Neural Processing Letters (2024) 56:15
https://doi.org/10.1007/s11063-024-11481-8

Knowledge Graph Embedding via Triplet Component Interactions

Tao Wang[1,2] · Bo Shen[1,2] · Jinglin Zhang[1,2] · Yu Zhong[1,2]

Accepted: 28 October 2023 / Published online: 7 February 2024

Abstract

In knowledge graph embedding, multidimensional representations of entities and relations are learned in vector space. Although distance-based graph embedding methods have shown promise in link prediction, they neglect context information among the triplet components, i.e., the head_entity, relation, and tail_entity, limiting their ability to describe multivariate relation patterns and mapping properties. Such context information denotes the entity structural association inside the same triplet and implies the correlation between entities that are not directly connected. In this work, we propose a novel knowledge graph embedding model that explicitly considers context information in graph **e**mbedding via **t**riplet **c**omponent **i**nteractions (TCIE). To build connections between components and incorporate contextual information, entities and relations are represented as vectors comprised of two specialized parts, enabling comprehensive interaction. By simultaneously interacting with one-hop related head and tail entities, TCIE strengthens the connections between distant entities and enables contextual information to be transmitted across the knowledge graph. Mathematical proofs and experiments are performed to analyse the modelling ability of TCIE in knowledge graph embedding. TCIE shows a strong capacity for modelling four relation patterns (i.e., symmetry, antisymmetry, inverse, and composition) and four mapping properties (i.e., one-to-one, one-to-many, many-to-one, and many-to-many). The experimental evaluation of ogbl-wikikg2, ogbl-biokg, FB15k, and FB15k-237 shows that TCIE achieves state-of-the-art results in link prediction.

Keywords Knowledge graph embedding · Knowledge graph completion · Triplet component interaction · Context information

✉ Bo Shen
bshen@bjtu.edu.cn

[1] School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China

[2] Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing 100044, China

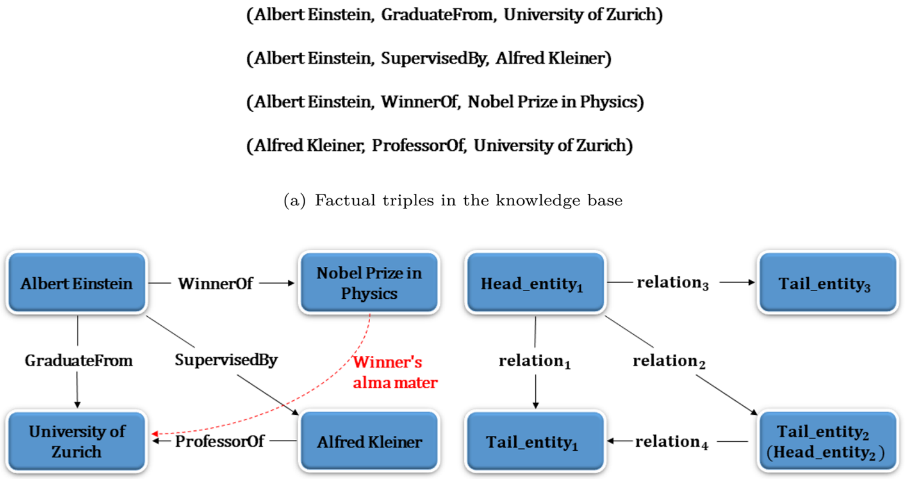
1 Introduction

Knowledge graphs (KGs), a kind of knowledge base, such as WordNet [1], Freebase [2], YAGO [3], and DBpedia [4], store large amounts of structured data in the form of triplets (head_entity, relation, tail_entity), abbreviated as (h, r, t). KGs concisely represent entities and their relationships, providing an effective way to organize, manage and use a vast amount of information on the internet. Therefore, KGs have attracted much attention and play crucial roles in many deep learning domains, including recommender systems [5], question answering [6], and natural language processing [7]. Although knowledge graphs contain a large number of entities and relations, there are still many missing links between entities, which causes KGs to face the dilemma of incompleteness. To solve this problem, predicting missing relations or entities, known as knowledge graph completion or link prediction, is becoming a hot research topic. Knowledge graph embedding (KGE) encodes entities and relations as multidimensional vectors in space, which is considered an efficient link prediction method.

Distance-based KGE models determine the rationality of a triplet by measuring the distances among the head_entity, relation, and tail_entity. TransE heuristically embeds the positive triplets with the translation rule $h + r \approx t$ [8], and can model multiple relation patterns, including antisymmetry, inversion, and composition. TransH [9] defines the relations on the hyperplane and enforces constraints by projecting the entities onto the hyperplane. In TransR [10], when two entities share similar meanings, they are expected to be in close proximity within the physical space. TransH [9] and TransR [10] can model 1-to-N and N-to-1 mapping properties well but cannot model inversion and composition. Inspired by the Euler formula $e^{i\theta} = \cos\theta + i \sin\theta$, RotatE [11] assumes that each relation r is the rotation from the head_entity h to the tail_entity t. RotatE [11] is a promising model for encoding symmetry, antisymmetry, inverse, and composition relation patterns but has difficulties dealing with N-to-1 relations. These models are intuitive and mathematically interpretable, but their performance in simultaneously modelling four relation patterns, i.e., symmetry, antisymmetry, inverse, and composition, and four mapping properties, i.e., one-to-one, one-to-many, many-to-one, and many-to-many, is limited. Distance-based approaches focus on the conditions that should be met among the head_entity, relation, and tail_entity of a positive triplet while ignoring the correlation between entities that are not in the same triplet. We refer to the potential information stored in the knowledge graph structure that reflects the relation patterns and mapping properties as context information.

Figure 1 depicts a straightforward example of a knowledge base and a knowledge graph. As shown in Fig. 1b, the triplet *(Albert Einstein, WinnerOf, Nobel Prize in Physics)*, and *(Albert Einstein, GraduateFrom, University of Zurich)* hold at the same time. Despite the fact that the entities *Nobel Prize in Physics* and *University of Zurich* are not directly connected, they are interconnected and may have a potential relationship, i.e., *Winner's alma mater*. This hypothesis is based on the possible relationship between multihop connected entities in two different triplets. Thus, context information among the head_entity, relation, and tail_entity is indispensable for link prediction. However, most distance-based KGE methods are only committed to properly representing the connections between entities and relations in the positive triplets and do not account for context information when embedding entities and relations.

Therefore, to add contextual information to knowledge graph embeddings, we represent the head_entity and tail_entity as vectors consisting of two specialized parts, $h = [h_r, h_c]$ and $t = [t_r, t_c]$, respectively. The part associated with c is used to capture the contextual information, and the r-related part is used to build the connection between the entity and


(Albert Einstein, GraduateFrom, University of Zurich)
(Albert Einstein, SupervisedBy, Alfred Kleiner)
(Albert Einstein, WinnerOf, Nobel Prize in Physics)
(Alfred Kleiner, ProfessorOf, University of Zurich)
Albert Einstein
WinnerOf
Nobel Prize in Physics
GraduateFrom
SupervisedBy
Winner's alma mater
University of Zurich
ProfessorOf
Alfred Kleiner
Head_entity$_1$
relation$_3$
Tail_entity$_3$
relation$_1$
relation$_2$
Tail_entity$_1$
relation$_4$
Tail_entity$_2$ (Head_entity$_2$)

(a) Factual triples in the knowledge base

(b) Entities and relations in the knowledge graph

(c) Symbols of head_entities, relations, and tail_entities in the knowledge graph

Fig. 1 An example of a knowledge base and knowledge graph. The red dotted line indicates the potential relationship *Winner's alma mater*

relation in the positive triplet. From Fig. 1c, we find that the relation connects the head and tail entities directionally. Therefore, the relation is represented as $r = [r_h, r_t]$. r_h is specifically used to connect to the head_entity, and r_t is used to connect to the tail entity. The entity *Alfred Kleiner* shown in Fig. 1b is the head_entity in triplet *(Alfred Kleiner, ProfessorOf, University of Zurich)* and the tail_entity in triplet *(Albert Einstein, SupervisedBy, Alfred Kleiner)*. This shows that relations can assign different roles to the same entity and have a critical effect on the transmission of context information. Motivated by latent associations among the triplet components, we opine that the embeddings of entities and relations should embody context information about related entities and relations. To achieve this goal, we devise an interaction for h_c, r_h and t_c, r_t to specifically learn context information. In addition, we design another interaction for h_r, r_h, r_t, and t_r, building associations between entities and relations in the positive triplets.

Herein, we propose a new KGE model named TCIE, which conducts interactions to build the connection between entity and relation in the positive triplet and capture the context information stored in the graph structure. The contributions of our paper are summarized as follows:

1. We propose a novel model for knowledge graph embedding, TCIE, which learns contextual information among the triplet components and establishes appropriate associations between entities and relations in positive triplets.
2. TCIE explicitly incorporates context information into the graph embeddings and shows a solid ability to model four relation patterns and mapping properties.
3. Extensive experiments are conducted to evaluate the performance of TCIE in terms of link prediction on six benchmark datasets: ogbl-wikikg2 [18], ogbl-biokg [18], FB15k [8], FB15k-237 [19], WN18 [8], and YAGO3-10 [15]. The experimental results show that with the help of triplet component interactions, TCIE obtains highly competitive results compared with those of state-of-the-art models.

Table 1 Details of some knowledge graph embedding models

Models	Score function	Parameters
TransE [8]	$- \lVert h + r = t \rVert_{1/2}$	$h,\ t,\ r \in \mathbb{R}^k$
TransH [9]	$\lVert (h - {w_r}^{\top} h w_r) + d_r - {w_r}^{\top} t w_r) \rVert_2^2$	$h,\ t,\ w_r,\ d_r \in \mathbb{R}^k$
TransR [10]	$\lVert M_r h + r - M_r t \rVert_2^2$	$h,\ t \in \mathbb{R}^k,\ r \in \mathbb{R}^d,\ M_r \in \mathbb{R}^{k \times d}$
DistMult [12]	$h^{\top} diag(r) \bar{t}$	$h,\ t,\ r \in \mathbb{R}^k$
ComplEx [13]	$Re(h^{\top} diag(r) \bar{t})$	$h,\ t,\ r \in \mathbb{C}^k$
HolE [14]	$r^{\top} (h \star t)$	$h,\ t,\ r \in \mathbb{R}^k$
ConvE [15]	$\langle \sigma(vec(\sigma([\bar{r}, \bar{h}] * \omega))W), t \rangle$	$h,\ t,\ r \in \mathbb{R}^k$
RotatE [11]	$- \lVert h \circ r - t \rVert_1$	$h,\ t,\ r \in \mathbb{C}^k, \lvert \mathrm{r_i} \rvert = 1$
PairRE [16]	$- \lVert h \circ r^H - t \circ r^T \rVert_1$	$h,\ r^H,\ r^T,\ t \in \mathbb{R}^k$
StructurE [17]	$- \lVert h + r - t^c_{edge} \rVert_{1/2} - \lVert t - r - h^c_{edge} \rVert_{1/2}$	$h,\ h_c,\ r,\ h^c_{edge},\ t^c_{edge},\ t,\ t_c \in \mathbb{R}^k$
TCIE (Ours)	$- \lVert h_r \circ r_h - t_r \circ r_t \rVert_1 - \lVert h_c \circ r_t - t_c \rVert_1$	
	$- \lVert t_c \circ r_h - h_c \rVert_1$	$h_r,\ h_c,\ r_h,\ r_t,\ t_r,\ t_c \in \mathbb{R}^k$

$\langle \cdot \rangle$ denotes the generalized dot product

σ denotes activation function and $*$ denotes 2D convolution

$\circ$ denotes the Hadamard product and $\star$ denotes the circular correlation

$\bar{\cdot}$ denotes conjugate for complex vectors, and 2D reshaping for real vectors in the ConvE

2 Related Work

A knowledge graph is a directed heterogeneous multigraph $G = (E, R, T)$, where E, R, and T are sets of entities, relations, and triplets, respectively. A triplet is denoted as *(h, r, t)* $\subseteq E \times R \times E$, where *h* represents the head_entity, *r* represents the relation, and *t* represents the tail_entity. Given a knowledge graph, which is described as a list of fact triplets, the knowledge graph embedding method defines a score function $f_r(h, t)$ to measure the plausibility of a triplet. Knowledge graph completion mainly predicts the tail of a given head and relation *(h, r, ?)* and the head of a given tail and relation *(?, r, t)*.

Typical knowledge graph embedding methods properly encode entities and relations in vector space via the score function [20, 21]. Table 1 summarizes the score functions and parameters of some recent knowledge graph embedding models. We can roughly divide these models into three categories [22]: distance-based models, semantic matching models, and neural network models.

2.1 Distance-Based Models

Distance-based models develop distance-based score functions and usually measure the distance between two entities after a relation has been transformed [23]. TransE [8] was the first approach to use translation distance constraints, which assumes that entities and relations satisfy h + r ≈ t. When the pattern of a relation is symmetric, its vector will be encoded by 0, thus, TransE unable to distinguish symmetric relations. TransH [9], which was proposed to compensate for this demerit of TransE [8], interprets a relation as a translation operation on a hyperplane and has advantages in modelling N-to-1, 1-to-N, and N-to-N mapping properties. In TransR [10], it is assumed that entities have multiple attributes, and various relations can focus on different attributes of entities. TransR models entities and relations in entity space and relation-specific entity space and transforms them into the corresponding relation space. TransR has more parameters than TransE and TransH. Because of its complexity, TransR is challenging to apply to large-scale knowledge graphs. TransD [24]simplifies TransR by setting up two projection matrices, which project the head and tail entities into the relational space. TransD considers the diversity of entities and the low complexity of the model, which is suitable for large-scale knowledge graphs.

RotatE [11] defines each relation as a rotation from the source entity to the target entity in a complex vector space and can model symmetry, asymmetry, inversion, and composition relation patterns. However, it does not perform well in N-to-1 relations. HAKE [25] considers that elements in the knowledge graphs belong to different hierarchical levels and captures semantic hierarchies by mapping entities into a polar coordinate system. PairRE [16] represents each relation with paired vectors and can deal with complex relations and more relation patterns. These translation-based models do not make significant use of context information, which is critical for modelling multivariate relationships. StructurE [17] uses a dual-interaction model to capture both relational structure-context information and edge structure-context information. To model complex relationships, StructurE defines different scoring functions for different relationships, but there is no unified scoring function, and a large number of parameters are needed.

2.2 Semantic Matching Models

Semantic matching models utilize similarity-based score functions. These models measure the plausibility of facts by matching the underlying semantics of entities and relations contained in a vector space representation. RESCAL [26] associates each entity with a vector to capture its potential semantics. Each relation is represented as a matrix that models pairwise interactions between potential factors. DistMult [12] simplifies RESCAL by limiting $\mathrm{M_r}$ to diagonal matrices. For each relation r, DistMult introduces an embedding vector $r \in \mathbb{R}^k$ and requires that $\mathrm{M_r} = \mathrm{diag(r)}$. Nevertheless, only the interactions between the components of h and t in the same dimension are captured. HolE [14] combines the expressive power of RESCAL with the efficiency and simplicity of DistMult. ComplEx [13] extends DistMult by introducing complex-valued embeddings of entities and relations to better model antisymmetric relations. However, despite the significantly increased space and time complexity, ComplEx cannot model composition patterns. QuatE [27] is an extension of ComplEx in hypercomplex space and provides a better spatial interpretation. Semantic matching models can reflect the credibility of semantic information of triplets, but they have defects in encoding relation patterns [28].

2.3 Neural Network Models

Driven by deep learning and machine learning, neural network models for knowledge graph embedding are developing rapidly. ConvE [15] outputs vectors on an input feature map using 2D convolution filters and computes the triplet fraction through the inner product of the output vector and the embedded tail entities. HypER [29] fully convolves head entity embeddings using relation-specific convolutional filters generated by the hypernetwork. Since a hypernetwork is a fully connected network, the interactions and relationships between entities are increased at the expense of the parameters. Neural network models start from the distributed representation of entities and relations. Some utilize complex neural structures such as tensor networks (NTN [30]), graph convolution networks (SACN [31] and R-GCNs [32]), recurrent networks (RSNs [33]), and transformers (CoKE [34] and KG-BERT [35]) to learn richer representation. These neural network models achieve outstanding results but are opaque and lack interpretability.

3 Proposed Method

3.1 Motivation and Overview

Examples of knowledge bases and knowledge graphs are illustrated in Fig. 1. There are four triplets e.g., *(Albert Einstein, WinnerOf, Nobel Prize in Physics)* in Fig. 1b. To properly represent entities and their relationships during graph embedding, first, we need to reasonably construct the appropriate association among the head_entity, relation, and tail_entity in the positive triplet. Second, due to the potential relationships between indirectly related entities, for instance, the entities *Nobel Prize in Physics* and *University of Zurich* may have a potential relationship, i.e., *Winner's alma mater* , such context information between triplet components can provide favourable evidence for link prediction. Naturally, we believe that the representations of entities and relations should not only simulate the association among h, r, and t in the positive triplet *(h, r, t)*, but also contain context information stored in the graph structure.

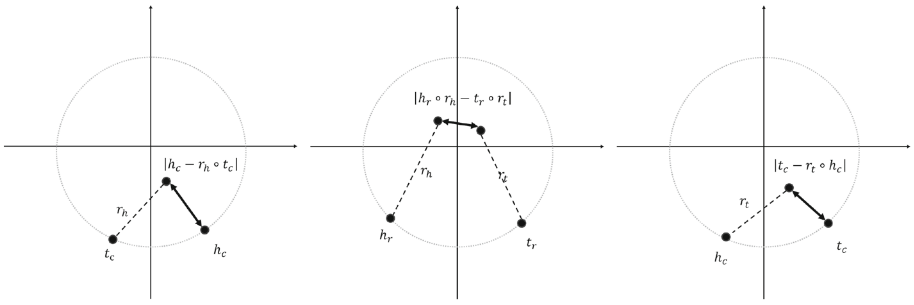
Consequently, to address this challenge, we propose a targeted KGE model, TCIE, which represents entities and relations as vectors consisting of two parts, such as $h = [h_r, h_c]$, $r = [r_h, r_t]$, and $t = [t_r, t_c]$, and conducts three interactions among them to establish appropriate associations between entities and relations in positive triplets and capture contextual information in the graph. For head_entity, $\mathrm{h_c} \circ \mathrm{r_t} \approx \mathrm{t_c}$; for relation, $\mathrm{h_r} \circ \mathrm{r_h} \approx \mathrm{t_r} \circ \mathrm{r_t}$; for tail_entity, $\mathrm{t_c} \circ \mathrm{r_h} \approx \mathrm{h_c}$. An illustration of TCIE is shown in Fig. 2. A relation interaction is devised to build a connection among h, r, and t in the positive triplet, and the head_entity and tail_entity interactions are designed to capture the context information from the related entities and relations.

3.2 TCIE

To facilitate building connections in positive triplets and capturing contextual information, all entity and relation vectors are composed of two parts according to their attributes in knowledge graphs. For relations, head_entity and tail_entity are simultaneously and directly connected. Therefore, the embedding vector of the relation is expected to contain semantic information about the head_entity and tail_entity. Therefore, in TCIE, the embedding vectors of relations should be composed of two parts related to the head_entity and tail_entity, which are denoted as:

$$r = [r_h, r_t] \tag{1}$$

As shown in Fig. 1c, $tail\_entity_2$ is the tail of the triplet ($head\_entity_1$, $relation_2$, $tail\_entity_2$) and the head of the triplet ($head\_entity_2$, $relation_4$, $tail\_entity_1$). Therefore, the embedding vector of $tail\_entity_2$ also contains the context information of these two linked triplets. In addition, $tail\_entity_2$ directly connects $relation_2$ and $relation_4$, which indicates that the embedding vector of $tail\_entity_2$ also contains information about relations. Hence, the embedding vector of tail_entity is designed to include the related relation



(a) Head Entity (b) Relation (c) Tail Entity

Fig. 2 Illustration of TCIE. For head_entity, the embedding vector of h_c will be close to the Hadamard product of r_h and t_c to learn contextual information of related entities and relations. For relation, the vector of h_r and r_h will be close to t_r and r_t after the Hadamard product to build the semantic connection between entities and relations. For tail_entity, the embedding vector of t_c will approach the Hadamard product of r_h and h_c to learn the context information. TCIE employs three interactions to build the association among the h, r, and t in the positive triplet and capture context information stored in the graph structure

information and context information, which is denoted as:

$$t = [t_r, t_c] \tag{2}$$

Head_entity is also directly connected with relations and contains the information of related entities. Therefore, its embedding vector should contain two parts, the relevant relation and context information, and is denoted as:

$$h = [h_r, h_c] \tag{3}$$

During knowledge graph embedding, to let the vectors of the entities and relations explicitly learn context information from the remaining triplet components, we design two interactions. $\circ$ denotes the Hadamard product. For head_entities, h_c is devoted to learning context information from other tail_entities through r_h. The interaction for the head_entity is represented as:

$$\mathrm{h_c} \approx \mathrm{r_h} \circ \mathrm{t_c} \tag{4}$$

For tail_entities, t_c is devised to learn context information from other head_entities through r_t. The interaction for tail_entity is represented as:

$$\mathrm{t_c} \approx \mathrm{r_t} \circ \mathrm{h_c} \tag{5}$$

To build the proper connection among the head_entity, relation, and tail_entity in the positive triplet, we design another interaction for relations. r_h is designed to connect with head entities, and r_t is used to connect with tail entities. The relation is a bridge to connect the head entity and tail entity. Therefore, the interaction for the relation is represented as:

$$\mathrm{h_r} \circ \mathrm{r_h} \approx \mathrm{t_r} \circ \mathrm{r_t} \tag{6}$$

TCIE makes the associated embeddings in triplet components close in geometric space to build associations in the positive triplets and capture context information. An illustration of TCIE is shown in Fig. 2. From Fig. 2a, for the head_entity, the embedding vector of h_c will be close to the Hadamard product of r_h and t_c to learn contextual information of related entities and relations. From Fig. 2b, for the relation, the vector of h_r and r_h will be close to t_r and r_t after the Hadamard product to build the semantic connections between entities and relations. From Fig. 2c, for the tail_entity, the embedding vector of t_c will approach the Hadamard product of r_h and h_c to learn the context information. These interactions are conducive to TCIE capturing context information and strengthening the triplets' associations to better model relation patterns and mapping properties.

3.3 Score Function

We combine the interactions for head_entities, relations, and tail_entities as the score function of our KGE model to measure the plausibility of facts by calculating the Euclidean distance between vector components. The projection operation is the Hadamard product between these two vectors. The distance of two projected vectors will be computed as the plausibility of the triplet. In this paper, the L_1-norm is chosen to measure this distance. The score function is defined as follows:

$$f_r(h,t) = - \parallel h_r \circ r_h - t_r \circ r_t \parallel_1 - \parallel h_c - r_h \circ t_c \parallel_1 - \parallel t_c - r_t \circ h_c \parallel_1, \tag{7}$$

3.4 Loss Function

Knowledge graphs only contain positive triplets. Therefore, we need to randomly replace the head_entity or tail_entity of an existing triplet to construct negative triplets, known as negative sampling. The distances between positive samples should be shorter, and the distances between negative samples should be longer. Many negative sampling methods have been proposed [36–38], among which the self-adversarial negative sampling method [11] dynamically adjusts the weights of negative samples according to the scores during training. We use this self-adversarial negative sampling loss as the objective for training:

$$\begin{aligned} L = &- \log \sigma(\gamma - f_r(h,t)) \\ &- \sum_{i=1}^{n} p(h'_i, r, t'_i) \log \sigma(f_r(h'_i, t'_i) - \gamma), \end{aligned} \tag{8}$$

where σ is the sigmoid function and γ is a fixed margin. (h'_i, r, t'_i) is the i^{th} negative triplet and $p(h'_i, r, t'_i)$ represents the weight of this negative sample. $p(h'_i, r, t'_i)$ is defined as follows:

$$p((h'_i, r, t'_i) \mid (h, r, t)) = \frac{\exp f_r(h'_i, t'_i)}{\sum_j \exp f_r(h'_j, t'_j)}, \tag{9}$$

3.5 Proofs of Modeling Relation Pattern

We deduce the conditions for our TCIE to be able to model the four relation patterns. The main results are as follows:

Proposition 1 *Our model can encode symmetry/antisymmetry relation patterns.*

Proof If $(h_1, r_1, t_1) \in$ T and $(t_1, r_1, h_1) \in$ T, we have

$$\begin{cases} h_{1r} \circ r_{1h} = t_{1r} \circ r_{1t} \\ h_{1c} \circ r_{1t} = t_{1c} \\ t_{1c} \circ r_{1h} = h_{1c} \end{cases} \tag{10}$$

$$\begin{cases} t_{1r} \circ r_{1h} = h_{1r} \circ r_{1t} \\ t_{1c} \circ r_{1t} = h_{1c} \\ h_{1c} \circ r_{1h} = t_{1c} \end{cases} \tag{11}$$

$$\Rightarrow \begin{cases} h_{1r} \circ r_{1h} = t_{1r} \circ r_{1t} \wedge t_{1r} \circ r_{1h} = h_{1r} \circ r_{1t} \\ h_{1c} \circ r_{1t} = t_{1c} \wedge t_{1c} \circ r_{1t} = h_{1c} \\ t_{1c} \circ r_{1h} = h_{1c} \wedge h_{1c} \circ r_{1h} = t_{1c} \end{cases}$$

$$\Rightarrow {r_{1h}}^2 = {r_{1t}}^2 = 1 \tag{12}$$

If r_1 satisfies the symmetry relation pattern, we need ${r_1}_h^2 = {r_{1t}}^2 = 1$.
If $(h_1, r_1, t_1) \in$ T and $(t_1, r_1, h_1) \notin$ T, we have

$$\begin{cases} h_{1r} \circ r_{1h} = t_{1r} \circ r_{1t} \\ h_{1c} \circ r_{1t} = t_{1c} \\ t_{1c} \circ r_{1h} = h_{1c} \end{cases} \tag{13}$$

$$\begin{cases} t_{1r} \circ r_{1h} \neq h_{1r} \circ r_{1t} \\ t_{1c} \circ r_{1t} \neq h_{1c} \\ h_{1c} \circ r_{1h} \neq t_{1c} \end{cases} \tag{14}$$

$$\Rightarrow \begin{cases} h_{1r} \circ r_{1h} = t_{1r} \circ r_{1t} \wedge t_{1r} \circ r_{1h} \neq h_{1r} \circ r_{1t} \\ h_{1c} \circ r_{1t} = t_{1c} \wedge t_{1c} \circ r_{1t} \neq h_{1c} \\ t_{1c} \circ r_{1h} = h_{1c} \wedge h_{1c} \circ r_{1h} \neq t_{1c} \end{cases}$$

$$\Rightarrow {r_{1h}}^2 \neq {r_{1t}}^2 \neq 1 \tag{15}$$

If r_1 satisfies the antisymmetry relation pattern, we need ${r_1}_h^2 \neq {r_{1t}}^2 \neq 1$. □

Proposition 2 *Our model can encode inverse relation pattern.*

Proof If $(h_1, r_1, t_1) \in$ T and $(t_1, r_2, h_1) \in$ T we have

$$\begin{cases} h_{1r} \circ r_{1h} = t_{1r} \circ r_{1t} \\ h_{1c} \circ r_{1t} = t_{1c} \\ t_{1c} \circ r_{1h} = h_{1c} \end{cases} \tag{16}$$

$$\begin{cases} t_{1r} \circ r_{2h} = h_{1r} \circ r_{2t}, \\ t_{1c} \circ r_{2t} = h_{1c}), \\ h_{1c} \circ r_{2h} = t_{1c}) \end{cases} \tag{17}$$

$$\Rightarrow \begin{cases} h_{1r} \circ r_{1h} = t_{1r} \circ r_{1t} \wedge t_{1r} \circ r_{2h} = h_{1r} \circ r_{2t} \\ h_{1c} \circ r_{1t} = t_{1c} \wedge t_{1c} \circ r_{2t} = h_{1c} \\ t_{1c} \circ r_{1h} = h_{1c} \wedge h_{1c} \circ r_{2h} = t_{1c} \end{cases}$$

$$\Rightarrow r_{1h} \circ r_{2h} = r_{1t} \circ r_{2t} = 1 \tag{18}$$

If r_1, r_2 satisfy the inverse relation, we need ${r_1}_h \circ {r_2}_h = r_{1t} \circ r_{2t} = 1$. □

Proposition 3 *Our model can encode composition relation pattern.*

Table 2 Statistical information of the datasets used in experiments

Datasets	#E	#R	#Train	#Valid	#Test
ogbl-wikikg2	2,500,604	535	1,610,912	429,456	598,543
ogbl-biokg	45,085	51	4,762,678	1,628,866	162,870
FB15k	14,951	1345	483,142	50,000	59,071
FB15k-237	14,541	237	272,115	17,535	20,466
WN18	40,943	18	141,442	5000	5000
YAGO3-10	123,182	37	1,079,040	5000	5000

#Test, #Valid and #Train denote the number of test, valid and training triplets, respectively; #R and #E denote the number of relations and entities, respectively

Proof If $(h_1, r_1, t_1) \in \mathrm{T}$, $(t_1, r_2, t_2) \in \mathrm{T}$, and $(h_1, r_3, t_2) \in \mathrm{T}$ we have

$$\begin{cases} h_{1r} \circ r_{1h} = t_{1r} \circ r_{1t} \\ h_{1c} \circ r_{1t} = t_{1c} \\ t_{1c} \circ r_{1h} = h_{1c} \end{cases} \tag{19}$$

$$\begin{cases} t_{1r} \circ r_{2h} = t_{2r} \circ r_{2t} \\ t_{1c} \circ r_{2t} = t_{2c} \\ t_{2c} \circ r_{2h} = t_{1c} \end{cases} \tag{20}$$

$$\begin{cases} h_{1r} \circ r_{3h} = t_{2r} \circ r_{3t} \\ h_{1c} \circ r_{3t} = t_{2c} \\ t_{2c} \circ r_{3h} = h_{1c} \end{cases} \tag{21}$$

$$\begin{aligned} &\Rightarrow \begin{cases} h_{1r} \circ r_{1h} = t_{1r} \circ r_{1t} \wedge t_{1r} \circ r_{2h} = t_{2r} \circ r_{2t} \\ \wedge h_{1r} \circ r_{3h} = t_{2r} \circ r_{3t} \\ h_{1c} \circ r_{1t} = t_{1c} \wedge t_{1c} \circ r_{2t} = t_{2c} \wedge h_{1c} \circ r_{3t} = t_{2c} \\ t_{1c} \circ r_{1h} = h_{1c} \wedge t_{2c} \circ r_{2h} = t_{1c} \wedge t_{2c} \circ r_{3h} = h_{1c} \end{cases} \\ &\Rightarrow r_{1h} \circ r_{2h} = r_{3h}, r_{1t} \circ r_{2t} = r_{3t} \end{aligned} \tag{22}$$

If r_1, r_2, and r_3 satisfy the composition relation pattern, we need $r_{1\,h} \circ r_{2\,h} = r_{3\,h}, r_{1t} \circ r_{2t} = r_{3t}$. □

4 Experiments

4.1 Datasets

Six commonly used standard datasets are used in our link prediction experiments: ogbl-wikikg2 [18], ogbl-biokg [18], FB15k [8], FB15k-237 [19], WN18 [8], and YAGO3-10 [15]. The statistical information of these datasets is summarized in Table 2.

ogbl-wikikg2 is extracted from the Wikidata knowledge base [39]. The main challenge for this dataset is complex relations.

ogbl-biokg contains data from a large number of biomedical data repositories. The main challenge for this dataset is symmetry relations.

FB15k is a subset of Freebase [2], and the main relation patterns are inverse, symmetric and antisymmetric.

FB15k-237 is a subset of FB15k and has no inverse relations. The challenge of link prediction on FB15k-237 is how to handle composition patterns.

WN18 is extracted from WordNet [1], the main challenges of link prediction on WN18 are to model inversion and symmetry relations.

YAGO3-10 is a YAGO3 sample proposed by [15]. This dataset was built by identifying all entities with at least ten different relations in the KG and extracting all the corresponding facts.

4.2 Evaluation Protocol

We use the mean rank (MR), mean reciprocal rank (MRR), and Hits@N (the proportion of correct entities ranked in the top N, where N = 1, 3, and 10) as the evaluation metrics. The lower the MR is, the better, while the higher the MRR and Hits@N are, the better. For ogbl-wikikg2 and ogbl-biokg, we follow the settings in the work by [18], using the average of multiple experiments to reflect the volatility of the results. For the other datasets, we follow the mainstream practice to select the best results for comparison. The embedding dimension is the same for h, r, and t in our model. The embedding size of the vector components h_r, h_c, r_h, r_t, t_r, and t_c is half that of h, r, and t.

Through the grid search method, hyperparameters are selected according to the performance of the model on the validation dataset. We set the range of the hyperparameter as follows: temperature of sampling a ∈ { 0.5, 1.0 }, embedding size k ∈ { 100, 200, 500, 1500, 2000, 2500 }, batch size b ∈ { 512, 1024 }, fixed margin $\gamma \in \{ 3, 4, 5, 6, 7, 8, 12 \}$, $\delta \in \{ 0.2, 0.4, 0.6, 0.8, 0.9 \}$, and number of negative samples for each observed triplet n ∈ { 128, 256, 512, 1024 }. We add two additional coefficients to the score function at training, i.e. $f_r(h, t) = -\lambda_1 \parallel h_r \circ r_h - t_r \circ r_t \parallel_1 -\lambda_2 \parallel h_c - r_h \circ t_c \parallel_1 -\lambda_2 \parallel t_c - r_t \circ h_c \parallel_1$, where $\lambda_1, \lambda_2 \in [0, 1], \lambda_1 + 2\lambda_2 = 1$. The optimal configurations of our model are:

a=1.0, k=100, b=1024, n=128, λ_1=0.6, λ_2=0.2, γ=4 on ogbl-wikikg2;
a=1.0, k=200, b=1024, n=128, λ_1=0.6, λ_2=0.2, γ=6 on ogbl-wikikg2;
a=1.0, k=400, b=1024, n=128, λ_1=0.4, λ_2=0.3, γ=12 on ogbl-biokg;
a=1.0, k=2000, b=1024, n=128, λ_1=0.4, λ_2=0.3, γ=12 on ogbl-biokg;
a=1.0, k=2500, b=1024, n=256, λ_1=0.9, λ_2=0.05, γ=17 on FB15k;
a=1.0, k=1500, b=1024, n=256, λ_1=0.8, λ_2=0.1, γ=4 on FB15k-237;
a=0.5, k=500, b=512, n=1024, λ_1=0.9, λ_2=0.05, γ=8 on WN18;
a=1.0, k=500, b=1024, n=512, λ_1=0.8, λ_2=0.1, γ=17 on YAGO3-10.

4.3 Main Results

The experimental results of knowledge graph completion on the Open Graph Benchmark [18] datasets are shown in Table 3. To incidentally explore the effect of dimensionality on the model performance, we used two embedding sizes for all models. TCIE outperforms all baselines significantly on two large datasets, ogbl-wikikg2 and ogbl-biokg. Compared with the test MMRs obtained using PairRE, our model obviously improves the test MRRs by 23% on ogbl-wikikg2 (dimension 100), 18% on ogbl-wikikg2 (dimension 200), and 0.67% on ogbl-biokg (dimension 2000). On the ogbl-wikikg2 and ogbl-biokg2 dataset, TCIE performs best using both finite and increased embedding sizes. As the dimension increases, the test

Table 3 Link prediction results on ogbl-wikikg2 and ogbl-biokg

Model	ogbl-wikikg2			ogbl-biokg		
	Dim	Test MRR	Valid MRR	Dim	Test MRR	Valid MRR
TransE [8]	100	0.2622± 0:0045	0.2465±0:0020	◇ 400	0.6996±0.0003	0.7002 ±0.0004
DistMult [12]	100	0.3447±0:0082	0.3150±0:0088	◇ 400	0.7987±0.0005	0.7998 ±0.0003
ComplEx [13]	50	0.3804±0:0022	0.3534±0:0052	◇ 200	0.8004±0.0002	0.8005 ±0.0004
RotatE [11]	50	0.2530±0:0034	0.2250±0:0035	◇ 400	0.7495±0.0003	0.7497 ±0.0005
PairRE [16]	100	0.4849±0.0029	0.4941±0.0035	◇ 400	0.7982±0.0008	0.7988 ±0.0007
TCIE	100	**0.5975± 0.0027**	**0.6113± 0.0032**	400	**0.8026±0.0003**	**0.8034 ±0.0005**
TransE [8]	500	0.4256± 0.0030	0.4272± 0.0030	2000	0.7452± 0.0004	0.7456± 0:0003
DistMult [12]	500	0.3729± 0.0045	0.3506± 0.0042	2000	0.8043± 0.0003	0.8055± 0.0003
ComplEx [13]	250	0.4027± 0.0027	0.3759± 0.0016	1000	0.8095± 0.0007	0.8105± 0.0001
RotatE [11]	250	0.4332±0.0025	0.4353±0.0028	1000	0.7989±0.0004	0.7997±0.0002
PairRE [16]	200	0.5208±0.0027	0.5423±0.0020	2000	0.8164±0.0005	0.8172±0.0005
TCIE	200	**0.6159±0.0024**	**0.6245±0.0023**	2000	**0.8219±0.0006**	**0.8228±0.0005**

Results marked with ◇ are our experimental results by using the code from [16], other results are taken from [16]. The best results are in bold

Table 4 Link prediction results on FB15k and FB15k-237

Model	FB15k					FB15k-237				
	MR	MRR	Hit@1	Hit@3	Hit@10	MR	MRR	Hit@1	Hit@3	Hit@10
TransE [8]	34	0.737	0.650	0.799	0.874	172	0.334	0.238	0.371	0.523
TransH [9]	**32**	0.748	0.661	0.813	0.884	168	0.339	0.243	0.375	0.531
TransD [24]	33	0.750	0.664	0.817	0.886	172	0.330	0.235	0.365	0.518
DistMult [12]	59	0.789	0.730	0.830	0.887	301	0.311	0.225	0.341	0.485
ComplEx [13]	63	0.809	0.757	0.846	0.894	376	0.313	0.227	0.342	0.486
ConvE [15]	64	0.745	0.670	0.801	0.873	246	0.316	0.239	0.350	0.491
RotatE [11]	40	0.797	0.746	0.830	0.884	177	0.338	0.241	0.375	0.533
PairRE [16]	38	0.811	0.765	0.845	0.896	160	0.351	0.256	0.387	0.544
StructurE [17] †	44	0.805	0.750	0.843	0.893	160	0.351	0.252	0.390	0.546
TCIE	48	**0.814**	**0.767**	**0.846**	**0.896**	**158**	**0.357**	**0.261**	**0.394**	**0.546**

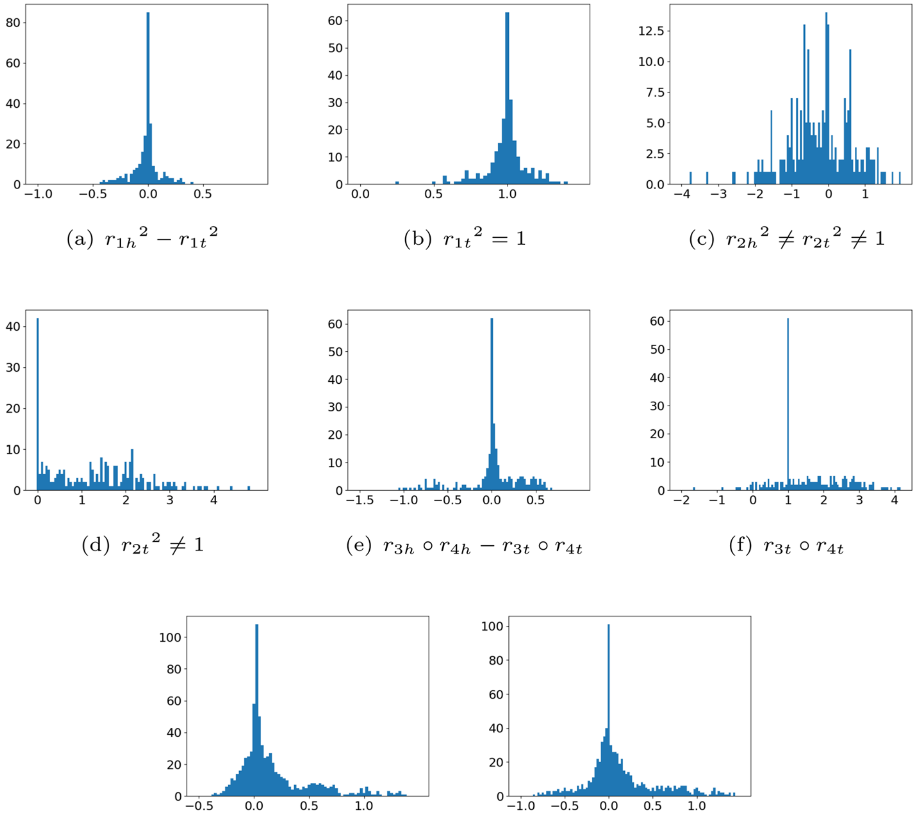
† represents results are from [17], and other results are taken from [40]

Table 5 Link prediction results on WN18 and YAGO3-10

Model	WN18					YAGO3-10				
	MR	MRR	Hit@1	Hit@3	Hit@10	MR	MRR	Hit@1	Hit@3	Hit@10
TransE [8]	**145**	0.821	0.713	0.930	0.955	‡ 1384	0.302	0.194	0.357	0.519
TransH [9]	452	0.823	0.721	0.929	0.954	‡ 1609	0.378	0.265	0.441	0.589
TransD [24]	261	0.822	0.722	0.926	0.956	‡ 1658	0.372	0.256	0.439	0.585
DistMult [12]	496	0.810	0.225	0.694	0.922	◇ 5926	0.34	0.24	0.38	0.54
ComplEx [13]	531	0.948	0.945	0.949	0.953	◇ 6351	0.36	0.26	0.40	0.55
ConvE [15]	504	0.942	0.935	0.947	0.955	◇ 1671	0.44	0.35	0.49	0.62
RotatE [11]	309	0.949	0.944	0.952	0.959	◇ 1767	0.495	0.402	0.550	0.670
PairRE [16]	260	0.946	9.939	0.949	0.957	1358	0.522	0.434	0.576	0.682
StructurE [17] †	249	**0.951**	0.945	0.953	**0.960**	2744	**0.536**	**0.450**	**0.592**	**0.687**
TCIE	260	0.950	**0.945**	**0.953**	0.958	**1166**	0.528	0.437	0.582	0.685

The results of PairRE and those marked with ‡ are obtained using the corresponding source code. Results marked with ◇ are from [41], † represents results are from [17], and other results are taken from [40]

The best results are in bold



(a) ${r_{1h}}^2 - {r_{1t}}^2$

(b) ${r_{1t}}^2 = 1$

(c) ${r_{2h}}^2 \neq {r_{2t}}^2 \neq 1$

(d) ${r_{2t}}^2 \neq 1$

(e) $r_{3h} \circ r_{4h} - r_{3t} \circ r_{4t}$

(f) $r_{3t} \circ r_{4t}$

(g) $r_{5h} \circ r_{6h} - r_{7h}$

(h) $r_{5t} \circ r_{6t} - r_{7t}$

Fig. 3 Histograms of relation embeddings for different relation patterns. r_1 is relation *_similar_to*. r_2 is relation *_part_of*. r_3 is relation *_hyponym*. r_4 is relation *_hypernym*. r_5 is relation */location/location/adjoin_s./location/adjoining_relationship/adjoins*. r_6 is relation */location/statistical_region/gdp_real./measurement_unit/adjusted_money_value/adjustment_currency*. r_7 is relation */location/statistical_region/gdp_nominal./measurement_unit/dated_money_value/currency*

MRR improvement using TCIE weakens, indicating that our model does not need many high dimensions to achieve competitive results.

Table 4 shows the experimental results of link prediction on the FB15k and FB15k-237 datasets. Compared with the results of recent competitive models, TCIE shows clear improvements on FB15k-237 for all evaluation metrics. On FB15k, our model outperforms all baselines in terms of almost all metrics. One exception is that TransH [9] performs better than TCIE in terms of the MR. The relations in FB15k and FB15k-237 are mainly inversion and composition patterns. Thus, TCIE is more suitable for capturing context information in such relationships.

Table 5 displays the experimental results of link prediction on WN18 and YAGO3-10. The major relations in WN18 are inversion and symmetry patterns. TCIE achieves comparable results to those of the state-of-the-art models and scores the highest in terms of Hit@1 and Hit@3. These comparisons demonstrate the strong ability of our model to encode inverse and symmetry relations. YAGO3-10 has 37 kinds of relations, among which the triplets involved in $isAffiliatedTo$ and $playsFor$ account for 35% and 30%, respectively, in the training set. This may be because their (subject, object) pairs basically overlap, resulting in nonoptimal results with our model.

4.4 Model Analysis

4.4.1 Analysis of the Relation Patterns

The propositions in Sect. 3.5 prove that TCIE can model multiple relationships. We investigate the relation embeddings of different patterns (500 dimensions on WN18 and 1500 dimensions on FB15k-237), the histograms are shown in Fig. 3.

Symmetry Pattern. Figure 3a and b show a symmetry relation r_1 *_similar_to* from WN18. We can see that most elements in Fig. 3a are close to or equal to 0, and the absolute values of the r_{1t} elements are very close to 1 in Fig. 3b. The embeddings of r_1 basically satisfy ${r_1}_h^2 = {r_1}_t^2 = 1$

Antisymmetry Pattern. Figure 3c and d show an antisymmetry relation r_2 *_part_of* from WN18. We observe that the elements in Fig. 3c are not concentrated at approximately 0, and the value of r_{2t} elements is not close to 1 in Fig. 3d. These results demonstrate that the embeddings of r_2 satisfy ${r_2}_h^2 \neq {r_2}_t^2 \neq 1$.

Inverse Pattern. Figure 3e and f show the inverse relation r_3 *_hyponym* and r_4 *_hypernym* from WN18. Most elements in Fig. 3e are approximately 0, and the Hadamard products of r_{3t} and r_{4t} are much closer to 1 in Fig. 3f. These results indicate that the embeddings of r_3 and r_4 nearly satisfy ${r_3}_h \circ {r_4}_h = r_{3t} \circ r_{4t} = 1$.

Composition Pattern. Figure 3g and h show the composition relation r_5 */location/location/adjoin_s./location/adjoining_relationship/adjoins*, r_6 */location/statistical _region/gdp_real./measurement_unit/adjusted_money_value/adjustment_currency*, and r_7 */location/statistical_region/gdp_nominal./measurement_unit/dated_money_value/currency* from FB15k-237. Most elements in Fig. 3g and h are close to or equal to 0, which shows that these three relations are close to satisfying ${r_5}_h \circ {r_6}_h = {r_7}_h, r_{5t} \circ r_{6t} = r_{7t}$.

4.4.2 Analysis of the Complex Relations

We study the performance of TCIE in different relation categories. Table 6 summarizes the precise results by relation category on FB15k-237, which shows that our model achieves highly competitive performance on different mapping properties. TCIE is capable of modeling complex mapping properties and performs pretty well on 1-to-N, N-to-1, and N-to-N relations.

4.4.3 Analysis of Embedding Dimension

The embedding dimension will impact the performance of KGE on the knowledge graph. We further conduct experiments to explore the influence of the embedding dimension on TCIE. On the ogbl-wikikg2 dataset, for TCIE in all embedding sizes, λ_1=0.6, λ_2=0.2 and γ=4. On FB15k-237, for TCIE in all embedding sizes, λ_1=0.8, λ_2=0.1 and γ=4. As shown in Fig. 4a, with the increase in dimensions in the ogbl-wikig2 dataset, the MRR increases and then remains unchanged. The same law is found in Fig. 4b on FB15k-237, and when the dimension increases to 2000, the MRR even decreases. Thus, we have learned that the improvements brought by adding dimensions are limited in link prediction. The embedding

Table 6 The detailed link prediction results by relation category on FB15k-237

Relation Category	1-to-1	1-to-N	N-to-1	N-to-N	1-to-1	1-to-N	N-to-1	N-to-N
Tasks	Prediction Head (Hits@10)				Prediction Tail(Hits@10)			
TransE [8]	0.594	0.659	0.162	0.458	0.583	0.138	0.879	0.606
ComplEx [13]	0.521	0.655	0.170	0.454	0.531	0.126	0.862	0.591
RotatE [11]	0.594	0.658	0.167	0.463	0.563	0.131	0.880	0.609
PairRE [16]	**0.604**	0.675	0.212	0.493	**0.594**	0.133	0.880	0.618
TCIE	0.599	**0.677**	**0.216**	**0.496**	0.593	**0.134**	**0.881**	**0.619**
Tasks	Prediction Head (MRR)				Prediction Tail(MRR)			
TransE [8]	0.490	0.450	0.081	0.248	0.481	0.070	0.746	0.364
ComplEx [13]	0.367	0.463	0.091	0.249	0.362	0.064	0.737	0.357
RotatE [11]	0.496	0.463	0.086	0.253	0.482	0.072	0.757	0.369
PairRE [16]	0.496	0.476	0.114	0.277	0.492	0.073	0.762	0.381
TCIE	**0.501**	**0.479**	**0.116**	**0.287**	**0.494**	**0.075**	**0.764**	**0.385**

The results of PairRE are our experimental results by using their source codes, and other results are taken from [42]

The best results are in bold

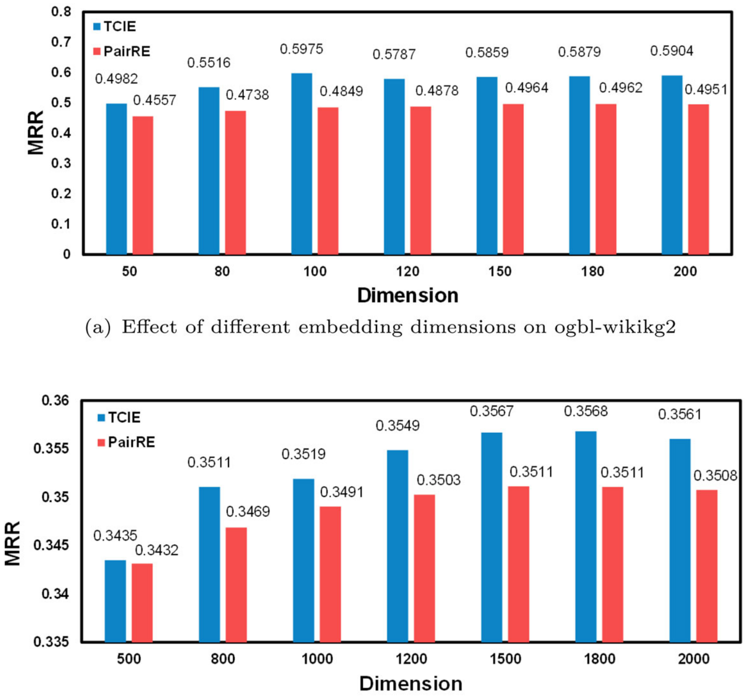
Table 7 Ablation experiment results on FB15K-237 and ogbl-wikikg2

Score Function	Test MRR	
	ogbl-wikikg2	FB15k-237
$score_1$	0.4872	0.3492
$score_2$	0.4385	0.3089
$score_3$	0.2608	0.2866
$score_1 + score_2 + score_3$	0.5975	0.3567

dimension in Fig. 4 is the entity dimension. PairRE's relation dimension is twice that of the entity. In TCIE, the dimensions of entities and relations are the same. Our model needs a lower relation dimension compared with that need for PairRE but achieves better performance.

4.4.4 Analysis of the Score Function

We perform an ablation study on combinations of score functions. We set $score_1 = - \parallel h_r \circ r_h - t_r \circ r_t \parallel_1$, $score_2 = - \parallel h_c - r_h \circ t_c \parallel_1$, $score_3 = - \parallel t_c - r_t \circ h_c \parallel_1$. The $score_1 + score_2 + score_3 = -\lambda_1 \parallel h_r \circ r_h - t_r \circ r_t \parallel_1 - \lambda_2 \parallel h_c - r_h \circ t_c \parallel_1 - \lambda_2 \parallel t_c - r_t \circ h_c \parallel_1$. For ogbl-wikikg2, λ_1=0.6, λ_2=0.2. For FB15k-237, λ_1=0.8, λ_2=0.1. As shown in Table 7, better results are achieved using $score_1$ than using $score_2$ or $score_3$; however, the lowest MRR is achieved using $score_3$. These results indicate that the interactions initiated by the relation are more useful than those of the head and tail; more information may be learned from the interactions initiated by the head than those initiated by the tail. The best result is achieved with our score function ($score_1 + score_2 + score_3$) obtains the best results. This proves that aggregating the interactions among the head, relation, and tail will help capture more triplet semantic information and obtain better knowledge representations.



(a) Effect of different embedding dimensions on ogbl-wikikg2

(b) Effect of different embedding dimensions on FB15k-237

Fig. 4 Results of different embedding dimensions on ogbl-wikikg2 and FB15k-237

Table 8 Parameter statistics of baselines and TCIE, where n_e denotes the number of entities and n_r and denotes the number of relations, d is the embedding dimension

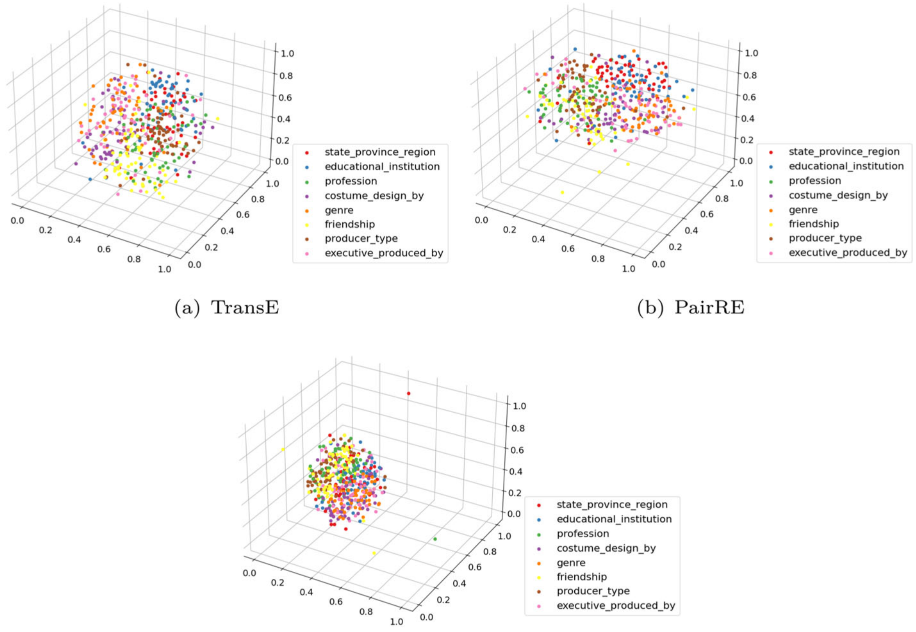
Model	Number of parameters
TrasnE	$(n_e+n_r)\times d$
Complex	$(2n_e+n_r)\times d$
RotatE	$(2n_e+n_r)\times d$
PairRE	$(n_e+2n_r)\times d$
StructurE	$(2n_e+3n_r)\times d$
TCIE	$(n_e+n_r)\times d$

4.4.5 Analysis of the Parameters

The number of parameters of the baseline models is shown in Table 8. TCIE and TransE require the same number of parameters. However, the performance of TCIE is significantly improved. Our model requires less parameters than those required by the recent state-of-the-art model StructurE [17] and performs better on FB15k and FB15k-237.

4.4.6 Analysis of the Entity Embeddings

To explore the distribution of entity embeddings in vector space, we project the learned entity embeddings from TransE [8], PairRE [16], and TCIE into a 3D space by t-SNE [43].



(a) TransE

(b) PairRE

(c) TCIE

Fig. 5 3D scatter plot of entity embedding on FB15k-237. Each color represents a relation and each point represents an entity

As shown in Fig. 5, points with the same colour are related to the same relation. We can observe that points of the same colour tend to be grouped together in TransE, PairRE, and TCIE. However, the points in TransE and PairRE are more dispersed than those in TCIE. In Fig. 5c, all the points are clustered together and very dense. This demonstrates that the distance between entities is closer through the interactions. A closer distance indicates that the context information has been learned and that the entity embeddings have similarities. This will help to strengthen the connections between entities and infer potential relationships in link prediction.

5 Conclusion

The contextual information between the triplet components reflects the correlation between entities that are not directly connected and is essential for inferring potential relations between entities. To explicitly account for this context information in knowledge graph embedding, we propose a new model, TCIE, which designs three interactions for entity and relation representations consisting of two specialized parts. These interactions are used separately for building semantic connections between entities and relations in the positive triplets and capturing contextual information. In addition, we design a unified score function to combine the interactions for semantic connections and contextual information. With these two key factors, TCIE shows a strong capacity for modelling four relation patterns and four mapping properties. Compared with distance-based KGE models, TCIE achieves state-of-the-art results on multiple standard datasets in link prediction.

In future work, we plan to study the following problems. (1) We will integrate the general semantic information in language models into TCIE. (2) We intend to extend the TCIE to KGE downstream tasks, such as graph-to-text generation and complex question answering over knowledge bases.

Acknowledgements This work was supported by the National Key R&D Program of China (grant number 2021YFC3300204).

Author Contributions TW: Conceptualization, Methodology, Software, Data curation, Original draft preparation, Visualization, Investigation, Validation, Formal analysis; BS: Supervision, Resources, Project administration, Funding acquisition, Reviewing and Editing; JZ: Software, Supervision, Formal analysis Reviewing and Editing; YZ: Project administration, Formal analysis. All authors read and approved the final manuscript.

Funding This work was supported by the National Key R&D Program of China (Grant Number 2021YFC3300204).

Code Availability Some or all models, code that support the findings of this study are available from the corresponding author upon reasonable request.

Declarations

Conflict of interest The authors declare that they have no competing interests.

Ethical Approval and Consent to participate Not Applicable.

Consent for publication Not Applicable.

Human and Animal Ethics Not Applicable.

References

1. Miller GA (1995) Wordnet: a lexical database for English. Commun ACM 38(11):39–41
2. Bollacker K, Evans C, Paritosh P et al (2008) Freebase: a collaboratively created graph database for structuring human knowledge. In: Proceedings of the 2008 ACM SIGMOD international conference on Management of data, pp 1247–1250
3. Suchanek FM, Kasneci G, Weikum G (2007) Yago: a core of semantic knowledge. In: Proceedings of the 16th international conference on World Wide Web, pp 697–706
4. Lehmann J, Isele R, Jakob M et al (2015) Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. Semantic web 6(2):167–195
5. Zhang F, Yuan NJ, Lian D et al (2016) Collaborative knowledge base embedding for recommender systems. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, pp 353–362
6. Hao Y, Zhang Y, Liu K et al (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), pp 221–231
7. Yang B, Mitchell T (2019) Leveraging knowledge bases in lstms for improving machine reading. arXiv:1902.09091

8. Bordes A, Usunier N, Garcia-Duran A et al (2013) Translating embeddings for modeling multi-relational data. In: Advances in neural information processing systems, vol 26
9. Wang Z, Zhang J, Feng J et al (2014) Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the twenty-eighth AAAI conference on artificial intelligence, pp 1112–1119
10. Lin Y, Liu Z, Sun M et al (2015) Learning entity and relation embeddings for knowledge graph completion. In: Proceedings of the twenty-ninth AAAI conference on artificial intelligence, pp 2181–2187
11. Sun Z, Deng ZH, Nie JY et al (2019) Rotate: Knowledge graph embedding by relational rotation in complex space. arXiv:1902.10197
12. Yang B, Yih Wt, He X et al (2014) Embedding entities and relations for learning and inference in knowledge bases. arXiv:1412.6575
13. Trouillon T, Welbl J, Riedel S et al (2016) Complex embeddings for simple link prediction. In: International conference on machine learning. PMLR, pp 2071–2080
14. Nickel M, Rosasco L, Poggio T (2015) Holographic embeddings of knowledge graphs. arXiv:1510.04935
15. Dettmers T, Minervini P, Stenetorp P et al (2017) Convolutional 2d knowledge graph embeddings. arXiv:1707.01476
16. Chao L, He J, Wang T et al (2020) Pairre: knowledge graph embeddings via paired relation vectors. arXiv:2011.03798
17. Zhang Q, Wang R, Yang J et al (2022) Structural context-based knowledge graph embedding for link prediction. Neurocomputing 470:109–120
18. Hu W, Fey M, Zitnik M et al (2020) Open graph benchmark: datasets for machine learning on graphs. Adv Neural Inf Process Syst 33:22,118-22,133
19. Toutanova K, 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, pp 57–66
20. Wang Q, Mao Z, Wang B et al (2017) Knowledge graph embedding: a survey of approaches and applications. IEEE Trans Knowl Data Eng 29(12):2724–2743
21. Ji S, Pan S, Cambria E et al (2021) A survey on knowledge graphs: representation, acquisition, and applications. IEEE Trans Neural Netw Learn Syst 33(2):494–514
22. Li W, Peng R, Li Z (2022) Improving knowledge graph completion via increasing embedding interactions. Appl Intell 52:9289–9307
23. Zhang S, Sun Z, Zhang W (2020) Improve the translational distance models for knowledge graph embedding. J Intell Inf Syst 55(3):445–467
24. Ji G, He S, Xu L et al (2015) 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
25. Zhang Z, Cai J, Zhang Y et al (2019) Learning hierarchy-aware knowledge graph embeddings for link prediction. arXiv:1911.09419
26. Nickel M, Tresp V, Kriegel HP (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, pp 809–816
27. Zhang S, Tay Y, Yao L et al (2019) Quaternion knowledge graph embeddings. In: Advances in neural information processing systems, vol 32
28. Yu L, Luo Z, Liu H et al (2022) Triplere: knowledge graph embeddings via tripled relation vectors. arXiv:2209.08271
29. Balažević I, Allen C, Hospedales TM (2019) Hypernetwork knowledge graph embeddings. In: International conference on artificial neural networks. Springer, pp 553–565
30. Socher R, Chen D, Manning CD et al (2013) Reasoning with neural tensor networks for knowledge base completion. In: Advances in neural information processing systems, vol 26
31. Shang C, Tang Y, Huang J et al (2019) End-to-end structure-aware convolutional networks for knowledge base completion. In: Proceedings of the thirty-third AAAI conference on artificial intelligence and thirty-first innovative applications of artificial intelligence conference and ninth AAAI symposium on educational advances in artificial intelligence, pp 3060–3067
32. Schlichtkrull M, Kipf TN, Bloem P et al (2018) Modeling relational data with graph convolutional networks. In: European semantic web conference. Springer, pp 593–607
33. Guo L, Sun Z, Hu W (2019) Learning to exploit long-term relational dependencies in knowledge graphs. In: International conference on machine learning. PMLR, pp 2505–2514
34. Wang Q, Huang P, Wang H et al (2019) Coke: contextualized knowledge graph embedding. arXiv:1911.02168
35. Yao L, Mao C, Luo Y (2019) Kg-bert: bert for knowledge graph completion. arXiv:1909.03193
36. Cai L, Wang WY (2017) Kbgan: adversarial learning for knowledge graph embeddings. arXiv:1711.04071

37. Zhang Y, Yao Q, Shao Y et al (2019) Nscaching: simple and efficient negative sampling for knowledge graph embedding. In: 2019 IEEE 35th international conference on data engineering (ICDE). IEEE, pp 614–625
38. Wang P, Li S, Pan R (2018) Incorporating GAN for negative sampling in knowledge representation learning. In: Proceedings of the thirty-second AAAI conference on artificial intelligence and thirtieth innovative applications of artificial intelligence conference and eighth AAAI symposium on educational advances in artificial intelligence, pp 2005–2012
39. Vrandečić D, Krötzsch M (2014) Wikidata: a free collaborative knowledgebase. Commun ACM 57(10):78–85
40. Peng Y, Zhang J (2020) Lineare: Simple but powerful knowledge graph embedding for link prediction. In: 2020 IEEE international conference on data mining (ICDM). IEEE, pp 422–431
41. Song T, Luo J, Huang L (2021) Rot-pro: Modeling transitivity by projection in knowledge graph embedding. Adv Neural Inf Process Syst 34:24695–24706
42. Zhang Q, Wang R, Yang J et al (2022) Knowledge graph embedding by reflection transformation. Knowl Based Syst 238(107):861
43. Van der Maaten L, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(11):2579–2605