Information Sciences 648 (2023) 119548

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Knowledge graph completion method based on quantum embedding and quaternion interaction enhancement

LinYu Li [a], Xuan Zhang [a,b,c,*], Zhi Jin [d,e], Chen Gao [f], Rui Zhu [a,b,c], YuQin Liang [a], YuBing Ma [a]

[a] School of Software, Yunnan University, Yunnan 650091 China
[b] Key Laboratory of Software Engineering of Yunnan Province, Yunnan 650091 China
[c] Engineering Research Center of Cyberspace, Yunnan 650091 China
[d] School of Computer Science, Peking University, Beijing 100871 China
[e] Key Lab. of High-Confidence Software Technologies (MoU), Peking University, Beijing 100871 China
[f] School of Information Science and Engineering, Yunnan University, Yunnan 650091 China

ARTICLE INFO

Keywords:
Knowledge graph completion
Link prediction
Quantum embedding
Quaternion
Knowledge graph

ABSTRACT

Knowledge graphs (KG) are used for many downstream tasks in artificial intelligence (AI). However, owing to accuracy issues associated with information extraction, KGs are often incomplete. This has led to the emergence of knowledge graph completion (KGC) tasks. Their purpose is to learn known facts to infer the missing entities in triples. Traditional embedding-based methods usually only focus on the information of individual triples and do not use the deep logical relationships of the KG. In this study, we propose a new KGC method referred to as QIQE-KGC. It uses quantum embedding and quaternion space interaction to capture the external logical relationship between triples in a KG and enhance the connection between entities and relations within a single triple to model and represent the KG. The proposed QIQE-KGC model can capture richer logical information and has more powerful and complex relationship modeling capabilities. Extensive experimental results using QIQE-KGC on 11 datasets demonstrate that the model achieves outstanding performance. Compared to the baseline models, QIQE-KGC produced the best results on most datasets.

1. Introduction

In 2012, Google proposed the concept of a knowledge graph (KG) [1]. Since then, the KG has played an important role in AI services such as question answering [2], semantic search, and recommendation systems [3][4]. However, when building a KG, the entities and relationships need to be extracted from a large text corpus. The limitations of natural language processing technology and the noise of text data usually affect the accuracy of information extraction to a certain extent. Consequently, real-world knowledge graphs are often incomplete. Owing to the incompleteness of knowledge graphs, many downstream AI applications are likely to be severely affected. To solve this problem, Knowledge Graph Completion (KGC) [5] has been proposed. Formally speaking, it is also known as a link prediction task in a KG environment. The key solution is to predict the correct tail entity t when given an incomplete triple $(h, r, ?)$, or predict the correct head entity h when given an incomplete triple $(?, r, t)$.

* Corresponding author at: School of Software, Yunnan University, Yunnan 650091 China.
E-mail address: zhxuan@ynu.edu.cn (X. Zhang).

https://doi.org/10.1016/j.ins.2023.119548
Received 12 April 2023; Received in revised form 10 August 2023; Accepted 13 August 2023

Available online 19 August 2023

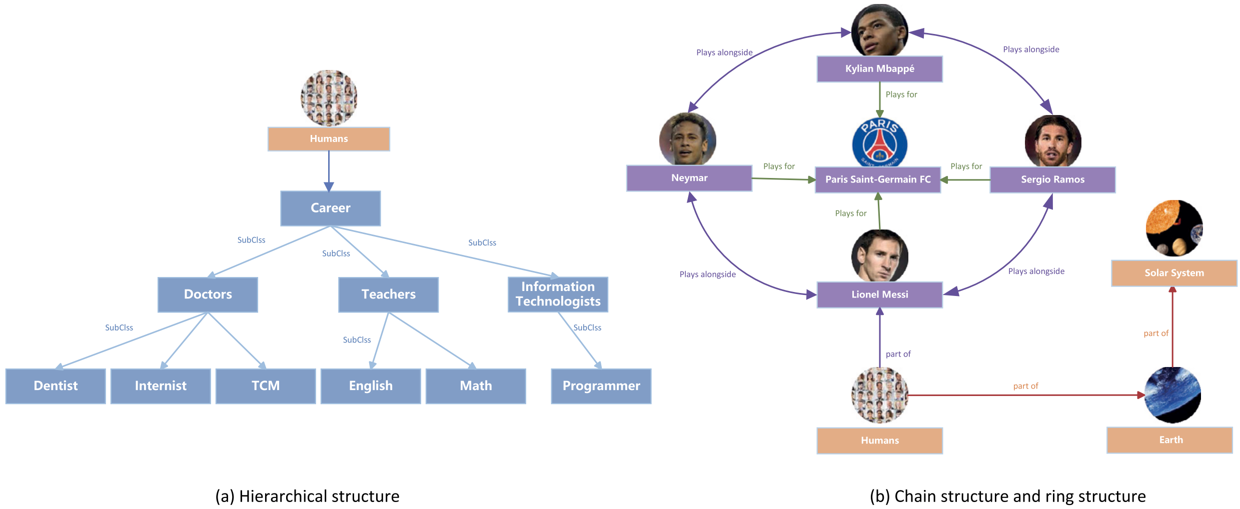

Fig. 1. Example of knowledge graph architecture.

Traditional KGC models are mainly divided into translation models, such as TransE [6], TransR [7], TransD [8] based on knowledge graph embedding (KGE), network models such as ConvE [9], and KGC models based on natural language processing such as KGBert [10], StAR [11], and SimKGC [12]. KGE models assign high scores to valid triples and low scores to invalid triplets in each iterative training round. However, they often focus only on independent information inside each triple; that is, any given KG embeds the entity and relation into the vector space by statistically calculating the similarity between the entity and the relation. These methods ignore the hidden logical dependencies between triples in the KG complex structures. Examples of hidden logical relationships are shown in Fig. 1. Triplet A in (b): Plays alongside (Lionel Messi, Neymar) and Triplet B: Plays alongside (Lionel Messi, Sergio Ramos), Triplet C should be reasoned as Plays alongside (Neymar, Sergio Ramos), i.e. $A \bigcap B \Rightarrow C$; similar logical dependencies are also observed in (a): $Dentist \Rightarrow Doctor \Rightarrow Career$. A recent work E2R [13] embeds the symbolic knowledge base into the vector space by maintaining the logical structures. This method is inspired by quantum logic theory [14][15] and satisfies the axioms of quantum logic. A single fact (i.e., a triple) is treated as an atom, and the connection between different facts is modeled as a complex formula. E2R also allows logical operations (e.g., negation, conjunction, disjunction, and implication) to be performed directly on vectors in a manner similar to Boolean logic, a process known as quantum embedding.

The structure of a knowledge graph is highly complex and typically encompasses circular, hierarchical, and chained structures, as illustrated in Figs. 1(a) and (b). It is difficult to effectively model the KG using single modeling methods. The complex number space is used in quantum logic; thus, we considered using the hypercomplex number space to jointly model quantum embeddings. Quaternion space was used in QuatE [16] and BiQUE [17], which were combined with interactive quantum embedding for KGC. These existing quaternion methods, QuatE, and other models also use the head entity, relation, and tail entity to measure the score of the triples in isolation. They were not entirely efficient in capturing the association between different types of head and tail entities. For example, in the triplet $(LionelMessi, played for, ParisSaint-Germain)$ in Fig. 1 (b), the entity type represented by the head entity Lionel-Messi is a person and the entity type represented by the tail entity Paris Saint-Germain is a club. These two entities differ greatly in terms of entity attributes, Lionel Messi's attributes include height and weight, etc. Paris Saint-Germain's attributes include the number of players, staff, etc. If the head entity, relation, and tail entity are used to measure the score of triples in isolation, it will increase the noise of the model to a certain extent. Inspired by the two-dimensional planar embedding model [7][8], we enhance the representation learning between different types of head and tail entities by adding the Hadamard product rotation operation of the head and tail entities under the same relation. The purpose is to pull the head-tail entity representations of triples with different attribute types closer but with actual factual associations, to obtain more powerful representation capabilities.

Because KGs are usually composed of hybrid structures such as (a) and (b) in Fig. 1, we propose a joint model QIQE-KGC based on the interaction of quantum embedding and quaternion space to solve the above problems. This new KGC method combines the excellent modeling ability of entities and relations within triples in the quaternion space and the logical capture ability of quantum embedding for deep dependencies between different triples in KGs. Thus, QIQE-KGC interactively performs multi-task learning to obtain a more powerful model effect.

The contributions of model QIQE-KGC are summarized as follows:

- To the best of our knowledge, this is the first study that uses quantum embedding of the joint quaternion space for the KGC task. It performs multi-task learning through the advantages of interactive quantum embedding and quaternion space; thus, the model has a powerful reasoning ability in complex KGs.
- We propose an enhanced method for learning the representations of different types of head and tail entities under the same relation in the process of quaternion modeling in KGC.

- Through many experiments on 11 different types of datasets, QIQE-KGC demonstrated good KGC capabilities in most of the datasets, with many improvements compared with the original baseline models.

The remainder of this paper is organized as follows. Related studies are introduced in Section 2. In Section 3, we introduce the details of the QIQE-KGC model based on the interaction between the quantum embedding and the quaternion. We perform experiments and analyze the results in Section 4. A summary of the study and an outlook for the future are presented in Section 5.

2. Related work

For better understanding, we first introduce some common KGC models in Sections 2.1, 2.2, and 2.3, including translation-based KGC models, natural language processing (NLP)-based KGC models, and graph neural network (GNN)-based KGC models. Related concepts of quantum embeddings and quaternions are introduced in Sections 2.4 and 2.5.

2.1. KGC model based on translation models

As the most classic and widely used series of KGC representation learning methods, translation models perform extremely well on KGC tasks. Their core idea is to encode the relation between entities into translation by representing entities as low-dimensional embedding vectors. These models define a scoring function for the translation between relations and entities and measure the probability of each triplet using a distance metric. Generally, the distance score reflects the accuracy of a triplet.

TransE, as the earliest and most classic translation KGC model, projects entities and relations into a continuous low-dimensional vector space, where the tail entity t in the triple (h, r, t) can be seen as the result of the translation operator between the head entity h and the relation r, i.e., $v_h + v_r \approx v_t$, and defines the relationship with the entity: $s(h, r, t) = \|v_h + v_r - v_t\|_{l_{1/2}}$. However, TransE's assumptions about the translation process are too simple, which will severely limit the ability to model complex relations, causing TransE to only model pure one-to-one simple relations in units of KGs. Affected by this, some series of derivative models were proposed, such as TransR [7] that projects entities onto a relation-specific hyperplane and performs translation operations on that hyperplane to model more complex relations. RotatE [18] uniformly models and infers three relational schemas by introducing complex number spaces: symmetric/antisymmetric, inversion, and composition. As most translation-based embedding models for KGC do not take into account neighborhood and textual information in KGs, other types of KGC models have been developed to compensate for the inadequacy of translation-based models.

2.2. KGC model based on NLP

The NLP-based KGC model, KG-BERT [10], is a new KGC model that was proposed in recent years. It suggests using a pre-trained language model for the KGC, where the triples are treated as a sequence of text, taking the entity and relation descriptions of the triples as input, and modeling these triples using a language model to compute a scoring function for the triples. NLP-based KGC models have been developed, including StAR [11] which combines NLP and translation models and embeds semantic and structural information of KGs into the natural language descriptions of triples. Recent research on SIM-KGC [12], introduced contrastive learning into the NLP-based KGC model and achieved excellent results on some datasets. However, through careful observation, we found that NLP-based KGC models usually perform poorly in datasets with a large number of relationships, such as FB15K-237.

2.3. KGC model based on GNN

In contrast to the translation embedding model and NLP-based KGC model, the GNN-based KGC model uses a GNN as an encoder and then uses the traditional neural network-based KGC model ConvE [9]to act as a decoder. GNN-based models typically use a encoder-decoder framework to learn the connectivity of the entire KG to better capture its neighborhood information structure. SACN [19] introduced GCNs with weights and defined the connection strengths between neighboring nodes with the same relation type to capture structured and useful information in KGs using the structure of nodes, attributes of nodes, and different relation types. In our previous study, we proposed CLGAT-KGC [20], which introduced a graph attention mechanism and defined an enhanced representation of relations in a multi-relational setting and two contrastive learning methods for enhanced representation learning of KGs. In addition, some of the other latest work on KG representation learning for recommendation systems [3][4] also uses contrastive learning to better learn the representation of entities and relationships in KGs by contrasting positive triples against negative triples. In general, the GNN-based KGC model has exhibited performance improvements. However, it has performed consistently at an ordinary level in experiments, indicating that there is still significant room for improvement.

2.4. Quantum embedding

Garg et al. proposed a new approach to quantum embedding known as Embed2Reason (E2R) [13], which embeds symbolic KGs into vector spaces in a manner that preserves the logical structure and represents each triple as a fact. It was inspired by the quantum logic used to explain the mechanisms of quantum mechanics [14], and is essentially an embedding model based on algebraic logic. It projects all elements into an identical vector space that contains some constraints from syntax and semantics and is multi-valued instead of binary as in Boolean logic, precisely so that quantum embeddings will also be more suitable to operate in continuous vector spaces. More formally, quantum embeddings are theoretically compatible with quaternion space embeddings. Nevertheless, E2R possesses inherent restrictions, given its sole consideration of the logical information within KGs. Its performance decreases when addressing KGs with complex structures, an aspect that we will detail in Section 4.2.3. Moreover, E2R integrates a multitude of loss functions, many of which are inapplicable to KGC tasks. Thus, in QIQE-KGC, we propose quantum embedding as a task-specific variant, that focuses on binary relations. We used negative sampling methodologies to augment the compatibility of quantum embeddings with KGC tasks.

2.5. Quaternion

The quaternion is a system of supercomplex numbers first proposed by Hamilton [21]. Many theories exist regarding the birth of quaternions. A familiar explanation is that we know that multiplication between vectors has inner and outer products but both operations are imperfect, i.e., the condition of the group is not satisfied. So the problems that cannot be solved in three dimensions are mapped to four dimensions, which is the cause and effect of the birth of quaternions. The advantages of processing 3D rotations with quaternions over matrices are undeniable. Its applications are in aerospace, game design, animation effects, machine learning, and many other fields. Quaternion representation is helpful for augmenting convolutional neural networks for a variety of tasks such as image classification and automatic speech recognition [22][23] and KGC [16][17]. Formally, the core motivation behind these models is to use quaternions to allow neural networks to encode potential inter and intra-dependencies between multidimensional input features, leading to more compact interactions and better representational capabilities.

3. QIQE-KGC model

This study does not consider the deeper quantum logic behind the quaternion and with quantum computation, but is concerned with how it powerfully expresses logical relationships algebraically. In this section, we present the proposed QIQE-KGC model in detail. The task-oriented quantum embedding module is introduced in Section 3.1, the quaternion space-based module in Section 3.2, and the entire model architecture in Section 3.3.

3.1. Task-oriented quantum embedding module

Unlike quantum embedding in E2R, the datasets in the KGC task that we are studying exist in the form of triplets; thus, there is no need to consider loss functions for unary relationships. In addition, we used negative sampling to generate negative samples, which also addressed the issue of subspace collapse in the E2R. This allowed us to reduce the computational complexity of the model without affecting its performance. We refer to this process as task-oriented quantum embedding.

Representation learning In the model, both entities and relations are represented as a d-dimensional complex vector. e.g., i is assumed to be an imaginary unit, and $\mathbb{C}^d$ and $\mathbb{R}^d$ denote the d-dimensional complex and real spaces, respectively. Taking the triplet (covid-19, subClass, RNA-Virus) as an example, for the entities covid-19 and RNA-Virus, they can be represented by the embedding vector as $\mathbf{e}_{covid-19} = \mathbf{v}_{covid-19} + i\mathbf{v}_0$ and $\mathbf{e}_{RNA-Virus} = \mathbf{v}_{RNA-Virus} + i\mathbf{v}_0$ where $\mathbf{v}_0$ denotes a zero vector of dimension d. In simple terms, the imaginary part of a single entity is zero, but it is non-zero for an entity pair, and these relations will map to different linear subspaces. A triple $(covid-19, subClass, RNA-Virus)$ can be represented as a fact that covid-19 is a subclass of RNA-Virus. This binary relationship can be defined as an entity pair: subClass$(covid-19, RNA-Virus)$. In terms of representation, this can be expressed by projecting it into vector space as $\mathbf{R}_{subClass} = \mathbf{v}_{\text{Covid}-19} + i\mathbf{v}_{\text{RNN}-\text{Virus}}$, where $\mathbf{v}_{\text{Covid}-19}, \mathbf{v}_{\text{RNN}-\text{Virus}} \in \mathbb{R}^d$ are entity representations of the entities Covid-19 and RNA-Virus, respectively, relation $\mathbf{R}_{\text{subClass}} \in \mathbb{C}^d$. Consistent with the previous work [13][24], the complex vector space of this relation is assumed to be isomorphic to the real space of entities $\mathbb{R}^{2d}$, which leads to a dual mapping relation representation: $T : \mathbb{C}^d \mapsto \mathbb{R}^{2d}$. To better understand the representation learning of the model, the KG is defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ represents the set of entities, $\mathcal{R}$ represents the set of relations, and $\mathcal{T}$ represents the set of edges can also be understood as an entity pair. The per-entity pair $\mathbf{R}_r(\mathbf{e}_h, \mathbf{e}_t)$ denotes that the head entity h to the tail entity t is real in the case of the relation $r \in \mathcal{R}$, then the entity pair can be expressed as $\mathbf{R}_r = \begin{pmatrix} \mathbf{v}_h \\ \mathbf{v}_t \end{pmatrix}, \mathbf{R}_r \in \mathcal{T}$. A single entity $\mathbf{e}_h, \mathbf{e}_t \in \mathcal{E}$ can then be represented as $e_h = \begin{pmatrix} \boldsymbol{v}_h \\ \mathbf{0}_{2d} \end{pmatrix}, \quad e_t = \begin{pmatrix} \boldsymbol{v}_t \\ \mathbf{0}_{2d} \end{pmatrix}$. Taking subClass$(Covid-19, RNA-Virus)$ as an example, it can be expressed as: $\mathbf{R}_{sub} = \begin{pmatrix} \mathbf{v}_{Covid-19} \\ \mathbf{v}_{RNN-Virus} \end{pmatrix}$, where the above variables $\mathbf{R}_{sub}$, $\mathbf{e}_{covid-19}$, $\mathbf{e}_{RNA-Virus} \in \mathbb{R}^{2d}$; $\mathbf{v}_{covid-19}$, $\mathbf{v}_{RNA-Virus} \in \mathbb{R}^d$. $\mathbf{0}_{2d}$ and $\mathbf{1}_{2d}$ represent the two-dimensional vectors containing only 0 and only 1.

Score and loss functions The score function is a quantitative calculation used by the KGC model to predict whether an unknown triplet (a fact) is correct or not. Considering that a single fact is considered as an atom and the connection between different facts is modeled as a complex logical formula, it is necessary to model this type, according to the orthocomplemented lattice syntax of quantum logic [15], where every proposition $\mathbf{p}$ and its own inverse proposition $\neg\mathbf{p}$ in quantum logic must be mutually orthogonal. Thus, in this model, the two complex vectors $\mathbf{p}$ and $\neg\mathbf{p}$ must satisfy the orthogonal formula: $\mathbf{p} \cdot \neg\mathbf{p} = 0$ and $\neg\boldsymbol{P} = \mathbf{1}_{2d} - \boldsymbol{P}$. In more formal terms, the more a triplet (a fact) in KG satisfies this orthogonal formula, the higher the score of this triplet should be. It is defined as:

$$f_{\text{Quantum}}\left(\mathbf{E}_r\right) = \left\|\left(\mathbf{1}_{2d} - \mathbf{E}_r\right) \cdot \mathbf{E}_r\right\|^2 \tag{1}$$

where $\mathbf{E}_r$ represents the entity pair representation of triples, and $\|\|$ represents the two-norm. Unlike the models [6][16] that construct loss functions based on a distance score function, E2R constructs constraints based on the syntax and semantics of quantum logic as a loss function. For the model to predict normal operation, it is necessary to formulate multiple constraints to act as loss functions, so there will be several term loss functions, such as logical loss, entity loss, membership loss, and so on. Compared with the E2R model, we do not consider the NLP reasoning task but only consider the link prediction task in KGs, and the datasets are all KG data in the form of triplets. Thus, a large number of loss functions are unnecessary, so the loss function can be defined for:

$$\mathcal{L}_{\text{Quantum}} = \sum_{(h,r,t)\in\{\mathcal{G}\cup\mathcal{G}'\}} \left(\mathcal{L}_{\text{ELoss}} + \mathcal{L}_{\text{LLoss}} + \mathcal{L}_{\text{MLoss}}\right) \tag{2}$$

where $\mathcal{G}\cup\mathcal{G}'$ represents the set of positive samples and negative samples of triplets that have been randomly replaced in [6]; $\mathcal{L}_{\text{ELoss}}$ represents entity loss, $\mathcal{L}_{\text{LLoss}}$ represents logical relationship loss, and $\mathcal{L}_{\text{MLoss}}$ represents membership loss, all of which are in the appendix Eqs. (26)(31)(32).

In addition to the loss function described above, since the datasets are all triples (pairs of entities) and are considered complex logical formulas, there is no need to consider the unary universal loss in [13]. Owing to the use of random substitution in the negative sampling method, there is no need to consider the subspace collapse problem in previous studies, reducing the overhead of the model.

3.2. Quaternion module

The quaternion Q is an upgraded version of the traditional complex number and is a super complex number. Unlike a complex number consisting of a real part and an imaginary part, a quaternion q is composed of a real part and three imaginary parts. Thus, a quaternion is usually defined as $q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} \in \mathbb{H}^n$, where $a, b, c, d \in \mathbb{R}^n$, $\mathbf{i}, \mathbf{j}, \mathbf{k}$ are imaginary units and satisfy $\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$. We begin with some basic symbolic operations on quaternions.

Quaternion parametrization:

$$|q| = \sqrt{a^2 + b^2 + c^2 + d^2} \tag{3}$$

Conjugate operation:

$$\bar{q} = a - b\mathbf{i} - c\mathbf{j} - d\mathbf{k} \tag{4}$$

Quaternion addition:

$$q_1 \pm q_2 = \left[a_1 \pm a_2, \mathbf{v_1} \pm \mathbf{v}_2\right] \tag{5}$$

Quaternion inner product:

$$q_1 \cdot q_2 = a_1 \cdot a_2 + b_1 \cdot b_2 + c_1 \cdot c_2 + d_1 \cdot d_2 \tag{6}$$

Quaternion normalization:

$$q^{\triangleleft}(a, b, c, d) = \frac{q}{|q|} = \frac{a_r + b_r\mathbf{i} + c_r\mathbf{j} + d_r\mathbf{k}}{\sqrt{a_r^2 + b_r^2 + c_r^2 + d_r^2}} \tag{7}$$

Hamilton product:

$$\begin{aligned} q_1 \otimes q_2 = &\left(a_1 a_2 - b_1 b_2 - c_1 c_2 - d_1 d_2\right) \\ &+ \left(a_1 b_2 + b_1 a_2 + c_1 d_2 - d_1 c_2\right)\mathbf{i} \\ &+ \left(a_1 c_2 - b_1 d_2 + c_1 a_2 + d_1 b_2\right)\mathbf{j} \\ &+ \left(a_1 d_2 + b_1 c_2 - c_1 b_2 + d_1 a_2\right)\mathbf{k} \end{aligned} \tag{8}$$

The Hamiltonian product of quaternions is not exchangeable and does not satisfy the traditional exchange law of multiplication, i.e., $q_1 \otimes q_2$ is not equal to $q_2 \otimes q_1$.

Consistent with the quantum embedding module, the KG is defined as $\mathcal{G} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$, where $\mathcal{E}$ represents a collection of entities, and $\mathcal{R}$ represents a set of relations, and $\mathcal{T}$ represents a set of edges that can also be understood as a triplet set. A triple is expressed as (h, r, t), where $h, t \in \mathcal{E}$ represent the head entity and tail entity respectively, and $r \in \mathcal{R}$ represents the relation. We need to represent entities and relations as embeddings and compute their scoring function and use the scoring function to measure the prediction score of triples.

Existing embedding models obtain triple scoring using a single head entity, tail entity, and relation. There are obvious shortcomings to this method. The ability of the model to capture the representation and feature interaction between triplet entities and relations is relatively weak. Because it relies only on the strict calculation of the three embedding vectors, it cannot capture different entities under the same relation types of potential dependencies. Inspired by TransD [8] and TransR [7], we propose a method to overcome this shortcoming by enhancing the interactive ability of head and tail entities through relation rotation. We use the representation method in [16] to represent a triplet (h, r, t) embedding in the quaternion space; h, r, t are represented as:

Algorithm 1 The Training Process for link prediction in QIQE-KGC.

```
Input: Knowledge graph 𝒢 = (ℰ, ℛ, 𝒯), training set 𝒮 = (h, r, t) | h, t ∈ ℰ, r ∈ ℛ; batch size b; learning rate a; hyperparameter α and β; Epoch number N.
Output: Triplet prediction probability score ŷ(·)
Initialize Quantum logic embedded module: R_ht, e_h, e_t ∈ ℝ^{2d}.
Initialize Quaternion embedded module: Q_h, W_r, Q_t ∈ ℍ^n.
for n = 1 → N do
    quantum logic embedded model:
    Obtain Entity Element Loss ℒ_ELoss with Eq. (23)-Eq. (25)
    Obtain Logical Relationship Loss ℒ_LLoss with Eq. (27)-Eq. (31)
    Obtain Membership Loss ℒ_MLoss with Eq. (32)
    Obtain Quantum Embedding Score f_Quantum with Eq. (2)
    Quaternion model:
    Obtain the rotated entity representation Q_h → Q_r,h, Q_t → Q_r,t with Eq. (14) and Eq. (15)
    Obtain Quaternion Loss ℒ_Quaternion with Eq. (17)
    Obtain Quaternion Score f(h, r, t)_Quaternion with Eq. (16)
    QIQE-KGC:
    Get the total loss of the model by combining the losses of the two parts with Eq. (18)
    Get the total score of the model by combining the scores of the two parts with Eq. (19)
end for
Repeat:Training is repeated to get best-predicted value
Until:Converges
Return:Triplet prediction probability score ŷ·
```

$$\mathbf{Q_h} = a_h + b_h\mathbf{i} + c_h\mathbf{j} + d_h\mathbf{k} \tag{9}$$

$$\mathbf{W_r} = a_r + b_r\mathbf{i} + c_r\mathbf{j} + d_r\mathbf{k} \tag{10}$$

$$\mathbf{Q_t} = a_t + b_t\mathbf{i} + c_t\mathbf{j} + d_t\mathbf{k} \tag{11}$$

where $\mathbf{Q_h}, \mathbf{W_r}, \mathbf{Q_t} \in \mathbb{H}^n$, and $a_h, b_h, c_h, d_h \in \mathbb{R}^d$; a_r, b_r, c_r, $d_r \in \mathbb{R}^d$, a_t, b_t, c_t, $d_t \in \mathbb{R}^d$.

To solve these problems, we use two quaternion vectors $\mathbf{W}_{r,h}$, $\mathbf{W}_{r,t} \in \mathbb{H}^n$ for each relation to performing the Hadamard product rotation operation. It is used to enhance the representation of learning between different types of head-tail entities with the same relation. In this way, the representation distance of head and tail entities with different attribute types but actual fact associations can be closer:

$$\mathbf{W}_{r,h} = a_{r,h} + b_{r,h}\mathbf{i} + c_{r,h}\mathbf{j} + d_{r,h}\mathbf{k} \tag{12}$$

$$\mathbf{W}_{r,t} = a_{r,t} + b_{r,t}\mathbf{i} + c_{r,t}\mathbf{j} + d_{r,t}\mathbf{k} \tag{13}$$

where $a_{r,h}, b_{r,h}, c_{r,h}, d_{r,h}, a_{r,t}, b_{r,t}, c_{r,t}, d_{r,t} \in \mathbb{R}^d$.

More formally speaking, the Hamilton product is used to rotate the representations of the head entity $\mathbf{Q_h}$ and the tail entity $\mathbf{Q_t}$ through the normalized vectors $\mathbf{W}_{r,\mathbf{h}}^{\triangleleft}$ and $\mathbf{W}_{r,\mathbf{t}}^{\triangleleft}$ respectively:

$$\mathbf{Q_{r,h}} = \mathbf{Q_h} \otimes \mathbf{W}_{r,\,\mathrm{h}}^{\triangleleft} \tag{14}$$

$$\mathbf{Q_{r,t}} = \mathbf{Q_t} \otimes \mathbf{W}_{r,t}^{\triangleleft} \tag{15}$$

Our score and loss functions were similar in form to those of QuatE [16]. The core is to use the head entity and relation to execute the rotation operation, and then perform the inner product operation with the tail entity to calculate the score function. The specific operation is shown in part of Algorithm 1. The correlation of the head and tail entities is enhanced by the formula (14) (15) to improve the representation ability of the model. So the scoring of the quaternion module is defined as:

$$f(h,r,t)_{\text{Quaternion}} = \left(\mathbf{Q_{r,h}} \otimes \mathbf{W_r}^{\triangleleft}\right) \cdot \mathbf{Q_{r,t}} \tag{16}$$

where $\cdot$ represents the quaternion inner product. The loss function is defined as:

$$\begin{aligned} &\mathcal{L}_{\text{Quaternion}} = \textstyle\sum_{(h,r,t)\in\{\mathcal{G}\cup\mathcal{G}'\}} \log\left(1 + \exp\left(-t_{(h,r,t)} \cdot f(h,r,t)_{\text{Quaternion}}\right)\right) \\ &\text{in which, } t_{(h,r,t)} = \begin{cases} 1 \text{ for } (h,r,t) \in \mathcal{G} \\ -1 \text{ for } (h,r,t) \in \mathcal{G}' \end{cases} \end{aligned} \tag{17}$$

Consistent with the quantum embedding module, $\mathcal{G} \cup \mathcal{G}'$ represents the set of positive samples and negative samples of triplets after a random replacement operation similar to [6].

3.3. QIQE-KGC

QIQE-KGC is based on these two basic modules, and its core is to interactively perform multitask learning. It uses KG to represent entities and relations through embedding methods of different theories. The model work and training process are summarized in the following steps as in Algorithm 1. We first generate spatial embeddings for entity set E and relation set R in the KG, including embeddings in quantum space and embeddings in quaternion space. Then QIQE-KGC uses the method of enhancing the correlation

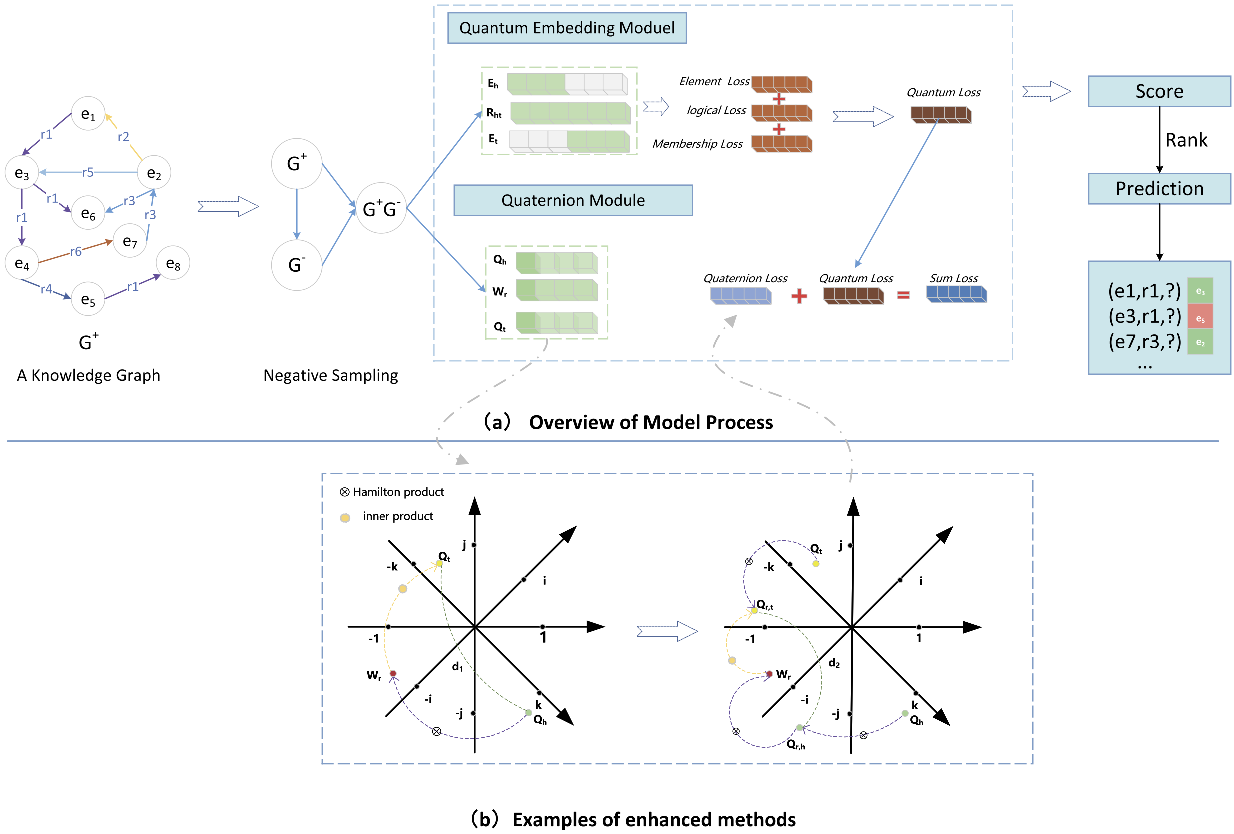

Fig. 2. Simple example of QIQE-KGC.

between head and tail entity types to enhance the expression of the head, tail entity, and relation in quaternion space. Finally, in order to capture the characteristics of different spaces, the model minimizes the $\mathcal{L}_{\text{QIQuate-kgc}}$ loss function, i.e., minimizing $\mathcal{L}_{\text{Quantum}}$ and $\mathcal{L}_{\text{Quaternion}}$ loss. Therefore, the model is trained according to the interactive loss function and score function so that it has the inference ability to capture the characteristics of the two models at the same time.

The model diagram is shown in Fig. 2. Model interactive multi-task learning plays an important role in capturing both internal and external features of elements. In a more formal language, it captures two parts of feature information at the same time. The first part is the information on the triplet (a fact) of entities and relations within a single triplet, and the second part is the information on the external logical relationship between different triplets. In Fig. 2(b), the representation learning enhancement method of different types of head and tail entities under the same relation is abstracted, the purpose is to shorten the distance between $\mathbf{Q_h}$ and $\mathbf{Q_t}$ distance d_1 to $\mathbf{Q_{r,h}}$ and $\mathbf{Q_{r,t}}$ distance d_2 (green dotted line). To ensure that the training process can achieve convergence and reasonableness, and to realize the multi-task interactive learning of the model, we define the loss function of QIQE-KGC in formula (18) and generate negative samples by randomly replacing the head and tail entities [6].

$$\mathcal{L}_{\text{QIQE-KGC}}(h,r,t) = \alpha\mathcal{L}_{\text{Quantum}}(h,r,t) + \beta\mathcal{L}_{\text{Quaternion}}(h,r,t) \tag{18}$$

where $\mathcal{L}_{\text{Quantum}}$ and $\mathcal{L}_{\text{Quaternion}}$ correspond to the losses of the task-oriented quantum embedding module and quaternion modules, respectively. α and β are hyperparameters used to adjust the loss balance. These hyperparameters are used to find an appropriate loss function scale. The final scoring function of the model is the weighted sum of the quantized scores of the two parts:

$$f_{\text{QIQE-KGC}}(h,r,t) = \alpha f_{\text{Quantum}}(h,r,t) + \beta f_{\text{Quaternion}}(h,r,t) \tag{19}$$

where $f_{\text{Quaternion}}$ and f_{Quantum} represent the scores of the quaternion module and the task-oriented quantum embedding module.

4. Experiments

To evaluate the effectiveness of QIQE-KGC, we conducted extensive large-scale experiments. The detailed experimental settings are provided in Section 4.1, along with an analysis of the results of each dataset experiment for the link prediction task and comparisons with existing baseline models. In addition, we conducted experimental analyses on the model, including experimental analysis in environments of transductive and inductive learning, fine-grained relation analysis, model parameter sensitivity analysis, ablation

Table 1
Dataset statistics.

Dataset	#entity	#relation	#train	#valid	#test
Kinships	104	26	8544	1068	1074
UML-S	135	46	5,216	652	661
CN-100K	78,334	34	100,000	1,200	1,200
Atomic	304,388	9	610,536	87,700	87,701
FB15K	14,951	1,345	483,142	50,000	59,071
FB15K-237	14,541	237	272,115	17,535	20,466
WN18	40,943	18	141,442	5,000	5,000
WN18RR	40,943	11	86,835	3,034	3,134
YAGO3-10	123,182	37	1,079,040	5,000	5,000
Wikidata5M-Trans	4,594,485	822	20,614,279	5,163	5,163
Wikidata5M-Ind	4,579,609	822	20,496,514	6,699	6,894

experiments, and computational cost analysis, to show the effect of each module improvement. We present a case study that provides a detailed analysis of the proposed model.

4.1. Experiment settings

4.1.1. Datasets

To analyze the effects of the model fully, we used many datasets to conduct link prediction experiments. These include two small datasets (Kinships and UML-S), two common-sense KG datasets (CN-100K and Atomic), and five medium-scale datasets (FB15K, FB15K-237, WN18, WN18RR, and Yoga3-10). We also used two large datasets (Wikidata5M-Trans and Wikidata5M-Ind). The detailed statistics of all datasets are presented in Table 1 and are from [25][12][26]:

- Kinships [25] and UML-S [27]: these two are small datasets, the KinShip [27] dataset is a dataset focusing on traditional Australian Aboriginal culture, and the UMLS [28] dataset is a dataset focusing on the relation between healthy living and biomedical information.
- CN-100K [28] and Atomic [29] Common KG datasets: Compared with traditional KGs, a common sense KG is slightly different. Its nodes are usually composed of more free-text formats and tend to be larger in scale and sparser than traditional KGs. CN-100K contains general knowledge facts about the world, and the original version includes open-minded common sense from ConceptNet. Atomic contains social common sense about current events. The detailed Statistics for these two datasets are shown in Table 1.
- WN18, FB15K, YAGO3-10 datasets: YAGO3-10 [30] is a KG based on Wikipedia, which expresses real-world facts and is a subset optimized by YAGO. WN18 and FB15K are two datasets composed of WordNet and Freebase respectively, which were proposed by Bordes et al. [6]. However, with the continuous utilization of the dataset, some later work [31][9] pointed out that these two datasets have problems such as test set leakage and relationship redundancy, and proposed WN18RR and FB15k-237 by removing the inverse relationship. They are composed of approximately 41,000 synsets and 11 relations from WordNet and approximately 15,000 entities and 237 relations from Freebase.
- Wikidata5M-Trans and Wikidata5M-Ind super-large-scale datasets: Wikidata5M [32] dataset is a dataset containing Wikipedia KG data and Wikipedia page data. The scale is much larger than other common datasets, with about 500 million entities and 20 million triples and more complex. Wikidata5M provides datasets of two schemes: transductive and inductive datasets. For Wikidata5M-Ind, all entities in the test set appear in the training set, while for Wikidata5M-Trans, there is no entity overlap between the training and test sets.

4.1.2. Evaluation metrics

The evaluation metrics of QIQE-KGC to measure link prediction follow the same evaluation method of the previous work [24][13]. They are: MRR, MR, and Hits@N (N = 1,3,10), and their calculation formulas are shown in (20)(21)(22).

$$\mathbf{MRR}=\frac{1}{|S|}\sum_{i=1}^{|S|}\frac{1}{\text{rank}_i}=\frac{1}{|S|}\left(\frac{1}{\text{rank}_1}+\frac{1}{\text{rank}_2}+\ldots+\frac{1}{\text{rank}_{|S|}}\right) \tag{20}$$

$$\mathbf{MR}=\frac{1}{|S|}\sum_{i=1}^{|S|}\text{rank}_i=\frac{1}{|S|}\left(\text{rank}_1+\text{rank}_2+\ldots+\text{rank}_{|S|}\right) \tag{21}$$

where S is the set of triples, $|S|$ is the number of sets of triples, and $rank_i$ is the link prediction rank of the i-th triple. The larger the MRR metric, the better the model is. The MR metric is just the opposite, the smaller the value, the better the model effect.

$$\mathbf{HITS@}n=\frac{1}{|S|}\sum_{i=1}^{|S|}\mathrm{II}\left(\text{rank}_i\leqslant n\right) \tag{22}$$

Table 2
Experimental results of proposed model performing link prediction on KinShips and UNL-S datasets.

Model	KinShips					UML-S				
	MRR	MR↓	Hit@1	Hit@3	Hit@10	MRR	MR↓	Hit@1	Hit@3	Hit@10
ComplEx [44]	0.838	-	75.40	-	98.000	0.894	-	82.300	-	99.500
NTP−λ [38]	0.800	-	76.000	82.000	89.000	0.930	-	87.000	<u>98.000</u>	**1.000**
ConvE [9]	0.871	-	79.700	-	98.100	0.957	1.51	93.200	-	99.400
KG-Bert [10]	-	-	-	-	-	-	1.47	-	-	99.000
DRGI [41]	0.847	1.9	76.500	91.500	98.100	0.898	1.5	83.800	94.800	98.800
StAR [11]	-	-	-	-	-	-	1.49	-	-	99.100
PRANN [36]	<u>0.952</u>	-	<u>91.800</u>	-	98.400	-	-	-	-	-
CoPER-CovE [42]	0.895	-	83.620	-	<u>98.420</u>	<u>0.971</u>	-	<u>95.460</u>	-	99.700
QIQE-KGC	**0.996**	**1.21**	**99.7672**	**99.7672**	**99.627**	**0.9972**	**1.08**	**99.697**	**99.697**	<u>99.773</u>

The function II (·) utilized here is an indicator function that outputs a value of 1 when the condition is met and 0 otherwise. The values of n used in this function are usually 1, 3, or 10. A higher output value from the indicator function indicates better performance of the model.

Consistent with the earliest work of KG representation learning [6][24], QIQE-KGC adopts a technique of randomly replacing head and tail entities for each triplet used in testing to generate corrupted sample triplets that serve as negative samples for our model. These corrupted samples were filtered if they still existed in the triplet set prior to training the model. The trained model output the scores for each triplet, which were sorted in descending order to obtain a ranking of the correctly predicted triplets within our candidate triplet entities.

4.1.3. Training protocol

Our model code is based on the OpenKE,[1] an open-source framework library and E2R [13] quantum logic, and uses PyTorch for the specific implementation. The experiments were performed on a Linux-based Nvidia GTX 3090 GPU PC server. The best hyperparameters of QIQE-KGC on each dataset were selected using a grid search. These optimal hyperparameters were based on MRR and Hits@N. For a comprehensive determination.

4.1.4. Compared model

To demonstrate the superior performance of the QIQE-KGC model on the KGC link prediction task on multiple datasets, and considering that different studies used different datasets, we chose the model in the following papers as our baseline models for comparison: embedding model baseds on Euler space [6,18,33,34]; models based on neural network and GNN [9,35,20,36,37,26,32,38,39], which design convolutional neural networks and specific GNNs to realize KGC tasks; models based on NLP and logical rule reasoning [11,12,10,40–42]; models that also used quaternion and complex space models [16,43–45,17,46], usually based on distance translation function representation operations for KGs; models based on quantum embeddings [13,24].

4.2. Experimental results and analysis

In this subsection, we comprehensively analyze the performance of QIQE-KGC for link prediction using different datasets. The superiority of QIQE-KGC is demonstrated through extensive experiments on 11 datasets.

4.2.1. Link prediction experiment results and analysis

The link prediction task results of KGC are presented in Table 2-7. The bold data in each column indicate the optimal results, the underline data indicate the suboptimal results; and the "−" symbol indicates that in the relevant original paper, there were no experimental results.

Performance Analysis of Link Prediction on Two Small KG Datasets. Table 2 summarizes the comparison of our proposed model with the baseline models on the KinShips and UML-S small datasets. The results of the baseline models are all taken from the respective original papers. The complexity of these two datasets is not high and the relation is relatively simple, so most of the models have shown very good results. For all metrics, our model has achieved the best results in the KinShips dataset and compared with other baseline models on the Hits@1 metric, there is a performance improvement of 1%−20%. In the experiment on the UML-S data set, the proposed model obtained four optimal results, and the suboptimal result evaluation index Hist@10 metric is also very close to the optimal one.

Performance Analysis of Link Prediction on Common Sense KG Datasets. Table 3 summarizes the comparison of our proposed model with the baseline models on two common sense KG datasets CN-100K and Atomic. The results of the baseline models were obtained from SIM [40], InductiveE [26], and their respective original papers. Compared with the baseline models, we can see clear advantages of QIQE-KGC. QIQE-KGC achieved four of the best results in the five metrics on the two datasets. For the CN-100K dataset, compared with the existing InductivE, which has the best effect, the MRR, Hits@1, and Hits@10 metrics for our model are increased

[1] https://github.com/thunlp/OpenKE.

Table 3
Experimental results of proposed model performing link prediction on Common-Sense KG datasets.

Model	CN-100K					Atomic				
	MRR	MR↓	Hit@1	Hit@3	Hit@10	MRR	MR↓	Hit@1	Hit@3	Hit@10
DistMult [39]	0.106	-	-	10.94	22.54	0.123	-	-	15.18	18.3
ComplEx [44]	0.115	-	-	12.40	20.31	0.142	-	-	14.13	15.96
ConvE [9]	0.208	-	-	22.91	34.02	0.100	-	-	10.29	13.37
RotatE [18]	0.247	-	-	28.2	45.41	0.111	-	-	11.54	15.60
BiQUE [17]	0.32	-	21.60	35.9	55.3	0.191	-	17.1	19.6	23.00
SIM(1) [40]	0.300	-	21.33	33.46	46.75	0.138	-	11.5	14.44	18.38
SIM(2) [40]	0.511	-	39.42	59.58	73.59	0.103	-	8.41	10.79	13.86
InductivE [26]	0.573	-	64.5	-	78	0.142	-	14.82	-	20.57
CKBC [37]	0.439	**169**	30.75	51.54	69.34	-	-	-	-	-
QIQE-KGC	**0.847**	3907	**84.79**	**84.79**	**84.791**	**0.633**	**45325**	**63.36**	**63.36**	**63.36**

Table 4
Experimental results of proposed model performing link prediction on FB15k and WN18 datasets.

Model	FB15K					WN18				
	MRR	MR↓	Hit@1	Hit@3	Hit@10	MRR	MR↓	Hit@1	Hit@3	Hit@10
TransE [6]	0.463	-	29.70	57.80	74.90	0.495	251	11.30	88.80	89.20
ConvE [9]	0.745	-	67.00	80.10	87.30	0.942	-	93.50	94.70	95.50
RotatE [18]	0.797	40	74.60	83.00	88.40	**0.949**	309	94.40	95.20	95.90
DistMult [39]	0.654	-	54.60	73.30	82.40	0.822	-	72.80	91.40	93.60
TransGare [33]	0.831	**32**	75.60	-	91.40	-	-	-	-	-
DA-GCN [35]	0.698	-	54.00	77.30	84.80	0.849	-	89.20	95.80	96.60
Rule-IC [47]	0.7732	38	71.78	81.64	88.85	0.888	**240**	93.76	94.85	96.65
QLogicE [24]	0.969	-	96.90	-	96.9	0.914	-	91.42	-	91.42
E2R [13]	0.960	72	96.40	-	96.4	0.710	5780	71.10	-	71.10
QIQE-KGC	**0.981**	54	**98.06**	**98.07**	**98.08**	0.979	363	**97.99**	**97.99**	**97.99**

by 47%, 31%, and 8% respectively. Compared with the NLP-based KGC model SIMKGC, the GNN-based CKBC, the translation-based embedding model RotatE, and the quaternion-based BiQUE, our model has a very obvious improvement. Compared with the baseline optimal model SIM on the Atomic dataset, Hits@10 increased by 173%, and the MRR increased by 231%. However, the performance of QIQE-KGC's MR metrics on the two datasets was not as good as expected. The reason may be that QIQE-KGC usually has an excellent ability to find the first ranking, but once the model cannot find the correct answer, the model cannot rank the correct result very well. Because of this, the MR of our model will not perform very well in both CN-100K and Atomic, which will be a key point that we need to improve in the future. In addition, compared with small datasets, we can clearly see that the performance of our model is not so good for common sense KGs. The essential reason is that the common sense KG is more complex and uses more free text format to describe nodes, the number of nodes is larger than that of traditional KGs, and the density of triples is sparser, which is another challenge in future KGC research.

Performance Analysis of Link Prediction on Traditional Datasets. Tables 4 and 5 summarize the experimental results for five traditional datasets. The QIQE-KGC model has very good results on FB15K, FB15K-237, WN18RR, and Yaga3-10. Although the performance of the WN18 dataset is not very good, it is still competitive. Compared to E2R [13], which also uses quantum logic embedding, the MRR, and MR metrics increased by 30.9% and 527%, respectively. On the FB15K dataset, compared with the baseline models, QIQE-KGC obtained three optimal and one suboptimal result for the five metrics. Compared with the E2R and QLogicE [24] models, which are also based on quantum logic, the MRR was improved by 1.66% and 1.23%; Compared with E2R, the MR metric increased by 33%; Compared with the logic rule-based model Rule-IC and the translation-based model RotatE, the Hits@10 metric was improved by 10.36% and 10.1% respectively.

It also achieved excellent results for the FB15K-237, WN18RR, and Yoga3-10 datasets. Compared with QLogicE, the MR metrics of the three datasets for our model increased by 3.6%, 1.1%, and 4.6%, respectively. Compared with the optimal model SimKGC [12] based on NLP and contrastive learning, Hits@10 in WN18RR for our model also has a 14.9% improvement, and it has a relatively better improvement compared to QuatE [16] BiQUE [17], and DualE [43], which are also based on quaternion space. All these results further illustrate the superiority of the QIQE-KGC model.

Performance Comparison of Link Prediction on Large-Scale Datasets. The experimental results of the QIQE-KGC model on two large-scale datasets are summarized in Table 6. In general, the experimental results of transductive learning should be better than inductive learning because all entities in the Wikidata5M-Transductive test set appear in the training set, while for Wikidata5M-Inductive there is no entity overlap between the training set and the test set. But what surprised us was that all models performed much better on Wikidata5M-Inductive than Wikidata5M-Transductive, which is clearly the opposite of the usual situation. After careful study of the dataset, we conclude that Wikidata5M-Inductive has only 201 relations in the test set, while Wikidata5M-Transductive has 822 relations in the test set. Although the entities have appeared in the training set, their representations differ

Table 5
Experimental results of proposed model performing link prediction on FB15k-237,WN18RR and YOGA3-10 datasets.

Model	FB15K-237			WN18RR			YAGO3-10		
	MRR	MR↓	Hit@10	MRR	MR↓	Hit@10	MRR	MR↓	Hit@10
TransE [6]	0.294	357	46.5	0.226	3384	50.1	0.501	-	67.39
RotatE [18]	0.358	177	53.3	0.476	3340	57.1	0.498	-	67.07
CLGAT-KGC [20]	0.364	160	55.1	0.484	2104	56.6	-	-	-
QuatE [16]	0.348	<u>87</u>	55	0.488	2314	58.2	-	-	-
DualE [43]	0.365	91	55.9	0.482	2270	58.4	-	-	-
BiQUE [17]	0.365	-	55.5	0.504	-	58.8	0.581	-	71.3
MorsE(TransE) [34]	-	-	<u>96.43</u>	-	-	76.46	-	-	-
Rule-IC [47]	0.355	166	55.2	0.436	3304	54.45	-	-	-
StAR [11]	0.365	117	56.2	0.551	**46**	73.2	-	-	-
ComplEx-SFBR [45]	0.374	-	56.7	0.5	-	58.4	0.584	-	71.3
PRANN [36]	0.66	-	-	-	-	-	-	-	-
SimKGC [12]	0.336	-	36.5	0.671	-	81.7	-	-	-
QLogicE [24]	<u>0.949</u>	-	94.89	<u>0.928</u>	-	<u>92.79</u>	<u>0.937</u>	-	<u>93.74</u>
QIQE-KGC	**0.984**	**17**	**98.5659**	**0.9391**	<u>946</u>	**93.9056**	**0.981**	**1035**	**98.19**

Table 6
Experimental results of proposed model performing link prediction on super large scale datasets.

Model	Wikidata5M-Trans					Wikidata5M-Ind				
	MRR	MR↓	Hit@1	Hit@3	Hit@10	MRR	MR↓	Hit@1	Hit@3	Hit@10
TransE [6]	0.253	-	17.0	0.311	39.2	-	-	-	-	-
RotatE [18]	0.29	-	23.4	32.2	39.0	-	-	-	-	-
KEPLER-Wiki [32]	0.154	**14454**	10.5	17.4	24.4	0.351	<u>32</u>	15.4	46.9	71.9
KEPLER-Cond [32]	0.210	<u>2</u>0267	17.3	22.4	27.7	0.402	**28**	22.2	51.4	73.0
SimKGC-I [12]	0.353	-	30.1	37.4	44.8	0.603	-	39.5	77.8	<u>92.3</u>
SimKGC-I+P [12]	0.354	-	30.2	37.3	44.8	0.602	-	39.4	77.7	**92.4**
SimKGC-I+S [12]	0.356	-	31.0	37.3	43.9	0.713	-	60.7	78.7	91.3
SimKGC-I+P+S [12]	0.358	-	31.3	37.6	44.1	0.714	-	60.9	78.5	91.7
QIQE-KGC	**0.657**	882254	**65.065**	**65.064**	**65.064**	**0.906**	35810	**90.587**	**90.587**	90.587

significantly across different triplets. In addition, the complexity of the relations in the test set of Wikidata5M-Inductive is much lower than that of Wikidata5M-Transductive, which explains the infrequent occurrences observed in Table 6.

From Table 6, in the comparison between QIQE-KGC and the baseline models on the Wikidata5M-Trans dataset, four of the five metrics have achieved the best performance and there are 81.5% and 47.3% higher than the MRR and Hits@10 metrics of SimKGC, respectively. However, the performance of the Hits@10 metric on Wikidata5M-Ind is slightly lower than that of SimKGC. This may be because the NLP-based KGC model SimKGC has a better generalization ability for entities that appear in the training set after adding text information. As we can see from the previous NLP-based KGC models [10][11], their experimental results in a less-relational environment are indeed excellent.

Summary 1. The effectiveness of QIQE-KGC was demonstrated in the link prediction experiments; and it has certain advantages for all kinds of KGC models, especially for the quantum embedding-based models E2R, QLogicE, and the quaternion space-based models QuatE, BiQUE, DualE, all of which have very considerable improvement. This shows that our model does indeed combine the excellent triadic modeling ability of quaternion space with the logical capture of deep dependencies in the KG by quantum embedding, and then interactively perform multi-task learning to obtain more powerful model results. In addition, we should also note that the model is not particularly good for MR metrics and the values of Hits@1 and Hits@3, and Hits@10 in the quantum logic-based models E2R [13] and QLogicE [24] are somewhat approximate but not exactly consistent. We explain this intuitively in Section 4.3 through a case study. When QIQE-KGC does not predict the correct result, the model tends to rank the result of the correct option at a lower position. In calculation formulas (20) and (21) for MRR and MR, we can clearly see that when there are many triplets with a large $rank_s$ value, the MR metric will become very large, while the impact on the MRR metric is relatively small. Thus, MRR is usually regarded as a good indicator in the KGC task; MR is used as an indicator to measure the degree of fluctuation in the model's predicted value.

4.2.2. Fine-grained analysis

To understand which types of relations QIQE-KGC performs better in actual modeling, we conducted fine-grained experiments, demonstrating the performance of QIQE-KGC in different conditions. We used the KG dataset WN18RR for the experiments. Table 7 presents the MRR metric performance of each relation for WN18RR. QIQE-KGC achieved outstanding results for most relation types, although it did not perform well in $similar\_to, verbgroup$, and other relations. We believe these relation types account for no more than 1.5% of the triples in the entire KG dataset, which is very small, and $similar\_to$ relation only accounts for 0.09%. However, the improvement in hierarchical logic relations and tree structure logic relations ($hypernym$ and $instance\ hypernym$) is very good

Table 7
Experimental results of the proposed model performance of fine-grained relation effects on WN18RR.

Relation_Name	RotatE	QuatE	BiQUE	Ours
hypernym	0.154	0.172	0.306	**1.000**
instance_hypernym	0.324	0.362	0.602	**1.000**
member_meronym	0.255	0.236	0.453	**1.000**
synset_domain_topic_of	0.334	0.395	0.539	**0.544**
has_part	0.205	0.210	0.392	**1.000**
member_of_domain_usage	0.277	0.372	**0.563**	0.333
member_of_domain_region	0.243	0.140	**0.500**	0.038
derivationally_related_form	0.957	0.952	0.970	**1.000**
also_see	0.627	0.607	**0.750**	0.001
verb_group	0.968	0.930	**0.974**	0.001
similar_to	1.000	1.000	**1.000**	0.001

Table 8
Results of ablation experiments with FB15K, WN18RR, and FB15K-237 datasets.

	FB15K			FB15K-237			WN18RR		
	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10	MRR	Hits@1	Hits@10
E2R	0.964	96.400	96.400	0.584	58.400	58.400	0.477	47.700	47.700
QuatE	0.808	75.100	89.600	0.367	26.900	56.300	0.493	43.900	59.200
QIQE-KGC*	0.971	97.110	97.214	0.957	95.712	95.712	0.9291	92.901	92.901
QIQE-KGC	0.981	98.067	98.067	0.984	98.480	98.565	0.939	93.905	93.905

compared to the QuatE and BiQUE models of pure quaternion space, which shows that our idea of interaction between quantum logic and quaternion space is feasible and very effective. Overall, these fine-grained analysis results, show our proposed scenario, effectively improving the model and achieving excellent performance.

4.2.3. *Ablation study and computational costs analysis*

To further enhance the understanding of the effectiveness of each module of QIQE-KGC, we conducted ablation experiments. The experimental results are presented in Table 8; the results of E2R [13] and QuatE [16] are from the original papers. QIQE-KGC* represents an incomplete model that did not use head-tail entity-type correlation enhancement methods. The MRR metrics on FB15K, FB15K-237, and WN18RR datasets increased by 1.13%, 2.82%, and 1.07% respectively through the method of head-tail entity correlation enhancement. Experiments show that our head-tail entities correlation enhancement method can actually enhance the interaction ability of the head and tail entities to improve the effectiveness of the model. In addition, QIQE-KGC, after learning the interaction between quantum embedding and quaternion space, compared with the simple quaternion space model QuatE and E2R, has improved the MRR metric by 21.41% and 1.7% respectively in FB15K. On the other two datasets, an astonishing two-fold effect is achieved, which strongly proves that our proposed KGC model based on the interaction of quantum embeddings and quaternion space is very effective.

Through the ablation study, we can further discover the limitations of E2R. When dealing with more complex datasets (FB15K-237/WN18RR), the performance of E2R is obviously poor. QIQE-KGC combines quantum embedding with quaternion space, obtaining the excellent modeling capabilities of entities and relations within triplets in quaternion space and logical capture of deep dependencies between different triplets in KGs by quantum embedding. This effectively addresses the shortcomings of E2R and yields impressive results.

To further analyze the computational cost of the models, we compared the training time of E2R and QIQE-KGC on two public datasets (FB15K, WN18). We set identical parameters and conducted the experiments on the same PC server. As shown in Fig. 3, the training time required for QIQE-KGC on both datasets was longer than that of E2R, which is very reasonable and in line with expectations. To some extent, we sacrificed time complexity to achieve better performance. The incorporation of the quaternion module enables the model to handle more complex triple relationships and achieve better results. However, it also introduces additional computational costs. In the future, we will consider how to shorten the time required for model training while maintaining excellent model results.

4.2.4. *Parameter sensitivity experiment*

To explore how well the model responds to parameter sensitivity, as shown in Fig. 4, we analyzed the ratio of the quantum embedding module to the quaternion module in the loss function on two small datasets KinShips and UML-S. The ratio of the model on MR and Hits@10 shows that the lower the MR metric, the better the model effect, and the higher the Hits@10, the better the effect. Combining the MR and Hits@10 metrics in Fig. 4, we find that when $\alpha : \beta$ is close to 1:1 and the loss weight for the two modules is similar, the model obtains better results, which confirms that both modules are important. In addition, Fig. 4 also shows that the model will not cause large fluctuations in the effect of the model with slight fluctuations in parameters, which shows the robustness of our model.

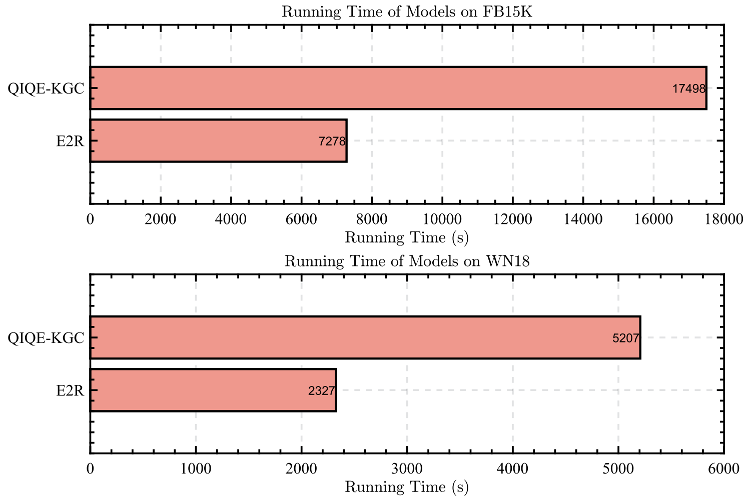

Fig. 3. Computational costs of FB15K and WN18 datasets.

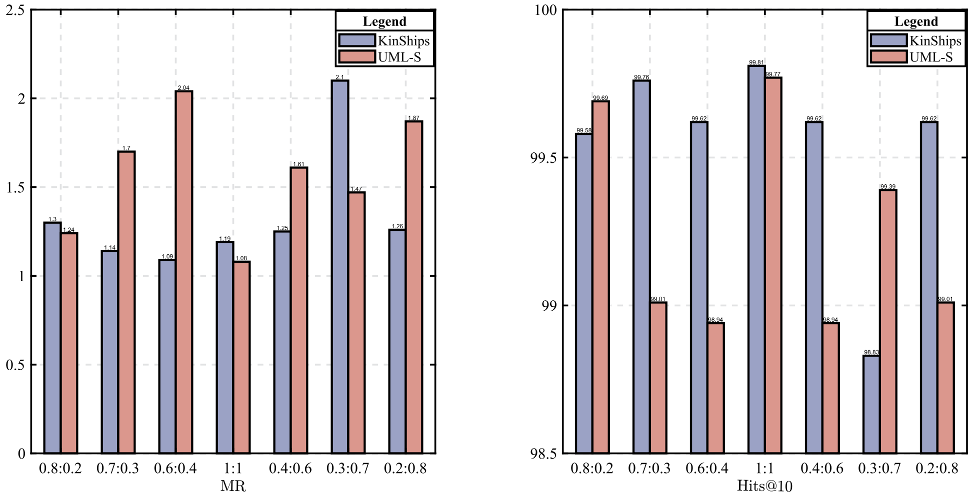

Fig. 4. The results of parameter sensitivity experiments on the Kinship and UML-S datasets.

4.3. Case study

To better understand the QIQE-KGC model, several examples are provided to demonstrate how the model works and its practical effects. Fig. 5 shows case studies in the UML-S dataset with four correct and incorrect predictions. In the experiment, the entities and relationships were formed into a mapping ID and stored in the mapping file. The cases in the figure show the intuitive results obtained after parsing the file. As shown in Fig. 5, to predict a tail entity, the head entity, and the relation are embedded in a known feature vector to find the possible correct tail entity in the form of $(h,r,?)$. In the entire test set, all entities except the head entity appear in predicates as potential candidate entities. When the model runs and receives the input h and r, scores are calculated for all candidate entities and ranked. The missing tail entities are predicted perfectly in triplets 1, 2, and 3. This also implies that the respective correct tail entity ranks first in the set of prediction candidates.

Triple 4 also successfully predicts the correct missing head entity. Similar to predicting the tail entity, after receiving the input r and t, the model considers all entities in the entire test set as candidates except the tail entity itself. Then the model will give all the scores and ranks of candidate entities.

To explain why Hits@1≈Hits@3≈Hits@10 in the QIQE-KGC model evaluation metrics, Triples 5 and 6 are selected and analyzed as the prediction errors in the experiment. As shown in Fig. 5, the correct tail entity is ranked at 64 in Triples5, when predicting the tail entities in the form of triple(laboratory_procedure, analyzes,?). Similarly in Triples6, the correct candidate triples are ranked at 24, when predicting the triple (?, property_of,family_group). That is the explanation why our Hits@1≈Hits@3≈Hits@10 values. The correct candidate is usually ranked first when the model is able to predict the missing head and tail entities in the triplet.

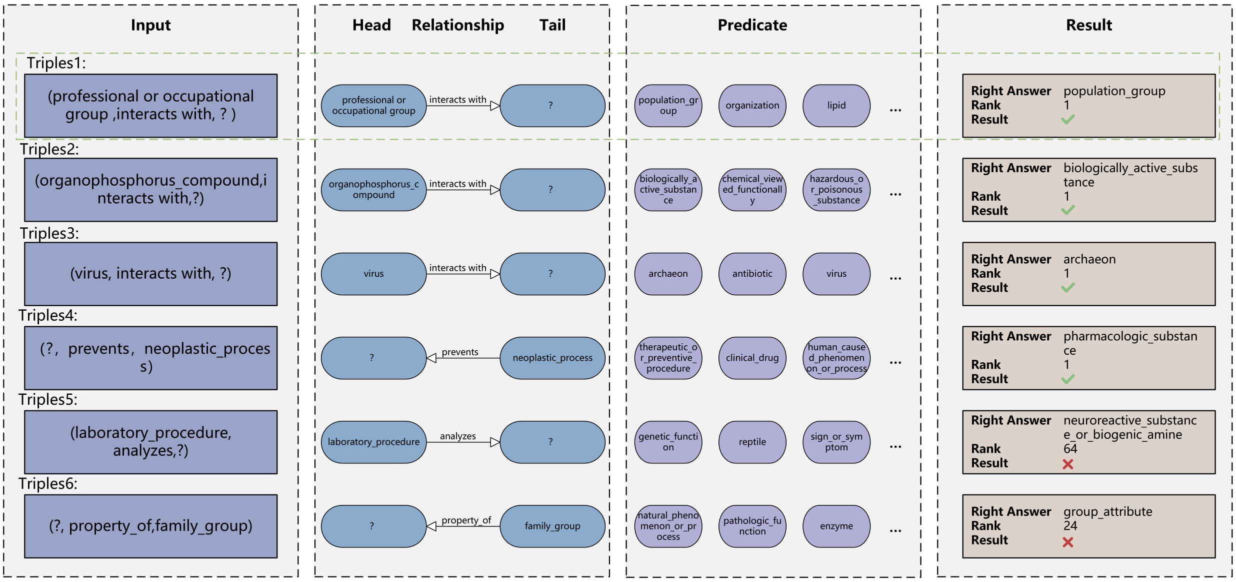

Fig. 5. Case study example of UML-S dataset.

Correspondingly, the model usually ranks the correct candidate more towards the bottom than the top 10, when the model does not have the ability to predict the missing head or tail entities.

5. Conclusions and future research

In this study, we proposed a knowledge graph completion model QIQE-KGC based on the interaction of quantum embeddings and quaternion space. By jointly capturing the internal semantic information structure of individual triplets and the external deep logical structure between different triplets in a KG, QIQE-KGC achieves excellent performance on numerous datasets. Our proposed method of enhancing the correlation between head-tail entity types in relations also contributes to more comprehensive modeling in quaternion space. However, the performance of our model on ultra-large-scale KGs and commonsense KGs is still not satisfactory. In the future, we will consider improving these two modules to extend our model and make it more robust. In addition, we will deeply study MR metrics and improve the performance of the model on MR metrics in the future.

In recent years, Multimodal Knowledge Graphs (MKGs) [48][49] have been gradually gaining popularity as a novel method for knowledge representation. MKGs extend the traditional entities and relations encapsulated in conventional KG, incorporating different modalities of information such as text, images, and audio. MKGs organize and integrate diverse types of information sources in a structured manner, offering a new perspective and methodology for complex information processing and understanding. We perceive the tasks pertaining to MKG completion [50] as compelling areas of investigation. Thus, our future research agenda aims to explore the study of representation learning in the context of MKGs.

CRediT authorship contribution statement

LinYu Li: Conceptualization, Methodology, Software, Writing - Original Draft. **Xuan Zhang:** Supervision, Writing - Review & Editing. **Zhi Jin:** Supervision, Writing - Review & Editing. **Chen Gao:** Writing - Review. **Rui Zhu:** Writing - Reviewing. **YuQin Liang:** Visualization, Investigation. **YuBin Ma:** Writing - Review.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This work was supported by the National Key Research and Development Program of China under Grant No. 2020AAA0109400; The Science Foundation of Young and Middle-aged Academic and Technical Leaders of Yunnan under Grant No. 202205AC160040;

The Science Foundation of Yunnan Jinzhi Expert Workstation under Grant No. 202205AF150006; The Major Project of Yunnan Natural Science Foundation under Grant No. 202302AE09002003; The Yunnan Provincial Department of Education Scientific Research Fund Project under Grant No. 2023Y0256; The Science and Technology Project of Yunnan Power Grid Co., Ltd. under Grant No. YNKJXM20222254.

Appendix A

In the quantum embedding module, we use the basic idea in the previous work [13], abandoning the regularization loss term and the unary relationship loss term.

The loss function of the entity element loss $\mathcal{L}_{ELoss}$ is contained as:

$$L_{E_i} = \left\| e_i \odot \begin{pmatrix} \mathbf{0}_d \\ \mathbf{1}_d \end{pmatrix} \right\|^2 \tag{23}$$

where $\odot$ means element-wise multiplication.

$$\mathcal{L}_{R_i} = \left(\min \left\{ 0, \left(\gamma_i^\top \begin{pmatrix} \mathbf{0}_d \\ \mathbf{1}_d \end{pmatrix} - 1 \right) \right\} \right)^2 + \left(\min \left\{ 0, \left(\gamma_i^\top \begin{pmatrix} \mathbf{1}_d \\ \mathbf{0}_d \end{pmatrix} - 1 \right) \right\} \right)^2 \tag{24}$$

where L_{R_i} represents the binary relationship loss function, γ_i represents the metric vector as a learnable parameter, and the value of the function $\min\{0, x\}$ is not less than 0.

$$L_{\gamma_i} = \| \gamma_i \odot \overline{\gamma_i} \|^2 \tag{25}$$

In order to ensure that the metric vector works normally, the L_{γ_i} loss function is used to constrain enforcement.

$$\mathcal{L}_{ELOSS} = L_{E_i} + L_{R_i} + L_{\gamma_i} \tag{26}$$

The loss function of the logical loss $\mathcal{L}_{LLoss}$ is important, they contained logic inclusion, logic conjunction, logic disjunction, and logic negation. According to the quantum logic embedding in [13], the logic inclusion loss term of binary relationship form is:

$$L_{R_i \sqsubseteq R_j} = \left\| \boldsymbol{\gamma}_i \odot \overline{\boldsymbol{\gamma}_j} \right\|^2 \tag{27}$$

According to the quantum logic embedding in [13], the logic conjunction loss term of binary relationship form is:

$$L_{(R_i = R_j \sqcap R_k)} = \left\| \boldsymbol{\gamma}_i - (\boldsymbol{\gamma}_j \odot \boldsymbol{\gamma}_k) \right\|^2 \tag{28}$$

According to the quantum logic embedding in [13], the logic disjunction loss term of binary relationship form is:

$$L_{(R_i = R_j \sqcup R_k)} = \left\| \gamma_i - \max(\gamma_j, \gamma_k) \right\|^2 \tag{29}$$

According to the quantum logic embedding in [13], the logic negation loss term of binary relationship form is:

$$L_{(R_i = \neg R_j)} = (\boldsymbol{\gamma}_i^\top \boldsymbol{\gamma}_j)^2 + \left(\overline{\boldsymbol{\gamma}}_i^\top \overline{\boldsymbol{\gamma}}_j \right)^2 \tag{30}$$

$$\mathcal{L}_{LLoss} = L_{R_h \sqsubseteq R_t} + L_{(R_i = R_j \sqcap R_k)} + L_{(R_i = R_j \sqcup R_k)} + L_{(R_i = \neg R_j)} \tag{31}$$

The logical loss function can be understood as the sum of the above loss functions.

$$\begin{aligned} L_{R_i}(E_h, E_t) = & \| \boldsymbol{\gamma}_i \odot \boldsymbol{E}_{ht} \|^2 \\ & + \left(\mathbf{1}_{2d}^\top \left(\left(\begin{pmatrix} \mathbf{0}_d & \mathbf{1}_d \\ \mathbf{1}_d & \mathbf{0}_d \end{pmatrix} \boldsymbol{\gamma}_i \right) \odot \boldsymbol{E}_{ht} \right) \right)^2 \\ & + \left\| \begin{pmatrix} \mathbf{1}_d \\ \mathbf{0}_d \end{pmatrix} \odot \boldsymbol{E}_{ht} - \boldsymbol{E}_h \right\|^2 \\ & + \left\| \begin{pmatrix} \mathbf{0}_d \\ \mathbf{1}_d \end{pmatrix} \odot \boldsymbol{E}_{ht} - \boldsymbol{E}_t \right\|^2 \end{aligned} \tag{32}$$

According to the quantum logic embedding in [13], the membership loss term is constructed from the remaining length of the projection. The membership relationship here is actually a multi-hop relationship, which means that the model learns their features through the loss term by projecting this relationship into the vector space.

References

[1] S. Ji, S. Pan, E. Cambria, P. Marttinen, S.Y. Philip, A survey on knowledge graphs: representation, acquisition, and applications, IEEE Trans. Neural Netw. Learn. Syst. 33 (2) (2021) 494–514.

[2] H. Cui, T. Peng, F. Xiao, J. Han, R. Han, L. Liu, Incorporating anticipation embedding into reinforcement learning framework for multi-hop knowledge graph question answering, Inf. Sci. 619 (2023) 745–761.

[3] Y. Yang, C. Huang, L. Xia, C. Li, Knowledge graph contrastive learning for recommendation, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 1434–1443.
[4] D. Zou, W. Wei, Z. Wang, X.-L. Mao, F. Zhu, R. Fang, D. Chen, Improving knowledge-aware recommendation with multi-level interactive contrastive learning, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 2817–2826.
[5] T. Shen, F. Zhang, J. Cheng, A comprehensive overview of knowledge graph completion, Knowl.-Based Syst. (2022) 109597.
[6] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Adv. Neural Inf. Process. Syst. 26 (2013).
[7] Y. Lin, Z. Liu, M. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2015, pp. 2181–2187.
[8] G. Ji, S. He, L. Xu, K. Liu, J. Zhao, 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), 2015, pp. 687–696.
[9] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, Convolutional 2d knowledge graph embeddings, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2018, pp. 1811–1818.
[10] L. Yao, C. Mao, Y. Luo, Kg-bert: Bert for knowledge graph completion, arXiv preprint arXiv:1909.03193, 2019.
[11] B. Wang, T. Shen, G. Long, T. Zhou, Y. Wang, Y. Chang, Structure-augmented text representation learning for efficient knowledge graph completion, in: Proceedings of the Web Conference 2021, 2021, pp. 1737–1748.
[12] L. Wang, W. Zhao, Z. Wei, J. Liu, Simkgc: simple contrastive knowledge graph completion with pre-trained language models, arXiv preprint arXiv:2203.02167, 2022.
[13] D. Garg, S. Ikbal, S.K. Srivastava, H. Vishwakarma, H. Karanam, L.V. Subramaniam, Quantum embedding of knowledge for reasoning, Adv. Neural Inf. Process. Syst. 32 (2019).
[14] G. Birkhoff, J. Von Neumann, The logic of quantum mechanics, Ann. Math. (1936) 823–843.
[15] V.S. Varadarajan, Geometry of Quantum Theory, vol. 1, Springer, 1968.
[16] S. Zhang, Y. Tay, L. Yao, Q. Liu, Quaternion knowledge graph embeddings, Adv. Neural Inf. Process. Syst. 32 (2019).
[17] J. Guo, S. Kok, Bique: biquaternionic embeddings of knowledge graphs, arXiv preprint arXiv:2109.14401, 2021.
[18] Z. Sun, Z.-H. Deng, J.-Y. Nie, J. Tang, Rotate: knowledge graph embedding by relational rotation in complex space, arXiv preprint arXiv:1902.10197, 2019.
[19] Y.-C. Wu, F. Yin, X.-Y. Zhang, L. Liu, C.-L. Liu Scan, Sliding convolutional attention network for scene text recognition, arXiv preprint arXiv:1806.00578, 2018.
[20] L. Li, X. Zhang, Y. Ma, C. Gao, J. Wang, Y. Yu, Z. Yuan, Q. Ma, A knowledge graph completion model based on contrastive learning and relation enhancement method, Knowl.-Based Syst. 256 (2022) 109889.
[21] W.R. Hamilton, Lxxviii. on quaternions; or on a new system of imaginaries in algebra: to the editors of the philosophical magazine and journal, Lond. Edinb. Dublin Philos. Mag. J. Sci. 25 (169) (1844) 489–495.
[22] C.J. Gaudet, A.S. Maida, Deep quaternion networks, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.
[23] T. Parcollet, Y. Zhang, M. Morchid, C. Trabelsi, G. Linarès, R. De Mori, Y. Bengio, Quaternion convolutional neural networks for end-to-end automatic speech recognition, arXiv preprint arXiv:1806.07789, 2018.
[24] P. Chen, Y. Wang, X. Yu, R. Feng, Qlogice: quantum logic empowered embedding for knowledge graph completion, Knowl.-Based Syst. 239 (2022) 107963.
[25] C. Kemp, J.B. Tenenbaum, T.L. Griffiths, T. Yamada, N. Ueda, Learning systems of concepts with an infinite relational model, in: AAAI, vol. 3, 2006, p. 5.
[26] B. Wang, G. Wang, J. Huang, J. You, J. Leskovec, C.-C.J. Kuo, Inductive learning on commonsense knowledge graph completion, in: 2021 International Joint Conference on Neural Networks (IJCNN), IEEE, 2021, pp. 1–8.
[27] A.T. McCray, An upper-level ontology for the biomedical domain, Comp. Funct. Genomics 4 (1) (2003) 80–84.
[28] X. Li, A. Taheri, L. Tu, K. Gimpel, Commonsense knowledge base completion, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 1445–1455.
[29] M. Sap, R. Le Bras, E. Allaway, C. Bhagavatula, N. Lourie, H. Rashkin, B. Roof, N.A. Smith, Y. Choi, Atomic: an atlas of machine commonsense for if-then reasoning, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 3027–3035.
[30] F. Mahdisoltani, J. Biega, F. Suchanek, Yago3: a knowledge base from multilingual wikipedias, in: 7th Biennial Conference on Innovative Data Systems Research, CIDR Conference, 2014.
[31] K. Toutanova, D. Chen, P. Pantel, H. Poon, P. Choudhury, M. Gamon, Representing text for joint embedding of text and knowledge bases, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 1499–1509.
[32] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, J. Tang Kepler, A unified model for knowledge embedding and pre-trained language representation, Trans. Assoc. Comput. Linguist. 9 (2021) 176–194.
[33] J. Yuan, N. Gao, J. Xiang, Transgate: knowledge graph embedding with shared gate structure, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2019, pp. 3100–3107.
[34] M. Chen, W. Zhang, Y. Zhu, H. Zhou, Z. Yuan, C. Xu, H. Chen, Meta-knowledge transfer for inductive knowledge graph embedding, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2022, pp. 927–937.
[35] J. Zhang, J. Huang, J. Gao, R. Han, C. Zhou, Knowledge graph embedding by logical-default attention graph convolution neural network for link prediction, Inf. Sci. 593 (2022) 201–215.
[36] B. Jagvaral, W.-K. Lee, J.-S. Roh, M.-S. Kim, Y.-T. Park, Path-based reasoning approach for knowledge graph completion using cnn-bilstm with attention mechanism, Expert Syst. Appl. 142 (2020) 112960.
[37] J. Ju, D. Yang, J. Liu, Commonsense knowledge base completion with relational graph attention network and pre-trained language model, in: Proceedings of the 31st ACM International Conference on Information & Knowledge Management, 2022, pp. 4104–4108.
[38] T. Rocktäschel, S. Riedel, End-to-end differentiable proving, Adv. Neural Inf. Process. Syst. 30 (2017).
[39] B. Yang, W.-t. Yih, X. He, J. Gao, L. Deng, Embedding entities and relations for learning and inference in knowledge bases, arXiv preprint arXiv:1412.6575, 2014.
[40] C. Malaviya, C. Bhagavatula, A. Bosselut, Y. Choi, Commonsense knowledge base completion with structural and semantic context, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 2925–2933.
[41] S. Liang, J. Shao, D. Zhang, J. Zhang, B. Cui, Drgi: deep relational graph infomax for knowledge graph completion, IEEE Trans. Knowl. Data Eng. (2021).
[42] G. Stoica, O. Stretcu, E.A. Platanios, T. Mitchell, B. Póczos, Contextual parameter generation for knowledge graph link prediction, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, pp. 3000–3008.
[43] Z. Cao, Q. Xu, Z. Yang, X. Cao, Q. Huang, Dual quaternion knowledge graph embeddings, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 6894–6902.
[44] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, Complex embeddings for simple link prediction, in: International Conference on Machine Learning, PMLR, 2016, pp. 2071–2080.
[45] Z. Liang, J. Yang, H. Liu, K. Huang, A semantic filter based on relations for knowledge graph completion, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 7920–7929.
[46] A. Saxena, A. Kochsiek, R. Gemulla, Sequence-to-sequence knowledge graph completion and question answering, arXiv preprint arXiv:2203.10321, 2022.
[47] Q. Lin, J. Liu, Y. Pan, L. Zhang, X. Hu, J. Ma, Rule-enhanced iterative complementation for knowledge graph reasoning, Inf. Sci. 575 (2021) 66–79.

[48] Z. Wang, L. Li, Q. Li, D. Zeng, Multimodal data enhanced representation learning for knowledge graphs, in: 2019 International Joint Conference on Neural Networks (IJCNN), IEEE, 2019, pp. 1–8.
[49] J. Zhu, C. Huang, P. De Meo, Dfmke: a dual fusion multi-modal knowledge graph embedding framework for entity alignment, Inf. Fusion 90 (2023) 111–119.
[50] D. Xu, T. Xu, S. Wu, J. Zhou, E. Chen, Relation-enhanced negative sampling for multimodal knowledge graph completion, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 3857–3866.