# Relation Adaptive Representation Learning Based on Factual Information Interaction for One-Shot Knowledge Graph Completion

Jinlin Li[1,2], Zikang Wang[1,2,*], Linjing Li[1,2], Daniel Dajun Zeng[1,2]
[1]Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
{lijinlin2021, wangzikang2016, linjing.li, dajun.zeng}@ia.ac.cn

***Abstract*—Few-shot, especially one-shot learning is a prominent research area in the field of knowledge graphs (KGs), aiming to utilize a limited number of triples with unseen relations as reference information for inferring missing knowledge. Recent research focuses on improving the semantic representation of entity pairs using interactions between their head and tail entities. However, this method only considers the reference information as the measurement criterion without taking into account the potential impact of it on the reasoning process of the model. In this paper, we propose a novel method that utilizes factual information interactions. Firstly, we learn static representations of entities based on their neighborhood information. Subsequently, we learn relation adaptive representations by incorporating the reference information. This interactive modeling strengthens the association between entity representations and task relations while suppressing irrelevant relations. Extensive experiments demonstrate that our model outperforms state-of-the-art methods on two public datasets. Remarkably, on the NELL-One dataset for one-shot link prediction, our model achieves an improvement of 11.8% in MRR compared to the best baseline model.**

***Index Terms*—Knowledge Graph, Few-shot learning, Factual Information Interaction**

## I. Introduction

Knowledge graphs (KGs) , such as WIKI [1], NELL [2], and YAGO [3], employ a structured approach to present facts. They are widely used in various domains, including recommendation systems [4] [5], question answering systems [6] [7], and pre-trained models [8]. However, the performance of these applications is constrained by the incompleteness of KGs [9].

To address this issue, Xiong et al. [10] propose the task Few-shot Knowledge Graph Completion (FKGC). FKGC aims to complete the missing facts with unseen relations based on a limited number of related referenced entity pairs. The facts that need to be completed is known as the query set, whereas the reference facts are known as the support set. Existing FKGC methods independently model query triples, learning fixed representations of relations between entity pairs [10] [11]. We argue that for queries with different relations,

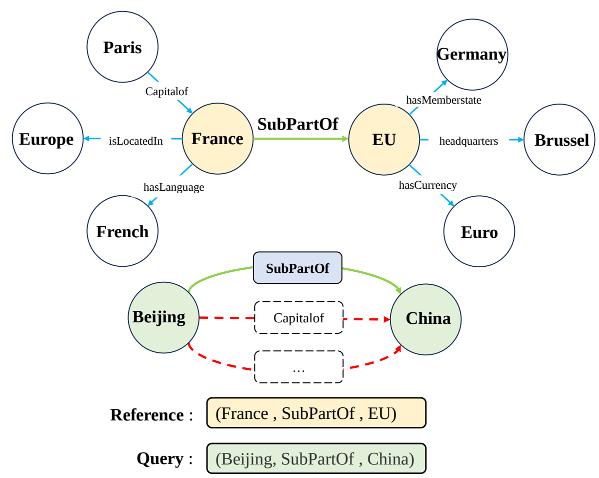


Fig. 1. An illustrative example of one-shot KG learning and the motivation of this paper. There are multiple relations between "`Beijing`" and "`China`" . Green solid lines denote task relations that require semantic strengthening, while red dashed lines represent irrelevant relations that require semantic weakening.

query entity pairs should also focus on different semantic information, requiring adaptive representations. In addition, as shown in Fig. 1, for the query with relation "`SubPartOf`", there may be multiple relations between the query entity pairs. Relations that fall outside of the task's scope may potentially undermine the model's accuracy.

We introduce a method to model the interaction of query entity pairs and support entity pairs in order to address the aforementioned issue. For example, for query triple (`Beijing, SubPartOf, China`) and support triple (`France, SubPartOf, EU`) in Fig. 1, a strong correlation between "`Beijing`" and "`France`" suggests that "`Beijing`" and "`France`" are semantically related, this correlation can be used to enhance the query representation. As the background knowledge of entities can be used to capture this kind of correlations, we use the neighborhood information of support triples in background knowledge to enhance the query representation.

* Corresponding author
This work was supported in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant #XDA27030100 and the National Natural Science Foundation of China under Grants #72293573, #72293575 and #62206282.

In this paper, we propose a novel **F**actual **I**nformation **I**nteraction **N**etwork (FIIN) that learns relation adaptive query representations for different task relations. Specifically, we consider neighborhood information of support entity pairs as factual information and provide it to query entity pairs. The factual information will have an impact on the representation of the entity pairs. This interactive process leads to the enhanced semantic information related to task relations while diminishing the effect of irrelevant relations. To further improve the quality of the representation, an adaptive negative sampling loss is introduced to expedite model optimization and enhance generalization, enabling FIIN to pay more attention to largely indistinguishable query triples during optimization. Experiments on two widely used datasets show that FIIN can largely improve the expressive capability of the learned representations, achieving state-of-the-art performance on FKGC.

In summary, our contributions can be summarized as three-fold:

- To the best of our knowledge, we are the first to explore the modeling of support and query sets interaction in the context of FKGC. Compared to previous methods, FIIN directly integrates task-relevant facts into the representation of the query triple, thereby adaptively generating relation-specific representations.
- Our study presents a novel approach that integrates adaptive loss and FKGC. As far as we know, this is the first attempt to combine these two techniques, leading to a notable enhancement in model performance.
- Extensive experiments on the NELL-One and Wiki-One datasets show that FIIN achieves state-of-the-art performance. Compared with baselines, our model relatively achieves 11.8% performance gains in MRR on NELL-One.

## II. Related Work

Significant progress has recently been achieved in research related to FKGC. The existing FKGC methods can be classified into three groups: metric learning-based methods, meta learner-based methods, and dual-process theory-based methods.

**Metric learning-based methods**. These methods primarily compute similarity scores between query triples and support triples. GMatching [10] first investigates few-shot learning in KGs. It aggregates the neighborhood relationship and the encoded entity information before calculating similarity between query sets and support sets via LSTM. FAAN [12] takes into account the dynamic information of entity pairs, and leverages it to calculate the attention of entities' neighborhoods to aggregate the entity neighborhood information. P-INT [13] expresses entity pair information through the path from the head entity to the tail entity, and uses an interaction-based model to match the paths of the support set and query set. CIAN [11] studies the interactive information of head and tail entities through two stages, and generates more discriminative entity representations. Yao et al. [14] augments data by generating new task relations and new triples. Informix-FKGC [15] augments data by merging the entity's one-hop neighbors, attributes and text description information. Jin et al. [16] use a two-branch feature extractor to capture complementary and comprehensive representations of entities, facilitating the differentiation of the few examples.

**Meta learner-based methods**. MetaR [17] proposes a meta relational learning framework by transferring relation-specific meta information between query set and support set. GANA [18] further improves MetaR by adaptively utilizing the neighborhood information of entities and modeling complex relationships.

**Dual-process theory-based methods**. Inspired by the dual-process theory of cognitive science, CogKR [19] integrate a summary module and a reasoning module to imitate the human cognition process.

The previously mentioned methods only use the neighborhood information of the entities, resulting in uniform query set representations for all task relation. Different from previous paradigms, FIIN takes into account the shared information of the query set and support set for the first time, and utilizes the neighborhood information of support set to enhance the semantic expression of query set. Therefore, the representation of query entity pairs is dynamic.

## III. Background

**KG.** We define a KG as $\mathcal{G} = \{(h, r, t)\} \in \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where $\mathcal{E}$ and $\mathcal{R}$ are the entity set and the relation set. The KGC task is to predict the tail entity $t^q$ when given the head entity $h^q$ and the query relation $r$, denoted as $(h^q, r, ?)$.

**Background KG.** To ensure basic semantics for entities in the KG, as shown in Fig. 1, we extract high-frequency relations $\mathcal{R}_b$ to build a background knowledge graph $\mathcal{G}_b$, where $\mathcal{G}_b \subseteq \mathcal{G}$, and $\mathcal{R}_b \subseteq \mathcal{R}$.

**FKGC.** For a query set $\mathcal{Q}_r = (h^q, r, ?)$ and a candidate entity set $\mathcal{C}_r$, FKGC aims to complete $\mathcal{Q}_r$ based on the given support set $\mathcal{S} = \{(h_i^s, r, t_i^s)\}_{i=1}^K$, where $K$ is a small number and usually set to $\{1, 3, 5\}$. The task is also called K-shot KGC.

To enhance the model's learning capability, we adopt the meta-learning framework. We split $\mathcal{R}$ into a relation set $\mathcal{R}_b$ of $\mathcal{G}_b$ and a meta-relation set $\mathcal{R}_{meta}$, where $\mathcal{R}_b \cap \mathcal{R}_{meta} = \emptyset$. For each relation $r \in \mathcal{R}_{meta}$, we build a meta-task $\mathcal{T}_{meta-r} = \{\mathcal{S}_r, \mathcal{Q}_r\}$, where $\mathcal{S}_r$ is a randomly sampled support set and $|\mathcal{S}_r| = K$. $\mathcal{R}_{meta}$ is divided into $\{\mathcal{R}_{train}, \mathcal{R}_{vaild}, \mathcal{R}_{test}\}$ for training, validation, and testing, respectively. In order to ensure that the model completes $\mathcal{Q}_r$ only based on $\mathcal{S}_r$ during meta-testing, $\{\mathcal{R}_{train}, \mathcal{R}_{vaild}, \mathcal{R}_{test}\}$ are independent of each other.

## IV. Model

This section introduces our method FIIN and shows its overall framework in Fig. 2. The principal framework of FIIN encompasses four components: (1) Entity encoder, which learns adaptive representations based on the $\mathcal{G}_b$; (2) Relation encoder, which leverages the semantics of entity pairs to compute representations of relations; (3) Matching processor, which calculates the semantic matching degree between the provided support set and the query set; (4) Adaptive negative sampling module, which assigns adaptive weights for negative samples while computing loss.

### *A. Entity Encoder*

Entity encoder aims to represent entities by utilizing their information in $\mathcal{G}_b$. Based on existing research [11], our entity encoder further considers the interaction information of the support set and the query set to enhance entity pair representation. Specifically, we take three steps for entity representation: (1) Using information of neighbor nodes and the dynamic information of the entity pairs, adaptively aggregate neighborhood information. (2) According to the interaction information of the head entity and the tail entity, the semantic information associated with the entity pair is enhanced. (3) Using information from the support set to enhance the representation of the query set for the task relation, while weakening the semantics of other relations.

The initial two phases are described as the adaptive neighbor encoder modules for entity pairs, with the subsequent phase labeled as the support and query interaction module.

*1) Adaptive Neighbor Encoder for Entity Pairs:* For an entity pair $(h, t)$, this module aims to enhance entity representations by incorporating their local neighborhood information in $\mathcal{G}_b$. Specifically, we randomly select $m$ one-hop neighbors and their corresponding relations for $h$ and $t$ respectively, denoted as $\mathcal{N} = \{(r_{hi}, e_{hi}), (r_{ti}, e_{ti}) | (r_{hi}, e_{hi}), (r_{ti}, e_{ti}) \in \mathcal{G}_b\}$.

Initially, we update the entity representations by aggregating their neighborhood information. For head entity $h$, we employ a linear transformation to get the neighborhood information matrix $N_h$:

$$N_h = \mathrm{ReLU}((\boldsymbol{r}_h \oplus \boldsymbol{e}_h)W_N), \tag{1}$$

where $\boldsymbol{r}_h, \boldsymbol{e}_h \in \mathbb{R}^{m \times d}$ are neighborhood information matrices formed by the embedding vectors of $r_{hi}$ and $e_{hi}$ in $\mathcal{N}$ respectively, $d$ is the embedding dimension of entities and relations, $W_N \in \mathbb{R}^{2d \times d}$ is a weight matrix, and $\oplus$ denotes the concatenation operator.

Inspired by CIAN [11] and Transformer [20], we utilize the dot-product attention to get the updated representation $\boldsymbol{h}_r$ based on entity pair $(h, t)$, representations of neighbor relations $r_h$ and neighborhood matrix $N_h$:

$$\boldsymbol{h}_r = W_r^s(\mathrm{softmax}(Q_r K_r^T)V_r), \tag{2}$$

where

$$Q_r = W_r^Q(W_r(\boldsymbol{h} \oplus \boldsymbol{t}) + b_r), \tag{3}$$

$$K_r = W_r^K \boldsymbol{r}_h, \tag{4}$$

$$V_r = W_r^V N_h. \tag{5}$$

Here, $W_r^Q, W_r^K, W_r^V, W_r^s \in \mathbb{R}^{d \times d}$, $W_r \in \mathbb{R}^{2d \times d}$, and $b_r \in \mathbb{R}^d$ are all trainable parameters. $\boldsymbol{h}$ and $\boldsymbol{t}$ are the embedding vectors of $h$ and $t$. $Q_r$ is determined by the relation calculated based on the entity pair.

To emphasize the representations of the entities themselves, we perform a residual-like computation:

$$\boldsymbol{h}_z = \boldsymbol{h} + \mathrm{ReLU}(W_1 \boldsymbol{h}_r + W_2 \boldsymbol{h}), \tag{6}$$

where $W_1, W_2 \in \mathbb{R}^{d \times d}$ are trainable and shared parameters.

After updating the entity representation based on its own neighbors, we further update the entity representation by encoding the neighborhood information from the paired entity.

For entity $h$ in pair $(h, t)$, its paired entity neighborhood information matrix is $N_t$. The updated representation $\boldsymbol{h}_c$ is calculated in the same way as Eq. (2) based on the representation $\boldsymbol{t}_z$ of the paired entity $t$, representations of neighbor relations $\boldsymbol{r}_h$, and neighborhood matrix $N_h$. $Q, K, V$ are redefined as follows:

$$Q_c = W_c^Q \boldsymbol{t}_z, \tag{7}$$

$$K_c = W_c^K \boldsymbol{r}_h, \tag{8}$$

$$V_c = W_c^V N_h, \tag{9}$$

where $W_c^Q, W_c^K, W_c^V \in \mathbb{R}^{d \times d}$ are parameters to be learned.

Finally, we get the updated representation $\boldsymbol{h}_e$ of $h$ as:

$$\boldsymbol{h}_e = \boldsymbol{h} + \mathrm{ReLU}(W_1 \boldsymbol{h}_c + W_2 \boldsymbol{h}). \tag{10}$$

*2) Support and Query Interaction:* Given support entity pairs $(h^s, t^s)$ and query entity pairs $(h^q, t^q)$, the support and query interaction module aims to leverage the semantic information of $\mathcal{S}_r$ to enhance the representation of $\mathcal{Q}_r$ on specific relation $r$.

For query head entity $h^q$, we calculate the neighborhood information $N_h^s$ of entity $h^s$ in the support entity pair based on Eq. (1):

$$N_h^s = \mathrm{ReLU}((\boldsymbol{r}_h^s \oplus \boldsymbol{e}_h^s)W_I), \tag{11}$$

where $\boldsymbol{r}_h^s, \boldsymbol{e}_h^s \in \mathbb{R}^{m \times d}$ are neighborhood information matrices of $h^s$, and $W_I \in \mathbb{R}^{2d \times d}$ is a learnable parameter.

We use the neighborhood information of support triples in $\mathcal{G}_b$ as factual interaction information. The input to the attention module is as follows:

$$Q_f = W_f^Q \boldsymbol{h}_e^q, \tag{12}$$

$$K_f = W_f^K N_h^s, \tag{13}$$

$$V_f = W_f^V N_h^s, \tag{14}$$

$W_f^Q, W_f^K, W_f^V \in \mathbb{R}^{d \times d}$ are parameters to be learned. According to the calculation of Eq. (2), we denote the output as $\boldsymbol{h}_f^q$.

Finally, we get the enhanced entity representation $\boldsymbol{h}_o^q$ as follows:

$$\boldsymbol{h}_o^q = \boldsymbol{h}^q + \boldsymbol{h}_e^q + \mathrm{ReLU}(W_1 \boldsymbol{h}_f^q + W_2 \boldsymbol{h}^q). \tag{15}$$

In order to ensure that support entity pairs and query entity pairs are treated equally, entities in the support set are re-aggregated using their own neighbors. For the 3-shot and 5-shot cases, we directly merge neighborhood information of multiple support entities and provide it to the query entity. We leave more efficient integration methods to future work.

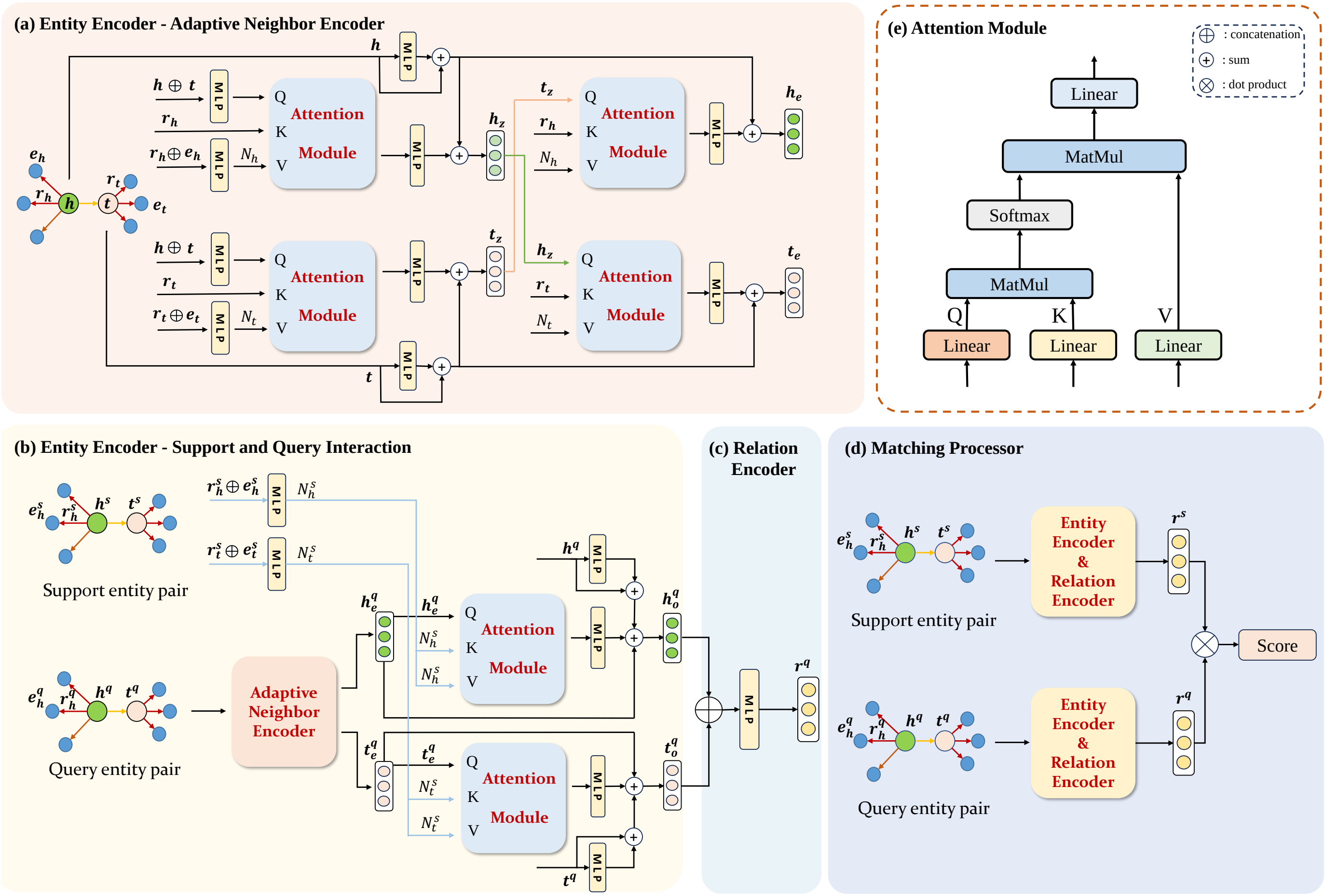


Fig. 2. The overview of FIIN: (a) Adaptive neighbor encoder for entity pairs; (b) Support and query interaction; (c) Relation Encoder; (d) Matching processor; (e) Dot-product attention module. Our entity encoder is composed of (a) and (b). We employ superscripts to differentiate support facts from queries.

### *B. Relation Encoder*

For both support entity pairs and query entity pairs, we calculate the corresponding relations between them by applying a two-layer feed-forward network:

$$\boldsymbol{r} = W_{r2}(\mathrm{ReLU}(W_{r1}(\boldsymbol{h}_o \oplus \boldsymbol{t}_o) + b_{r1})), \tag{16}$$

where $W_{r1}, W_{r2} \in \mathbb{R}^{2d \times 2d}$, $b_{r1} \in \mathbb{R}^{2d}$ are trainable parameters.

### *C. Matching Processor*

Although our experiments mainly focus on the one-shot case, we include an adaptive matching module to the model to compare with the baseline models in the 3-shot and 5-shot cases.

For an unseen relation $r$, since the size of support set is limited and different support entity pairs may focus on expressing different semantics of $r$, this leads to a large variance in the relation representation when using different support entity pairs. Inspired by FAAN [12], we use an attention-based approach to holistically represent the K-shot references. For the relation representation $\boldsymbol{r}^q$ of a query triple, the overall representation $\boldsymbol{M}_{r^q}$ of K-shot support set is:

$$\boldsymbol{M}_{r^q} = \sum_{i=1}^{K} \beta_i \boldsymbol{r}_i^s, \tag{17}$$

$$\beta_i = \frac{e^{\delta(\boldsymbol{r}^q, \boldsymbol{r}_i^s)}}{\sum_{j=1}^{K} e^{\delta(\boldsymbol{r}^q, \boldsymbol{r}_j^s)}}, \tag{18}$$

where $\beta_i$ denotes the attention score between $\boldsymbol{r}^q$ and $\boldsymbol{r}_i^s$, $\boldsymbol{r}_i^s$ denotes the relation representation of the $i$-th support triple. $\delta(\boldsymbol{r}^q, \boldsymbol{r}_i^s) = \boldsymbol{r}^q \cdot \boldsymbol{r}_i^s$, where $\delta(\cdot,\cdot)$ measures the semantic similarity of $\boldsymbol{r}^q$ and $\boldsymbol{r}_i^s$. Finally, we define a metric function $\phi(r^q, \mathcal{S}_r)$ to calculate the matching score of the query $r^q$ with the K-shot support set $\mathcal{S}_r$:

$$\phi(r^q, \mathcal{S}_r) = \boldsymbol{r}^q \cdot \boldsymbol{M}_{r^q}. \tag{19}$$

When K=1, the metric function $\phi(\cdot,\cdot)$ can be defined as follows:

$$\phi(r^q, \mathcal{S}_r) = \boldsymbol{r}^q \cdot \boldsymbol{r}^s, \tag{20}$$

where $\boldsymbol{r}^s$ is representation of the relation in the support triple.

In FKGC, higher score of $\phi(r^q, \mathcal{S}_r)$ corresponds to stronger semantic similarity between $r^q$ and $\mathcal{S}_r$. This indicates that the likelihood increases of $r$ being present in the query entity pairs.

### *D. Adaptive Negative Sampling Loss and Model Training*

For an unseen relation $r$, we collect a batch of triples as the positive query set $\mathcal{Q}_r^+ = \{(h^q, t^{q+})|(h^q, r, t^{q+}) \in \mathcal{G}\}$, and construct a batch of negative query triples $\mathcal{Q}_r^- = \{(h^q, t^{q-})|(h^q, r, t^{q-}) \notin \mathcal{G}\}$ by polluting the tail entities of $\mathcal{Q}_r^+$.

Existing models simply use a margin-based scoring function to ensure that $\mathcal{Q}_r^+$ has a higher similarity score $\mathcal{S}_r$ than $\mathcal{Q}_r^-$:

$$\mathcal{L} = \sum_r \sum_{r^{q+} \in \mathcal{Q}_r^+} \sum_{r^{q-} \in \mathcal{Q}_r^-} [\gamma + \phi(r^{q-}, \mathcal{S}_r) - \phi(r^{q+}, \mathcal{S}_r)]_+, \tag{21}$$

where $[x]_+ = \max(0, x)$ is the standard hinge loss, and the margin $\gamma$ is a hyperparameter to separate $\mathcal{Q}_r^+$ and $\mathcal{Q}_r^-$.

However, this loss function treats all negative samples equally, which is unreasonable. Only a small number of strong negative samples significantly influence the model's optimization direction. Therefore, we propose an adaptive loss computation mechanism that assign smaller weights to those negative samples that the model can already distinguish with ease, while assigning larger weights to those negative samples that are difficult for the model to distinguish. The weight $\alpha_{r_i}$ for negative sample $r_i^{q-}$ is:

$$\alpha_{r_i} = \frac{e^{\phi(r_i^{q-}, \mathcal{S}_r)}}{\sum_{j=1}^{n} e^{\phi(r_j^{q-}, \mathcal{S}_r)}}, \tag{22}$$

where $n$ is the number of negative samples.

Finally, the negative sampling loss $\mathcal{L}$ is calculated as follows:

$$\mathcal{L} = \sum_r \sum_{r^{q+} \in \mathcal{Q}_r^+} [\gamma + \sum_{i=1}^{n} \alpha_{r_i} \phi(r_i^{q-}, \mathcal{S}_r) - \phi(r^{q+}, \mathcal{S}_r)]_+. \tag{23}$$

To minimize $\mathcal{L}$, we consider each unseen relation as a meta-task and sample batches of meta-tasks to train the network.

## V. EXPERIMENTS

This section presents extensive experiments to verify the effectiveness of FIIN. Furthermore, the ablation tests are conducted to demonstrate the impact of each individual key component of FIIN.

### *A. Datasets*

We conduct experiments on two widely used benchmarks: NELL-One and Wiki-One [10]. According to the dataset settings, relations with less than 500 but more than 50 triples are selected as meta-tasks in FKGC, and other relations with the corresponding triples constitute the $\mathcal{G}_b$. We use 51/5/11 and 133/16/34 relations for training/validation/testing in NELL-One and Wiki-One, respectively. The specific statistics are listed in Table I.

TABLE I
STATISTICS OF THE DATASETS. # DENOTES 'THE NUMBER OF'.

| Dataset | # Ent. | # Rel. | # Triples | # Tasks |
|---|---|---|---|---|
| NELL-One | 68,545 | 358 | 181,109 | 67 |
| Wiki-One | 4,838,244 | 822 | 5,859,240 | 183 |

### *B. Evaluation Metrics*

We use two traditional metrics, MRR and Hits@N, in KGC to evaluate the performance of the models. MRR is the mean reciprocal rank of the correct entities. Hits@N is the proportion of correct entities ranked in top $N$, with $N = 1, 5, 10$. Higher MRR and Hits@N indicates better model performance.

### *C. Baselines*

**KG embedding models.** We adopt four widely used methods as baselines: TransE [21], DistMult [22], ComplEx [23], and RESCAL [24]. As traditional embedding models require a sizeable number of training triples, their performance is limited in few-shot tasks. Therefore, we primarily compare the performance of FIIN to FKGC models.

**FKGC models.** We adopt seven FKGC models as baselines: GMatching [10], MetaR [17], CogKR [19], FAAN [12], GANA [18], FCC [16], and CIAN [11]. For MetaR, the results under In-train and Pre-train settings are provided separately. All of these methods represent the query set and support set independently, without considering the interaction between them. Due to different evaluation criteria, we will not compare with P-INT [13]. For fairness, we do not use the models with additional information as baselines,such as InforMix-FKGC [15].

### *D. Implementation Details*

In order to reduce the training time of FIIN, we use embeddings pre-trained by TransE to initialize entity and relation embeddings on both datasets. We set the maximum number of neighbors as 100 on both datasets. The embedding dimensions of entity and relation are set to 100 and 50 for NELL-One and Wiki-One, respectively. To enhance the generalization ability of the model, we apply dropout layer with dropout rate equals 0.2. The number of negative samples is set to 16. We set Adam [25] as the optimizer, and set the initial learning rate of the optimizer to be $8e^{-5}$ and $3e^{-4}$ for NELL-One and Wiki-One, respectively. The margin $\gamma$ is set to 5.0. We evaluate FIIN for every 5k training steps. We use MRR as an indicator to select the optimal parameters based on the validation set within 100k steps.

### *E. Results and Analysis*

We mainly study the feasibility of enhancing the representation of entities through the information interaction between the support set and the query set in the one-shot case. Table II

TABLE II
RESULTS OF ONE-SHOT LINK PREDICTION ON NELL-ONE AND WIKI-ONE. †RESULTS IS OBTAINED ACCORDING TO THE OFFICIAL CODE PROVIDED BY THE AUTHORS. ‡ RESULTS COME FROM [10]. THE REMAINING RESULTS WERE REPORTED IN THE ORIGINAL PAPERS.

| | **NELL-One** | | | | **Wiki-One** | | | |
|---|---|---|---|---|---|---|---|---|
| | MRR | Hits@10 | Hits@5 | Hits@1 | MRR | Hits@10 | Hits@5 | Hits@1 |
| RESCAL‡ | .140 | .229 | .186 | .089 | .072 | .082 | .062 | .051 |
| TransE‡ | .093 | .192 | .141 | .043 | .035 | .052 | .043 | .025 |
| DistMult‡ | .102 | .177 | .126 | .066 | .048 | .101 | .070 | .019 |
| ComplEX‡ | .131 | .223 | .086 | .086 | .069 | .121 | .092 | .040 |
| GMatching(ComplEx) | .185 | .313 | .260 | .119 | .200 | .336 | .272 | .120 |
| CogKR | .256 | .353 | .314 | .205 | .288 | .366 | .334 | .249 |
| MetaR(Pre-Train) | .164 | .331 | .238 | .093 | .314 | .404 | .375 | .266 |
| MetaR(In-Train) | .250 | .401 | .336 | .170 | .193 | .280 | .233 | .152 |
| FAAN† | .174 | .322 | .250 | .099 | .206 | .363 | .275 | .134 |
| GANA | .307 | .483 | .409 | .211 | .301 | .416 | .350 | .231 |
| CIAN† | .291 | .456 | .356 | .215 | .337 | .451 | .404 | .261 |
| FCC | .331 | **.531** | - | - | .343 | .460 | - | - |
| FIIN | **.370** | .507 | **.443** | **.297** | **.359** | **.481** | **.433** | **.289** |

reports the performance of all models on NELL-One and Wiki-One datasets. It reveals that FIIN significantly outperforms all baseline models on both datasets. For one-shot link prediction, compared with the results of best baseline model, FIIN achieves relative improvements of 11.8% in MRR on NELL-One, and improvements of 4.7% on Wiki-One, respectively. This proves that using information from the neighborhood of the support set to enhance the representation of query entity pairs can effectively improve the performance of FKGC.

We conducted an experimental study to examine the effect of the size of negative samples on the model. Figure 3 illustrates the performance of FIIN at different sizes of negative samples. Due to limitations in GPU memory, we did not conduct larger-scale experiments. According to the experimental results, it can be concluded that FIIN still has the potential for performance improvement.

In order to explore the adaptability of FIIN in 3-shot and 5-shot cases, we report the results of FIIN and existing FKGC models on NELL-One in Table III. We observe that the performance of FIIN in the one-shot case is similar to the best baseline in the 5-shot case. Although our model still outperforms baselines for 3-shot and 5-shot cases, the improvement is not as significant as that in one-shot case. We argue that direct aggregation of neighborhood information of multiple support entities may introduce bias in the representation for query.

### *F. Ablation Studies*

Compared to the existing models, FIIN improves the performance by extracting the interaction information between the support set and the query set and adding an adaptive negative sampling module. To investigate the contribution of these components, we provide variant models of FIIN for ablation tests on NELL-One. One remove the interaction information between the support set and the query set, denoted as $-s$, one remove the adaptive negative sampling module, denoted as $-l$, and the other remove both parts, denoted as $-l-s$. The results of the ablation experiments are shown in Table IV. We can see that the two components have a significant impact on the model performance.

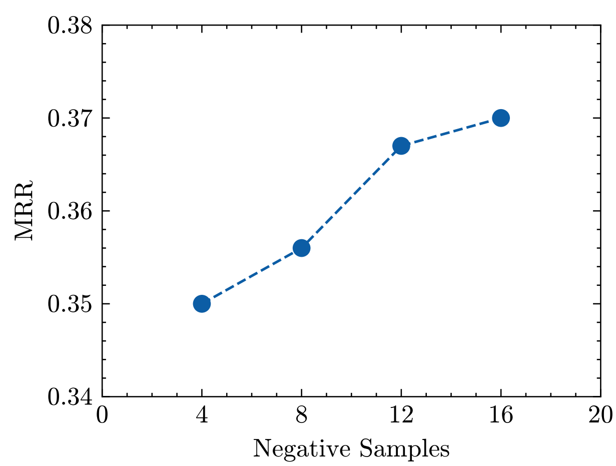


Fig. 3. Impact of negative samples size on NELL dataset.

To demonstrate the generalizability of our proposed method, we further introduced these two modules into FAAN [12], and the results are presented in Table V. The improvements brought to FAAN show that the ideas proposed by FIIN have universal applicability.

TABLE III
RESULTS OF 3-SHOT AND 5-SHOT LINK PREDICTION ON NELL-ONE.

| | MRR | | Hits@10 | | Hits@5 | | Hits@1 | |
|---|---|---|---|---|---|---|---|---|
| | 3-shot | 5-shot | 3-shot | 5-shot | 3-shot | 5-shot | 3-shot | 5-shot |
| GMatching(ComplEx) | - | .200 | - | .325 | - | .269 | - | .133 |
| MetaR(Pre-Train) | .210† | .209 | .386† | .355 | .311† | .280 | .119† | .141 |
| MetaR(In-Train) | .245† | .261 | .456† | .437 | .360† | .350 | .144† | .168 |
| FAAN | .247† | .279 | .369† | .428 | .309† | .364 | .183† | .200 |
| GANA | .322 | .344 | .510 | .517 | .432 | .437 | .225 | .246 |
| CIAN | .344 | .376 | .484 | .527 | .417 | .453 | .266 | .298 |
| FCC | .346 | .374 | **.553** | **.566** | - | - | - | - |
| FIIN | **.386** | **.398** | .506 | .537 | **.446** | **.485** | **.317** | **.319** |

TABLE IV
ABLATION STUDY OF ONE-SHOT LINK PREDICTION ON NELL-ONE.

| Model | MRR | Hits@10 | Hits@5 | Hits@1 |
|---|---|---|---|---|
| FIIN | **.370** | **.507** | **.443** | **.297** |
| $-s$ | .342 | .479 | .401 | .263 |
| $-l$ | .346 | .482 | .411 | .265 |
| $-l-s$ | .299 | .451 | .379 | .208 |

TABLE V
ONE-SHOT LINK PREDICTION RESULTS OF THE FAAN VARIANT MODELS ON NELL-ONE.

| Model | MRR | Hits@10 | Hits@5 | Hits@1 |
|---|---|---|---|---|
| FAAN | .174 | .322 | .250 | .099 |
| $+s$ | .205 | .346 | .276 | .130 |
| $+l$ | .203 | .339 | .267 | .133 |
| $+l+s$ | **.241** | **.360** | **.307** | **.174** |

### G. Results on Different Relations

We show the prediction performance for various relations on NELL-One in Table VI. #Can. denotes the number of candidate entities. Notably, there is a significant variance in results across different meta-tasks, which is a prevalent challenge encountered in FKGC methods. This issue is possibly due to the variation in candidate entity count and the limitations of $\mathcal{G}_b$. $\mathcal{G}_b$ arises from the fact that not all task relations are inherently implicit within it, consequently constraining the accuracy of representation in FKGC models. Nevertheless, it is worth highlighting that FIIN outperforms other methods in most task relations, showing its strong generalization ability.

In order to demonstrate the advantages of our model more intuitively, we visualized the representations of entity pairs, focusing on the "`Producedby`" relationship as an example. We analyzed the representation distributions of various entity pairs, including reference, positive, and negative candidate entity pairs, as shown in Fig. 4. The negative candidate entity pairs in FIIN are positioned significantly farther from the reference entity pairs within the vector space. On the other hand, in the case of CIAN, while there are discernible distinctions in the vector distributions of positive and negative candidate entity pairs, the reference entity vectors struggle in making this distinction. The similarity ranking of positive samples is adversely affected as numerous negative sample vectors show similarity to the reference entity pair vectors. Conversely, FIIN considers the interaction information during modeling, which results in an improved similarity between positive samples and reference vectors.

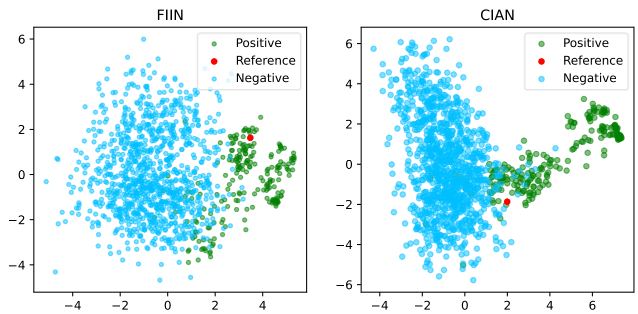


Fig. 4. The visualization of entity pair relation representations.

TABLE VI
ONE-SHOT LINK PREDICTION RESULTS (MRR) OF GANA, CIAN AND FIIN FOR EACH RELATION (RID) ON NELL-ONE.

| | | MRR | | |
|---|---|---|---|---|
| RID | #Can. | GANA | CIAN | FIIN |
| 1 | 123 | .974 | .973 | **.980** |
| 2 | 299 | .085 | .092 | **.127** |
| 3 | 786 | .253 | .447 | **.472** |
| 4 | 1084 | .421 | .421 | **.515** |
| 5 | 2100 | .488 | **.546** | .536 |
| 6 | 2160 | **.250** | .204 | .240 |
| 7 | 2222 | .121 | **.140** | .135 |
| 8 | 3174 | .338 | .356 | **.585** |
| 9 | 5716 | .139 | .111 | **.176** |
| 10 | 10569 | .179 | .196 | **.383** |
| 11 | 11618 | **.211** | .048 | .134 |

## VI. CONCLUSION

This paper proposes a one-shot KGC method that learns representations adaptively by leveraging factual information interactions between a support set and a query set. Different from previous studies that independently learn representations for each set, neglecting their mutual interactivity, FIIN extracts information from the support set and applies it to enhance the specificity of the query set's representation concerning certain relationships between entities. Experimental results on two public datasets demonstrate the superior performance of our model compared to current state-of-the-art methods. Future research will involve enhancing entity representation with additional information and efficiently integrating data from multiple support entity pairs.

## Acknowledgements

We extend our sincere gratitude to the anonymous reviewers for their insightful and valuable suggestions, which have significantly enhanced the quality of this paper.

## References

[1] D. Vrandečić and M. Krötzsch, "Wikidata: a free collaborative knowledgebase," *Communications of the ACM*, vol. 57, no. 10, pp. 78–85, 2014.

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

[3] F. M. Suchanek, G. Kasneci, and G. Weikum, "Yago: a core of semantic knowledge," in *Proceedings of the 16th international conference on World Wide Web*, 2007, pp. 697–706.

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

[5] Y. Cao, X. Wang, X. He, Z. Hu, and T.-S. Chua, "Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences," in *The world wide web conference*, 2019, pp. 151–161.

[6] M. Shi, "Knowledge graph question and answer system for mechanical intelligent manufacturing based on deep learning," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–8, 2021.

[7] A. Saxena, A. Tripathi, and P. Talukdar, "Improving multi-hop question answering over knowledge graphs using knowledge base embeddings," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 4498–4507.

[8] D. Yu, C. Zhu, Y. Yang, and M. Zeng, "Jaket: Joint pre-training of knowledge graph and language understanding," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 10, 2022, pp. 11 630–11 638.

[9] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo, "Knowledge graph embedding for link prediction: A comparative analysis," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 2, pp. 1–49, 2021.

[10] W. Xiong, M. Yu, S. Chang, X. Guo, and W. Y. Wang, "One-shot relational learning for knowledge graphs," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 1980–1990.

[11] Y. Li, K. Yu, X. Huang, and Y. Zhang, "Learning inter-entity-interaction for few-shot knowledge graph completion," in *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, 2022, pp. 7691–7700.

[12] J. Sheng, S. Guo, Z. Chen, J. Yue, L. Wang, T. Liu, and H. Xu, "Adaptive attentional network for few-shot knowledge graph completion," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 1681–1691.

[13] J. Xu, J. Zhang, X. Ke, Y. Dong, H. Chen, C. Li, and Y. Liu, "P-int: A path-based interaction model for few-shot knowledge graph completion," in *Findings of the Association for Computational Linguistics: EMNLP 2021*, 2021, pp. 385–394.

[14] Y. Yao, Z. Zhang, Y. Xu, and C. Li, "Data augmentation for few-shot knowledge graph completion from hierarchical perspective," in *Proceedings of the 29th International Conference on Computational Linguistics*, 2022, pp. 2494–2503.

[15] Q. Li, J. Yao, X. Tang, H. Yu, S. Jiang, H. Yang, and H. Song, "Capsule neural tensor networks with multi-aspect information for few-shot knowledge graph completion," *Neural Networks*, vol. 164, pp. 323–334, 2023.

[16] J. Liu, C. Fan, F. Zhou, and H. Xu, "Complete feature learning and consistent relation modeling for few-shot knowledge graph completion," *Expert Systems with Applications*, vol. 238, p. 121725, 2024.

[17] M. Chen, W. Zhang, W. Zhang, Q. Chen, and H. Chen, "Meta relational learning for few-shot link prediction in knowledge graphs," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 4217–4226.

[18] G. Niu, Y. Li, C. Tang, R. Geng, J. Dai, Q. Liu, H. Wang, J. Sun, F. Huang, and L. Si, "Relational learning with gated and attentive neighbor aggregator for few-shot knowledge graph completion," in *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021, pp. 213–222.

[19] Z. Du, C. Zhou, J. Yao, T. Tu, L. Cheng, H. Yang, J. Zhou, and J. Tang, "Cogkr: cognitive graph for multi-hop knowledge reasoning," *IEEE Transactions on Knowledge and Data Engineering*, 2021.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

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

[22] B. Yang, S. W.-t. Yih, X. He, J. Gao, and L. Deng, "Embedding entities and relations for learning and inference in knowledge bases," in *Proceedings of the International Conference on Learning Representations (ICLR) 2015*, 2015.

[23] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, and G. Bouchard, "Complex embeddings for simple link prediction," in *International conference on machine learning*. PMLR, 2016, pp. 2071–2080.

[24] M. Nickel, V. Tresp, and H.-P. Kriegel, "A three-way model for collective learning on multi-relational data," in *Proceedings of the 28th International Conference on International Conference on Machine Learning*, 2011, pp. 809–816.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.