant entities can effectively assist SuperRL in generating few-shot relation embeddings on Wiki-One. On Nell-One, we observed a more remarkable decline when removing triplet-level relevant entity encoders. This suggests that triplet-level relevant entities are more crucial for learning few-shot relation embeddings on Nell-One. Overall, SuperRL performs the worst compared to other variants on both datasets when removing the cascaded embedding enhancement network (w/o Enh-Network). This result highlights the significance of both directly and indirectly relevant entities, which play an essential role in relational learning for few-shot KG completion.

Ablation on Supervised Relational Learning Module. To verify the effectiveness of supervised relational learning in SuperRL, we removed dual contrastive learning (w/o $\mathcal{L}_{Dual}$) and metric learning (w/o $\mathcal{L}_{cls}$), respectively. We find that the performance of SuperRL declines significantly when removing metric learning. This demonstrates the necessity of metric learning, which generates few-shot relation embeddings from reference triplets to facilitate effective few-shot KG completion. After removing dual contrastive learning, the performance of SuperRL decreases to some extent. This verifies the role of dual contrastive learning, which can learn high-quality entity pair embeddings to further improve the performance of SuperRL.

Table 3. The ablation study on SuperRL under 5-shot link prediction. The best results are highlighted in bold, and the runner-up results are underlined for each metric.

Configuration	Nell-One			Wiki-One		
	MRR	Hits@5	Hits@1	MRR	Hits@5	Hits@1
SuperRL	**0.330**	**0.441**	**0.234**	**0.388**	**0.458**	**0.320**
SuperRL w/o Con-Encoders	0.303	0.375	<u>0.226</u>	0.338	0.438	0.248
SuperRL w/o Tri-Encoders	0.287	0.370	0.198	0.348	0.425	0.274
SuperRL w/o Enh-Network	0.125	0.141	0.069	0.180	0.226	0.111
SuperRL w/o $\mathcal{L}_{Dual}$	<u>0.306</u>	<u>0.408</u>	0.209	<u>0.367</u>	<u>0.452</u>	<u>0.288</u>
SuperRL w/o $\mathcal{L}_{cls}$	0.197	0.272	0.120	0.193	0.216	0.165

5.6 Parameter Sensitivity

We conducted sensitivity tests under 3/5-shot settings for the number of enhancement layers L and the hyperparameter λ on both datasets, which further reveals the impact

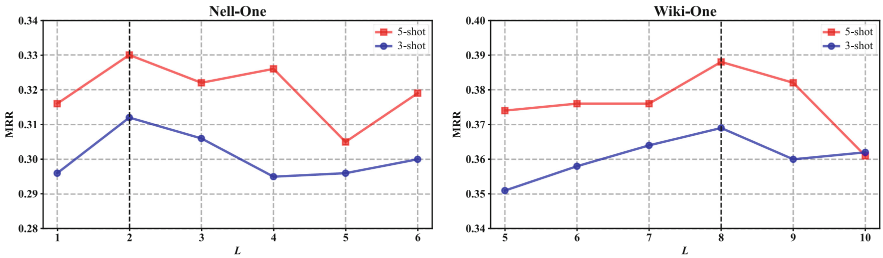

Fig. 4. Sensitivity experiments regarding the number of enhancement layers L on Nell-One and Wiki-One.

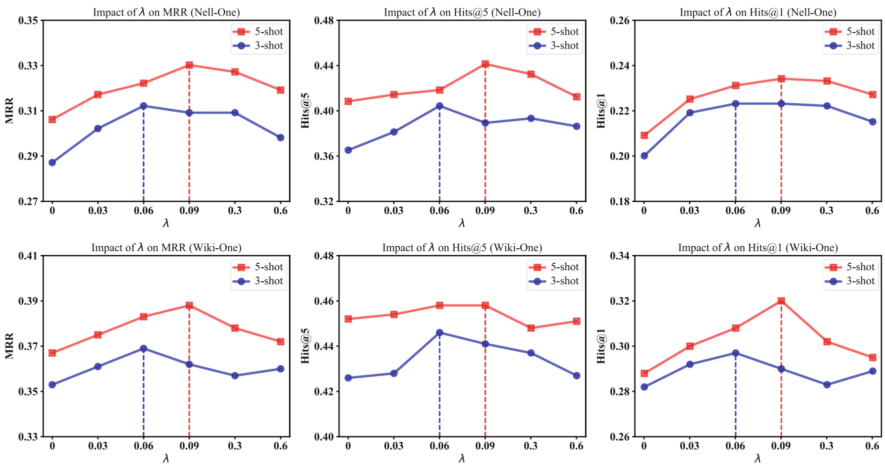

Fig. 5. Sensitivity experiments regarding the hyperparameter λ on Nell-One and Wiki-One.

of N-order indirectly relevant entities and dual contrastive learning in few-shot KG completion.

For the hyperparameter L, we performed sensitivity tests by varying L from 1 to 6 and 5 to 10 on Nell-One and Wiki-One, respectively. As shown in Fig. 4, the MRR of SuperRL reaches a peak and then declines, as the value of L increases on both datasets. The increase of MRR demonstrates the significance of indirectly relevant entities in few-shot KG completion when the order N is relatively small. However, indirectly relevant entities introduce irrelevant noise as N increases further. This reveals the duality of introducing indirectly relevant entities in few-shot KG completion. Besides, we find that the optimal L on Wiki-One is 8, which is higher than the optimal value $L = 2$ on Nell-One. One main reason is that, higher-order indirectly relevant entities may play a crucial role in few-shot KG completion on a more sparse dataset like Wiki-One.

For the hyperparameter λ, since the value of contrastive loss is significantly larger than that of the classification loss, λ should be small to ensure effective supervision from the classification loss. Thus, by varying λ in $\{0, 0.03, 0.06, 0.09, 0.3, 0.6\}$, we performed sensitivity tests on both datasets. In Fig. 5, SuperRL achieves the best results when $\lambda = 0.06$ under the 3-shot setting and $\lambda = 0.09$ under the 5-shot setting on both datasets. This highlights the significance of dual contrastive learning, which is more beneficial for few-shot KG completion as the number of reference triplets increases.

6 Conclusion

This paper explores the role of relevant neighbor entities and their impact on supervised relational learning for few-shot KG completion. We categorize these relevant neighbor entities into two main categories: directly relevant entities and indirectly relevant entities. Based on this, we propose a supervised relational learning model SuperRL leveraging these relevant neighbor entities for few-shot KG completion. SuperRL comprises a cascaded embedding enhancement network to capture the information of both directly and indirectly relevant entities, along with a supervised relational learning module that applies dual contrastive learning and metric learning to provide supervision signals. In experiments, our evaluations not only show that SuperRL significantly outperforms the state-of-the-art baselines in different evaluation metrics, but also confirms the significance of directly relevant entities and indirectly relevant entities as well as the effectiveness of our supervised relational learning model.

Acknowledgements. This work is supported by National Natural Science Foundation of China (Grant No. 62376058, 52378009, 62276063), the Fundamental Research Funds for the Central Universities (2242022R40045), ZhiShan Young Scholar Program of Southeast University, and the Big Data Computing Center of Southeast University.

References

1. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of NeurlIPS, pp. 2787–2795 (2013)
2. Chen, M., Zhang, W., Zhang, W., Chen, Q., Chen, H.: Meta relational learning for few-shot link prediction in knowledge graphs. In: Proceedings of EMNLP-IJCNLP, pp. 4217–4226 (2019)
3. Chen, T., Kornblith, S., Norouzi, M., Hinton, G.: A simple framework for contrastive learning of visual representations. In: Proceedings of ICML (2020)
4. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2D knowledge graph embeddings. In: Proceedings of AAAI (2018)
5. Du, H., Le, Z., Wang, H., Chen, Y., Yu, J.: COKG-QA: multi-hop question answering over COVID-19 knowledge graphs. Data Intell. **4**(3), 471–492 (2022)
6. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: Proceedings of ICML, pp. 1126–1135 (2017)
7. Han, X., et al.: OpenKE: an open toolkit for knowledge embedding. In: Proceedings of EMNLP, pp. 139–144 (2018)
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2015)
9. Lacroix, T., Usunier, N., Obozinski, G.: Canonical tensor decomposition for knowledge base completion. In: Proceedings of ICML, pp. 2863–2872 (2018)
10. Li, X., Yang, X., Ma, Z., Xue, J.H.: Deep metric learning for few-shot image classification: a review of recent developments. Pattern Recogn. **138**(C) (2023)

11. Li, Y., Yu, K., Huang, X., Zhang, Y.: Learning inter-entity-interaction for few-shot knowledge graph completion. In: Proceedings of EMNLP, pp. 7691–7700 (2022)
12. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29, no. 1 (2015)
13. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.: A novel embedding model for knowledge base completion based on convolutional neural network. In: Proceedings of NAACL, pp. 327–333 (2018)
14. Niu, G., et al.: Relational learning with gated and attentive neighbor aggregator for few-shot knowledge graph completion. In: Proceedings of SIGIR, pp. 213–222 (2021)
15. Sheng, J., et al.: Adaptive attentional network for few-shot knowledge graph completion. In: Proceedings of EMNLP, pp. 1681–1691 (2020)
16. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: RotatE: knowledge graph embedding by relational rotation in complex space. In: Proceedings of ICLR (2019)
17. Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., Bouchard, G.: Complex embeddings for simple link prediction. In: Proceedings of ICML, pp. 2071–2080 (2016)
18. Vrandečić, D., Krötzsch, M.: Wikidata: a free collaborative knowledgebase. Commun. ACM **57**(10), 78–85 (2014)
19. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: a survey of approaches and applications. IEEE Trans. Knowl. Data Eng. **29**(12), 2724–2743 (2017)
20. Wang, X., Lin, T., Luo, W., Cheng, G., Qu, Y.: CKGSE: a prototype search engine for Chinese knowledge graphs. Data Intell. **4**(1), 41–65 (2022)
21. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of AAAI, pp. 1112–1119 (2014)
22. Wu, H., Yin, J., Rajaratnam, B., Guo, J.: Hierarchical relational learning for few-shot knowledge graph completion. In: Proceedings of ICLR (2023)
23. Wu, T., et al.: AsdKB: a Chinese knowledge base for the early screening and diagnosis of autism spectrum disorder. In: Proceedings of ISWC, Part II, pp. 59–75 (2023)
24. Wu, T., et al.: Knowledge graph construction from multiple online encyclopedias. World Wide Web **23**, 2671–2698 (2020)
25. Xiong, W., Yu, M., Chang, S., Guo, X., Wang, W.Y.: One-shot relational learning for knowledge graphs. In: Proceedings of EMNLP, pp. 1980–1990 (2018)
26. Yang, B., Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. In: Proceedings of ICLR (2015)
27. Zhang, C., Yao, H., Huang, C., Jiang, M., Li, Z., Chawla, N.V.: Few-shot knowledge graph completion. In: Proceedings of AAAI, pp. 3041–3048 (2020)