Abstract

Knowledge graph embeddings (KGEs) learn low-dimensional representations of entities and relations to predict missing facts based on existing ones. Quantum-based KGEs utilize variational quantum circuits for link prediction and score triples via the probability distribution of measuring the qubit states. However, there exists another best measurement for training variational quantum circuits. Besides, current quantum-based methods ignore theoretical analysis which are essential for understanding the model performance and applying for downstream tasks such as reasoning, path query answering, complex query answering, etc. To address measurement issue and bridge theory gap, we propose QubitE whose score of a triple is defined as the similarity between qubit states. Here, our measurements are viewed as kernel methods to separate the qubit states, while preserving quantum advantages. Furthermore, we show that (1) QubitE is full-expressive; (2) QubitE can infer various relation patterns including symmetry/antisymmetry, inversion, and commutative/non-commutative composition; (3) QubitE subsumes serveral existing approaches, e.g. DistMult, pRotatE, RotatE, TransE and ComplEx; (4) QubitE owns linear space complexity and linear time complexity. Experiments results on multiple benchmark knowledge graphs demonstrate that QubitE can achieve comparable results to the state-of-the-art classical models.

1 Introduction

Knowledge graphs (KGs) consist of nodes (entities) and edges (relationships between entities), which have been widely applied for knowledge-driven tasks such as question answering, recommendation system, and search engine. However, KGs are incomplete and this problem affects the performance of any algorithm related to KGs. Knowledge graph embeddings (KGEs) are prominent approaches to predict missing links for KG completion.
to be a fixed circuit, and the training adapts the measurement basis. By contrast, we note that if the entities are well-separated in Hilbert space, the best measurements, that distinguish whether the entities are the tails of the tuple \((h, r, ?)\) or not, are known as follows: The best measurement for the entities separated by the trace distance is the Helstrom minimum error measurement, and the best measurement for the Hilbert-Schmidt distance is the fidelity or overlaps measurement between the semantics of embedded entities. Therefore, we argue that, the adaptive training of the quantum circuit should focus on the metric that carries out a maximally separating embedding.

In this paper, we propose a new quantum-based KGE for knowledge graph completion to explore the performance of different measurements. We numerically investigate different measurements for training quantum embeddings on four standard datasets. Extensive experiments demonstrate the efficacy of our model.

In addition, we analysis our model theoretically, including subsumption, full expressiveness, patterns inference and space&time complexity. We prove that QubitE is fully expressive and deriving a bound on the embedding dimensionality for full expressiveness, which is the crucial property that indicates well-separation of the data. We show that QubitE subsumes TransE, RotatE, pRotatE, ComplEx and DisMult. We also prove that QubitE allows to learn composition, inverse and symmetric relation patterns. Besides, QubitE owns linear space complexity and linear time complexity.

We summarise our contributions as follows:

- **KGE**: We propose QubitE, a new linear quantum-based KGE model for link prediction on knowledge graphs, that is simple and expressive to explore the performance of different measurements.

- **Theoretical Analysis**: We fully analysis QubitE theoretically in subsumption, full expressiveness, patterns inference and space&time complexity.

- **Experiments**: We conduct extensive experiments on four standard public datasets to demonstrate the efficacy of our model. The source code is available online \(^1\).

2 Related Work

The KG embedding is divided into the following categories, Euclidean geometric model, non-Euclidean geometric model, tensor decomposition model, neural network model, etc.

**Euclidean KG Embedding.**

TransE (Bordes et al., 2013) models the relationship as a distance transformation from the head entity to the tail entity; TransR (Lin et al., 2015) proposes to design a projection matrix for each relationship, in order that entities have different embedding vectors under different relationships; RotatE (Sun et al., 2019) defines the relationship as rotation transformation from head entities to tail entities in the two-dimensional complex space; QuatE (Zhang et al., 2019) uses the quaternion method to extend the rotation to three-dimensional complex space; 5*E (Nayyeri et al., 2021) proposes a model based on projective geometry that provides a unified method for simultaneously representing translation, rotation, homomorphism, inversion, and reflection.

**Non-Euclidean KG Embedding.**

MuRP (Balazevic et al., 2019b) models both in hyperbolic space and Euclidean space, and combines relationship vectors, which can handle the multiple types of relationships that exist in the graph; ATTH (Chami et al., 2020) uses the expressiveness of hyperbolic space and attention-based geometric transformation to learn improved KG representation in low-dimensional space.

**Tensor Decomposition KG Embedding.**

DistMult (Yang et al., 2015) relaxes the constraint on the relationship matrix and uses a diagonal matrix to represent the relationship matrix; ComplEx (Trouillon et al., 2016) extends to the complex space, which can solve both symmetric and asymmetric relationships at the same time; Simple (Kazemi and Poole, 2018) proposed a simple Canonical Polyadic (CP) enhancement to allow the two embeddings of each entity to be learned independently; HypER (Balazevic et al., 2019a) uses a hypergraph network to generate a one-dimensional convolution filter for each relationship, in order to extract the specific characteristics of the relationship; TuckER (Balazevic et al., 2019c) proposes a model that uses Tucker decomposition to perform link prediction on the binary tensor representation of KG.

**Neural Network KG Embedding.**

ConvE (Dettmers et al., 2018) uses a convolu-

\(^1\)https://github.com/LinXueyuanStudio/QubitE
tional neural network to define the scoring function; CoPER (Stoica et al., 2020) generates contextual parameters into neural network to predict links.

**Quantum Embedding.**

Ma et al. (2019) proposes two types of variational quantum circuits (QCE and F-QCE) for knowledge graph embedding. Lloyd et al. (2020) proposes a quantum embedding model that represents classical data points as quantum states in a Hilbert space via quantum feature map. A classical datapoint $x$ is translated into a set of gate parameters in a quantum circuit $\psi$, creating a quantum state $|x\rangle$ such that $\psi : x \rightarrow |x\rangle$. However, our method is quite different. Firstly, we compare the quantum states via trace distance rather than the probability distribution of measuring the qubit states. Secondly, entities in KG are assigned tunable parameters directly to create quantum states instead of using parametric quantum circuits.

### 3 Preliminaries

**Knowledge Graph Embeddings.** A KG is a multi-relational directed graph $KG = (E, R, T)$ where $E$ is the set of nodes (entities) and $R$ is the set of edges (relations between entities). The set $T = \{(h, r, t)\} \subseteq E \times R \times E$ contains all triples as $\text{(head, relation, tail)}$, e.g. $(\text{smartPhone, hypernym, iPhone})$. To apply learning methods on KGs, a KGE learns vector representations of entities ($E$) and relations ($R$). A vector representation denoted by $(h, r, t)$ is learned by the model per triple $(h, r, t)$, where $h, t \in \mathbb{R}^d, r \in \mathbb{R}^d$ ($\mathbb{R}^d$ is a $d$-dimensional vector space). TransE (Bordes et al., 2013) considers $V = \mathbb{R}$ while ComplEx (Trouillon et al., 2016) and RotatE use $V = \mathbb{C}$ (complex space) and QuatE (Zhang et al., 2019) considers $V = \mathbb{H}$ (quaternion space). In this paper, we choose two-dimensional Hilbert space to embed the graph i.e. $V = \mathbb{C}^2$. Most KGE models are defined via a relation-specific transformation function $g_r : \mathbb{R}^d \rightarrow \mathbb{R}^d$ which maps head entities to tail entities, i.e. $g_r(h) = t$. On top of such a transformation function, the score function $f : \mathbb{R}^d \times \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ is defined to measure the plausibility for triples: $f(h, r, t) = p(g_r(h), t)$. Generally, the formulation of any score function can be either $p(g_r(h), t) = -\|g_r(h) - t\|$ or $p(g_r(h), t) = \langle g_r(h), t \rangle$.

**Qubit.** A classical bit can exist in one of two states denoted as $0$ and $1$. A quantum bit or qubit can exist not only in these two discrete states but in all possible linear superpositions of them. Mathematically, the quantum state of a qubit is represented as a state vector in a two-dimensional Hilbert space $\mathbb{C}^2$, whose basis vectors are denoted in the Dirac notation as

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

Let the vector $|0\rangle$ correspond to the classical value $0$, while $|1\rangle$ to $1$. The state vector of a qubit is written as

$$|\psi\rangle = a|0\rangle + b|1\rangle$$

where $a, b \in \mathbb{C}, |a|^2 + |b|^2 = 1$. The complex numbers $a$ and $b$ are called quantum amplitudes.

According to quantum mechanics, if we make measurement on $|\psi\rangle$ to see whether it is in $|0\rangle$ or $|1\rangle$, the outcome will be $0(1)$ with the probability $|a|^2(|b|^2)$ and state $|0\rangle(|1\rangle$) immediately. The density matrix $\rho$ of state $|\psi\rangle$ is given by:

$$\rho = |\psi\rangle \langle \psi|$$

**Quantum Gates.** Quantum gates essentially transform the system from one state to another state. When measurements are not made, the time evolution of a state is described by the Schrödinger equation. Because of the probabilistic interpretation of quantum mechanics, state vectors are normalized to $1$. Thus the time development is unitary. Quantum gate $U$ holds $UU^\dagger = U^\dagger U = I$, where $U^\dagger$ is the conjugate transpose of matrix $U$. The general expression of a $2 \times 2$ unitary matrix is

$$U = \begin{pmatrix} a & -e^{i\psi}b^* \\ b & e^{i\psi}a^* \end{pmatrix}$$

where $a, b \in \mathbb{C}, |a|^2 + |b|^2 = 1$ and $\psi$ is the angle. $a^*$ is the complex conjugate of $a$.

### 4 Method

#### 4.1 Model Formulation

Given a triple $(h, r, t)$, the head and tail entities $h, t \in E$ are embedded into a $d$ dimensional Hilbert space $i.e. \ h, t \in \mathbb{C}^{2d}$ where each element is a 2-dimensional complex value vector. A relation $r \in R$ is embedded into a $d$ dimensional vector $r$ where each element is a $2 \times 2$ complex value unitary matrix. $r$ contains two complex vectors $r_a$ and $r_b \in \mathbb{C}^d$. With $r_{ai}, r_{bi}, h_{ai}, h_{bi}, t_{ai}, t_{bi}$, we refer to the $i$th element of $r_a, r_b, h_a, h_b, t_a, t_b$ respectively.
4.1.1 Entity-specific Qubit Embedding
We use standard representation of the state of qubit to represent an entity in \( \mathbb{C}^{2d} \). The \( i \)-th element of entity embedding vector \( \mathbf{h} \) is given by
\[
\mathbf{h}_i = \mathbf{h}_{ai} |0\rangle + \mathbf{h}_{bi} |1\rangle = \left( \begin{array}{c} \mathbf{h}_{ai} \\ \mathbf{h}_{bi} \end{array} \right),
\]
where \( d \) is entity embedding dimension, \( \mathbf{h}_{ai}, \mathbf{h}_{bi} \in \mathbb{C} \) and \( |\mathbf{h}_{ai}|^2 + |\mathbf{h}_{bi}|^2 = 1 \) such that \( \mathbf{h} = [\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_d] \).

Respectively, the density matrix of entity \( h \) is
\[
\rho_h = |\mathbf{h}_i\rangle \langle \mathbf{h}_i| = \left( \begin{array}{cc} |\mathbf{h}_{ai}|^2 & \mathbf{h}_{ai}^* \mathbf{h}_{bi}^* \\ \mathbf{h}_{bi} \mathbf{h}_{ai}^* & |\mathbf{h}_{bi}|^2 \end{array} \right).
\]

4.1.2 Relation-specific Quantum Gate
We use relation-specific transformation to map the head entity \( h \) from a source to a target Hilbert space. Since quantum gates are unitary, we write the parameterized unitary matrix of \( i \)-th element of relation embedding vector \( r \) as
\[
r_i = \mathcal{U}_{ri} = \left( \begin{array}{cc} \mathbf{r}_{ai} & -e^{i\psi} \mathbf{r}_{bi}^* \\ \mathbf{r}_{bi} & e^{i\psi} \mathbf{r}_{ai}^* \end{array} \right),
\]
where \( d \) is relation embedding dimension, \( \mathbf{r}_{ai}, \mathbf{r}_{bi} \in \mathbb{C} \) and \( |\mathbf{r}_{ai}|^2 + |\mathbf{r}_{bi}|^2 = 1 \), so that \( r = [\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_d] \). This implies \( \det(\mathcal{U}_{ri}) = e^{i\psi} \neq 0 \), i.e. \( \mathcal{U}_{ri} \) is invertible.

To apply quantum gate to the qubit, i.e. to apply relation-specific transformation \( r \) to the head entity \( h \), we perform element-wise transformation via matrix multiplication to compute the transformed entity representation \( h_r \):
\[
h_{ri} = g_{ri}(h_i) = \mathcal{U}_{ri} h_i = \left( \begin{array}{cc} \mathbf{r}_{ai} \mathbf{h}_{ai} - e^{i\psi} \mathbf{r}_{bi} \mathbf{h}_{bi} \\ \mathbf{r}_{bi} \mathbf{h}_{ai} + e^{i\psi} \mathbf{r}_{ai} \mathbf{h}_{bi} \end{array} \right),
\]
\( i = 1, 2, \cdots, d \)

which implies \( h_r = [h_{r1}, h_{r2}, \cdots, h_{rd}] \).

4.1.3 Score Function
In our method, we do not need to exactly measure the states. Instead, we separate the states by kernel methods.

The score of a triple in KG is the similarity \( \langle h_r, t \rangle \) between the relation-specific transformed head \( h_r \) and tail \( t \). The model aims to minimize the distance between \( h_r \) and tail \( t \), i.e. their similarity \( \langle h_r, t \rangle \) is maximized for positive triples. Otherwise, it is conversely minimized for sampled negative triples.

There are various ways to define the similarity \( \langle h_r, t \rangle \). In this paper, we choose the following definitions for experiments.

**Trace Distance.**
The trace distance measures the distinguishability between two states. Two states are more similar if their trace distance is smaller. We define the similarity as the negative of the trace distance as
\[
f(h, r, t) = -\frac{1}{2} tr(\sqrt{(\rho_{h_r} - \rho_t)(\rho_{h_r} - \rho_t)^\dagger})
\]
where \( \rho_{h_r}, \rho_t \) are the density matrices of states \( |h_r\rangle \) and \( |t\rangle \) respectively, \( tr(\rho) \) is the trace of density matrix \( \rho \), \( \rho^\dagger \) is the conjugate transpose of \( \rho \).

**Hilbert-Schmidt Distance.**
Hilbert-Schmidt distance between two states is known as \( l_2 \) distance, while the \( l_1 \) distance is trace distance. Similarly, we define the similarity as the negative of the Hilbert-Schmidt distance as
\[
f(h, r, t) = -tr(\rho_{h_r} - \rho_t)(\rho_{h_r} - \rho_t)^\dagger)
\]

We also explore more definitions that may contribute to the training procedure. Element-wise \( l_1 \) distance and element-wise inner product are two measurements that follows previous classic KGEs.

**Element-wise \( l_1 \) Distance.**
\[
f(h, r, t) = -\|h_r - t\|_1
\]
\( = -\sum_{i=1}^{d} \|h_{ri} - t_i\|_1 \)

where \( \|x\|_1 \) is the \( l_1 \) norm of the two-dimensional complex vector \( x \in \mathbb{C}^{2d} \).

**Element-wise Inner Product.**
\[
f(h, r, t) = Re(\langle h_r, t \rangle)
\]

where \( Re(x) \) is the real part of the two-dimensional complex vector \( x \in \mathbb{C}^{2d} \). \( \langle h_r, t \rangle \) is element-wise inner product.

4.1.4 Loss Function
In order to optimize the model, we formulate the link prediction task as a classification problem. Following (Sun et al., 2019), the model minimizes the
following loss:

\[
\text{Loss} = - \log(p(h, r, t)) - \sum_{i=1}^{K} p(h_i, r_i, t_i) \log \sigma(f(h_i, r_i, t_i) - \gamma)
\]

where \(\gamma\) is a fixed margin, \(K\) is the number of negative examples, \((h_i, r_i, t_i)\) is the \(i\)th negative triple, \(\sigma\) is the sigmoid function. Besides, \(p(h_i, r_i, t_i)\) is the distribution of sampling negative samples and it depends on negative sampling strategies such as uniform sampling, bernoulli sampling and adversarial sampling (Sun et al., 2019).

### 4.1.5 Initialization

For parameter initialization, we adopt a particular initialization algorithm to preserve quantum advantages and speed up model efficiency and convergence (Glorot and Bengio, 2010). The initialization of entities follows the rule:

\[
\begin{align*}
 a_{\text{real}} &= \cos(\theta) \\
 a_{\text{img}} &= \sin(\theta) \cos(\phi) \\
 b_{\text{real}} &= \sin(\theta) \sin(\phi) \cos(\varphi) \\
 b_{\text{img}} &= \sin(\theta) \sin(\phi) \sin(\varphi)
\end{align*}
\]

where \(a_{\text{real}}, a_{\text{img}}, b_{\text{real}}, b_{\text{img}}\) denote the scalar and imaginary coefficients of \(a\) and \(b\), respectively. \(\theta, \phi, \varphi\) are randomly generated from the interval \([-\pi, \pi]\). The initialization of relations follows an extended rule. The coefficients of \(a\) and \(b\) are initialized by the same rule as above, while the angle \(\psi\) is randomly generated from the interval \([-\pi, \pi]\). This initialization method is optional.

### 4.2 Theoretical Analysis

The Proposition 1 below illustrates the connection with classic KGE methods.

**Proposition 1.** qubit representation is equal to unit quaternion representation. In this way, special quantum gates are rotations in the quaternion space.

For each qubit representation, there are four free variables normalized to 1. There exists a natural one-to-one mapping \(\phi\):

\[
\phi : \mathbb{C}^2 \rightarrow \mathbb{H}^d
\]

\[
(a + bi) |0\rangle + (c + di) |1\rangle \rightarrow a + bi + c + di
\]

\(a^2 + b^2 + c^2 + d^2 = 1\)

That map each qubit to unit quaternion. Similarly, the relation representation is also mapped to unit quaternion if we limit the angle \(\psi = 0\) in unitary matrix.

\[
\varphi : \mathbb{C}^{2 \times 2 \times d} \rightarrow \mathbb{H}^d
\]

\[
\begin{pmatrix}
 a + bi & -c + di \\
 c + di & a - bi
\end{pmatrix} \rightarrow a + bi + c + di
\]

\(a^2 + b^2 + c^2 + d^2 = 1\)

Therefore, that special quantum gates acting on qubit states is equal to the Hamilton product of two unit quaternions. With \(\psi = 0\) we generate a variant of QubitE, namely QubitE2.

However, QuatE (Zhang et al., 2019) which represents entities as quaternion and relations as rotations in the quaternion space, subsumes QubitE but does not subsume QubitE, because the determine of unitary matrix representation of quantum gates of QubitE is \(e^{i\psi}\) rather than 1. In other words, the general quantum gates of QubitE are not equal to unit quaternions.

### 4.2.1 Subsumption

We show that QubitE subsumes other models and inherits their favorable characteristics in learning various graph patterns.

**Definition 1.** A model \(M_1\) subsumes \(M_2\) when any scoring over triples of a KG measured by model \(M_2\) can also be obtained by \(M_1\) (Wang et al., 2018).

**Proposition 2.** QubitE subsumes DistMult, pRotatE, RotatE, Trans and ComplEx.

### 4.2.2 Full Expressiveness

**Definition 2** (from (Kazemi and Poole, 2018)). A model \(M\) is fully expressive if there exist assignments to the embeddings of the entities and relations, that accurately separate correct triples for any given ground truth.

**Proposition 3.** QubitE is fully expressive.

### 4.2.3 Inference of Patterns

**Definition 3.** Relation \(r_2\) (e.g. StudentOf) is the inversion of relation \(r_1\) (e.g. SupervisorOf) if

\[
\forall x, y \in \mathcal{E}, (x, r_1, y) \in \mathcal{T} \Rightarrow (y, r_2, x) \in \mathcal{T}
\]

**Proposition 4.** Let \(r_2 \in \mathcal{R}\) be the inversion of \(r_1 \in \mathcal{R}\). QubitE infers this pattern with \(U_{r_2,i} = U_{r_1,i}^{-1}\) for \(i = 1, 2, \cdots, d\) where \(d\) is relation embedding dimension.
Definition 4. A relation $r$ is symmetric (antisymmetric) if

$$\forall x, y \in \mathcal{E}, (x, r, y) \in \mathcal{T} \Rightarrow (y, r, x) \in \mathcal{T}$$

$$((x, r, y) \in \mathcal{T} \Rightarrow (y, r, x) \notin \mathcal{T})$$

Proposition 5. Let $r \in \mathcal{R}$ be symmetric (antisymmetric). QubitE infers the symmetry (antisymmetry) pattern if $\mathcal{U}_{r,i} = \mathcal{U}_{r,i}^{-1}$ holds (does not hold) for $i = 1, 2, \cdots, d$ where $d$ is relation embedding dimension.

Definition 5. Relation $r_1$ and relation $r_2$ are commutative (non-commutative) if

$$\forall x, y \in \mathcal{E}, (x, r_1 \circ r_2, y) \in \mathcal{T}$$

$$\Rightarrow (x, r_2 \circ r_1, y) \in \mathcal{T}$$

$$(\exists x, y \in \mathcal{E}, (x, r_1 \circ r_2, y) \in \mathcal{T}$$

$$\Rightarrow (x, r_2 \circ r_1, y) \notin \mathcal{T})$$

where $\circ$ is the composition operator.

Definition 6. Relation $r_3$ (e.g. UncleOf) is the composition of relation $r_1$ (e.g. FatherOf) and relation $r_2$ (e.g. BrotherOf) if

$$\forall x, y, z \in \mathcal{E}, (x, r_1, y) \in \mathcal{T} \land (y, r_2, z) \in \mathcal{T}$$

$$\Rightarrow (x, r_3, z) \in \mathcal{T}$$

Proposition 6. Let $r_1, r_2, r_3 \in \mathcal{R}$ be relations and $r_3$ be a composition of $r_1$ and $r_2$. QubitE infers composition with $\mathcal{U}_{r_3,i} = \mathcal{U}_{r_1,i} \mathcal{U}_{r_2,i}$. If $r_1$ and $r_2$ are commutative, then $\mathcal{U}_{r_3,i} = \mathcal{U}_{r_1,i} \mathcal{U}_{r_2,i}$. If $r_1$ and $r_2$ are non-commutative, then $\mathcal{U}_{r_3,i} \mathcal{U}_{r_2,i} \neq \mathcal{U}_{r_1,i} \mathcal{U}_{r_2,i}$ for $i = 1, 2, \cdots, d$ where $d$ is relation embedding dimension.

With above propositions, we have the following theorem:

Theorem 1. QubitE can model the symmetry / antisymmetry, inversion, and commutative / non-commutative composition patterns.

4.2.4 Complexity Analysis

Table 1 compares the space and time complexity of QubitE with several popular models. It can be seen that QubitE is efficient and shares similar complexity with classical KGEs such as TransE, RotatE and QuatE, etc.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Space Complexity</th>
<th>Time Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>TransE</td>
<td>$O(</td>
<td>\mathcal{E}</td>
</tr>
<tr>
<td>TransH</td>
<td>$O(</td>
<td>\mathcal{E}</td>
</tr>
<tr>
<td>TransR</td>
<td>$O(</td>
<td>\mathcal{E}</td>
</tr>
<tr>
<td>RESCAL</td>
<td>$O(</td>
<td>\mathcal{E}</td>
</tr>
<tr>
<td>DistMult</td>
<td>$O(</td>
<td>\mathcal{E}</td>
</tr>
<tr>
<td>ComplEx</td>
<td>$O(</td>
<td>\mathcal{E}</td>
</tr>
<tr>
<td>RotatE</td>
<td>$O(</td>
<td>\mathcal{E}</td>
</tr>
<tr>
<td>QuatE</td>
<td>$O(</td>
<td>\mathcal{E}</td>
</tr>
<tr>
<td>5*E</td>
<td>$O(</td>
<td>\mathcal{E}</td>
</tr>
</tbody>
</table>

Table 1: Comparison in space and time complexity.

5 Experiments

5.1 Experimental Settings

Datasets We evaluated our model on four widely used benchmark datasets namely FB15k (Bollacker et al., 2008), FB15k-237 (Toutanova and Chen, 2015), WN18 (Bordes et al., 2013) and WN18RR (Dettmers et al., 2018). Table 2 summarises the statistics of these four datasets.

We evaluated our model on four widely used benchmark datasets namely FB15k (Bollacker et al., 2008), FB15k-237 (Toutanova and Chen, 2015), WN18 (Bordes et al., 2013) and WN18RR (Dettmers et al., 2018). Table 2 summarises the statistics of these four datasets.

FB15k is a standard benchmark created from the original FreeBase KG (Bollacker et al., 2008), WN18 (Bordes et al., 2013) is a lexical database with hierarchical collection for the English language that was derived from the original WordNet dataset (Miller, 1992). According to (Dettmers et al., 2018), FB15k and WN18 suffer from the test leakage problem. The training set contains a large number of inverse test triples. To solve the problem, FB15k-237 and WN18RR are proposed as sub-version of FB15k and WN18, respectively, with inverse relations removed. The FB15k-237 and WN18RR datasets both include several relational patterns such as composition (e.g. award nomineel/\ldots/nominatedfor), symmetry (e.g. derivationally-related-form in WN18RR), and anti-symmetry (e.g. has_part in WN18RR).

Evaluation Protocol In order to speed up evaluation, we score each triple with all entities at a time. In detail, firstly, for each test triples, we replace tail entity with all entities in the KG to obtain candidate triples. Then, we compute the scores of all candidate triples and sort them by scores ascending order. Finally, we store the rank of the correct triple. Following the best practices of evaluations for em-
Table 2: Dataset Statistics. Split of datasets in terms of number of triples.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#train</th>
<th>#valid</th>
<th>#test</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB15k</td>
<td>483,142</td>
<td>50,000</td>
<td>59,071</td>
</tr>
<tr>
<td>WN18</td>
<td>141,442</td>
<td>5,000</td>
<td>5,000</td>
</tr>
<tr>
<td>FB15k-237</td>
<td>272,115</td>
<td>17,535</td>
<td>20,466</td>
</tr>
<tr>
<td>WN18RR</td>
<td>86,835</td>
<td>3,034</td>
<td>3,134</td>
</tr>
</tbody>
</table>

For all metrics, the higher, the better.

Implementation Details We implement our model with PyTorch (Paszke et al., 2017). The model is trained and tested on one GTX1080 graphic card. We use Adam as a gradient optimizer. We do not use Dropout because it may lead to normalization to 0 and destroy our normalization. See Appendix A.2 for more details.

Baselines We compare QubitE with a number of strong baselines. For Euclidean KG Embedding, we reported TransE (Bordes et al., 2013), TransR (Lin et al., 2015), RotatE (Sun et al., 2019), QuatE (Zhang et al., 2019), 5*E (Nayyeri et al., 2021) and HopfE (Bastos et al., 2021). For Non-Euclidean KG Embedding, we reported MuRP (Balazevic et al., 2019b) and ATTH (Chami et al., 2020). For Tensor Decomposition KG Embedding, we reported DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), SimplE (Kazemi and Poole, 2018), HypER (Balazevic et al., 2019a). For Neural Network KG Embedding, we reported ConvE (Dettmers et al., 2018), CoPER (Stoica et al., 2020). For Quantum KG Embedding, we reported QCE (Ma et al., 2019) and its variant FQCE (Ma et al., 2019).

5.2 Experimental Results and Analysis

We study the performance of our method on link prediction task. Table 3 shows the results on WN18RR and FB15k-237, and Table 4 summarizes the results on WN18 and FB15k. Overall, QubitE achieves extremely competitive results compared to the state-of-the-art classical models on all metrics across all datasets.

FB15k-237 and WN18RR mainly contain inference patterns of symmetry/antisymmetry and composition. For Euclidean KGEs, TransE and TransR perform the worst because they cannot infer antisymmetry or inversion patterns. RotatE and its variant pRotatE perform better for their inference ability. But QubitE subsumes RotatE and not surprisingly has better performance than RotatE. From RotatE, QuatE to HopfE, the MRR and Hits@10 steadily improve with the promotion on the complex space, quantization space, etc. For Tensor Decomposition KGEs, ComplEx and DistMult perform poorly since they cannot infer the composition pattern. For Neural Network KGEs, ConvE and CoPER utilise convolution neural network and contextual parameter generate neural network to score triples. But these two methods require too many parameters when compared to the linear model QubitE. On the whole, the improvement of our method demonstrate the high expressiveness of QubitE.

FB15k and WN18 mainly contain inference patterns of symmetry/antisymmetry and inversion. For Euclidean KGEs, TransE and TransR perform poorly on these two datasets because TransE cannot handle symmetry patterns and TransR cannot infer inversion patterns. RotatE converts the relation into the rotation in complex space, while QuatE in quaternion space. But as QuatE observes, the normalization of the relation to unit quaternion is a critical step for the embedding performance. And QubitE satisfies the normalization constraint naturally for quantum advantages, thus performing much better. All in all, QubitE preserves the quantum advantages and efficiently separates the qubit states.

As a quantum-based method, QubitE outperforms the two representative quantum-based models QCE and F-QCE significantly. Compared with QCE and F-QCE, QubitE gains 50% improvements in average across all metrics on FB15k and WN18. We believe the improvement of QubitE originate from its pattern inference ability, full-expressiveness, subsumption and the correct application of quantum mechanism on link prediction task.

6 Conclusion

In this paper, we propose a novel KG embedding model named QubitE to apply quantum mechanics for knowledge graph completion. QubitE models entities as qubit states and represents relations as quantum gates. With fine-grained initialization algorithm and scoring function, QubitE can preserve quantum advantages and separate the triples properly. With detailed theoretical analysis,
Table 3: Link prediction results on WN18RR and FB15k-237. Results are grouped from top to bottom by Euclidean KGE, Non-Euclidean KGE, Tensor Decomposition KGE, Neural Network KGE and Quantum KGE. Best results are in bold, second best results are underlined, third best results are italic. [◇]: Results are taken from (Dettmers et al., 2018). Other results are taken from their original papers.

Table 4: Link prediction results on WN18 and FB15k. Results are grouped from top to bottom by Euclidean KGE, Tensor Decomposition KGE, Neural Network KGE. Best results are in bold, second best results are underlined, third best results are italic. [◇]: Results are taken from (Dettmers et al., 2018); Other results are taken from their original papers.

QubitE owns the advantages of full expressiveness, subsumption, pattern inference ability and linear space&time complexity. Empirical experimental evaluations on four well-established datasets show that QubitE achieves an overall comparable performance, outperforming multiple recent strong baselines.

In the future, we would like to explore the following research directions: (1) we plan to model logical rules from the KG by using the learned embedding; (2) we plan to model complex logical query with more types of quantum gates.
References


George Stoica, Otilia Stretcu, Emmanouil Antonios Platanios, Tom Mitchell, and Barnabás Póczos.
A.1.1 Subsumption

Before our proof for Proposition 2, we give the proposition below:

**Proposition 7.** ∀ unit quaternion \( q = a + bi + cj + dk \), we can write:

\[
\begin{align*}
    a &= \cos(\theta) \\
    b &= \sin(\theta) \cos(\phi) \\
    c &= \sin(\theta) \sin(\phi) \cos(\varphi) \\
    d &= \sin(\theta) \sin(\phi) \sin(\varphi)
\end{align*}
\]

where \( \theta, \phi, \varphi \in [-\pi, \pi] \). Our goal is to generate

\[ \phi(q) = a' + 0i + b'j + 0k \text{ where } a', b' \in \mathbb{R}. \]

First, we can generate \( a' \) from \( a \) with

\[ a' = \frac{a}{1 - a^2}. \]

which implies \( a' \in \mathbb{R} \).

Second, we note that

\[
\frac{c}{b} = \tan(\phi) \cos(\varphi), \quad \frac{d}{b} = \tan(\phi) \sin(\varphi)
\]

\[
\frac{c^2}{b^2} + \frac{d^2}{b^2} = \tan^2(\phi)
\]

\[
\frac{c^2}{b^2} + \frac{d^2}{b^2} = b(c^2 + d^2)
\]

\[
\phi : \mathbb{H} \rightarrow \mathbb{C}
\]

\[ a + bi + cj + dk \rightarrow a' + 0i + b'j + 0k \]

\[ a' = \frac{a}{1 - a^2} \quad \text{and} \quad b' = \frac{c^2}{b} + \frac{d^2}{b} \]
quaternions, we also prove that there exists a surjection that maps to complex numbers (See Proposition 7). Let $z_e = a_e' + 0i + b_e'j + 0k$ where $e$ represents qubit states, $z_e$ is the projected quaternion format of $e$. Therefore, we obtain the following equation:

$$f(h, r, t) = Re(\langle h_r, t \rangle)$$

$$= Re(\langle z_{he}, z_t \rangle)$$

$$= \sum_{i=1}^{d} Re(\langle z_{hi}, z_{t_i} \rangle)$$

$$= \sum_{i=1}^{d} Re(\langle z_{hi}, z_{ri}, z_{ti} \rangle)$$

$$= f_{ComplEx}(h, r, t)$$

which shows that QubitE subsumes ComplEx. By removing the imaginary parts of $z_e$, the scoring function becomes $f(h, r, t) = \sum_{i=1}^{d} (Re(z_{hi}), Re(z_{ri}), Re(z_{ti}))$, degrading to DistMult in this case. On the other hand, we also have the following equation:

$$f(h, r, t) = -\|h_r - t\|$$

$$= -\|z_{he} - z_t\|$$

$$= -\|z_h \circ z_r - z_t\|$$

$$= f_{RotatE}(h, r, t)$$

which shows that QubitE subsumes RotatE. From (Sun et al., 2019) we know RotatE subsumes pRotatE and TransE. So QubitE also subsumes pRotatE and TransE. \qed

### A.1.2 Full Expressiveness

Here we prove Proposition 3, that QubitE is fully expressive.

**Proof.** The proof contains two steps. First, we show that QubitE is expressive. Second, we show that the expressiveness is full.

In formulation, first, we show that QubitE can express any ranking tensor $A \in \mathbb{R}^{n_e \times n_e \times n_r}$ where $n_e$ is the number of entities and $n_r$ is number of relations in KG. The $ijk$-th element of $A$, denoted $\alpha_{ijk}$, corresponds to the triple $(h_i, r_k, t_j)$. The ranking tensor gives lower rank to the triple $(h_i, r_k, t_j)$ than to $(h'_i, r'_k, t'_j)$ if the model scores the triple $(h_i, r_k, t_j)$ higher than $(h'_i, r'_k, t'_j)$. Second, for any boolean tensor $B \in \{0, 1\}^{n_e \times n_e \times n_r}$, QubitE obtains a ranking tensor which is consistent with $B$. That is, for $\beta_{ijk} = 1$ where the triple $(h_i, r_k, t_j)$ is positive and $\beta_{ijk} = 0$ where the triple $(h'_i, r'_k, t'_j)$ is negative, we have $\alpha_{ijk} > \alpha_{ij'k'}$ to correctly separate the triples.

For the first step, Wang et al. (2018) proved that the ComplEx model can obtain score tensor $A^{n_e \times n_e \times n_r}$ that fulfills the ranking rules. The model gives score $\mu_{ijk} = f(h_i, r_k, t_j)$ for triple $(h_i, r_k, t_j)$, such that $\mu_{ijk} < \mu_{ij'k'}$ holds for the definition of ranking tensor $A$. In the subsumption 2 we proved that QubitE subsumes ComplEx. Therefore, there is a vector assignment to embeddings of entities and relations such that QubitE obtains a ranking tensor.

For the second step, Wang et al. (2018) show that for a given boolean matrix $B$, there exists a ranking matrix consistent with $B$. Therefore, it is also true for QubitE to obtain a ranking matrix consistent with $B$.

With the first and the second step, we conclude that there exists an assignment to entity and relation embeddings such that for any ground truth, QubitE can separate the triples correctly. This means QubitE is fully expressive. \qed

### A.1.3 Inference of Patterns

#### Symmetry/Antisymmetry

**Definition 7.** A relation $r$ is symmetric (antisymmetric) if

$$\forall x, y \in \mathcal{E}, (x, r, y) \in \mathcal{T} \Rightarrow (y, r, x) \in \mathcal{T}$$

$$((x, r, y) \in \mathcal{T} \Rightarrow (y, r, x) \notin \mathcal{T})$$

**Proposition 8.** Let $r \in \mathcal{R}$ be symmetric (antisymmetric). QubitE infers the symmetry (antisymmetry) pattern if $\mathcal{U}_{r,i} = \mathcal{U}_{r,i}^{-1}$ holds (does not hold) for $i = 1, 2, \cdots, d$ where $d$ is relation embedding dimension.

**Proof.** Firstly, we consider the situation that relation $r$ is symmetric.

According to Definition 7, a model infers the symmetry pattern when for all given entities $x, y$, if $(x, r, y)$ is represented as positive, then $(y, r, x)$ is also represented as positive. That is

$$g_{r,i}(x_i) = y_i$$

then $g_{r,i}(y_i) = x_i$. From Equation 24, we have $y_i = g_{r,i}(x_i) = \mathcal{U}_{r,i}^{-1}x_i$. Since $g_{r,i}$ is the quantum gate whose matrix representation $\mathcal{U}_{r,i}$ is unitary and invertible, we can make the assumption $\mathcal{U}_{r,i} = \mathcal{U}_{r,i}^{-1}$ following Proposition 8. Then we have

$$y_i = g_{r,i}^{-1}(x_i)$$
which equals to $x_i = g_{r,i}(y_i)$. This means that the triple $(y, r, x)$ must be positive, i.e. inferred as positive.

Secondly, if relation $r$ is antisymmetric, we just make the assumption $\Sigma_{r,i} \neq \Sigma_{r,i}^{-1}$ to get $x_i \neq g_{r,i}(y_i)$, which means that the triple $(y, r, x)$ is inferred as negative.

\[\text{Inversion}\]

**Definition 8.** Relation $r_2$ (e.g. StudentOf) is the inversion of relation $r_1$ (e.g. SupervisorOf) if

\[
\forall x, y \in \mathcal{E}, (x, r_1, y) \in \mathcal{T} \Rightarrow (y, r_2, x) \in \mathcal{T}
\]

**Proposition 9.** Let $r_2 \in \mathcal{R}$ be the inversion of $r_1 \in \mathcal{R}$. QubitE infers this pattern with $\Sigma_{r_2,i} = \Sigma_{r_1,i}^{-1}$ for $i = 1, 2, \ldots, d$ where $d$ is relation embedding dimension.

\[\text{Proof.}\] According to Definition 8, a model infers the inversion pattern when for all given entities $x, y$, if $(x, r_1, y)$ is represented as positive, then $(y, r_2, x)$ is also represented as positive. That is

\[
g_{r_1,i}(x_i) = y_i
\]

then $g_{r_2,i}(y_i) = x_i$. From Equation 26, we have $y_i = g_{r_1,i}(x_i) = \Sigma_{r_1,i}^{-1}x_i$. Since $r_1$ is the quantum gate whose matrix representation $\Sigma_{r_1,i}$ is unitary and invertible, we can make the assumption $\Sigma_{r_2,i} = \Sigma_{r_1,i}^{-1}$ following Proposition 9. Then we have

\[
y_i = g_{r_2,i}^{-1}(x_i)
\]

which equals to $x_i = g_{r_2,i}(y_i)$. This means that the triple $(y, r_2, x)$ must be positive, i.e. inferred as positive.

\[\text{Commutative/Non-commutative Composition}\]

**Definition 9.** Relation $r_1$ and relation $r_2$ are commutative (non-commutative) if

\[
\forall x, y \in \mathcal{E}, (x, r_1 \circ r_2, y) \in \mathcal{T} \Rightarrow (x, r_2 \circ r_1, y) \in \mathcal{T}
\]

\[
\exists x, y \in \mathcal{E}, (x, r_1 \circ r_2, y) \notin \mathcal{T} \Rightarrow (x, r_2 \circ r_1, y) \in \mathcal{T}
\]

where $\circ$ is the composition operator.

**Definition 10.** Relation $r_3$ (e.g. UncleOf) is the composition of relation $r_1$ (e.g. FatherOf) and relation $r_2$ (e.g. BrotherOf) if

\[
\forall x, y, z \in \mathcal{E}, (x, r_1, y) \in \mathcal{T} \land (y, r_2, z) \in \mathcal{T} \Rightarrow (x, r_3, z) \in \mathcal{T}
\]

**Proposition 10.** Let $r_1, r_2, r_3 \in \mathcal{R}$ be relations and $r_3$ be a composition of $r_1$ and $r_2$. QubitE infers composition with $\Sigma_{r_3,i} = \Sigma_{r_1,i} \Sigma_{r_2,i}$. If $r_1$ and $r_2$ are commutative, then $\Sigma_{r_3,i} = \Sigma_{r_1,i}\Sigma_{r_2,i}$. If $r_1$ and $r_2$ are non-commutative, then $\Sigma_{r_3,i} \neq \Sigma_{r_1,i}\Sigma_{r_2,i}$ for $i = 1, 2, \ldots, d$ where $d$ is relation embedding dimension.

\[\text{Proof.}\] According to Definition 6, a model infers a composition pattern when for all given entities $x, y, z$, if the score of the model represents triples $(x, r_1, y)$ and $(y, r_2, z)$ as positive, it also represents $(x, r_3, z)$ as positive. In other words, when given

\[
g_{r_1,i}(x_i) = y_i
\]

\[
g_{r_2,i}(y_i) = z_i
\]

then it holds $g_{r_3,i}(x_i) = z_i$ for $i = 1, 2, \ldots, d$ where

\[
g_{r,j}(h_i) = \Sigma_{r,j,i}h_i,
\]

\[
j = 1, 2, 3; \quad i = 1, 2, \ldots, d
\]

From Equation 28, we insert $y_i = g_{r_1,i}(x_i)$ into $g_{r_2,i}(y_i) = z_i$, which gives $g_{r_2,i}(g_{r_1,i}(x_i)) = z_i$. Therefore, we have

\[
g_{r_2,i} \circ g_{r_1,i}(x_i) = \Sigma_{r_2,i}\Sigma_{r_1,i}x_i = z_i.
\]

Considering the Proposition 6 and assuming $\Sigma_{r_2,i}\Sigma_{r_1,i} = \Sigma_{r_3,i}$, we have $g_{r_2,i} \circ g_{r_1,i}(x_i) = g_{r_3,i}(x_i) = z_i$. This means that the triple $(x, r_3, z)$ must be positive, i.e. inferred to be positive. If $r_1$ and $r_2$ are commutative, then $\Sigma_{r_2,i}\Sigma_{r_1,i} = \Sigma_{r_1,i}\Sigma_{r_2,i}$. If $r_1$ and $r_2$ are non-commutative, then $\Sigma_{r_2,i}\Sigma_{r_1,i} \neq \Sigma_{r_1,i}\Sigma_{r_2,i}$. \[\square\]

\[\text{A.2 Implementation Details}\]

We implement our model with PyTorch (Paszke et al., 2017). The model is trained and tested on one GTX1080 graphic card. We use Adam as a gradient optimizer. We do not use Dropout because it may lead normalization to 0 and destroy our normalization. We use grid search to obtain the best hyperparameters according to MRR on the validation set. The hyperparameters are selected as follows: embedding dimension $n \in \{100, 200, 500, 1000\}$, fixed margin $\gamma \in \{3, 6, 9, 12, 24\}$, self-adversarial sampling temperature $\alpha \in \{0.5, 1.0\}$, batch size $B \in \{256, 512, 1024\}$.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>lr</th>
<th>dr</th>
<th>dc</th>
<th>p</th>
<th>neg</th>
<th>strategy</th>
</tr>
</thead>
<tbody>
<tr>
<td>FB15k</td>
<td>0.00005</td>
<td>0.99</td>
<td>500</td>
<td>500</td>
<td>24</td>
<td>256</td>
</tr>
<tr>
<td>FB15k-237</td>
<td>0.0005</td>
<td>0.995</td>
<td>500</td>
<td>500</td>
<td>12</td>
<td>256</td>
</tr>
<tr>
<td>WN18</td>
<td>0.0001</td>
<td>0.995</td>
<td>500</td>
<td>500</td>
<td>12</td>
<td>256</td>
</tr>
<tr>
<td>WN18RR</td>
<td>0.00005</td>
<td>1.0</td>
<td>500</td>
<td>500</td>
<td>6</td>
<td>256</td>
</tr>
</tbody>
</table>

Table 5: Hyper-parameter values for QubitE across all datasets.

A.3 Limitation

On the one hand, one entity is only represented by one qubit. There exists multi qubits system, that represents entities as multi qubits and brings more favorable features, though the theoretical analysis becomes difficult. On the other hand, the convergence is really slow because of the slow sampling procedure.