Exploring Relational Semantics for Inductive Knowledge Graph Completion

Changjian Wang, [1,2] **Xiaofei Zhou,** [1,2]* **Shirui Pan,** [3] **Linhua Dong,** [1,2] **Zeliang Song,** [1,2] **Ying Sha** [4]

[1] Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] Faculty of Information Technology, Monash University, Melbourne, Australia
[4] College of Informatics, Huazhong Agricultural University, Wuhan, China
{wangchangjian,zhouxiaofei,donglinhua,songzeliang}@iie.ac.cn, shirui.pan@monash.edu, shaying@mail.hzau.edu.cn

Abstract

Knowledge graph completion (KGC) aims to infer missing information in incomplete knowledge graphs (KGs). Most previous works only consider the transductive scenario where entities are existing in KGs, which cannot work effectively for the inductive scenario containing emerging entities. Recently some graph neural network-based methods have been proposed for inductive KGC by aggregating neighborhood information to capture some uncertainty semantics from the neighboring auxiliary triples. But these methods ignore the more general relational semantics underlying all the known triples that can provide richer information to represent emerging entities so as to satisfy the inductive scenario. In this paper, we propose a novel model called CFAG, which utilizes two granularity levels of relational semantics in a coarse-grained aggregator (CG-AGG) and a fine-grained generative adversarial net (FG-GAN), for inductive KGC. The CG-AGG firstly generates entity representations with multiple semantics through a hypergraph neural network-based global aggregator and a graph neural network-based local aggregator, and the FG-GAN further enhances entity representations with specific semantics through conditional generative adversarial nets. Experimental results on benchmark datasets show that our model outperforms state-of-the-art models for inductive KGC.

Introduction

Knowledge graphs (KGs) store facts about the real world as collections of triples. Each triple $\langle h, r, t\rangle$ in KGs indicates a relation r between a head entity h and a tail entity t. In recent years, several KGs such as NELL (Carlson et al. 2010) and DBpedia (Lehmann et al. 2015) have been built and become extremely useful resources for various applications (Hao et al. 2017; Moon et al. 2019; Zhu et al. 2020). Since KGs are still far from completion to support the downstream tasks, many knowledge graph completion (KGC) methods have been proposed to predict the missing links, such as TransE (Bordes et al. 2013), DistMult (Yang et al. 2015), and ConvE (Dettmers et al. 2018). However, these traditional models are inherently transductive, which require that all entities should be seen during training. As we know that KGs may continuously evolve into an inductive scenario which emerges many new entities. So under the inductive scenario, the traditional embedding-based models hardly represent the emerging entities, and cannot work for the inductive KGC.

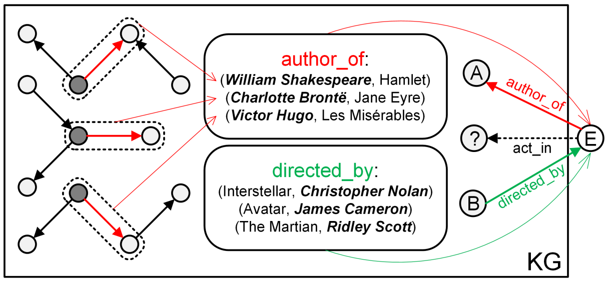

author_of:
(William Shakespeare, Hamlet)
(Charlotte Brontë, Jane Eyre)
(Victor Hugo, Les Misérables)
directed_by:
(Interstellar, Christopher Nolan)
(Avatar, James Cameron)
(The Martian, Ridley Scott)
A
author_of
?
act_in
E
B
directed_by
KG

Figure 1: An example of inductive KGC. E is an emerging entity and the task is $\langle E, act\_in, ?\rangle$. In addition to limited local neighborhood information, through relations, we can utilize coarse-grained relational semantics (global head entities of $author\_of$ and global tail entities of $directed\_by$) and fine-grained relational semantics (query relation act_in) to represent the emerging entity E.

The inductive KGC is more challenging than the traditional task, due to the uncertainty of emerging entity representations. Traditional KGC methods can learn the representations of entities more accurately by fully training, but the inductive KGC methods need to learn a general model through as much known information as possible to represent emerging entities. Recently, some methods (Hamaguchi et al. 2017; Albooyeh, Goel, and Kazemi 2020) based on graph neural networks (GNNs) (Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2017) have been proposed for inductive KGC, which generate the representation of an emerging entity by aggregating the information of its local neighbors. Such methods can capture some uncertainty semantics from the neighboring auxiliary triples, but face two issues. Firstly, they obtain coarse-grained entity representations by gathering the local information, but ignore the possible global information. Secondly, for specific query relations, they obtain fine-grained entity representations through training on known triples to balance with the coarse-grained representations, but cannot generalize better

*Corresponding Author

to emerging entity representations. This paper considers the relational semantics in KG to solve the two issues.

As we known, the relational semantics can be reflected by the fact that entities with a same relation are usually semantically similar. For example, given a relation $author\_of$ and some corresponding triples of the form $\langle X, author\_of, Y \rangle$, the head entities ($X$) mean writers and the tail entities (Y) mean books. Through the relational semantics we can capture more useful information to represent emerging entities. Therefore we consider the relational semantics at two granularity levels for our work. At coarse granularity, an entity related to multiple relations may have multiple semantics (e.g., E in Figure 1 is both a writer and a director). We consider the global head or tail entities related to these relations in the known triples to capture global information, so as to obtain entity representations with multiple semantics. At fine granularity, an entity under a certain query relation should have specific semantics (e.g., for the task $\langle E, act\_in, ? \rangle$ in Figure 1, E is an actor). That existing methods enhancing entity representations under query relations by training on known triples is not enough to generalize to emerging entities. We consider a model that can map known entities given a certain query relation to a same semantic distribution, while being able to generalize to emerging entities, so as to generate more accurate entity representations with specific semantics.

In this paper, we propose CFAG, a novel method for inductive KGC, which consists of a coarse-grained aggregator (CG-AGG) and a fine-grained GAN (FG-GAN). To utilize coarse-grained relational semantics, we build the CG-AGG, in which we regard relations as hyperedges and construct a hypergraph using triples in KGs. The CG-AGG uses a hypergraph neural network (HGNN)-base global aggregator and a graph neural network (GNN)-based local aggregator to capture global and local entity information, so that it can obtain entity representations with multiple semantics. To utilize fine-grained relational semantics, we build the FG-GAN based on conditional generative adversarial nets (CGANs). The generator of FG-GAN uses the entity representations generated by the CG-AGG as an input, and generates new entity representations conditioned on certain query relations. We associate each relation with a prior distribution and enforce the generated representations to match the prior distributions, so as to generate more accurate entity representations with specific semantics. To keep predictive capabilities, we also train our model using triples to predict target entities. We evaluate our proposed CFAG model for inductive KGC on two benchmark datasets, and the experimental results demonstrate the effectiveness of relational semantics and the superiority of our model.

In summary, our contributions are as follows:

- We propose a novel model called CFAG for inductive KGC. CFAG consists of CG-AGG and FG-GAN, and it utilizes coarse-grained and fine-grained relational semantics in KGs to generate the representations of emerging entities.
- The CG-AGG, as a coarse-grained aggregator, regards relations as hyperedges and construct a hypergraph on a KG. It can capture both global and local entity information to obtain entity representations with multiple semantics.
- The FG-GAN, as a fine-grained net, associates each relation with a prior distribution and enforces the generated representations to match the prior distributions. It enhances our model to generate more accurate entity representations with specific semantics.
- We evaluate our CFAG on two benchmark datasets FB15k-237 and NELL-995. Experimental results show that our model significantly and consistently outperforms state-of-the-art methods.

Related Work

In recent years, the most successful methods for KGC are embedding-based methods (Nayyeri et al. 2021; Zhou et al. 2021; Rossi et al. 2021). These methods map entities and relations into a continuous space and define a scoring function to infer the missing information. Translational models, such as TransE (Bordes et al. 2013) and its extensions (Wang et al. 2014; Lin et al. 2015), represent entities and relations as vectors and treat relations as translations from head entities to tail entities. Bilinear models, such as RESCAL (Nickel, Tresp, and Kriegel 2011) and its extensions (Yang et al. 2015; Trouillon et al. 2016), represent relations as matrices and combine head and tail entities by multiplication. Compared with these shallow models, some more expressive models which based on CNNs (Dettmers et al. 2018; Nguyen et al. 2018) or GNNs (Schlichtkrull et al. 2018; Vashishth et al. 2020) have recently received widespread attention and achieved better performance. However, these models are used for transductive KGC, which needs all the test entities have been seen in the training set.

Actually, KGs are constantly evolving, and there are many emerging entities that are not in the original KGs. Transductive KGC methods are failed to handle these emerging entities. Although some methods use external resources such as text (Shi and Weninger 2018; Daza, Cochez, and Groth 2021) and images (Xie et al. 2017) to learn the representations of emerging entities, these resources may be hard to acquire. Some GNNs such as GraphSAGE (Hamilton, Ying, and Leskovec 2017) and GraphSAINT (Zeng et al. 2020) show their inductive representation learning ability in graphs. Follow this direction, some GNN-based inductive KGC methods have been proposed. Hamaguchi et al. (2017) proposed a tailored GNNs to compute the embeddings of emerging entities by aggregating the representations of their neighbor entities and relations. To distinguish the importance of neighbor entities and relations, Wang et al. (2019) introduced a novel aggregator to aggregate neighbors using the attention mechanism. He et al. (2020) developed a virtual neighbor prediction method to reduce the neighbor sparsity problem. Since previous models are not explainable, Bhowmik and de Melo (2020) utilized a variant of Graph Transformer encoder and the reinforcement learning for inductive representation learning while preserving explainability. Albooyeh et al. (2020) extended current transductive KGC methods by developing new training algorithms which use aggregation functions to handle emerging entities. However, these methods mainly utilize limited local neighborhood information, while we utilize the relational semantics, which can capture more useful information in known KGs to represent emerging entities.

We also note some related lines of studies on emerging entities or relations. Some works focus on the few-shot scenario of entities (Baek, Lee, and Hwang 2020) or relations (Zhang et al. 2020; Jiang, Gao, and Lv 2021) in KGs, and some researches (Teru, Denis, and Hamilton 2020; Chen et al. 2021) focus on predicting missing relations on an entirely new KGs. In this paper, we focus the inductive scenario which contains emerging entities connected with original KGs by auxiliary triples, and predict missing entities. There are also some works which are close to ours. Fatemi et al. (2020) extends the hypergraph to KGs with non-binary relations, while we still focus on the general binary relational KGs. ReInceptionE (Xie et al. 2020) uses global entity information for a triple through attention mechanism, while we use global entity information for the emerging entities through HGNNs. In addition, it focuses on transductive KGC while we focus on inductive KGC. ARGA (Pan et al. 2018) regularizes graph embedding through an adversarial training scheme and enforce the latent codes to match a prior distribution. Unlike this model, we focus on the inductive scenario of multi-relational graphs, and we enforce the entity representations to match multiple semantic distributions.

Problem Definition

A knowledge graph can be formalized as a set of triples $\mathcal{K} = \{\langle h, r, t\rangle\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$, where $\mathcal{E}$ and $\mathcal{R}$ are the sets of existing entities and relations respectively. For our inductive scenario, we define the set of emerging entities (i.e. unseen entities when testing) as $\widetilde{\mathcal{E}}$, where each emerging entity $\widetilde{e} \in \widetilde{\mathcal{E}}$ is associated with some existing entities by auxiliary triples $\{\langle \widetilde{e}, r, e\rangle \ or \langle e, r, \widetilde{e}\rangle \mid \widetilde{e} \in \widetilde{\mathcal{E}}, e \in \mathcal{E}, r \in \mathcal{R}, \mathcal{E} \cap \widetilde{\mathcal{E}} = \emptyset\}$.

Knowledge graph completion (KGC) typically refers to the task of predicting either the tail entity t given h and r ($\langle h, r, ?\rangle$) or the head entity h given r and t ($\langle ?, r, t\rangle$). By adding the inverse relation set $\mathcal{R}^{-1}$ to $\mathcal{R}$, $\langle ?, r, t\rangle$ can be represented as $\langle t, r^{-1}, ?\rangle$, so we unify these two tasks into the task of predicting $o \in \mathcal{E}$ given an entity s and a query relation q ($\langle s, q, ?\rangle$). For traditional KGC, $s \in \mathcal{E}$ is an existing entity, while for inductive KGC, $s \in \widetilde{\mathcal{E}}$ is an emerging entity.

Methodology

In this section, we will describe the two parts (CG-AGG and FG-GAN) of our model in detail. The CG-AGG aims at generating entity representations with multiple semantics. The FG-GAN focuses on generating more accurate entity representations with specific semantics, and keeping predictive capabilities. Figure 2 shows the architecture of CFAG. In the following, we take the task $\langle s, q, ?\rangle$ as an example, and the target entity is o.

CG-AGG

In order to use coarse-grained relational semantics, we first convert a KG to a hypergraph, and then use a HGNN-based global aggregator to capture global entity information and a GNN-based local aggregator to capture local entity information. Finally, we combine the two aggregators to obtain entity representations with multiple semantics.

Hypergraph Construction In the hypergraph, a hyperedge is a non-empty subset of a vertex set. We treat each relation r as two hyperedges ζ_r^h and ζ_r^t, which represent the head entity set and the tail entity set of r respectively. The hypergraph on a KG is defined as $\mathcal{HG} = (\mathcal{E}, \mathcal{Z})$, where $\mathcal{Z} = \{\zeta_r^h \mid r \in \mathcal{R}\} \cup \{\zeta_r^t \mid r \in \mathcal{R}\}$ is the set of hyperedges. Let $\mathcal{E} = \{e_1, e_2, \ldots, e_{|\mathcal{E}|}\}$ and $\mathcal{Z} = \{\zeta_1, \zeta_2, \ldots, \zeta_{2|\mathcal{R}|}\}$, $\mathcal{HG}$ can be denoted by an incidence matrix $\mathbf{H}$, with entries defined as:

$$H_{ij} = \begin{cases} 1, & \text{if } e_i \in \zeta_j \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Figure 2(a) shows an example of constructing $\mathbf{H}$ using triples (e.g. $H_{21} = 1$ since e_2 is a head entity of r_1).

Global Aggregator To model complex graph data more efficiently, in recent years, HGNNs (Feng et al. 2019; Yadati et al. 2019; Bai, Zhang, and Torr 2021), the hypergraph based GNNs, have emerged as promising solutions. In this part, we build our global aggregator based on a one-layer HGNN to capture global entity information. As is shown in Figure 2(b), at first, the global aggregator gathers the global entity information according to the hyperedges to form the hyperedge features, and then aggregates the hyperedge features related to the entities as their representations.

Formally, we first gather the representations of entities which belong to each hyperedge to form the hyperedge representations:

$$\mathbf{Z} = \mathbf{W}\mathbf{D}^{-1}\mathbf{H}^{\mathrm{T}}\mathbf{E}, \tag{2}$$

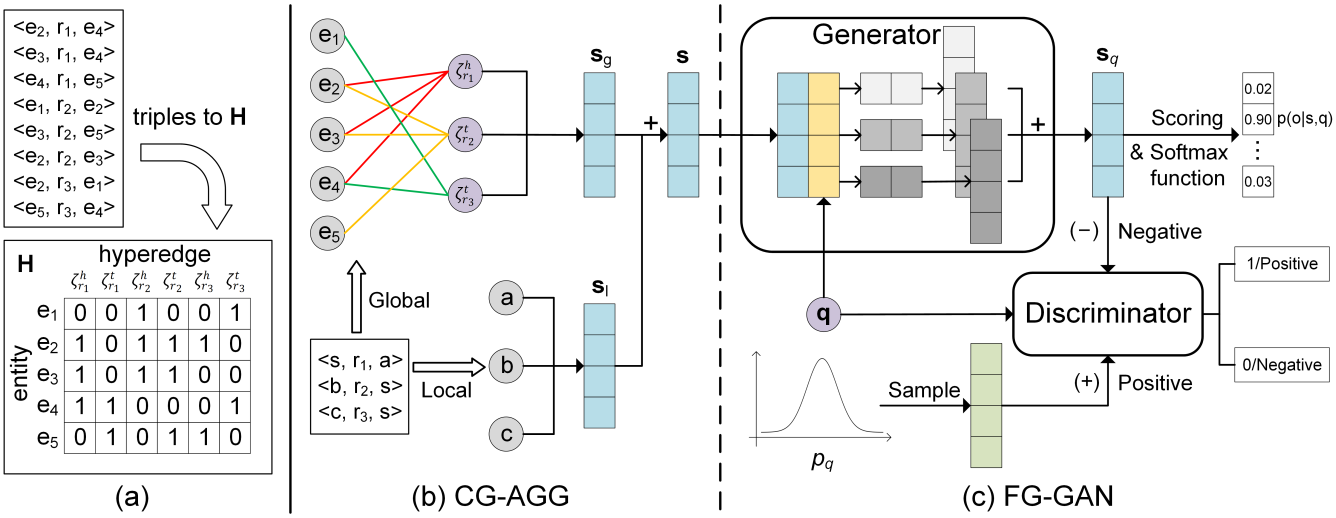
where $\mathbf{E} \in \mathbb{R}^{|\mathcal{E}| \times d_E}$ is the entity embedding matrix. $\mathbf{H} \in \mathbb{R}^{|\mathcal{E}| \times 2|\mathcal{R}|}$ is the incidence matrix. $\mathbf{D} \in \mathbb{R}^{2|\mathcal{R}| \times 2|\mathcal{R}|}$ is a diagonal matrix of hyperedge degree and it is defined as $D_{jj} = \sum_i H_{ij}$. $\mathbf{W} \in \mathbb{R}^{2|\mathcal{R}| \times 2|\mathcal{R}|}$ is a diagonal matrix of hyperedge weights, and we initialize it as an identity matrix, which means equal weights for all hyperedges.

Then we get the representation of s by aggregating the representations of hyperedges related to s:

$$\mathbf{s}_{\mathrm{g}} = \sigma\left(\frac{1}{\sum_j h_{s,j}}\mathbf{h}_s^{\mathrm{T}}\mathbf{Z}\mathbf{\Theta}_{\mathrm{g}}\right), \tag{3}$$

where $\sigma(\cdot)$ is an activation function. $\mathbf{\Theta}_{\mathrm{g}} \in \mathbb{R}^{d_E \times d_A}$ is a trainable weight matrix. $\mathbf{h}_s \in \mathbb{R}^{2|\mathcal{R}|}$ is an incidence vector, i.e., $h_{s,j} = 1$ if $s \in \zeta_j$, and otherwise $h_{s,j} = 0$. The representation is normalized by the vertex degree of s, i.e., the number of hyperedges related to s.

Local Aggregator As a powerful method for processing graph-structured data, GNNs have been widely used in KGs. Similar to traditional GNN-based methods for inductive KGC, we build our local aggregator based on a one-layer GNN to capture local entity information as basic information. As is shown in Figure 2(b), the main idea of our



	$\zeta^h_{r_1}$	$\zeta^t_{r_1}$	$\zeta^h_{r_2}$	$\zeta^t_{r_2}$	$\zeta^h_{r_3}$	$\zeta^t_{r_3}$
e1	0	0	1	0	0	1
e2	1	0	1	1	1	0
e3	1	0	1	1	0	0
e4	1	1	0	0	0	1
e5	0	1	0	1	1	0

Figure 2: (a) is an example of constructing $\mathbf{H}$ using triples. (b) and (c) are the architecture of CFAG. We use the hypergraph constructed in (a), and take the triple $\langle s, q, o\rangle$ as an example. Given some auxiliary triples related to s, the CG-AGG generates s's representation $\mathbf{s}$ using a GNN-based local aggregator and a HGNN-based global aggregator. $\mathbf{s}$ will be used as an input for the generator of FG-GAN. The FG-GAN enforces the generator to generate $\mathbf{s}_q$ conditioned on q to match a corresponding prior distribution p_q. The generator also uses the scoring function and the softmax function to predict the probability of o, so that it can perform KGC task.

local aggregator is to generate the representation of an entity by aggregating the representations of its neighbors. Formally, the local neighborhood aggregator is defined as:

$$\mathbf{s}_\text{l} = \sigma\left(\frac{1}{\sum_i n_{s,i}}\mathbf{n}_s^\text{T}\mathbf{E}\mathbf{\Theta}_\text{l}\right), \tag{4}$$

where $\mathbf{\Theta}_\text{l} \in \mathbb{R}^{d_E \times d_A}$ is a trainable weight matrix. $\mathbf{n}_s \in \mathbb{R}^{|\mathcal{E}|}$ is an incidence vector about s's neighbor entity set $N(s)$, i.e., $n_{s,i} = 1$ if $e_i \in N(s)$, and otherwise $n_{s,i} = 0$.

Finally, we get the new representation of s through the aggregator A which combines the representations obtained from the global aggregator and the local aggregator:

$$\mathbf{s} = A(s) = \alpha \mathbf{s}_\text{l}^\text{T} + (1-\alpha)\,\mathbf{s}_\text{g}^\text{T}, \tag{5}$$

where α is a hyperparameter.

FG-GAN

In order to use fine-grained relational semantics, we associate relations with prior distributions, where different distributions mean different semantics, and use conditional GANs (Mirza and Osindero 2014) to construct our FG-GAN to make entity representations with the same query relation match the same distribution. We build the generator of FG-GAN based on CNNs, and hope it can generate entity representations with specific semantics while have the ability to perform KGC task, which is achieved by jointly optimizing their objective functions in training procedure.

Generator and Discriminator The entity representation obtained by the aggregator mixes multiple semantics, but for a specific query relation q, the entity should have specific semantics. To achieve this, we use a generator G to get an entity representation conditioned on a certain relation. Similar to some methods (Dettmers et al. 2018; Nguyen et al. 2018) for traditional KGC, we build the generator using CNN since it is expressive and efficient:

$$\mathbf{s}_q = G\left(A(s), q\right) = \sigma\left(\sum_{i=1}^{K}\left([\mathbf{s}; \mathbf{R}^T\mathbf{q}]\right) * \mathbf{\Omega}_i\right), \tag{6}$$

where $\mathbf{R} \in \mathbb{R}^{|\mathcal{R}| \times d_A}$ is the relation embedding matrix and $\mathbf{q} \in \mathbb{R}^{|\mathcal{R}|}$ is the one-hot vector of relation q. $*$ denotes a convolution operator. K is the number of filters. $\mathbf{\Omega}_i \in \mathbb{R}^{1\times 2}$ is the i-th filter. The generator takes a query relation q and the output of CG-AGG $A(s)$ as inputs. As is shown in Figure 2(c), it first combines the representations of entity and relation into a matrix, and then performs the convolution operation on this matrix using multiple filters. Finally, it adds the feature maps generated by the filters to a vector, and applies an activation on the vector as the output.

Different relations usually represent different semantics, and entities under the same query relation usually have some similarities. In order to distinguish this difference and similarity, we associate each relation with a prior distribution and make entity representations with the same query relation match the same distribution. We achieve this by an adversarial training procedure with a discriminator:

$$D(\mathbf{x}, q) = \text{Sigmoid}\left(\mathbf{W}_1(\mathbf{W}_0(\mathbf{x}||\mathbf{q}) + b_0) + b_1\right), \tag{7}$$

where $||$ denotes a concatenation operator. $\mathbf{W}$ and b are the parameters of this multi-layer perceptron (MLP). The discriminator is a binary classifier, which can distinguish whether the representation $\mathbf{x}$ is from the prior distribution p_q (positive) or from the generator G (negative). In this paper, we choose the multivariate Gaussian distribution as the prior distribution $p_q(\mathbf{z}) = \mathcal{N}(\mathbf{z}; \boldsymbol{\mu}_q, \boldsymbol{\Sigma}_q)$ where $\boldsymbol{\mu}_q$ and $\boldsymbol{\Sigma}_q$ are the pre-defined mean vector and covariance matrix of relation q respectively (see the experimental setup part for the details of parameter selection).

Objective Function One objective of our model is to enforce the generator to generate the entity representation conditioned on a query relation $\mathbf{s}_q$ that match the prior distribution p_q. We treat the representations generated by the generator as negative examples, and the examples sampled from the prior distributions as positive examples. When training, the generator tries to generate positive examples to deceive the discriminator, while the discriminator tries to distinguish negative examples from positive examples. This adversarial training procedure optimizes the min-max objective function as follows:

$$\begin{aligned}\min_{G,A} \max_{D} V(G, A, D) = \mathbb{E}_{\mathbf{z} \sim p_q}[\log D(\mathbf{z}, q)] + \\ \mathbb{E}_{s \sim p_{data}}[\log(1 - D(G(A(s), q), q)].\end{aligned} \tag{8}$$

Another objective of our model is to perform KGC task. We first define a scoring function which can measure the plausibility of triples being valid:

$$f(s, q, o) = \mathbf{s}_q^T \mathbf{o}, \tag{9}$$

where $\mathbf{o}$ is generated by A. Then we compute the probability of o using the softmax function:

$$p\left(o|s, q\right) = \frac{\exp\left(f\left(s, q, o\right)\right)}{\sum_{o' \in \mathcal{E}} \exp\left(f\left(s, q, o'\right)\right)}. \tag{10}$$

The objective function is defined as a cross-entropy loss:

$$\min_{G,A} \mathcal{L}(G, A) = -\sum_{\langle s,q,o \rangle \in \mathcal{K}} \log p\left(o|s, q\right). \tag{11}$$

Considering these two objectives, the final objective function is:

$$\min_{G,A} \max_{D} \mathcal{L}(G, A) + V(G, A, D) \tag{12}$$

Algorithm 1 shows the whole training procedure of our model. We jointly optimize these two objectives according to Eq. (8) and Eq. (11). Finally, the optimized aggregator and generator will be used to represent the emerging entities and perform KGC task.

Algorithm 1: Model Training

```
Require: a knowledge graph $\mathcal{K}$, the number of iterations $T$,
    the number of samples $m$, $|\mathcal{R}|$ prior distributions $p = \{p_q \mid q \in \mathcal{R}\}$
Ensure: aggregator $A$, generator $G$
for iterator = $1$ to $T$ do
    Sample $m$ positive examples with corresponding relations $\{(\mathbf{z}^i, r^i)\}_{i=1}^m$ from the prior distributions $p$;
    Sample $m$ entities $S = \{s^i\}_{i=1}^m$ from $\mathcal{E}$;
    Sample $m$ relations $\{q^{\ i}\}_{i=1}^m$ from $\mathcal{R}$;
    Update $D$ by Eq. (8) to maxmize:
        $\frac{1}{m}\sum_{i=1}^m \left[\log D(\mathbf{z}^i, r^i) + \log(1 - D(G(A(s^i), q^i), q^i)\right]$
    Update $G$ and $A$ by Eq. (8) to minimize:
        $\frac{1}{m}\sum_{i=1}^m \left[\log(1 - D(G(A(s^i), q^i), q^i)\right]$
    Extract training examples $\{\langle s, q, o \rangle \mid s \in S\}$ from $\mathcal{K}$;
    Update $G$ and $A$ according to Eq. (11);
end for
```

Dataset	$\|\mathcal{E}\|$	$\|\widetilde{\mathcal{E}}\|$	$\|\mathcal{R}\|$	|train|	|aux|	|valid|	|test|
FB15k-237-Sub	10,468	4,072	237	86,629	96,126	5,365	3,735
FB15k-237-Obj	10,728	2,135	237	80,591	115,350	4,025	3,969
NELL-995-Sub	34,770	2,071	200	35,963	17,097	5,281	2,728
NELL-995-Obj	28,812	1,550	200	26,090	14,551	3,640	2,276

Table 1: Statistics of the experimental datasets.

Experiments

Datasets

To evaluate our proposed model, we adopt two benchmark datasets: FB15k-237 (Toutanova and Chen 2015) and NELL-995 (Xiong, Hoang, and Wang 2017). FB15k-237 is a subset of Freebase, and NELL-995 is a subset of NELL. Freebase and NELL are two large knowledge bases that contain a large number of general facts. They cover a variety of relations, such as cities, companies, things, and people.

We adapt the original datasets to fit inductive KGC task in a similar way to (Hamaguchi et al. 2017) and (Wang et al. 2019). We first randomly select 30% of the triples from the original testing set. We choose two different settings to get the initial emerging entities. One is *Subject*, which chooses the head entities in the selected triples as emerging entities. The other is *Object*, which chooses the tail entities in the selected triples as emerging entities. Some initial emerging entities which have no connection with the original training set will be filtered out. The remaining entities constitute the emerging entity set $\widetilde{\mathcal{E}}$. We construct new datasets according to $\widetilde{\mathcal{E}}$.

The original training set is split into the training set and the auxiliary set, where the training set does not contain the emerging entities and each triple in the auxiliary set contains only one emerging entity. The validation set is constructed by removing triples which contain emerging entities from the original validation set. The testing set retains those triples in the selected triples where only the head entities (for *Subject* setting) or the tail entities (for *Object* setting) are emerging entities. Finally, we construct the following datasets for inductive KGC task: FB15k-237-Sub, FB15k-237-Obj, NELL-995-Sub, and NELL-995-Obj, where Sub and Obj represent adopting *Subject* setting and *Object* setting respectively. Table 1 presents the statistics of these datasets.

Evaluation Protocol

When testing, for a test triple $\langle s, q, ? \rangle$, we calculate the score for each possible triple $\langle s, q, o' \rangle$ using the scoring function f. Then we rank the possible entities o' according to the scores in a descending order. Our evaluation is performed on these rankings.

To measure the performance of different methods, we choose two common evaluation metrics, including MRR and Hits@N. MRR is the mean reciprocal ranking of the target entities. Hits@N denotes the proportion of the triples whose target entities are ranked within top N. Higher MRR and Hits@N indicate better performance. We use the filtered

Model	FB15k-237-Sub				FB15k-237-Obj				NELL-995-Sub				NELL-995-Obj			
		Hits@N				Hits@N				Hits@N				Hits@N		
	MRR	1	3	10	MRR	1	3	10	MRR	1	3	10	MRR	1	3	10
MEAN	0.376	27.9	42.2	56.6	0.128	6.6	13.6	25.5	0.296	17.3	33.9	57.4	0.253	17.3	28.3	40.6
LAN	0.359	25.6	40.9	56.1	0.138	7.5	15.1	26.3	0.313	17.5	37.8	60.7	0.250	16.0	29.7	42.6
ELPE	0.371	26.6	41.8	53.3	0.126	6.4	13.0	24.1	0.287	16.7	32.7	57.1	0.248	16.2	27.5	40.2
OOS	0.325	23.8	37.1	48.9	0.132	8.1	14.3	23.7	0.263	16.6	28.6	51.4	0.290	22.0	32.3	42.1
CFAG	**0.393**	**29.5**	**44.0**	**57.5**	**0.156**	**9.4**	**17.0**	**27.6**	**0.360**	**23.9**	**41.9**	**62.0**	**0.312**	**23.8**	**34.2**	**45.9**

Table 2: Experimental results on FB15k-237 and NELL-995.

setting (Bordes et al. 2013) to filter out some fake negatives before ranking.

Baselines

To compare our model against others, we choose four state-of-the-art and open source models as baselines:

- MEAN (Hamaguchi et al. 2017) uses a tailored GNN to compute the embeddings of emerging entities.
- LAN (Wang et al. 2019) introduces a novel aggregator which aggregates neighbors using a both rule- and network-based attention mechanism.
- ELPE (Bhowmik and de Melo 2020) aggregates neighborhood information through a variant of Graph Transformer encoder and provides explainability through reinforcement learning.
- OOS (Albooyeh, Goel, and Kazemi 2020) extends current transductive KGC models to the inductive scenario via aggregation functions that can compute representations of emerging entities.

Experimental Setup

We train our models using Adam (Kingma and Ba 2015) optimizer and use grid search to select the hyperparameters of our model. Hyperparameter ranges are as follows: learning rate lr in $\{0.1, 0.01, 0.005, 0.001\}$, α in Eq. (5) in $\{0, 0.3, 0.5, 0.7, 1.0\}$, embedding size d_E in $\{100, 200, 300\}$, output size of CG-AGG d_A in $\{200, 500, 1000\}$, the number of filters K in $\{10, 50, 100\}$. The optimal hyperparameter configurations are $lr = 0.001, \alpha = 0.5, d_E = 100, d_A = 200$ for all datasets, $K = 50$ for FB15k-237-Sub and FB15k-237-Obj, $K = 10$ for NELL-995-Sub and NELL-995-Obj. We use ReLU as the activation function and multiple multivariate Gaussian distributions as the prior distributions. Before training, we sample $|\mathcal{R}|$ vectors from the uniform distribution on $[0, 1]^{d_A}$. Each vector is associated with a relation, and is regarded as the mean vector of a prior distribution. We choose $0.1I$ as the covariance matrix for all distributions, where I is the identity matrix. Best models are selected by using early stopping according to MRR on the validation sets. The code and datasets are available at https://github.com/changjianw/CFAG.

Results

Experimental results on benchmark datasets are shown in Table 2. We can see that CFAG consistently outperforms all baselines on four datasets. Compared with the second best result, CFAG has the highest relative improvement on Hits@1 in four datasets, by a margin of 5.7%, 16.0%, 36.6%, and 8.1% in FB15k-237-Sub, FB15k-237-Obj, NELL-995-Sub, and NELL-995-Obj respectively. Our model significantly outperforms baselines, especially on Hits@1 metric, which means CFAG can predict the real missing entities more accurately. We also find that even the simplest GNN-based methods MEAN has obtained competitive results on multiple datasets. Experimental results demonstrate our model which utilizes both coarse-grained and fine-grained relational semantics performs better than these neighborhood aggregation baselines. We will also investigate the impact of each part of CFAG in the following parts.

Model	MRR	Hits@1	Hits@3	Hits@10
CFAG w/o A	0.348	23.6	39.7	56.4
CFAG w/o G	0.342	23.5	38.3	56.8
CFAG w/o A, G	0.334	22.3	37.8	54.4
CFAG	**0.360**	**23.9**	**41.9**	**62.0**

Table 3: Ablation results on NELL-995-Sub.

Ablation Studies

In order to investigate the effectiveness of coarse-grained and fine-grained relational semantics, we conduct ablation studies on NELL-995-Sub. More specifically, we remove the global aggregator in CG-AGG (denoted as w/o A) and the adversarial training procedure (i.e. removing the discriminator) in FG-GAN (denoted as w/o G) to construct three new models, and compare them with our full model CFAG. Table 3 shows the results. Compared with CFAG, the performance of "CFAG w/o A, G" declines on all metrics, up to 12.3% relative decrease on Hits@10. Both "CFAG w/o A" and "CFAG w/o G" are better than "CFAG w/o A, G" and worse than CFAG. Experimental results show that coarse-grained and fine-grained relational semantics all contribute to the performance improvement of our model, and considering both can obtain better performance.

To further explore the role of adversarial training, we select four query relations and corresponding head entities from the validation set of NELL-995-Sub, and visualize the representations of entities under different query relations

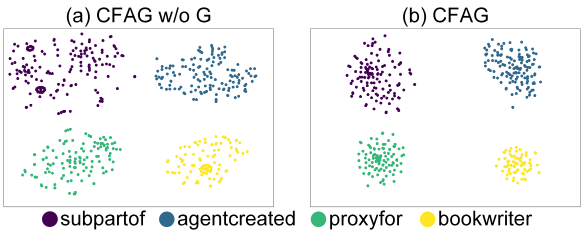


Figure 3: Visualization of entity representations under different query relations.

Model	MRR	Hits@1	Hits@3	Hits@10
TransE*	0.294	-	-	46.5
DistMult+	0.241	15.5	26.3	41.9
R-GCN	0.248	15.3	25.8	41.4
ConvE	0.316	23.9	35.0	49.1
ReInceptionE	**0.349**	-	-	52.8
CFAG	0.342	**24.9**	**37.8**	**52.9**

Table 4: Transductive KGC results on FB15k-237. Results of $*$ are taken from (Nguyen et al. 2018) and results of $+$ are taken from (Dettmers et al. 2018).

using the t-SNE (Van der Maaten and Hinton 2008) algorithm. The results are shown in Figure 3. We can see that the representations generated by CFAG are more compact than “CFAG w/o G”, which means that the adversarial training procedure can help our model generate more accurate entity representations with specific semantics.

Transductive Knowledge Graph Completion

Our CFAG mainly focuses on the inductive KGC, but it can also be used for the transductive KGC. We compare our model with several transductive KGC methods on FB15k-237 and the results are shown in Table 4. Compared with the baselines, our CFAG achieves competitive performance on the transductive KGC. It should be noted that the transductive KGC methods directly train the embeddings of known entities, while our CFAG mainly trains the aggregator A and the generator G which can generalize to the representations of emerging entities. Compared with the transductive KGC methods, our CFAG has more application scenarios.

Impact of the Proportion of Emerging Entities

As mentioned in the datasets part, we mainly select 30% triples from the original testing sets to construct the new testing sets. In this part, we will investigate the impact of the selection percentage. Higher selection percentage means more emerging entities and indicates that the observed knowledge graph becomes sparser. We conduct our experiments on NELL-995 under the *Subject* setting. The datasets are constructed with 10%, 20%, and 30% selection percentages, denoted as Sub10, Sub20, and Sub30 respectively. We experiment on these datasets and compare our model with two baselines LAN and OOS. Table 5 shows the results. We can see that our model consistently outperforms OOS and LAN on the datasets with different selection percentages. We note that as the proportion of emerging entities increases, the performance of all models decreases with varying degrees on the most strict overall metrics MRR and fine-grained metrics Hits@1, but our model performs best. For example, on Hits@1, CFAG drops by 0.8% from S10 to S30, while LAN and OOS drop by 2.0% and 3.6% respectively. The experimental results show that our model is more robust than LAN and OOS on sparse knowledge graphs.

Dataset	Model	MRR	Hits@1	Hits@3	Hits@10
Sub10	LAN	0.320	19.5	37.2	57.4
	OOS	0.286	20.2	30.5	46.7
	CFAG	**0.365**	**24.7**	**41.1**	**60.3**
Sub20	LAN	0.304	17.3	36.7	57.1
	OOS	0.277	18.1	31.1	49.3
	CFAG	**0.361**	**24.3**	**42.6**	**59.2**
Sub30	LAN	0.313	17.5	37.8	60.7
	OOS	0.263	16.6	28.6	51.4
	CFAG	**0.360**	**23.9**	**41.9**	**62.0**

Table 5: Experimental results on NELL-995 with different selection percentages.

Model	MRR	Hits@1	Hits@3	Hits@10
CFAG r/w TransE	0.350	23.6	40.4	58.7
CFAG r/w DistMult	0.319	21.0	36.1	55.9
CFAG	**0.360**	**23.9**	**41.9**	**62.0**

Table 6: Experimental results on NELL-995-Sub with different generators.

Impact of Different Generators

Our CFAG constructs the generator based on CNN. This module can also be replaced with other models. In this part, we investigate the impact of different generators on model performance. We choose TransE and DistMult to construct two new models, denoted as “CFAG r/w TransE” and “CFAG r/w DistMult” respectively. The model comparison results are shown in Table 6. We can see that our generator is a better choice.

Conclusion

In this paper, we focus on the inductive KGC task in KGs, which contains emerging entities that are not in the original KGs. We propose a model called CFAG, which utilizes the relational semantics in KGs at two granularity levels to better represent the emerging entities. CFAG consists of CG-AGG and FG-GAN. The CG-AGG utilizes coarse-grained relational semantics to obtain entity representations with multiple semantics. The FG-GAN utilizes fine-grained relational semantics to generate more accurate entity representations with specific semantics, while keeping predictive capabilities. Experimental results on benchmark datasets show that our model significantly and consistently outperforms state-of-the-art methods.

Acknowledgments

This work is supported by the National Natural Science Foundation of China No.62176252 and No.62102421.

References

Albooyeh, M.; Goel, R.; and Kazemi, S. M. 2020. Out-of-Sample Representation Learning for Knowledge Graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, 2657–2666.

Baek, J.; Lee, D. B.; and Hwang, S. J. 2020. Learning to Extrapolate Knowledge: Transductive Few-shot Out-of-Graph Link Prediction. *Advances in Neural Information Processing Systems*, 33.

Bai, S.; Zhang, F.; and Torr, P. H. 2021. Hypergraph convolution and hypergraph attention. *Pattern Recognition*, 110: 107637.

Bhowmik, R.; and de Melo, G. 2020. Explainable link prediction for emerging entities in knowledge graphs. In *International Semantic Web Conference*, 39–55. Springer.

Bordes, A.; Usunier, N.; Garcia-Duran, A.; Weston, J.; and Yakhnenko, O. 2013. Translating embeddings for modeling multi-relational data. *Advances in Neural Information Processing Systems*, 26.

Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka, E. R.; and Mitchell, T. M. 2010. Toward an architecture for never-ending language learning. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

Chen, J.; He, H.; Wu, F.; and Wang, J. 2021. Topology-Aware Correlations Between Relations for Inductive Link Prediction in Knowledge Graphs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 6271–6278.

Daza, D.; Cochez, M.; and Groth, P. 2021. Inductive Entity Representations from Text via Link Prediction. In *Proceedings of the Web Conference 2021*, 798–808.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, 3844–3852.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018. Convolutional 2d knowledge graph embeddings. In *Thirty-second AAAI Conference on Artificial Intelligence*.

Fatemi, B.; Taslakian, P.; Vázquez, D.; and Poole, D. 2020. Knowledge Hypergraphs: Prediction Beyond Binary Relations. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, 2191–2197.

Feng, Y.; You, H.; Zhang, Z.; Ji, R.; and Gao, Y. 2019. Hypergraph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 3558–3565.

Hamaguchi, T.; Oiwa, H.; Shimbo, M.; and Matsumoto, Y. 2017. Knowledge transfer for out-of-knowledge-base entities: a graph neural network approach. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 1802–1808.

Hamilton, W. L.; Ying, R.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 1025–1035.

Hao, Y.; Zhang, Y.; Liu, K.; He, S.; Liu, Z.; Wu, H.; and Zhao, J. 2017. An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 221–231.

He, Y.; Wang, Z.; Zhang, P.; Tu, Z.; and Ren, Z. 2020. VN Network: Embedding Newly Emerging Entities with Virtual Neighbors. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 505–514.

Jiang, Z.; Gao, J.; and Lv, X. 2021. MetaP: Meta Pattern Learning for One-Shot Knowledge Graph Completion. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2232–2236.

Kingma, D. P.; and Ba, J. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

Lehmann, J.; Isele, R.; Jakob, M.; Jentzsch, A.; Kontokostas, D.; Mendes, P. N.; Hellmann, S.; Morsey, M.; Van Kleef, P.; Auer, S.; et al. 2015. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2): 167–195.

Lin, Y.; Liu, Z.; Sun, M.; Liu, Y.; and Zhu, X. 2015. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 29.

Mirza, M.; and Osindero, S. 2014. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Moon, S.; Shah, P.; Kumar, A.; and Subba, R. 2019. Opendialkg: Explainable conversational reasoning with attention-based walks over knowledge graphs. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 845–854.

Nayyeri, M.; Vahdati, S.; Aykul, C.; and Lehmann, J. 2021. 5* Knowledge Graph Embeddings with Projective Transformations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, 9064–9072.

Nguyen, T. D.; Nguyen, D. Q.; Phung, D.; et al. 2018. A Novel Embedding Model for Knowledge Base Completion Based on Convolutional Neural Network. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 327–333.

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

Pan, S.; Hu, R.; Long, G.; Jiang, J.; Yao, L.; and Zhang, C. 2018. Adversarially regularized graph autoencoder for graph embedding. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2609–2615.

Rossi, A.; Barbosa, D.; Firmani, D.; Matinata, A.; and Merialdo, P. 2021. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 15(2): 1–49.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, 593–607. Springer.

Shi, B.; and Weninger, T. 2018. Open-world knowledge graph completion. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Teru, K.; Denis, E.; and Hamilton, W. 2020. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning*, 9448–9457. PMLR.

Toutanova, K.; and Chen, D. 2015. Observed versus latent features for knowledge base and text inference. In *Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality*, 57–66.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *International Conference on Machine Learning*, 2071–2080. PMLR.

Van der Maaten, L.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11).

Vashishth, S.; Sanyal, S.; Nitin, V.; and Talukdar, P. 2020. Composition-based Multi-Relational Graph Convolutional Networks. In *International Conference on Learning Representations*.

Wang, P.; Han, J.; Li, C.; and Pan, R. 2019. Logic attention based neighborhood aggregation for inductive knowledge graph embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 7152–7159.

Wang, Z.; Zhang, J.; Feng, J.; and Chen, Z. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Xie, R.; Liu, Z.; Luan, H.; and Sun, M. 2017. Image-embodied Knowledge Representation Learning. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, 3140–3146.

Xie, Z.; Zhou, G.; Liu, J.; and Huang, X. 2020. ReInceptionE: Relation-aware inception network with joint local-global structural information for knowledge graph embedding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 5929–5939.

Xiong, W.; Hoang, T.; and Wang, W. Y. 2017. DeepPath: A Reinforcement Learning Method for Knowledge Graph Reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 564–573.

Yadati, N.; Nimishakavi, M.; Yadav, P.; Nitin, V.; Louis, A.; and Talukdar, P. 2019. HyperGCN: a new method of training graph convolutional networks on hypergraphs. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 1511–1522.

Yang, B.; Yih, W.; He, X.; Gao, J.; and Deng, L. 2015. Embedding Entities and Relations for Learning and Inference in Knowledge Bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zeng, H.; Zhou, H.; Srivastava, A.; Kannan, R.; and Prasanna, V. 2020. GraphSAINT: Graph Sampling Based Inductive Learning Method. In *International Conference on Learning Representations*.

Zhang, C.; Yao, H.; Huang, C.; Jiang, M.; Li, Z.; and Chawla, N. V. 2020. Few-shot knowledge graph completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 3041–3048.

Zhou, X.; Niu, L.; Zhu, Q.; Zhu, X.; Liu, P.; Tan, J.; and Guo, L. 2021. Knowledge Graph Embedding by Double Limit Scoring Loss. *IEEE Transactions on Knowledge and Data Engineering*.

Zhu, Q.; Zhou, X.; Wu, J.; Tan, J.; and Guo, L. 2020. A knowledge-aware attentional reasoning network for recommendation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, 6999–7006.