Relational Symmetry based Knowledge Graph Contrastive Learning

Ke Liang*† 1, Yue Liu*† 1, Sihang Zhou2, Xinwang Liu 1, Wenzuan Tu 1
1 School of Computer, National University of Defense Technology, Changsha, Hunan, 410073
2 College of Intelligence Science and Technology, National University of Defense Technology, Changsha, Hunan, 410073
liangke200694@126.com

Abstract
Knowledge graph embedding (KGE) aims to learn powerful representations to benefit various artificial intelligence applications, such as question answering and recommendations. Meanwhile, contrastive learning (CL), as an effective mechanism to enhance the discriminative capacity of the learned representations, has been leveraged in different fields, especially graph-based models. However, since the structures of knowledge graphs (KGs) are usually more complicated compared to homogeneous graphs, it is hard to construct appropriate contrastive sample pairs. In this paper, we find that the entities within a symmetrical structure are usually more similar and correlated. This key property can be utilized to construct contrastive positive pairs for contrastive learning. Following the ideas above, we propose a relational symmetrical structure based knowledge graph contrastive learning framework, termed KGE-SymCL, which leverages the symmetrical structure information in KGs to enhance the discriminative ability of KGE models. Concretely, a plug-and-play approach is designed by taking the entities in the relational symmetrical positions as the positive samples. Besides, a self-supervised alignment loss is used to pull together the constructed positive sample pairs for contrastive learning. Extensive experimental results on benchmark datasets have verified good generalization and superiority of the proposed framework.

Introduction
Knowledge graphs (KGs) benefit many artificial intelligence applications, such as recommendation systems (Wang et al. 2018), logic reasoning (Teru, Denis, and Hamilton 2020), question answering (Ren et al. 2021), and text generation (Song et al. 2020). Motivated by their success, researchers have recently focused on developing better knowledge graph embedding (KGE) models to generate high-quality entity and relation representations for performance improvements.

The recent KGE models can be roughly divided into three categories (Wang et al. 2017; Ji et al. 2021) as follows; (1) translational distance models (e.g., TransE (Bordes et al. 2013), RotaE (Sun et al. 2019), QuatE (Zhang et al. 2019), DualE (Cao et al. 2021), HAKE (Zhang et al. 2020), etc.), (2) semantic matching models (e.g., RESCAL (Nickel, Tresp, and Kriegel 2011), DisMult (Yang et al. 2015), ComplEX (Trouillon et al. 2016), ConvE (Dettmers et al. 2018a), etc.) and (3) GNN-based models (e.g., RGCN (Schlichtkrull et al. 2018), KBGAT (Nathani et al. 2019), COMPGCN (Vashishth et al. 2019), etc.). Motivated by the great success of graph contrastive learning, researchers attempt to integrate contrastive learning mechanisms with KGE for more powerful representations. Since the essence of contrastive learning is to mine the hidden information between samples by pulling together similar samples and pushing away dissimilar samples, constructing high confidence contrastive pairs is important to the discriminative ability of the contrastive learning models. To this end, the existing models construct the contrastive pairs by calculating the semantic similarity estimated by language models, such as Bert (Devlin et al. 2019), etc. Samples with high semantic similarity tend to be combined as positive sample pairs. Although proved to be effective, the performance of these models can be easily affected by the adopted language model. As a consequence, the performance of these models would drop drastically when they are applied to new applications where language models are not finely trained.

To solve the problem, a more stable criterion should be developed. In this paper, we find that structural information could be a good choice. Specifically, entities located in relational symmetrical positions which can be commonly found in knowledge graphs are usually similar and correlated, and this property can be utilized to construct contrastive positive pairs. For example, the Bob Jones are structural symmetrical about Basketball in Fig. 1 which reveals the similar semantics between Bob and Jones (i.e., both playing basketball). In other words, such relational symmetrical structures will naturally bring positive contrastive pairs with similar semantics, even if the labels of the nodes and edges are unknown. However, the existing contrastive KGE models overlook it, thus leading to sub-optimal performance.

Following the above idea, we propose a relational symmetrical structure-based knowledge graph contrastive learning framework, termed KGE-SymCL. It leverages the symmetrical structural information to enhance the discriminative ability of KGE models. Concretely, a novel plug-and-play approach takes the entities in the relational symmetrical positions as the positive samples. Besides, a self-supervised alignment loss is leveraged to pull together the constructed
positive sample pairs for contrastive learning. Extensive experimental results on benchmark datasets have verified our proposed method’s good generalization and superiority. The proposed KGE-SymCL is easy to be adopted to the existing KGE models for different downstream tasks. The main contributions of this paper are summarized as follows:

- We propose a relational symmetrical structure based knowledge graph contrastive learning framework, KGE-SymCL, which is the first work to mine the similar semantics underlying the symmetrical structures in KGs for the contrastive knowledge graph learning.

- We design a plug-and-play strategy for constructing positive contrastive pairs based on the defined relational symmetrical structures, i.e., the entities in the relational symmetrical positions are treated as the positive pairs.

- We integrate our KGE-SymCL with multiple KGE models and conduct the experiments for entity classification and link prediction tasks. The promising performances verify the scalability and generalization of the proposed framework. Besides, the best KGE-SymCL models are also compared to other KGE models on two downstream tasks, demonstrating the superiority of our approach.

**Related Work**

**Knowledge Graph Embedding** Knowledge Graph Embedding (KGE) aims to encode the entities and relations to the low dimensional vector or matrix space. Recent existing KGE models can be roughly divided into three categories (Wang et al. 2017, Ji et al. 2021). (1) Translational distance models (e.g., TransE (Bordes et al. 2013), RotaE (Sun et al. 2019), QuatE (Zhang et al. 2019), DualE (Cao et al. 2021), HAKE (Zhang et al. 2020), etc) leverage distance-based scoring functions and treat the relation as an operation. For example, TransE (Bordes et al. 2013) treats the relation as the addition operation between entities, while RotaE (Sun et al. 2019) regards it as the rotation operation. (2) Semantic matching models are developed based on similarity scoring functions. RESCAL (Nickel, Tresp, and Kriegel 2011) utilizes a bilinear function to associate each entity with a vector to capture its latent semantics. Besides, DisMult (Yang et al. 2015) proposes a multiplicative model to represent the likelihood of the triplets. ConvE (Dettmers et al. 2018a) applies a neural network for similarity modeling. (3) GNN-based models, including RGCN (Schlichtkrull et al. 2018), COMPGCN (Vashishth et al. 2019), KGBAT (Nathani et al. 2019) and SCAN (Shang et al. 2019), leverage GNN to capture the structural characteristics of KGs. For example, RGCN (Schlichtkrull et al. 2018) introduces a relation-specific transformation to integrate relation information with the message aggregation. COMPGCN (Vashishth et al. 2019) proposes various composition operations for triplet scoring. The above KGE models constitute the baselines in our work.

**Graph Contrastive Learning** Graph Contrastive Learning (CL), which aims to mine the hidden information between samples in an unsupervised manner, has recently attracted significant attention. The early works (Velickovic et al. 2019, Hassani and Khasahmad 2020) demonstrate the effectiveness of the mutual information maximization principle (Hjelm et al. 2018) in the node and graph level tasks. After that, GRACE (Zhu et al. 2020), and GraphCL (You et al. 2020) are proposed to pull together the same samples...
across augmented views and push away the others. However, the large number of negative samples leads to high computational and memory costs. To solve these issues, researchers propose various negative-sample-free methods by redundancy reduction principles (Liu et al. 2022b; Bielak, Kajdanowicz, and Chawla 2021) and asymmetrical strategies (Thakoor et al. 2021; Lee, Lee, and Park 2022). Although verified effectiveness, the promising performance of previous works highly depends on the choice of data augmentation schemes, leading to cumbersome manual trial-and-error. Motivated by it, the learnable augmentation methods (You et al. 2021; Suresh et al. 2021) are increasingly proposed. In addition, the augmentations-free methods are also designed to replace augmentations by parameter un-shared encoders (Liu et al. 2022b), or discovering the local structural information and the global semantics information (Lee, Lee, and Park 2022).

Motivated by the success of CL on graphs, a few contrastive KGE models are proposed, but are still in the early stage. To this end, the existing models, such as KRACL (Tan et al. 2022) and SimKGC (Wang et al. 2022), construct the contrastive pairs by calculating the semantic similarity estimated by language models, such as Bert (Devlin et al. 2019), etc. Samples with high semantic similarity tend to be combined as positive sample pairs. Although proved to be effective, the performance of these models can be easily affected by the adopted language model. As a consequence, the performance of these models would drop drastically when they are applied to new applications where language models are not finely trained. Comparatively, the structural semantics, as more stable semantics underlying in all KGSs, are rarely used in the integration of KGE and CL. Besides, the models, such as KGCL (Yang et al. 2022), are developed for specific tasks, like recommendation systems. Thus these models have poor scalability on other tasks. To tackle these issues, our KGE-SymCL is a novel KG contrastive learning method leveraging the symmetrical structural semantics, which is also scaled well on various tasks.

Method

In this section, we will introduce the details of the proposed relational symmetrical structure based contrastive knowledge graph framework KGE-SymCL from two aspects, i.e., Relational Symmetrical Structure Extraction and Relational Symmetrical Contrastive Learning (See Fig. 1).

Preliminary

The knowledge graph is composed of the fact triplets, denoted as $KG = \{ (e_u, r_t, e_v) \mid e_u, e_v \in \mathcal{E}, r_t \in \mathcal{R} \}$, where $\mathcal{E}$ is the entity (i.e., node) set, $\mathcal{R}$ is the relation (i.e., edge label) set, $e_u$ is the head entity, $e_v$ is the tail entity and $r_t$ is the relation between them. Based on KGSs, we define the relation sequence extraction operation as follow, which is important to understand the relational symmetrical structure.

**Definition 1 (Relation Sequence Extraction)** Given the knowledge graph $KG = \{ (e_u, r_t, e_v) \mid e_u, e_v \in \mathcal{E}, r_t \in \mathcal{R} \}$ and the corresponding inverted knowledge graph $KG_{inv} = \{ (e_v, r_t, e_u) \mid \forall (e_u, r_t, e_v) \in KG \}$, Relation Sequence Extraction aims to get the relation sequence $RS(e_u, e_v)$ along the $i^{th}$ path between $e_u$ and $e_v$ on the KG $KG = KG_{\text{pos}}$ or KG$_{\text{inv}}$ if the $i^{th}$ path $P_{KG_{\text{pos}}}(e_u, e_v)$ between $e_u$ and $e_v$ exists in the KG. $RS(e_u, e_v) = \{ F(r_{ia, e}^1), F(r_{ia, e}^2), ..., F(r_{ia, e}^i) \}$, (1)

where $e^i$ represents the $n^{th}$ entity on the $i^{th}$ path, $r_{ia, e}^i$ represents the relation with the head entity $e^i$ and tail entity $e_{ia}^i$ in KG. Besides, $F$ is the symbol function defined as follows:

$$F(r_{ia, e}^i) = \begin{cases} r_{ia, e}^+, (e_u, e_{ia}, e_v, r_{ia}) \in KG \\ r_{ia, e}^-, (e_u, e_{ia}, e_v, r_{ia}) \in KG_{\text{inv}} \end{cases}$$ (2)

Relational Symmetrical Structure Extraction

Many existing graph contrastive learning frameworks (Lin et al. 2022; Hu et al. 2021; Dong et al. 2022) construct the positive pairs based on the structural-semantic similarity of the neighborhood information. However, due to the concrete yet complicated edge labels, the neighbors may not have similar semantics, such as the cases in Fig. 2.

We find that the more deep-in reason for the success of the neighbor-based graph contrastive learning frameworks may be that they find out the nodes with symmetrical positions based on symmetrical structures in homogeneous graphs (See Fig. 3(a)). Surprisingly, we observe that although such symmetrical similarities between entities cannot be found in the neighbors in KGSs, they can still be found in the relational symmetrical structures defined in Def. 2 where the relationships associate with the edge directions are symmetrical. As shown in Fig. 3(b), Bob and Jones have similar semantics since there is relational symmetrical patterns (they both play Basketball) between them (likewise for Bob and Andy). In other words, such relational symmetrical structures will naturally bring positive contrastive pairs with similar semantics. Thus, the contrastive pair construction is reformulated into the relational symmetrical structure extraction in KGSs.

**Figure 2:** Cases for neighbors of opposite semantics in KGSs.

**Figure 3:** Differences between neighborhood structures in homogeneous graphs and relational symmetrical structures in KGSs. Note that in relational symmetrical structures, the relationships (i.e., play, teach, student_of) associate with the directions are symmetrical.

Extraction aims to get the relation sequence $RS^i(e_u, e_v)$ along the $i^{th}$ path between $e_u$ and $e_v$ on the KG $KG = KG_{\text{pos}}$ or KG$_{\text{inv}}$, if the $i^{th}$ the path $P_{KG_{\text{pos}}}(e_u, e_v)$ between $e_u$ and $e_v$ exists in the KG. $RS^i(e_u, e_v) = \{ F(r_{ia, e}^{+1}), F(r_{ia, e}^{+2}), ..., F(r_{ia, e}^{+i}) \}$, (1)

where $e_{ia}^i$ represents the $n^{th}$ entity on the $i^{th}$ path, $r_{ia, e}^{+i}$ represents the relation with the head entity $e_{ia}^i$ and tail entity $e^i$ in KG. Besides, $F$ is the symbol function defined as follows:

$$F(r_{ia, e}^{+i}) = \begin{cases} r_{ia, e}^{+}, (e_u, e_{ia}, e_v, r_{ia}) \in KG \\ r_{ia, e}^{-}, (e_u, e_{ia}, e_v, r_{ia}) \in KG_{\text{inv}} \end{cases}$$ (2)
Figure 4: Illustration of k-hop relational symmetrical structure $RSym_k(e_a)$.

**Definition 2 (k-hop Relational Symmetrical Structure)**

Given the knowledge graph $KG = \{ (e_a, r_i, e_v) \mid e_a, e_v \in \mathcal{E}, r_i \in \mathcal{R} \}$, the $i^{th}$-hop relational symmetrical structure of an anchor entity $e_a$ is denoted as $RSym_k(e_a)$, iff the $i^{th}$ structure exists.

$$RSym_k(e_a) = \{ (e_a, e_p, e_i), RS^i(e_a, e_p) \},$$ (3)

where $e_p$ is the pivot entity, $e_i$ is the target entity which is symmetrical to $e_a$ about $e_p$. According to $e_p$, the structure can be split into two parts, where the relation sequence of the first part should be symmetrical to the second part (i.e., $RS^i(e_i, e_p) = RS^i(e_a, e_p)$). Besides, there are $k$ hops for both two parts, i.e., $len(RS^i(e_a, e_p)) = len(RS^i(e_p, e_i)) = k$.

Following the ideas above, we propose the relational symmetrical structure extraction module, which takes the knowledge graph $KG$, the anchor entity $e_a$ and the hyperparameter $K$ as inputs and outputs the target entity set $P_{e_a}$ for the anchor entity. Concretely, we traverse all of the $2k$-hop structures started from the anchor entity in the given $KG$ and only keep the structures satisfying Def. 2. Note that the uppercase letter $K$ is the upper bounds for the lowercase letter $k$, i.e., $k \leq K$. Assume the quantity of $k$-hop relational symmetrical structures for $e_a$ is $n_k$, we can first get the target structure set $T(e_a, K) = \bigcup_{k \in K} \bigcup_{i \in n_k} RSym_k(e_a)$. Then, the target entity set $P_{e_a}$ for the anchor entity $e_a$ is generated by picking out all the relational symmetrical entities in structures belonging to $T(e_a, K)$. Fig. 4 shows an example for the procedure regard to anchor entity Bob with $K = 2$.

**Relational Symmetrical Contrastive Learning**

We design a simple yet effective contrastive learning framework to leverage the hidden information in the relational symmetrical structures to improve the discriminative ability of the KGE models. The entities in the target entity set $P_{e_a}$ are also treated as the contrastive positive sample candidates. Based on that, we use a self-supervised alignment loss to pull together the positive pairs for contrastive learning. The details will be illustrated as follows.

**Knowledge Graph Encoding**

Our model can be easily scaled to various KGE models for entity encoding, such as RDF2Vec (Ristoski et al. 2019), RGCN (Schlichtkrull et al. 2018), COMPGCN (Vashishth et al. 2019), HAKE (Zhang et al. 2020), CompLEX-DURA (Zhang, Cai, and Wang 2020), and etc. The selected knowledge graph encoder $g(\cdot)$ aims to embed the entity $e$ into the embedding $h_e$ in the latent space.

$$h_e = g(e).$$ (4)

**Contrastive Positive Pair Construction**

The entities in the target entity set $P_{e_a}$ are treated as the positive candidates for the anchor entity $e_a$. Considering the time efficiency, we random sample $m$ entities within $P_{e_a}$ as the positive samples and feed them into the selected KGE model together with the $KG$. Therefore, positive pair set $CP_m(h_{e_a})$ of the anchor entity $e_a$ is generated as follows:

$$CP_m(h_{e_a}) = \{ (h_{e_a}, h_{e_a}^+) \mid e_a \in P_{e_a}, i \in [1, m] \},$$ (5)

where $h_{e_a}^+$ denotes the embedding of the $i^{th}$ positive sample. Fig. 1 shows an example with $m$ set as 6.

**Contrastive Loss**

We design the contrastive loss function based on a self-supervised alignment loss, i.e., MSE loss, used in previous negative-free GCL methods (Ermolov et al. 2021, Thakoor et al. 2021) to pull together the contrastive positive pair $(h_{e_a}, h_{e_a}^+)$ for training:

$$L_{\text{contrastive}} = \frac{1}{m} \sum_{i=1}^{m} \text{MSELoss}(h_{e_a}, h_{e_a}^+),$$

$$= \frac{1}{m} \sum_{i=1}^{m} \| h_{e_a} - h_{e_a}^+ \|^2,$$ (6)

$$= 2 - 2 \frac{1}{m} \sum_{i=1}^{m} \| h_{e_a} \|^2 - \| h_{e_a}^+ \|^2,$$

where $\| \cdot \|$ denotes the 2-norm. Our network is optimized by minimizing the contrastive loss. In this manner, we pull together the positive samples in the latent space, thus improving the discriminative capability of our network.

**Training Objective**

The total training objective of our proposed KGE-SymCL consists of two parts, i.e., the contrastive loss and task loss. It is formulated as follows:

$$L = L_{\text{task}} + \alpha \cdot L_{\text{contrastive}},$$ (7)

where $\alpha$ denotes the trade-off hyper-parameter. Our model is adopted for various task losses. For the entity classification, the cross-entropy loss (Schlichtkrull et al. 2018) is a widely used loss function. Besides, as for the link prediction, there are many types of the loss functions (e.g., the ranking losses (Bordes et al. 2013), the binary logistic regression loss (Ji et al. 2016), the sampled multi-class log loss (Trouillon et al. 2016), the binary cross-entropy loss (Vashishth et al. 2019), and etc.).

**Experiments and Analysis**

In this section, we first introduce the experiment setup. Then we show the performance comparison for two different downstream tasks. Afterward, we conduct the Hyper-parameter experiment to analyze the sensitivity of KGE-SymCL. Moreover, the statistics on relational symmetrical structures and transfer experiments on the SimKGC are introduced for comprehensive analysis on our KGE-SymCL.
Table 1: Four benchmark datasets for entity classification.

| Datasets  | Entities | Relations | Edges   | Classes |
|----------|----------|-----------|---------|---------|
| AIFB     | 8,285    | 45        | 29,043  | 4       |
| MUTAG    | 23,644   | 23        | 74,227  | 2       |
| BGS      | 333,845  | 103       | 916,199 | 2       |
| AM       | 1,666,764| 133       | 5,988,321| 11      |

Table 2: Three benchmark datasets for link prediction.

| Datasets  | Entities | Relations | Train Edges | Val. Edges | Test Edges |
|----------|----------|-----------|-------------|------------|------------|
| WN18RR   | 40,943   | 11        | 86,835      | 3,034      | 3,134      |
| FB15K-237| 14,541   | 237       | 272,115     | 17,535     | 20,466     |
| NELL-995 | 75,492   | 200       | 126,176     | 13,912     | 14,125     |

Table 3: Performance comparison w./w.o. the SymCL framework on entity classification.

| Methods | AIFB     | MUTAG    | BGS      | AM       |
|---------|----------|----------|----------|----------|
| RDF2Vec | 88.88    | 72.06    | 86.21    | 87.88    |
| RDF2Vec-SymCL | 88.88 | 73.53    | 89.66    | 88.99    |
| RGGCN   | 95.83    | 72.21    | 81.38    | 89.19    |
| RGGCN-SymCL | 96.11   | 72.35    | 82.35    | 89.60    |
| COMPGCN | 94.44    | 79.29    | 82.45    | 93.10    |
| COMPGCN-SymCL | 94.44 | 80.90    | 88.24    | 96.55    |

Table 4: Performance comparison between KGE-SymCL with other KGE baselines for entity classification.

| Methods | AIFB     | MUTAG    | BGS      | AM       |
|---------|----------|----------|----------|----------|
|Feat     | 55.55    | 77.94    | 72.41    | 66.66    |
|WL       | 80.55    | 80.88    | 86.20    | 87.37    |
|RDF2Vec  | 88.88    | 72.06    | 86.21    | 87.88    |
|RGGCN    | 95.83    | 72.21    | 81.38    | 89.19    |
| COMPGCN | 94.44    | 79.29    | 82.35    | 93.10    |
|RR-GCN   | 83.33    | 81.67    | 80.00    | 70.00    |
| RGGCN-SymCL | 96.11   | 72.35    | 83.45    | 89.60    |
| COMPGCN-SymCL | 94.44 | 80.90    | 88.24    | 96.55    |

Performance Comparison

Based on Table 3, we observe that our KGE-SymCL makes average 1.92% improvements in accuracy compared to the KGE baselines on the entity classification benchmark datasets. Besides, based on Table 4, we find that our KGE-SymCL makes average 1.25% improvements in accuracy compared to the previous state-of-the-art KGE models. In particular, the COMPGCN-SymCL model achieves better results on the BGS and AM datasets.

Performance Comparison on Entity Classification

Based on Table 3, we observe that our KGE-SymCL makes average 1.92% improvements in accuracy compared to the KGE baselines on the entity classification benchmark datasets. Besides, based on Table 4, we find that our KGE-SymCL makes average 1.25% improvements in accuracy compared to the previous state-of-the-art KGE models. In particular, the COMPGCN-SymCL model achieves better results on the BGS and AM datasets.

Performance Comparison on Link Prediction

Table 5 and Table 6 show the performance comparison between our KGE-SymCL with the KGE models on link prediction. Besides WN18RR and FB15K-237, these two commonly used datasets, we introduce a benchmark dataset, NELL-995, to evaluate KGE models for the diversity of the dataset. Table 5 indicates that our KGE-SymCL makes average boosts on all the metrics. In particular, our SymCL framework improves the DisMult and Hake’s performance on FB15K-237 average of 1.4%, 1.8%, 1.2%, 1.3% on MRR, Hit@1, Hit@3, and Hit@10 separately, and RGGCN-SymCL and COMPGCN-SymCL also achieve great performances on NELL-995. Besides, our KGE-SymCL also outperforms other SOTA KGE models (See Table 7).
TABLE 5: Performance comparison w./w.o. the SymCL framework on link prediction.

| Methods            | WN18RR     | FB15K-237 | NELL-995 |
|--------------------|------------|-----------|----------|
|                     | MRR | Hit@1 | Hit@3 | Hit@10 | MRR | Hit@1 | Hit@3 | Hit@10 | MRR | Hit@1 | Hit@3 | Hit@10 |
| TransE             | 0.231 | 0.021 | 0.409 | 0.533 | 0.289 | 0.193 | 0.326 | 0.473 | 0.249 | 0.095 | 0.377 | 0.471 |
| TransE-SymCL       | 0.233 | 0.022 | 0.411 | 0.535 | 0.290 | 0.195 | 0.326 | 0.486 | 0.255 | 0.102 | 0.379 | 0.475 |
| HAKE               | 0.497 | 0.453 | 0.515 | 0.582 | 0.335 | 0.237 | 0.371 | 0.530 | 0.415 | 0.313 | 0.464 | 0.612 |
| HAKE-SymCL         | 0.497 | 0.454 | 0.515 | 0.585 | 0.346 | 0.248 | 0.384 | 0.544 | 0.419 | 0.318 | 0.468 | 0.616 |
| DisMult            | 0.420 | 0.370 | 0.439 | 0.521 | 0.243 | 0.191 | 0.271 | 0.328 | 0.223 | 0.139 | 0.244 | 0.398 |
| DisMult-SymCL      | 0.421 | 0.371 | 0.441 | 0.522 | 0.260 | 0.215 | 0.282 | 0.339 | 0.224 | 0.140 | 0.248 | 0.401 |
| ComplEx-DURA       | 0.489 | 0.445 | 0.503 | 0.574 | 0.370 | 0.275 | 0.409 | 0.562 | 0.469 | 0.376 | 0.512 | 0.647 |
| ComplEx-DURA-SymCL | 0.491 | 0.448 | 0.504 | 0.576 | 0.371 | 0.276 | 0.411 | 0.566 | 0.469 | 0.376 | 0.512 | 0.647 |
| RGCN               | 0.427 | 0.382 | 0.446 | 0.510 | 0.248 | 0.153 | 0.258 | 0.414 | 0.382 | 0.272 | 0.435 | 0.590 |
| RGCN-SymCL         | 0.432 | 0.396 | 0.447 | 0.510 | 0.249 | 0.159 | 0.270 | 0.435 | 0.394 | 0.287 | 0.443 | 0.603 |
| COMPGCN            | 0.469 | 0.434 | 0.482 | 0.537 | 0.352 | 0.261 | 0.387 | 0.534 | 0.456 | 0.361 | 0.507 | 0.637 |
| COMPGCN-SymCL      | 0.471 | 0.437 | 0.484 | 0.537 | 0.354 | 0.262 | 0.389 | 0.537 | 0.469 | 0.378 | 0.520 | 0.649 |

Table 6: Performance comparison between KGE-SymCL with other KGE baselines for link prediction.

| Methods                   | WN18RR     | FB15K-237 | NELL-995 |
|---------------------------|------------|-----------|----------|
|                           | MRR | Hit@1 | Hit@3 | Hit@10 | MRR | Hit@1 | Hit@3 | Hit@10 | MRR | Hit@1 | Hit@3 | Hit@10 |
| **Ours Proposed Methods** |     |       |       |        |     |       |       |        |     |       |       |        |
| ComplEx-DURA-SymCL        | 0.491 | 0.448 | 0.504 | 0.576 | 0.371 | 0.276 | 0.411 | 0.566 | 0.469 | 0.376 | 0.511 | 0.645 |
| HAKE-SymCL                | 0.497 | 0.454 | 0.515 | 0.585 | 0.346 | 0.248 | 0.384 | 0.544 | 0.419 | 0.318 | 0.468 | 0.616 |
| COMPGCN-SymCL             | 0.471 | 0.437 | 0.484 | 0.537 | 0.354 | 0.262 | 0.389 | 0.537 | 0.469 | 0.378 | 0.520 | 0.649 |

In conclusion, the performance comparison on the above two downstream tasks between KGE-SymCL with other KGE baselines shown in this section demonstrate the generalizability and superiority of our model. In particular, as the most important attribute of the proposed contrastive learning framework, the generalizability of KGE-SymCL is verified from two aspects: (1) KGE-SymCL can be adopted to improve the expressive ability of various KGE models. (2) KGE-SymCL is scalable to various benchmark datasets for different downstream tasks. Moreover, the promising results suggest that the structural information leveraged in our KGE-SymCL indeed helps us to get more powerful and discriminative representations.

Hyper-parameter Experiment

We investigate the influence of the hyper-parameter hop $K$, sampling number $m$, and trade-off weight $\alpha$ in our KGE-SymCL. The COMPGCN-SymCL is selected as the model in this section. As for the scope of the hyper-parameters for both two tasks, i.e., entity classification and link prediction, $K$ is selected in $\{1, 2, 3\}$, and $\alpha$ is searched in $\{0.001, 0.01, 0.1\}$. However, since the benchmark datasets for entity classification are smaller than link prediction, the different scope of $m$ are used, i.e., $\{10, 50, 100, 1000\}$ for link prediction and $\{10, 50, 100\}$ for entity classification. As for $K$ and $m$, we observe that the performance will not fluctuate greatly when $K$ and $m$ are varying in Fig. 5(a) to Fig. 5(d). It demonstrates that KGE-SymCL is insensitive to $K$. 
and \( m \). The reason is that there are many symmetrical structures for each entity, which can enhance the discriminative capability of samples. As for the trade-off hyper-parameter \( \alpha \), we find our KGE-SymCL is much more sensitive to it (See Fig. 5 (e) and Fig. 5 (f)). It is because of the magnitude difference between the contrastive and task loss, i.e., the contrastive loss is usually a hundred times larger than the task loss. The best performances are generally reached when \( \alpha = 0.001 \).

**Extensive Experiment**

**Statistics on the Relational Symmetrical Structure** We counted the number and calculated the proportion of the 1-hop and 2-hop relational symmetrical structures in three link prediction benchmark datasets to demonstrate the universality of such structures. Fig. 5 suggests that there are many relational symmetrical (R-S) structures in these datasets, which suggests the feasibility of the motivation in this work. In particular, compared to WN18RR and FB15K-237, more R-S structures are found in NELL-995, which may also be a reason for the more apparent improvements made by our KGE-SymCL on NELL-995 datasets. Moreover, we intuitively show the existence of the structures with six structures from the real-world datasets (See Fig. 7).

**Transfer Experiments on SimKGC** To demonstrate that the structural information mined by our KGE-SymCL is also helpful for the existing knowledge graph contrastive learning frameworks, we further conduct the transfer experiment on SimKGC. Table 7 shows that our approach makes average 0.4%, 0.2%, 0.2% improvements on MRR, Hit@1 and Hit@3 on the SimKGC. It shows that our KGE-SymCL can be well scaled to other KGE-CL methods and further suggests that the leveraged structural information is a good supplementary.

**Conclusion**

In this paper, we propose a relational symmetrical structure-based knowledge graph contrastive learning framework KGE-SymCL, which leverages the symmetrical structural information in KGs to enhance the discriminative ability of KGE models. Experimental results on benchmark datasets have verified the excellent generalization and superiority of the proposed framework. In the future, we aim to continue...
investigating a method for negative pair construction based on the structural information in KGs. Moreover, we will also develop more powerful knowledge graph contrastive learning frameworks to leverage the different kinds of semantics.

References
Bielak, P.; Kajdanowicz, T.; and Chawla, N. V. 2021. Graph Barlow Twins: A self-supervised representation learning framework for graphs. arXiv preprint arXiv:2106.02466.

Bloehdorn, S.; and Sure, Y. 2007. Kernel methods for mining instance data in ontologies. In The Semantic Web, 58–71. 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.

Cao, Z.; Xu, Q.; Yang, Z.; Cao, X.; and Huang, Q. 2021. Dual quaternion knowledge graph embeddings. In Proceedings of the AAAI Conference on Artificial Intelligence, 6894–6902.

de Boer, V.; Wiellemaker, J.; van Gent, J.; Hildebrand, M.; Isaac, A.; van Ossenbruggen, J.; and Schreiber, G. 2012. Supporting Linked Data Production for Cultural Heritage Institutes: The Amsterdam Museum Case Study. In ESWC.

de Vries, G. K. 2013. A fast approximation of the Weisfeiler-Lehman graph kernel for RDF data. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 606–621. Springer.

Degraeve, V.; Vandewiele, G.; Ongenae, F.; and Van Hoecke, S. 2022. R-GCN: The R Could Stand for Random. arXiv.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018a. Convolutional 2d knowledge graph embeddings. In Proceedings of the AAAI conference on artificial intelligence.

Dettmers, T.; Minervini, P.; Stenetorp, P.; and Riedel, S. 2018b. Convolutional 2d knowledge graph embeddings. In Proceedings of the AAAI conference on artificial intelligence.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 4171–4186. Minneapolis, Minnesota: Association for Computational Linguistics.

Dong, W.; Wu, J.; Luo, Y.; Ge, Z.; and Wang, P. 2022. Node Representation Learning in Graph via Node-to-Neighbourhood Mutual Information Maximization.

Ermolov, A.; Siarohin, A.; Singineto, E.; and Sebe, N. 2021. Whitening for self-supervised representation learning. In International Conference on Machine Learning, 3015–3024. PMLR.

Hassani, K.; and Khasahmadi, A. H. 2020. Contrastive multi-view representation learning on graphs. In Proc. of ICML.

Hjelm, R. D.; Fedorov, A.; Lavoie-Marchildon, S.; Grewal, K.; Bachman, P.; Trischler, A.; and Bengio, Y. 2018. Learning deep representations by mutual information estimation and maximization. In International Conference on Learning Representations.

Hu, Y.; You, H.; Wang, Z.; Wang, Z.; Zhou, E.; and Gao, Y. 2021. Graph-MLP: Node Classification without Message Passing in Graph. CoRR, abs/2106.04051.

Ji, G.; Liu, K.; He, S.; and Zhao, J. 2016. Knowledge Graph Completion with Adaptive Sparse Transfer Matrix. In AAAI.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Philip, S. Y. 2021. A survey on knowledge graphs: Representation, acquisition, and applications. IEEE Transactions on Neural Networks and Learning Systems, 33(2): 494–514.

Lee, N.; Lee, J.; and Park, C. 2022. Augmentation-free self-supervised learning on graphs. In Proceedings of the AAAI Conference on Artificial Intelligence, 7372–7380.

Lin, Z.; Tian, C.; Hou, Y.; and Zhao, W. X. 2022. Improving Graph Collaborative Filtering with Neighborhood-enriched Contrastive Learning. In Proceedings of the ACM Web Conference 2022, 2320–2329.

Liu, Y.; Tu, W.; Zhou, S.; Liu, X.; Song, L.; Yang, X.; and Zhu, E. 2022a. Deep Graph Clustering via Dual Correlation Reduction. In Proceedings of the AAAI Conference on Artificial Intelligence, 7603–7611.

Liu, Y.; Yang, X.; Zhou, S.; and Liu, X. 2022b. Simple Contrastive Graph Clustering. arXiv preprint arXiv:2205.07865.

Liu, Y.; Zhou, S.; Liu, X.; Tu, W.; and Yang, X. 2022c. Improved Dual Correlation Reduction Network. arXiv preprint arXiv:2202.12533.

Nathani, D.; Chauhan, J.; Sharma, C.; and Kaul, M. 2019. Learning Attention-based Embeddings for Relation Prediction in Knowledge Graphs. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics.

Nickel, M.; Tresp, V.; and Kriegel, H.-P. 2011. A three-way model for collective learning on multi-relational data. In Icml.

Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. 2019. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32.

Paulheim, H.; and Fümkranz, J. 2012. Unsupervised generation of data mining features from linked open data. In Proceedings of the 2nd international conference on web intelligence, mining and semantics, 1–12.

Ren, H.; Dai, H.; Dai, B.; Chen, X.; Yasunaga, M.; Sun, H.; Schuurmans, D.; Leskovec, J.; and Zhou, D. 2021. Lego: Latent execution-guided reasoning for multi-hop question answering on knowledge graphs. In International Conference on Machine Learning, 8959–8970. PMLR.

Ristoski, P.; Rosati, J.; Noia, T. D.; Leone, R. D.; and Paulheim, H. 2019. RDF2Vec: RDF graph embeddings and their applications. Semantic Web, 10: 721–752.
Ristoski, P.; Vries, G. K. D. d.; and Paulheim, H. 2016a. A collection of benchmark datasets for systematic evaluations of machine learning on the semantic web. In *International semantic web conference*, 186–194. Springer.

Ristoski, P.; Vries, G. K. D. d.; and Paulheim, H. 2016b. A collection of benchmark datasets for systematic evaluations of machine learning on the semantic web. In *International semantic web conference*, 186–194. Springer.

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Berg, R. v. d.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, 593–607. Springer.

Shang, C.; Tang, Y.; Huang, J.; Bi, J.; He, X.; and Zhou, B. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 3060–3067.

Shervashidze, N.; Schweitzer, P.; Van Leeuwen, E. J.; Mehlhorn, K.; and Borgwardt, K. M. 2011. Weisfeiler-lehman graph kernels. *Journal of Machine Learning Research*, 12(9).

Song, L.; Wang, A.; Su, J.; Zhang, Y.; Xu, K.; Ge, Y.; and Yu, D. 2020. Structural Information Preserving for Graph-to-Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 7987–7998. Online: Association for Computational Linguistics.

Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; and Tang, J. 2019. RotateE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In *International Conference on Learning Representations*.

Suresh, S.; Li, P.; Hao, C.; and Neville, J. 2021. Adversarial graph augmentation to improve graph contrastive learning. *Advances in Neural Information Processing Systems*, 34: 15920–15933.

Tan, Z.; Chen, Z.; Feng, S.; Zhang, Q.; Zheng, Q.; Li, J.; and Luo, M. 2022. Contrastive Learning with Graph Context Modeling for Sparse Knowledge Graph Completion. *arXiv*.

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

Thakoor, S.; Tallec, C.; Azar, M. G.; Azabou, M.; Dyer, E. L.; Munos, R.; Velickovic, P.; and Valko, M. 2021. Large-scale representation learning on graphs via bootstrapping. *arXiv preprint arXiv:2102.06514*.

Toutanova, K.; Chen, D.; Pantel, P.; Poon, H.; Choudhury, P.; and Gamon, M. 2015. Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 1499–1509.

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.

Vashishth, S.; Sanyal, S.; Nitin, V.; and Talukdar, P. 2019. Composition-based Multi-Relational Graph Convolutional Networks. In *International Conference on Learning Representations*.

Velickovic, P.; Fedus, W.; Hamilton, W. L.; Liò, P.; Bengio, Y.; and Hjelm, R. D. 2019. Deep Graph Infomax. *ICLR (Poster)*, 2(3): 4.

Wang, H.; Zhang, F.; Xie, X.; and Guo, M. 2018. DKN: Deep knowledge-aware network for news recommendation. In *Proceedings of the 2018 world wide web conference*, 1835–1844.

Wang, L.; Zhao, W.; Wei, Z.; and Liu, J. 2022. SimKGC: Simple Contrastive Knowledge Graph Completion with Pre-trained Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 4281–4294. Dublin, Ireland: Association for Computational Linguistics.

Wang, Q.; Mao, Z.; Wang, B.; and Guo, L. 2017. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12): 2724–2743.

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.

Yang, B.; Yih, W.-t.; He, X.; Gao, J.; and Deng, L. 2015. Embedding entities and relations for learning and inference in knowledge bases. *International Conference on Learning Representations*.

Yang, Y.; Huang, C.; Xia, L.; and Li, C. 2022. Knowledge Graph Contrastive Learning for Recommendation. *SIGIR*.

You, Y.; Chen, T.; Shen, Y.; and Wang, Z. 2021. Graph contrastive learning automated. In *International Conference on Machine Learning*, 12121–12132. PMLR.

You, Y.; Chen, T.; Sui, Y.; Chen, T.; Wang, Z.; and Shen, Y. 2020. Graph contrastive learning with augmentations. *Advances in Neural Information Processing Systems*, 33: 5812–5823.

Zhang, S.; Tay, Y.; Yao, L.; and Liu, Q. 2019. Quaternion knowledge graph embeddings. *Advances in neural information processing systems*, 32.

Zhang, Z.; Cai, J.; and Wang, J. 2020. Duality-induced regularizer for tensor factorization based knowledge graph completion. *Advances in Neural Information Processing Systems*, 33: 21604–21615.

Zhang, Z.; Cai, J.; Zhang, Y.; and Wang, J. 2020. Learning hierarchy-aware knowledge graph embeddings for link prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 3065–3072.

Zhu, Y.; Xu, Y.; Yu, F.; Liu, Q.; Wu, S.; and Wang, L. 2020. Deep Graph Contrastive Representation Learning. In *ICML Workshop on Graph Representation Learning and Beyond*. 