CKGE: Improving Distance Based Knowledge Graph Embedding via Contrastive Learning

Yafei Liu, Shuaishuai Zu, and Li Li(✉)

School of Computer and Information Science, Southwest University, Chongqing, China
lily@swu.edu.cn

Abstract. Knowledge graph embedding (KGE) is critical in various downstream applications as it represents entities and relations in a knowledge graph as low-dimensional vectors. The embeddings of the entities and relations denote their semantics on the knowledge graph, which affects the effectiveness of the model. Recently, distance-based (DB) models have demonstrated great explanatory power in KGE. However, most existing DB models focus solely on single triples to independently optimize the scoring function, disregarding the interconnections among different triples. To address this issue, we propose CKGE, a novel contrastive learning approach that enhances the performance of DB models while remaining versatile enough to apply to different DB models. Specifically, CKGE improves the alignment and uniformity of DB models, meaning that the embedding of the same semantic entities should remain close under different relations, and embeddings for random entities should scatter on the hypersphere. Additionally, we present a supervised contrastive learning approach to optimize in-batch negative methods, thereby improving the learning of semantic entities. Extensive experiments on four benchmark datasets demonstrate that CKGE yields significant improvements in link prediction, especially for large-scale datasets such as ogbl-wikikg2.

Keywords: Knowledge graph embedding · Distance based model · Contrastive Learning

1 Introduction

Knowledge graphs usually represent structured human knowledge in the form of (head entity, relation, tail entity). Although knowledge graphs usually contain billions of triples, they still suffer from the incompleteness problem due to a lot of factual triples missing, which needs knowledge graph completion (KGC). Knowledge graph embedding (KGE) has been proposed for this problem, which embeds all entities and relations into a low dimensional space and aims to predict missing links between entities.

X. Yang et al. (Eds.): ADMA 2023, LNAI 14177, pp. 3–15, 2023.
https://doi.org/10.1007/978-3-031-46664-9_1

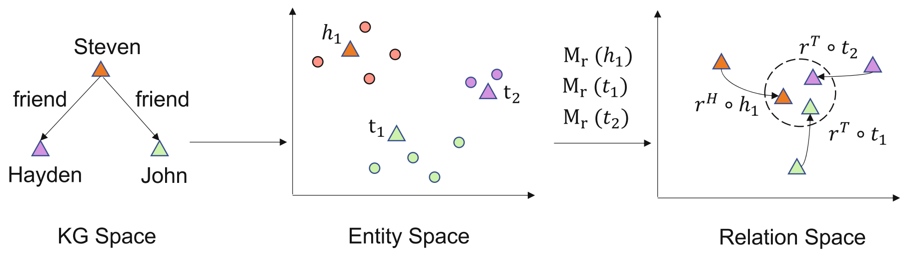


Fig. 1. A toy example showing how DB models can exhibit entities and relations representation on KGs.

Recently, some distance based (DB) models, which use the spatial distance of two entities after the transformation of the relation to judge whether two entities have a certain relation or not, have shown great explanation and power in KGE. In general, according to the way of manipulating relations, we divide the majority of DB models roughly into two groups: translation families and rotation families. In knowledge graph embeddings, translation models such as TransE [1], TransR [10], and TransD [8] primarily focus on addressing relation mappings including 1-to-N, N-to-1, and N-to-N relationships. Meanwhile, the newer rotation models, exemplified by RotatE [17] and PairRE [3], have expanded their scope to cater to a range of relation patterns including symmetry/antisymmetry, inverse, composition, and subrelation. For both categories, the underlying assumption for a valid triple (h, r, t) is that after undergoing a relational transformation r, the head entity h should be proximate to the tail entity t. Intuitively, the smaller the spatial distance between two entities post-transformation, the higher their likelihood of representing a valid relationship in reality.

However, most existing DB models only focus on single triples to independently optimize the scoring function while ignoring the interconnection among different triples. Motivated by the sentence embedding representation [6], alignment and uniformity are also observed by KGE. As shown in Fig. 1, suppose that some triples have different tail entities but share the same head entity and relation like *(Steven, friend, Hayden)* and *(Steven, friend, John)*. For alignment, the tail entities *Hayden* and *John* should be as close as possible to the head entity *Steven* after relation *friend* transformation respectively. While the entities *Steven*, *Hayden*, and *John* should also maintain distance uniformity with other entities in the knowledge graph after relation transformation, which is beneficial to the link prediction task as a measure of the quality of KGE.

Here we propose CKGE, a novel method that effectively constrains entities to improve the performance of KGE via contrastive learning, especially for the DB models. Specifically, Our motivation is based on the observation that the same semantic entities should keep alignment as shown in Fig. 1 and maintain distance uniformity with other entities in the knowledge graph. First, we abstract key procedures from mainstream DB models and present a unified DB model paradigm. Secondly, based on the above paradigm and analysis, we design an in-batch division method for positive and negative samples without extra data.

Thirdly, we further conduct a supervised contrastive learning method to optimize the in-batch negative method using semantic labels, which is able to learn the same semantic entities better. CKGE is widely applicable to various distance based models, including TransE, TransH, PairRE, etc. Experiments show that CKGE yields consistent and significant improvements in datasets for the knowledge graph completion task.

In summary, our main contributions are as follows:

- As far as we know, we are the first to propose the DB model framework with contrastive learning. The proposed CKGE constrains the representation of the same semantic entities in different triples.
- We present a unified DB model paradigm by abstracting diverse DB models, and theoretically prove that CKGE is widely applicable to various DB models.
- Experiments show that CKGE yields consistent and significant improvements on four benchmark datasets for link prediction tasks. It is worth noting that CKGE improves nearly 6% on the large scale knowledge graph (ogbl-wikikg2).

2 Related Work

2.1 KGE Model

Knowledge graph embedding models can be broadly classified into three categories [15]: *distance based models* (DB), *tensor decomposition models* (TDB), and *neural network based Models* (NN).

Distance Models project entities on the knowledge graph into space. Generally, the closer the spatial distance between two entities after the transformation of relation, the greater the probability of validity in the real world. And the score function have the formulation of $s(h_i, r_j, t_k) = -\|\Gamma(h_i, r_j, t_k)\|$, where Γ is a model-specific function. We divide the majority of DB models roughly into two families according to the way they manipulate relations. Moreover, although many research attempts to design more complicated scoring function [2,22], we think that the aforementioned DB models are powerful enough and our proposed unified DB model paradigm is based on these.

Tensor Decomposition Based Models formulate the KGC task as a triadic binary tensor completion challenge. Within the framework of RESCAL [14], each relationship is depicted using a matrix of full rank, with its scoring mechanism defined through a bilinear approach, which is $f_r(\mathbf{h}, \mathbf{t}) = \mathbf{h}^\top \mathbf{M}_r \mathbf{t}$. However, full-rank matrices are prone to overfitting, DistMult [21] defines $\mathbf{M}_r$ as a diagonal matrix to solve it. ComplEx [19] emerged to tackle DistMult's limitations in handling antisymmetric relations, integrating complex-valued embeddings. However, its ability to handle the composition pattern remains limited, and both its spatial and temporal complexities have grown significantly.

Neural Network Based Models leveraging neural architectures have also made good progress in recent years. ConvE [4], R-GCN [16], and KBGAT [12] have incorporated convolutional neural networks, graph convolutional networks, and graph attention networks into KGE, respectively. Yet, due to the opaque nature of NNs, they often lack clear interpretability.

2.2 Contrastive Learning

Contrastive learning seeks to optimize representations by drawing positive pairs closer and distancing negative pairs. This process ensures that similar samples cluster together while dissimilar ones remain distant. This approach has found broad applications in both computer vision and natural language processing. Acting akin to regularization techniques, contrastive learning leverages negative samples to stabilize the loss function within mini-batches. In the realm of knowledge graphs, [20] employs this method for efficient training with an expansive set of negative samples. Yet, their model relies on textual data and pre-trained models, overlooking semantically similar entities.

3 Methods

In this section, we present CKGE. Section 3.1 introduces a consolidated DB model paradigm, encapsulating essential processes of prevalent translation and rotation models. Subsequently, Sects. 3.2 and 3.3 detail unsupervised and supervised CKGE, respectively. The supervised approach utilizes semantically similar entities as labels, enhancing the unsupervised version.

3.1 A Unified DB Model Paradigm

We provide a unified view of several DB models, by showing that they are restricted versions under our paradigm. We first propose a unified version of the distance scoring function:

$$f_r(\boldsymbol{h},\boldsymbol{t}) = \left\|\boldsymbol{r}^{\mathbf{1}} \circ \mathbf{M_r}\left(\boldsymbol{h}\right) + \boldsymbol{b_r} - \boldsymbol{r}^{\mathbf{2}} \circ \mathbf{M_r}\left(\boldsymbol{t}\right)\right\|_{1/2} \tag{1}$$

where $\boldsymbol{h}, \boldsymbol{t}$ represent entities embedding. $\boldsymbol{M_r}(\cdot)$ represents the relation-specific projecting matrix of entity vectors. Inspired by PairRE [3], each relation is characterized as two weight vectors ($\boldsymbol{r}^{\mathbf{1}}$ and $\boldsymbol{r}^{\mathbf{2}}$), corresponding to the transformations towards the head and tail entities, respectively. The symbol $\circ$ illustrates the functional transition induced by relation r, understood as a rotation maneuver within a complex space in RotatE. Next, we show that our unified DB Model Paradigm can cover the ideas of most mainstream DB models.

TransE is the most classic translation based model, Given a triple $(h,\ r,\ t)$, in which entities and relations are projected in Euclidean Space. There is no entity mapping and relation translation. Compared to Eq. 1, the score function of TransE is trivial to be rewritten as removing $\boldsymbol{r}^{\mathbf{1}}$, $\boldsymbol{r}^{\mathbf{2}}$ and $\boldsymbol{M_r}$. The distance score is defined as:

$$f_{\mathbf{TransE}}(\boldsymbol{h},\boldsymbol{t}) = \|\boldsymbol{h} + \boldsymbol{r} - \boldsymbol{t}\|_{1/2} \tag{2}$$

TransX includes TransH, TransR, and TransD, which represent different projection operations for entities. For example, for TransH, $\boldsymbol{M_r}(\cdot)$ is supposed to be projected to the hyperplane with $\boldsymbol{M_r}(\boldsymbol{h}) = \boldsymbol{h} - \boldsymbol{w_r}^\top \boldsymbol{h}\boldsymbol{w_r}$. Compared to Eq. 1, the score function of TransX is trivial to be rewritten as removing $\boldsymbol{r^1}$ and $\boldsymbol{r^2}$. The distance score is defined as:

$$f_{\mathbf{TransX}}(\boldsymbol{h}, \boldsymbol{t}) = \|\mathbf{M_r}(\boldsymbol{h}) + \boldsymbol{b_r} - \mathbf{M_r}(\boldsymbol{t})\|_{1/2} \tag{3}$$

RotatE and PairRE study more relationships pattern compared by TransE and TransX. For a triple (h, r, t), the relation means the rotation of the entity, which is different from the translation in TransE. Compared to Eq. 1, the score function of RotatE/PairRE is trivial to verify as the score function can be rewritten as removing $\boldsymbol{M_r}$. PairRE is the complete model of RotatE with paired vectors for each relation representation, which rotates the head and tail entities separately to better model the sub-relationship. The distance score is defined as:

$$f_{\mathbf{RotatE/PairRE}}(\boldsymbol{h}, \boldsymbol{t}) = \left\|\boldsymbol{r^1} \circ \boldsymbol{h} + \boldsymbol{b_r} - \boldsymbol{r^2} \circ \boldsymbol{t}\right\|_{1/2} \tag{4}$$

3.2 Unsupervised CKGE

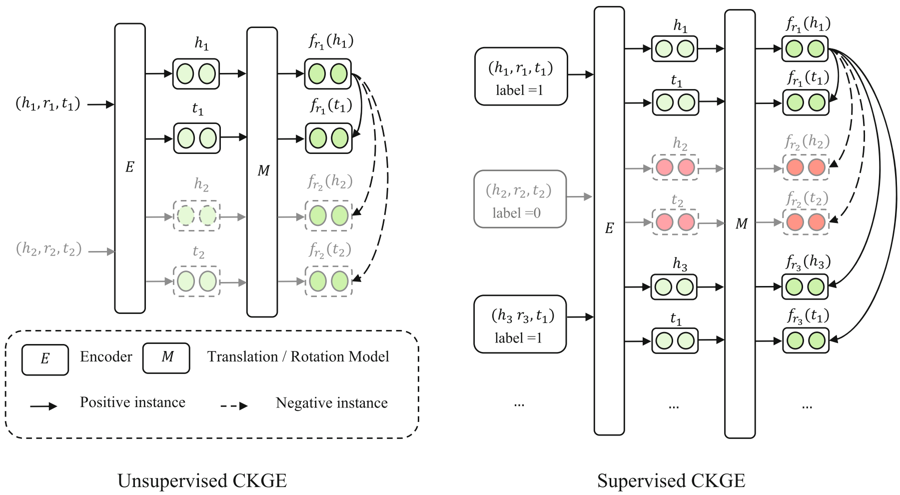


Fig. 2. Illustrations of unsupervised CKGE and supervised CKGE. The supervised CKGE considers similar semantics entities as labels to improve the unsupervised CKGE.

Now we introduce the unsupervised CKGE, and Fig. 2 shows the details. The basic idea of our approach is to treat each triple itself as a positive sample pair and different triples as negative sample pairs. Thus our method generates more negative samples to provide to the model for learning and is able to represent alignment and uniformity better for sparse entities.

For the unified DB Model framework, we would split the distance scoring function into two parts for (h, r, t): $\boldsymbol{r^1} \circ \mathbf{M_r}\ (\boldsymbol{h}) + \boldsymbol{b_r}$ and $\boldsymbol{r^2} \circ \mathbf{M_r}\ (\boldsymbol{t})$ (or $\boldsymbol{r^1} \circ \mathbf{M_r}\ (\boldsymbol{h})$ and $\boldsymbol{b_r} + \boldsymbol{r^2} \circ \mathbf{M_r}\ (\boldsymbol{t})$). Note that we use $f_r(h)$ and $f_r(t)$ to replace $\boldsymbol{r^1} \circ \mathbf{M_r}\ (\boldsymbol{h}) + \boldsymbol{b_r}$ and $\boldsymbol{r^2} \circ \mathbf{M_r}\ (\boldsymbol{t})$ respectively. For DB models, the score function factually uses the distance between $f_r(h)$ and $f_r(t)$ as a basis to judge whether (h, r, t) is valid. For each triple, $f_r(h)$ and $f_r(t)$, the head entity and tail entity after the mapping matrix and relation transformation, are used as positive sample pairs. For the remaining triples in the same mini-batch, the head/tail entity pairs are used as negative sample pairs. For unsupervised CKGE, triples within the same batch are widely used as negative samples. With a mini-batch of N pairs, we adopt the InfoNCE loss function to calculate a sample $f_{r_i}(h_i)$ and its positive sample $f_{r_i}(t_i)$ as the contrastive loss:

$$\mathcal{CL}(\boldsymbol{h_i}) = -\log \frac{e^{\mathrm{sim}(f_{r_i}(h_i), f_{r_i}(t_i))/\tau}}{\sum_{j=1}^{N} (e^{\mathrm{sim}(f_{r_i}(h_i), f_{r_j}(h_j))/\tau} + e^{\mathrm{sim}(f_{r_i}(h_i), f_{r_j}(t_j))/\tau})}, \tag{5}$$

where τ is a temperature hyperparameter and $\mathrm{sim}(f_r(h), f_r(t))$ is the cosine similarity. As shown in Fig. 2, it illustrates unsupervised contrastive learning in CKGE. For head entity and tail entity in the triple(h_i, r_i, t_i), we have:

$$\mathcal{CL}(\boldsymbol{h_i}, \boldsymbol{r_i}, \boldsymbol{t_i}) = \mathcal{CL}(\boldsymbol{h_i}) + \mathcal{CL}(\boldsymbol{t_i}) \tag{6}$$

Within a mini-batch, for each entity after relation transformation, similar semantic entities will have alignment, and the distribution of different semantic entities will have more uniformity.

3.3 Supervised CKGE

Unsupervised CKGE optimizes the different triples associated with one mini-batch, rather than separately during training. However, it still faces certain issues. To take this for example, consider the case where two triples, such as *(tigers, is, mammals)* and *(lions, is, mammals)*, are valid but happen to be in the same mini-batch. Due to *tigers* and *lions* having similar semantics in the query of *"Which entities are mammals?"*, we expect them to have similar embeddings. However, as no labels are available, positive pairs come from the same single triple, while negative pairs are chosen samples from the mini-batch. This causes tigers and lions to be pushed apart as negative pairs due to their presence in different triples, resulting in a negative gain.

To address the aforementioned issue, we draw inspiration from supervised contrastive learning [9] and employ a supervised contrastive learning loss (Eq. 7) to train the model. Specifically, we define the positive samples of those triples that share the same relation and head entity/tail entity as the label. The triples sharing the head or tail entities are treated as positive sample pairs like *(tigers,* *is, mammals)* and *(lions, is, mammals)*. For a given triple, we use all its positive samples in the same mini-batch, and define the improvement loss as follows:

$$\mathcal{CL}(\boldsymbol{h_i}) = -\log \frac{\sum\limits_{a \in P} (e^{\mathrm{sim}(f_{r_i}(h_i), f_{r_a}(h_a))/\tau} + e^{\mathrm{sim}(f_{r_i}(h_i), f_{r_a}(t_a))/\tau})}{\sum_{j=1}^{N} (e^{\mathrm{sim}(f_{r_i}(h_i), f_{r_j}(h_j))/\tau} + e^{\mathrm{sim}(f_{r_i}(h_i), f_{r_j}(t_j))/\tau})}, \tag{7}$$

where P is defined as the set in which triples have the same label in the same mini-batch.

3.4 Training Objective

For a given training triple (h_i, r_i, t_i) in a KG, our framework computes the joint loss as follows:

$$\mathcal{L}(h_i, r_i, t_i) = \mathcal{L}_s + \lambda \mathcal{L}_{cl} \tag{8}$$

where $\mathcal{L}_s$ signifies the scoring function of the DB model. $\mathcal{L}_{cl}$ represents the weighted contrastive loss discussed in Sects. 3.2 and 3.3. λ is a balanced hyper-parameter.

For the scoring function $\mathcal{L}_s$, the self-adversarial negative sampling loss [17] is typically employed for training purposes:

$$\mathcal{L}_s = -\log \sigma(\gamma - f_r(h, t)) - \sum_{i=1}^{n} p(h_i^{'}, r, t_i^{'}) \log \sigma(f_r(h_i^{'}, t_i^{'}) - \gamma) \tag{9}$$

where γ stands for a set margin, while σ denotes the sigmoid function. And we introduce a weighted contrastive loss that assigns λ as a hyper-parameter. $(h_i^{'},\ r,\ t_i^{'})$ refers to the i^{th} negative triple and $p(h_i^{'}, r, t_i^{'})$ indicates the weight assigned to this negative sample. $p(h_i^{'}, r, t_i^{'})$ is defined as follows:

$$p\left((h_i', r, t_i') \mid (h, r, t)\right) = \frac{\exp f_r\left(h_i', t_i'\right)}{\sum_j \exp f_r\left(h_j', t_j'\right)} \tag{10}$$

4 Experiments

4.1 Experimental Setting

Dataset Setting. Our model's effectiveness is assessed through link prediction across four benchmark knowledge graphs: FB15k-237 [18], WN18RR [4], YAGO3-10 [11], and ogbl-wikikg2 [7]. Table 1 offers a summary of these datasets' statistical data. Compared to FB15k and WN18, FB15k-237 and WN18RR have their inverse relations excluded, emphasizing primarily symmetry/antisymmetry and composition patterns. Ogbl-wikikg2, derived from the Wikidata knowledge base, surpasses the scale of other benchmarks by a considerable margin. Navigating complex relation mappings becomes an added challenge besides the standard relation patterns.

Evaluation Protocol. For link prediction evaluation, we employ MR, MRR, and Hit@N as metrics. Given a test triple (h, r, t), the objective is to replace a missing head or tail entity, resulting in pairs like $(h, r, ?)$ or $(?, r, t)$. KGE models rank these triples based on their scores. MR represents the mean rank of accurate entities; a lower value is preferable. MRR calculates the average inverse rank of these entities, while Hit@N determines the fraction of correct entities among the top n. For both MRR and Hit@N, higher values indicate better performance.

Table 1. Entity, relation, and triple counts for the datasets utilized in our experiments.

Dataset	#Entity	#Relation	#Train	#Valid	#Test
FB15k-237	14,541	237	272,115	17,535	20,466
WN18RR	40,943	11	86,835	3,034	3,134
YAGO3-10	123,182	37	1,079,040	5,000	5,000
ogbl-wikikg2	2,500,604	535	16,109,182	429,456	598,543

Implementation and Baseline. For the main experiments, we implement our models based on the implementation of PairRE [3], which is the recent state-of-the-art. In order to maintain a controlled test, all hyperparameters are kept the same with origin experiments except the hyperparameters related to comparative learning like t and λ. And We employ grid search, optimizing hyperparameters based on the validation datasets' performance. Specifically, we search temperature hyperparameter in { 0.5, 0.1, 0.05, 0.01, 0.005 }, and search contrastive learning loss coefficients in { 0.05, 0.1, 1 }. Our proposed models CKGE are implemented by PyTorch 1.12.0 and trained on a Linux server with GTX 3090. We compare the performance of CKGE against several KGE models with different families, including RotatE [17], RotatE3D [5], QuatE [22], DisMult [21], ComplEx [19], ConvE [4], R-GCN [16], and ConvKB [13].

4.2 Main Results

Comparisons for FB15k-237, WN18RR, and YAGO3-10 datasets are shown in Table 2. We can see that our model's performance yields consistent metrics improvements using different datasets compared to other models. Since our model shares the same hyper-parameter settings and implementation with PairRE, comparing it with this state-of-the-art model is fair to show the advantages and disadvantages of the proposed model. It is noteworthy that CKGE works very well on the WN18RR dataset at Hit@1 than PairRE, but is lower than RotatE. Compared to other datasets, the knowledge graph for the WN18RR dataset is denser because the number of relations has only 11. As we will demonstrate in the following experiments, CKGE can theoretically be integrated with any model that fits this paradigm in Sect. 3.1.

Table 2. Link prediction results on FB15k-237, WN18RR, and YAGO3-10. While CKGE's results are from our experiments, data for other models are sourced from their respective papers.

Model	FB5k-237				WN18RR				YAGO3-10		
	Hits		MR	MRR	Hits		MR	MRR	Hits		MRR
	@10	@1			@10	@1			@10	@1	
ConvE	0.491	0.239	246	0.316	0.520	0.400	4187	0.430	0.620	0.350	0.440
R-GCN	0.417	0.153	–	0.248	–	–	-	-	–	–	–
ConvKB	0.483	–	196	0.302	0.508	–	2741	0.220	-	–	–
DisMult	0.485	0.225	301	0.311	0.490	0.390	5100	0.430	0.540	0.240	0.340
ComplEx	0.486	0.227	376	0.313	0.510	0.410	5261	0.440	0.550	0.260	0.360
TransE	0.527	0.231	173	0.329	0.529	0.013	3414	0.223	0.673	0.391	0.492
TransH	0.534	0.236	171	0.335	0.499	0.010	3937	0.214	0.645	0.357	0.357
RotatE	0.533	0.241	177	0.338	0.552	0.417	2923	0.462	0.670	0.402	0.495
RotatE3D	0.543	0.250	165	0.347	**0.579**	**0.442**	3328	**0.489**	–	–	–
QuatE	0.495	0.221	176	0.311	0.564	0.436	3472	**0.481**	–	–	–
PairRE	0.544	0.256	160	0.351	0.522	0.400	2867	0.440	0.675	0.436	0.522
PairRE-CKGE	**0.550**	**0.260**	**155**	**0.355**	0.554	0.426	**2651**	0.463	**0.687**	**0.446**	**0.531**

Additionally, we run CKGE on many different relation mapping types, including 1-1, 1-N, N-1, and N-N. The results of CKGE on different relation categories on FB15k and ogbl-wikikg2 are shown in Table 3. We have observed that our model exhibits excellent performance in heterogeneous relationships such as 1-N, N-1, and N-N, particularly on the ogbl-wikikg2 dataset. This demonstrates that CKGE improves the alignment and uniformity of the DB models.

Table 3. Experimental results on FB15k and ogbl-wikikg2 by relation mapping.

Model	FB15k				ogbl-wikikg2			
	1-1	1-N	N-1	N-N	1-1	1-N	N-1	N-N
TransE	0.887	0.822	0.766	0.895	0.074	0.063	0.400	0.220
ComplEx	**0.939**	0.896	0.822	0.902	0.394	0.278	0.483	0.504
RotatE	0.923	0.840	0.782	0.908	0.164	0.144	0.431	0.261
PairRE	0.785	**0.899**	0.872	**0.940**	0.262	0.270	0.594	0.587
PairRE-CKGE	0.919	0.846	**0.964**	0.935	**0.589**	**0.549**	**0.696**	**0.759**

4.3 Model Analysis

To demonstrate the generality of our approach, we applied CKGE to TransE, TransH, and PairRE in the ogbl-wikikg2 dataset respectively. Table 4 shows the effectiveness of CKGE. We think CKGE will play a huge potential role in the large-scale model in the future. It is worth noting that TransE-CKGE gets an MRR score of 0.355, but TransH-CKGE only gets an MRR score of 0.337. Incorporated with CKGE, TransE gets a 9.2% improvement on MRR, which outperforms the improvement 7.3% score of TransH. Because for TransE, CKGE can constrain more samples, including triples with identical head entities and those with identical head entities and relations. But for TransH, the conditions for CKGE to be effective are more rigorous, only including the samples with the same head entity and relation.

Table 4. Added CKGE to TransE, TransH, and PairRE. Experiment results show that metrics MRR and Hit are significantly improved.

Model	ogbl-wikikg2			
	MRR	Hit@10	Hit@3	Hit@1
TransE	0.263	0.360	0.286	0.206
TranE-CKGE	**0.355**	**0.395**	**0.360**	**0.329**
TransH	0.264	0.360	0.287	0.208
TransH-CKGE	**0.337**	**0.388**	**0.347**	**0.304**
PairRE	0.522	0.621	0.539	0.469
PairRE-CKGE	**0.582**	**0.699**	**0.607**	**0.520**

We compare unsupervised CKGE and supervised CKGE in Table 5. The difference between unsupervised CKGE and supervised CKGE is used unsupervised contrastive learning and supervised contrastive learning. We add them to TransH and PairRE in the FB15k-237 dataset. For TransH-CKGE and PairRE-CKGE used the unsupervised contrastive learning method, the version that used the supervised contrastive learning method brings consistent improvements. Therefore, we adopt supervised CKGE with training as much as possible.

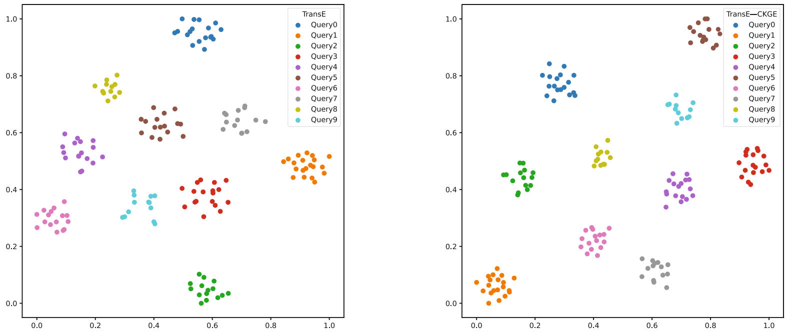
Table 5. Results on unsupervised CKGE and supervised CKGE for FB15k-237 dataset.

Model		FB5k-237				
		MR	MRR	Hit@1	Hit@3	Hit@10
TransH	+unsupervised CKGE	170	0.335	0.236	0.374	0.531
	+supervised CKGE	**168**	**0.336**	**0.238**	**0.375**	**0.533**
PairRE	+unsupervised CKGE	156	0.353	0.258	0.389	0.548
	+supervised CKGE	**155**	**0.355**	**0.260**	**0.394**	**0.550**

4.4 Visualization

We employed T-SNE to visualize triples, highlighting how CKGE promotes consistency and uniformity by aligning entities with analogous semantics post-relation transformation.

Given a query, $(h_i, r_j, ?)$ - with h_i and r_j as the head entities and relations respectively - the goal of link prediction is to determine the valid t_k. From ogbl-wikikg2, we randomly picked 10 queries having a 1-to-N relation mapping type. The entity embeddings generated by TransE are visualized using T-SNE. In Fig. 3, every entity is depicted as a 2D point; points of the same color and number represent different $(h_i, r_j, ?)$ contexts. As evident in Fig. 3, CKGE ensures that entities within the same (h_i, r_j) context have closely aligned and compact representations.



(a) TransE. (b) TransE-CKGE.

Fig. 3. We visualized the embeddings of tail entities using T-SNE. Points sharing a color and number correspond to the same (h, r) context.

5 Conclusion and Future Work

In this study, we introduce CKGE, a contrastive learning framework tailored for distance-based knowledge graph embedding models. We noted that in such models, positive pairs often diverge into subsets: one closely tied to the head entity and the other to the tail entity following a relationship shift. Experimental results reveal that CKGE enhances the efficiency of distance-based models on standard datasets, notably in large-scale graphs. The efficacy of contrastive learning suggests its potential applicability in various other domains, warranting future exploration. Given the power of contrastive learning, a potential avenue for future research is to extend this approach to models in other fields.

References

1. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Advances in Neural Information Processing Systems 26 (2013)
2. Cao, Z., Xu, Q., Yang, Z., Cao, X., Huang, Q.: Dual quaternion knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 6894–6902 (2021)
3. Chao, L., He, J., Wang, T., Chu, W.: Pairre: knowledge graph embeddings via paired relation vectors. arXiv preprint arXiv:2011.03798 (2020)
4. Dettmers, T., Minervini, P., Stenetorp, P., Riedel, S.: Convolutional 2d knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
5. Gao, C., Sun, C., Shan, L., Lin, L., Wang, M.: Rotate3d: representing relations as rotations in three-dimensional space for knowledge graph embedding. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, pp. 385–394 (2020)
6. Gao, T., Yao, X., Chen, D.: Simcse: simple contrastive learning of sentence embeddings. arXiv preprint arXiv:2104.08821 (2021)
7. Hu, W., et al.: Open graph benchmark: datasets for machine learning on graphs. Adv. Neural. Inf. Process. Syst. **33**, 22118–22133 (2020)
8. Ji, G., He, S., Xu, L., Liu, K., Zhao, J.: Knowledge graph embedding via dynamic mapping matrix. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (volume 1: Long papers), pp. 687–696 (2015)
9. Khosla, P., et al.: Supervised contrastive learning. Adv. Neural. Inf. Process. Syst. **33**, 18661–18673 (2020)
10. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. In: Twenty-Ninth AAAI Conference on Artificial Intelligence (2015)
11. Mahdisoltani, F., Biega, J., Suchanek, F.: Yago3: a knowledge base from multilingual wikipedias. In: 7th Biennial Conference On Innovative Data Systems Research. CIDR Conference (2014)
12. Nathani, D., Chauhan, J., Sharma, C., Kaul, M.: Learning attention-based embeddings for relation prediction in knowledge graphs. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pp. 4710–4723 (2019)
13. Nguyen, D.Q., Nguyen, T.D., Nguyen, D.Q., Phung, D.: A novel embedding model for knowledge base completion based on convolutional neural network. arXiv preprint arXiv:1712.02121 (2017)
14. Nickel, M., Tresp, V., Kriegel, H.P.: A three-way model for collective learning on multi-relational data. In: ICML (2011)
15. Rossi, A., Barbosa, D., Firmani, D., Matinata, A., Merialdo, P.: Knowledge graph embedding for link prediction: a comparative analysis. ACM Trans. Knowl. Dis. Data (TKDD) **15**(2), 1–49 (2021)
16. Schlichtkrull, M., Kipf, T.N., Bloem, P., van den Berg, R., Titov, I., Welling, M.: Modeling relational data with graph convolutional networks. In: Gangemi, A., et al. (eds.) ESWC 2018. LNCS, vol. 10843, pp. 593–607. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93417-4_38

17. Sun, Z., Deng, Z.H., Nie, J.Y., Tang, J.: Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv preprint arXiv:1902.10197 (2019)
18. Toutanova, K., Chen, D.: Observed versus latent features for knowledge base and text inference. In: Proceedings of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality, pp. 57–66 (2015)
19. Trouillon, T., Welbl, J., Riedel, S., Gaussier, É., Bouchard, G.: Complex embeddings for simple link prediction. In: International Conference on Machine Learning, pp. 2071–2080. PMLR (2016)
20. Wang, L., Zhao, W., Wei, Z., Liu, J.: Simkgc: simple contrastive knowledge graph completion with pre-trained language models. arXiv preprint arXiv:2203.02167 (2022)
21. Yang, B., Yih, W.t., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases. arXiv preprint arXiv:1412.6575 (2014)
22. Zhang, S., Tay, Y., Yao, L., Liu, Q.: Quaternion knowledge graph embeddings. In: Advances in Neural Information Processing Systems 32 (2019)