QubitE: Qubit Embedding for Knowledge Graph Completion

Anonymous ACL submission

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. But current quantum-based KGEs either lose quantum advantages during optimizing, or require a large number of parameters to store quantum states, thus leading to overfitting and low performance. Besides, they ignore theoretical analysis which are essential for understanding the model performance. To address performance issue and bridge theory gap, we propose QubitE which is lightweight and suitable for link prediction task. In addition, our model preserves quantum advantages which enable quantum logical computing based on semantics. Furthermore, we prove 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 several existing approaches, *e.g.* DistMult, pRotatE, RotatE, TransE and ComplEx; (4) QubitE owns linear space complexity and linear time complexity. Experiments 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.

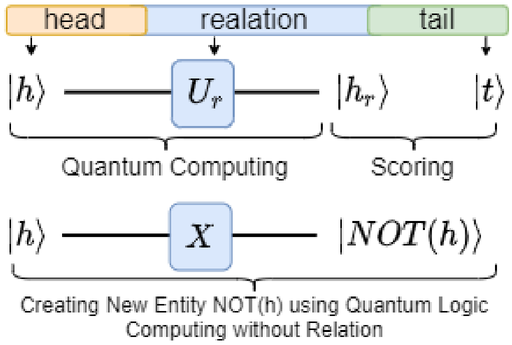

Figure 1: Visualization of the QubitE architecture. Upper explains that our socring function is based on relation-specific quantum gates acting on entity-specific quantum states. Lower illustrates a new feature to operate entity embedding regardless of relation.

Quantum-based KGEs are the application of quantum mechanics on knowledge representation learning field, but current research is still in its initial stage. With parametric quantum circuits, Ma et al. (2019) proposes the most classical quantum-based KGEs, including two types of variational quantum circuits KGEs.

The first type, *i.e.* **QCE**, considers latent features for entities as coefficients of quantum states, while predicates are characterized by parametric gates acting on the quantum states. The score of a triple depends on measurements on quantum states. However, measurements lead to information loss. The quantum advantages, *e.g.* normalization constraint of quantum states and quantum gates according to the probabilistic interpretation of quantum mechanics, disappear when optimizing the model.

The second type, *i.e.* **F-QCE**, generates embeddings of entities from parameterized quantum gates acting on the pure quantum states. The quantum embeddings can be trained efficiently meanwhile preserving the quantum advantages. However, it has to face the situation of parameter explosion, because it is expensive to prepare multi qubits for inference.

Additionally, both types are of low performance on knowledge graph completion (KGC) task. Therefore, we would like to propose a method that (1) preserves the quantum advantages, (2) is lightweight and (3) achieves high performance.

The reasons why we need the quantum advantages are listed below. Firstly, with the quantum advantages, we can operate on the semantics of entities through predefined quantum gates without the involvement of relations. This means that the model can perceive deeper, relation-independent, entity-specific semantic information. Secondly, it enables **reprogramming based on semantics**, creating new entities from the negation, intersection, union, etc. This is new feature when compared to previous classical KGEs. Lastly, quantum advantages simplify the model and allow to better study it theoretically. All in all, it is valuable for KGE community to explore new approchs with physical explanation.

In this paper, we propose a new quantum-based KGE for knowledge graph completion, namly QubitE. The entity embedding vectors are viewed as coefficients of quantum state, while preserving quantum advantages through activation function. The relation are modeled as parametric quantum gates acting on the quantum states. It is lightweight and suitable for link prediction task. Extensive experiments demonstrate the efficacy of our model.

In addition, we theoretically analysis our model, 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. Furthermore, we also prove that QubitE allows learning composition, inverse and symmetric relation patterns. Besides, QubitE owns linear space complexity and linear time complexity.

We summarize our contributions as follows:

- **KGE**: We propose QubitE, a new *linear* quantum-based KGE model for link prediction on knowledge graphs, that is lightweight, simple and expressive.
- **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

Classical KGEs. are divided into the following categories. **Euclidean geometric KGEs** includes 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), etc. **Non-Euclidean geometric KGEs** includes MuRP (Balazevic et al., 2019b) and ATTH (Chami et al., 2020). **Tensor decomposition KGEs** includes DistMult (Yang et al., 2015), ComplEx (Trouillon et al., 2016), SimplE (Kazemi and Poole, 2018), HypER (Balazevic et al., 2019a), TuckER (Balazevic et al., 2019c), etc. **Neural network KGEs** includes ConvE (Dettmers et al., 2018), CoPER (Stoica et al., 2020), etc. We provide short introduction of these methods in Appendix A.4.

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 data point x is translated into a set of gate parameters in a quantum circuit ψ, creating a quantum state $|x\rangle$ such that $\psi : x \to |x\rangle$. However, our method is quite different. Firstly, we distinguish quantum states by distance function on the embedding vector 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 $\mathcal{KG} = (\mathcal{E}, \mathcal{R}, \mathcal{T})$ where $\mathcal{E}$ is the set of nodes (entities) and $\mathcal{R}$ is the set of edges (relations between entities). The set $\mathcal{T} = \{(h, r, t)\} \subseteq \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ contains all triples as *(head, relation, tail)*, *e.g.* (*smartPhone, hypernym, iPhone*). To apply learning methods on KGs, a KGE learns vector representations of entities ($\mathcal{E}$) and relations ($\mathcal{R}$). A vector representation denoted by $(\mathbf{h}, \mathbf{r}, \mathbf{t})$ is learned by the model per triple (h, r, t), where $\mathbf{h}, \mathbf{t} \in \mathbb{V}^{d_e}$, $\mathbf{r} \in \mathbb{V}^{d_r}$ ($\mathbb{V}^d$

[1] https://anonymous.4open.science/r/QubitE-ACL2022/

is a d-dimensional vector space). TransE (Bordes et al., 2013) considers $\mathbb{V} = \mathbb{R}$ while ComplEx (Trouillon et al., 2016) and RotatE use $\mathbb{V} = \mathbb{C}$ (complex space) and QuatE (Zhang et al., 2019) considers $\mathbb{V} = \mathbb{H}$ (quaternion space). In this paper, we choose two-dimensional Hilbert space to embed the graph i.e. $\mathbb{V} = \mathbb{C}^2$. Most KGE models are defined via a relation-specific transformation function $g_r : \mathbb{V}^{d_e} \to \mathbb{V}^{d_e}$ which maps head entities to tail entities, *i.e.* $g_r(\mathbf{h}) = \mathbf{t}$. On top of such a transformation function, the score function $f : \mathbb{V}^{d_e} \times \mathbb{V}^{d_r} \times \mathbb{V}^{d_e} \to \mathbb{R}$ is defined to measure the plausibility for triples: $f(\mathbf{h}, \mathbf{r}, \mathbf{t}) = p(g_r(\mathbf{h}), \mathbf{t})$. Generally, the formulation of any score function can be either $p(g_r(\mathbf{h}), \mathbf{t}) = -\|g_r(\mathbf{h}) - \mathbf{t}\|$ or $p(g_r(\mathbf{h}), \mathbf{t}) = \langle g_r(\mathbf{h}), \mathbf{t}\rangle$.

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 superposition 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

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

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

$$|\psi\rangle = \mathbf{a}\,|0\rangle + \mathbf{b}\,|1\rangle \tag{2}$$

where $\mathbf{a}, \mathbf{b} \in \mathbb{C}$, $|\mathbf{a}|^2 + |\mathbf{b}|^2 = 1$. The complex numbers $\mathbf{a}$ and $\mathbf{b}$ 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 $|\mathbf{a}|^2(|\mathbf{b}|^2)$ and state $|0\rangle(|1\rangle)$ immediately. The density matrix ρ of state $|\psi\rangle$ is given by:

$$\rho = |\psi\rangle\,\langle\psi| \tag{3}$$

Quantum Gates. Essentially, quantum gates 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 2×2 unitary matrix is

$$U = \begin{pmatrix}\mathbf{a} & -e^{i\psi}\mathbf{b}^* \\ \mathbf{b} & e^{i\psi}\mathbf{a}^*\end{pmatrix} \tag{4}$$

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

4 Method

4.1 Model Formulation

Given a triple (h, r, t), the head and tail entities $h, t \in \mathcal{E}$ are embedded into a d dimensional Hilbert space *i.e.* $\mathbf{h}, \mathbf{t} \in \mathbb{C}^{2d}$ where each element is a 2-dimensional complex value vector. A relation $r \in \mathcal{R}$ is embedded into a d dimensional vector $\mathbf{r}$ where each element is a 2×2 complex value unitary matrix. $\mathbf{r}$ contains two complex vectors $\mathbf{r_a}$ and $\mathbf{r_b} \in \mathbb{C}^d$. With $\mathbf{r}_{ai}, \mathbf{r}_{bi}, \mathbf{h}_{ai}, \mathbf{h}_{bi}, \mathbf{t}_{ai}, \mathbf{t}_{bi}$, we refer to the ith element of $\mathbf{r}_a, \mathbf{r}_b, \mathbf{h}_a, \mathbf{h}_b, \mathbf{t}_a, \mathbf{t}_b$ 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 ith element of entity embedding vector $\mathbf{h}$ is given by

$$\mathbf{h}_i = \mathbf{h}_{ai}\,|0\rangle + \mathbf{h}_{bi}\,|1\rangle = \begin{pmatrix}\mathbf{h}_{ai}\\ \mathbf{h}_{bi}\end{pmatrix}, \quad i = 1, 2, \cdots, d \tag{5}$$

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

$$\rho_{\mathbf{h}_i} = |\mathbf{h}_i\rangle\,\langle\mathbf{h}_i| = \begin{pmatrix}|\mathbf{h}_{ai}|^2 & \mathbf{h}_{ai}\mathbf{h}_{bi}^* \\ \mathbf{h}_{bi}\mathbf{h}_{ai}^* & |\mathbf{h}_{bi}|^2\end{pmatrix}. \tag{6}$$

4.1.2 Relation-specific Quantum Gate

We use reletion-specific transformation to map the head entity $\mathbf{h}$ from a source to a target Hilbert space. Since quantum gates are unitary, we write the parameterized unitary matrix of ith element of relation embedding vector $\mathbf{r}$ as

$$\mathbf{r}_i = \mathfrak{U}_{ri} = \begin{pmatrix}\mathbf{r}_{ai} & -e^{i\psi}\mathbf{r}_{bi}^* \\ \mathbf{r}_{bi} & e^{i\psi}\mathbf{r}_{ai}^*\end{pmatrix}, \quad i = 1, 2, \cdots, d \tag{7}$$

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 $\mathbf{r} =$

Table 1: Scoring functions of state-of-the-art KGEs. "$\star$" denotes the circular correlation operation; "$\circ$" denotes Hadmard (or element-wise) product. "$\otimes$" denotes Hamilton product.

Model	Scoring Function	Parameters
TransE	$\| (\mathbf{h}+\mathbf{r})-\mathbf{t} \|$	$\mathbf{h},\mathbf{r},\mathbf{t} \in \mathbb{R}^d$
HolE	$\langle \mathbf{r}, \mathbf{h} \star \mathbf{t} \rangle$	$\mathbf{h},\mathbf{r},\mathbf{t} \in \mathbb{R}^d$
DistMult	$\langle \mathbf{r}, \mathbf{h}, \mathbf{t} \rangle$	$\mathbf{h},\mathbf{r},\mathbf{t} \in \mathbb{R}^d$
ComplEx	$\mathrm{Re}(\langle \mathbf{r}, \mathbf{h}, \bar{\mathbf{t}} \rangle)$	$\mathbf{h},\mathbf{r},\mathbf{t} \in \mathbb{C}^d$
RotatE	$\| \mathbf{h} \circ \mathbf{r} - \mathbf{t} \|$	$\mathbf{h},\mathbf{r},\mathbf{t} \in \mathbb{C}^d$
QuatE	$\mathbf{h} \otimes \mathbf{r} \cdot \mathbf{t}$	$\mathbf{h},\mathbf{r},\mathbf{t} \in \mathbb{H}^d$
5*E	$\| \frac{\mathbf{r}_1\mathbf{h}+\mathbf{r}_2}{\mathbf{r}_3\mathbf{h}+\mathbf{r}_4} - \mathbf{t} \|$	$\mathbf{h},\mathbf{t},\mathbf{r}_{1\ldots4} \in \mathbb{C}^d$
QubitE	$\langle \mathfrak{U}_r \mathbf{h}, \mathbf{t} \rangle$	$\mathbf{h},\mathbf{t} \in \mathbb{C}^{2d}$ $\mathfrak{U}_r \in \mathbb{C}^{2\times2\times d}$

$[\mathbf{r}_1, \mathbf{r}_2, \cdots, \mathbf{r}_d]$. This implies $det(\mathfrak{U}_{ri}) = e^{i\psi} \neq 0$ *i.e.* $\mathfrak{U}_{ri}$ is invertible.

To apply quantum gate to the qubit, *i.e.* to apply relation-specific transformation $\mathbf{r}$ to the head entity $\mathbf{h}$, we perform element-wise transformation via matrix multiplication to compute the transformed entity representation $\mathbf{h}_r$:

$$\mathbf{h}_{ri} = g_{ri}(\mathbf{h}_i) = \mathfrak{U}_{ri}\mathbf{h}_i = \begin{pmatrix} \mathbf{r}_{ai}\mathbf{h}_{ai} - e^{i\psi}\mathbf{r}_{bi}^*\mathbf{h}_{bi} \\ \mathbf{r}_{bi}\mathbf{h}_{ai} + e^{i\psi}\mathbf{r}_{ai}^*\mathbf{h}_{bi} \end{pmatrix}, \quad i = 1, 2, \cdots, d \tag{8}$$

which implies $\mathbf{h}_r = [\mathbf{h}_{r1}, \mathbf{h}_{r2}, \cdots, \mathbf{h}_{rd}]$.

4.1.3 Scoring Function

Table 1 summarizes scoring functions and parameters of several popular KGEs. TransE, HolE, and DistMult use Euclidean embeddings, while ComplEx and RotatE operate in the complex space. QuatE operates in the quaternion space. In contrast, our model uses quantum states and quantum gates which are in hyper-complex space.

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 $\langle \mathbf{h}_r, \mathbf{t} \rangle$ between the relation-specific transformed head $\mathbf{h}_r$ and tail $\mathbf{t}$. The model aims to minimize the distance between $\mathbf{h}_r$ and tail $\mathbf{t}$, *i.e.* their similarity ($\langle \mathbf{h}_r, \mathbf{t} \rangle$) is maximized for positive triples. Otherwise, it is conversely minimized for sampled negative triples.

There are various ways to define the similarity $\langle \mathbf{h}_r, \mathbf{t} \rangle$. 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

$$f(h,r,t) = -\frac{1}{2} tr(\sqrt{(\rho_{h_r} - \rho_t)^\dagger (\rho_{h_r} - \rho_t)}) \tag{9}$$

where ρ_{h_r}, ρ_t are the density matrices of states $|h_r\rangle$ and $|t\rangle$ respectively, $tr(\rho)$ is the trace of density matrix ρ, $\rho^\dagger$ is the conjugate transpose of ρ.

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

$$f(h,r,t) = -tr((\rho_{h_r} - \rho_t)^\dagger (\rho_{h_r} - \rho_t)) \tag{10}$$

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.

$$\begin{aligned} f(h,r,t) &= -\|\mathbf{h}_r - \mathbf{t}\|_1 \\ &= -\sum_{i=1}^{d} \|\mathbf{h}_{ri} - \mathbf{t}_i\|_1 \end{aligned} \tag{11}$$

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

Element-wise Inner Product.

$$f(h,r,t) = Re(\langle \mathbf{h}_r, \bar{\mathbf{t}} \rangle) \tag{12}$$

where $Re(\mathbf{x})$ is the real part of the two-dimensional complex vector $\mathbf{x} \in \mathbb{C}^{2d}$. $\langle \mathbf{h}_r, \bar{\mathbf{t}} \rangle$ 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:

$$\begin{aligned} Loss = &-\log(\gamma - f(h,r,t)) \\ &- \sum_{i=1}^{K} p(h_i, r_i, t_i) \log \sigma(f(h_i, r_i, t_i) - \gamma) \end{aligned} \tag{13}$$

where γ is a fixed margin, K is the number of negative examples, (h_i, r_i, t_i) is the ith negative triple, σ 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:

$$\begin{aligned}\mathbf{a}_{\text{real}} &= \cos(\theta)\\ \mathbf{a}_{\text{img}} &= \sin(\theta)\cos(\phi)\\ \mathbf{b}_{\text{real}} &= \sin(\theta)\sin(\phi)\cos(\varphi)\\ \mathbf{b}_{\text{img}} &= \sin(\theta)\sin(\phi)\sin(\varphi)\end{aligned} \tag{14}$$

where $\mathbf{a}_{\text{real}}, \mathbf{a}_{\text{img}}, \mathbf{b}_{\text{real}}, \mathbf{b}_{\text{img}}$ denote the scalar and imaginary coefficients of $\mathbf{a}$ and $\mathbf{b}$, respectively. θ, ϕ, φ are randomly generated from the interval $[-\pi, \pi]$. The initialization of relations follows an extended rule. The coefficients of $\mathbf{a}$ and $\mathbf{b}$ are initialized by the same rule as above, while the angle ψ 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 ϕ:

$$\begin{gathered}\phi : \mathbb{C}^2 \to \mathbb{H}\\ (a + b\mathbf{i})\,|0\rangle + (c + d\mathbf{i})\,|1\rangle \to a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}\\ a^2 + b^2 + c^2 + d^2 = 1\end{gathered} \tag{15}$$

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.

$$\begin{gathered}\varphi : \mathbb{C}^{2\times 2} \to \mathbb{H}\\ \begin{pmatrix} a + b\mathbf{i} & -c + d\mathbf{i} \\ c + d\mathbf{i} & a - b\mathbf{i}\end{pmatrix} \to a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}\\ a^2 + b^2 + c^2 + d^2 = 1\end{gathered} \tag{16}$$

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 QubitE_2.

However, QuatE (Zhang et al., 2019) which represents entities as quaternion and relations as rotations in the quaternion space, subsumes QubitE_2 but does not subsume QubitE, because $\psi \neq 0$. The determine of unitary matrix in 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

In this section, We show that QubitE subsumes other models and inherits their favorable characteristics in learning various graph patterns. We also provide full proofs in the AppendixA.1.

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, TransE 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

Proposition 4. *Let $r_2 \in \mathcal{R}$ be the inversion of $r_1 \in \mathcal{R}$. QubitE infers this pattern with $\mathfrak{U}_{r_2,i} = \mathfrak{U}_{r_1,i}^{-1}$ for $i = 1, 2, \cdots, d$ where d is relation embedding dimension.*

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

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 $\mathfrak{U}_{r_2,i}\mathfrak{U}_{r_1,i} = \mathfrak{U}_{r_3,i}$. If r_1 and r_2 are commutative, then $\mathfrak{U}_{r_2,i}\mathfrak{U}_{r_1,i} = \mathfrak{U}_{r_1,i}\mathfrak{U}_{r_2,i}$. If r_1 and r_2 are non-commutative, then $\mathfrak{U}_{r_2,i}\mathfrak{U}_{r_1,i} \neq \mathfrak{U}_{r_1,i}\mathfrak{U}_{r_2,i}$ for $i = 1, 2, \cdots, d$ where d is relation embedding dimension.*

With above propositions, we conclude that:

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

4.2.4 Complexity Analysis

Table 2 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.

Methods	Space Complexity	Time Complexity
TransE	$O(\vert\mathcal{E}\vert n+\vert\mathcal{R}\vert n)$	$O(n)$
TransH	$O(\vert\mathcal{E}\vert n+\vert\mathcal{R}\vert n)$	$O(n)$
TransR	$O(\vert\mathcal{E}\vert n+\vert\mathcal{R}\vert n^2)$	$O(n^2)$
RESCAL	$O(\vert\mathcal{E}\vert n+\vert\mathcal{R}\vert n^2)$	$O(n^2)$
DistMult	$O(\vert\mathcal{E}\vert n+\vert\mathcal{R}\vert n)$	$O(n)$
ComplEx	$O(\vert\mathcal{E}\vert n+\vert\mathcal{R}\vert n)$	$O(n)$
RotatE	$O(\vert\mathcal{E}\vert n+\vert\mathcal{R}\vert n)$	$O(n)$
QuatE	$O(\vert\mathcal{E}\vert n+\vert\mathcal{R}\vert n)$	$O(n)$
5*E	$O(\vert\mathcal{E}\vert n+\vert\mathcal{R}\vert n)$	$O(n)$
QubitE	$O(\vert\mathcal{E}\vert n+\vert\mathcal{R}\vert n)$	$O(n)$

Table 2: 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 3 summarizes the statistics of these four datasets. See Appendix A.2 for more details.

Dataset	#train	#valid	#test
FB15k	483,142	50,000	59,071
WN18	141,442	5,000	5,000
FB15k-237	272,115	17,535	20,466
WN18RR	86,835	3,034	3,134

Table 3: **Dataset Statistics.** Split of datasets in terms of number of triples.

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 embedding models, we consider the most-used metrics (Mean) Reciprocal Rank (MRR) and Hits@n (n = 1, 3, 10). 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. In addition, we adopt the same type constraint from QuatE (Zhang et al., 2019). See Appendix A.3 for more details about hyperparameters.

Baselines We compare QubitE with 17 strong baselines. For *Euclidean KGEs*, we report TransE, TransR, RotatE, QuatE, 5*E and HopfE. For *Non-Euclidean KGEs*, we compare to MuRP and ATTH. For *Tensor Decomposition KGEs*, we report DistMult, ComplEx, SimplE, HypER and TuckER. For *Neural Network KGEs*, we report ConvE and CoPER. For *Quantum KGEs*, we report QCE and its variant F-QCE. All these models are introduced in Appendix A.4.

5.2 Main Results

We study the performance of our method on link prediction task. Table 4 shows the results on WN18RR and FB15k-237, and Table 5 summarizes the results on WN18 and FB15k. Overall, QubitE achieves competitive results compared to the state-of-the-art classical models on all metrics across all datasets exepct WN18RR.

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. TuckER is much better because of its full expressiveness. For Neural Network KGEs, ConvE and CoPER utilize convolution neural network and contextual parameter generate

	FB15k-237				WN18RR			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
TransE (Bordes et al., 2013)	.294	.465	–	–	.226	.501	–	–
TransR (Lin et al., 2015)	–	.486	–	–	–	.503	–	–
RotatE (Sun et al., 2019)	.338	.533	.375	.241	.476	.571	.492	.428
QuatE (Zhang et al., 2019)	.348	*.550*	.382	.248	**.488**	.582	**.508**	.438
NagE (Yang et al., 2020)	.340	.530	.378	.244	*.477*	.574	*.493*	.432
5*E (Nayyeri et al., 2021)	.350	.530	.380	*.260*	.470	*.580*	.500	.410
HopfE (Bastos et al., 2021)	.343	.534	.379	.247	.472	**.586**	.500	.413
MuRP (Balazevic et al., 2019b)	.340	.520	.370	.240	.480	.570	.500	*.440*
ATTH (Chami et al., 2020)	.311	.488	.339	.223	.456	.526	.471	.419
DistMult◇ (Yang et al., 2015)	.241	.419	.263	.155	.430	.490	.440	.390
ComplEx◇ (Trouillon et al., 2016)	.247	.428	.275	.158	.440	.510	.460	.410
HypER (Balazevic et al., 2019a)	.341	.520	.376	.252	.465	.522	.477	.436
TuckER (Balazevic et al., 2019c)	*.358*	.544	*.394*	.266	.470	.526	.482	**.443**
ConvE◇ (Dettmers et al., 2018)	.325	.501	.356	.237	.430	.520	.440	.400
CoPER (Stoica et al., 2020)	.365	.504	–	**.295**	.465	.510	–	.427
QCE (Ma et al., 2019)	–	.350	.225	–	–	.323	.195	–
F-QCE (Ma et al., 2019)	–	.337	.198	–	–	.378	.274	–
QubitE (ours)	**.366**	.554	.400	.273	.467	.525	.478	.437
QubitE$_2$ (ours)	**.366**	**.555**	**.401**	.273	.471	.531	.482	.441

Table 4: Link prediction results on FB15k-237 and WN18RR. Results are grouped from top to bottom by Euclidean KGEs, Non-Euclidean KGEs, Tensor Decomposition KGEs, Neural Network KGEs and Quantum KGEs. 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$_2$ is the varient with $\psi = 0$.

	FB15k				WN18			
	MRR	Hits@10	Hits@3	Hits@1	MRR	Hits@10	Hits@3	Hits@1
TransE (Bordes et al., 2013)	.463	.749	.578	.297	.495	.943	.888	.113
TransR (Lin et al., 2015)	.198	.582	.404	.218	.427	.940	.876	.335
RotatE (Sun et al., 2019)	.797	.884	.830	*.746*	.949	.959	.952	.944
QuatE (Zhang et al., 2019)	.782	.900	.835	.711	*.950*	**.959**	.954	.945
NagE (Yang et al., 2020)	–	–	–	–	*.950*	**.960**	.953	.944
5*E (Nayyeri et al., 2021)	.730	.860	.780	.660	*.950*	**.960**	.950	**.950**
HopfE (Bastos et al., 2021)	–	–	–	–	.949	**.960**	.954	.938
DistMult◇ (Yang et al., 2015)	*.798*	*.893*	–	–	.797	.893	–	–
ComplEx (Trouillon et al., 2016)	.692	.840	.759	.599	.941	.947	.936	.936
SimplE (Kazemi and Poole, 2018)	.727	.838	.773	.660	.942	.947	.944	.939
HypER (Balazevic et al., 2019a)	.790	.885	.829	.734	.951	*.958*	**.955**	*.947*
TuckER (Balazevic et al., 2019c)	.795	.892	*.833*	.741	**.953**	*.958*	**.955**	.949
ConvE (Dettmers et al., 2018)	.657	.831	.723	.558	.943	.956	.946	.935
QubitE (ours)	.807	.894	.838	**.758**	*.950*	.957	.952	.945
QubitE$_2$ (ours)	**.818**	**.897**	**.846**	.753	*.950*	.959	*.952*	.946

Table 5: Link prediction results on FB15k and WN18. Results are grouped from top to bottom by Euclidean KGEs, Tensor Decomposition KGEs, Neural Network KGEs. 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$_2$ is the varient with $\psi = 0$.

neural network to socre 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, thus performing better. As QuatE observes, the normalization of the relation to unit quaternion is a critical step for the embedding performance. Exactly, because of quantum mechanics, QubitE satisfies the normalization constraint naturally to preserve quantum advantages, thus performing much better.

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. After all, QCE is not able to preserve quantum advantages in training, while F-QCE is faced with parameter explosion and overfitting. We believe the improvement of QubitE also originate from its pattern inference ability, full-expressiveness, subsumption and the correct application of quantum mechanism on link prediction task.

5.3 Model Analysis

Ablation Study on ψ. We constraint $\psi = 0$ to construct the variant QubitE$_2$, which is subsumed by QuatE mentioned in Section 4.2. From Table 4 and Table 5, we observe that QubitE$_2$ is slightly better than standard QubitE accross all datasets. The results demonstrate that ψ is not the core parameter that improves the performance. It also indicates that the other parameters, whcih make quantum advantages come true, are more important for high performance. By the way, there is another explanation that ψ does not affect the physical measurement of qubits, so it does not significantly affect the experimental results.

Impacts of Dimensionality. Our experiments also indicate that the selection of embedding dimension has substantial influence on both effectiveness and efficiency of QubitE. We train QubitE with embedding dimension $d \in \{100, 200, 400, 800, 1000, 1200\}$ and plot results based on the validation set, as shown in Figure 2. With the increase of d, the training time rises, while the model performance (indicated by MRR) increased slowly during $d = 100$ and $d = 400$ but fell sharply after $d = 400$. Therefore, we decide 400 as the best setting for WN18RR.

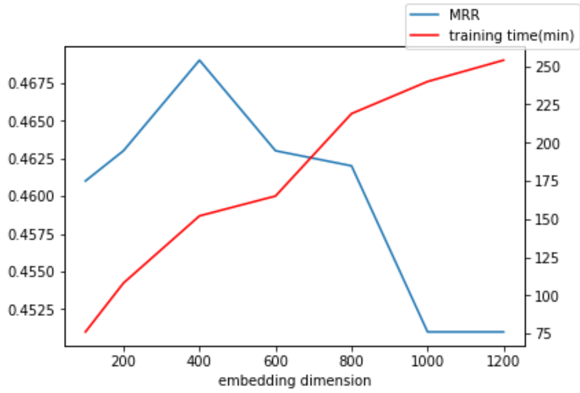

Figure 2: The convergence MRR and training time of QubitE on WN18RR.

Semantic Logic Computing. Logic computing is the favorable feature different from all previous classical KGEs. With the benification of quantum mechanics, we can perform quantum logic computing on the semantic of learned quantum embedding. For instance, given entity A, we can compute the semantic negation of entity A using NOT quantum circuit. In addition, we are able to get the semantic intersection of given entity A and B with the help of AND quantum gate. The NOT gate and AND gate are non-parametric, indicating that logic computing is relation-independent for entity quantum embedding. QubitE supports all quantum logic operators. AppendixA.6 gives the definitions of logic operator NOT for example, explains how to use it and visualizes the results.

6 Conclusion

In this paper, we propose a novel KGE 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, 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.

References

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019a. Hypernetwork knowledge graph embeddings. In *Artificial Neural Networks and Machine Learning - ICANN 2019 - 28th International Conference on Artificial Neural Networks, Munich, Germany, September 17-19, 2019, Proceedings - Workshop and Special Sessions*, volume 11731 of *Lecture Notes in Computer Science*, pages 553–565. Springer.

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019b. Multi-relational poincaré graph embeddings. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 4465–4475.

Ivana Balazevic, Carl Allen, and Timothy M. Hospedales. 2019c. Tucker: Tensor factorization for knowledge graph completion. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pages 5184–5193. Association for Computational Linguistics.

Anson Bastos, Kuldeep Singh, Abhishek Nadgeri, Saeedeh Shekarpour, Isaiah Onando Mulang, and Johannes Hoffart. 2021. Hopfe: Knowledge graph representation learning using inverse hopf fibrations. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, page 89–99, New York, NY, USA. Association for Computing Machinery.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

Antoine Bordes, Nicolas Usunier, Alberto García-Durán, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *NIPS 2013*.

Ines Chami, Adva Wolf, Da-Cheng Juan, Frederic Sala, Sujith Ravi, and Christopher Ré. 2020. Low-dimensional hyperbolic knowledge graph embeddings. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 6901–6914. Association for Computational Linguistics.

Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and Sebastian Riedel. 2018. Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 1811–1818. AAAI Press.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Seyed Mehran Kazemi and David Poole. 2018. Simple embedding for link prediction in knowledge graphs. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 4289–4300.

Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. 2015. Learning entity and relation embeddings for knowledge graph completion. In *AAAI 2015*.

Seth Lloyd, Maria Schuld, Aroosa Ijaz, Josh Izaac, and Nathan Killoran. 2020. Quantum embeddings for machine learning. *arXiv preprint arXiv:2001.03622*.

Yunpu Ma, Volker Tresp, Liming Zhao, and Yuyi Wang. 2019. Variational quantum circuit model for knowledge graph embedding. *Advanced Quantum Technologies*, 2(7-8):1800078.

George A. Miller. 1992. Wordnet: A lexical database for english. *Commun. ACM*, 38:39–41.

Mojtaba Nayyeri, Sahar Vahdati, Can Aykul, and Jens Lehmann. 2021. 5* knowledge graph embeddings with projective transformations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(10):9064–9072.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic Differentiation in PyTorch. In *NIPS-W*.

George Stoica, Otilia Stretcu, Emmanouil Antonios Platanios, Tom Mitchell, and Barnabás Póczos. 2020. Contextual parameter generation for knowledge graph link prediction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 3000–3008.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. In *International Conference on Learning Representations*.

Kristina Toutanova and Danqi Chen. 2015. Observed versus latent features for knowledge base and text inference.

Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. 2016. Complex embeddings for simple link prediction. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 2071–2080. JMLR.org.

Yanjie Wang, Rainer Gemulla, and Hui Li. 2018. On multi-relational link prediction with bilinear models. In *AAAI*.

Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. Embedding entities and relations for learning and inference in knowledge bases. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tong Yang, Long Sha, and Pengyu Hong. 2020. Nage: Non-abelian group embedding for knowledge graphs. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, page 1735–1742, New York, NY, USA. Association for Computing Machinery.

Shuai Zhang, Yi Tay, Lina Yao, and Qi Liu. 2019. Quaternion knowledge graph embeddings. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

A Appendix

A.1 Theoretical Proofs

A.1.1 Subsumption

Here we will prove Proposition 2. We will show that QubitE subsumes DistMult, pRotatE, RotatE, TransE and ComplEx and inherits their favorable characteristics in learning various graph patterns.

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

Proposition 7. *$\forall$ unit quaternion q, there exists a surjection $\phi : \mathbb{H} \to \mathbb{C}$ such that $\phi(q)$ is complex number. Moreover, $\phi(q)$ can be written in quaternion format $\phi(q) = a + 0\mathbf{i} + b\mathbf{j} + 0\mathbf{k}$, $a, b \in \mathbb{R}$, and the Hamilton product in quaternion space will also degrade to complex number multiplication.*

Proof. For any given unit quaternion $q = a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k}$, we can write:

$$\begin{aligned} a &= \cos(\theta) \\ b &= \sin(\theta)\cos(\phi) \\ c &= \sin(\theta)\sin(\phi)\cos(\varphi) \\ d &= \sin(\theta)\sin(\phi)\sin(\varphi) \end{aligned} \tag{17}$$

where $\theta, \phi, \varphi \in [-\pi, \pi]$. Our goal is to generate $\phi(q) = a' + 0\mathbf{i} + b'\mathbf{j} + 0\mathbf{k}$ where $a', b' \in \mathbb{R}$.

First, we can generate a' from a with

$$a' = \frac{a}{1 - a^2}. \tag{18}$$

which implies $a' \in \mathbb{R}$.

Second, we note that

$$\begin{aligned} \frac{c}{b} &= \tan(\phi)\cos(\varphi), \\ \frac{d}{b} &= \tan(\phi)\sin(\varphi) \\ \frac{c^2}{b^2} + \frac{d^2}{b^2} &= \tan^2(\phi) \\ \frac{c^2}{b} + \frac{d^2}{b} &= b\left(\frac{c^2}{b^2} + \frac{d^2}{b^2}\right) \\ &= \sin(\theta)\cos(\phi)\tan^2(\phi) \in \mathbb{R} \end{aligned} \tag{19}$$

Therefore, we can generate b' with b, c, d with

$$b' = \frac{c^2}{b} + \frac{d^2}{b} \tag{20}$$

which implies $b' \in \mathbb{R}$. The surjection is

$$\begin{aligned} \phi &: \mathbb{H} \to \mathbb{C} \\ a + b\mathbf{i} + c\mathbf{j} + d\mathbf{k} &\to a' + 0\mathbf{i} + b'\mathbf{j} + 0\mathbf{k} \\ a' &= \frac{a}{1 - a^2} \\ b' &= \frac{c^2}{b} + \frac{d^2}{b} \end{aligned} \tag{21}$$

and the Hamilton product in quaternion space will also degrade to complex number multiplication. $\square$

Then we can begin our proof for Proposition 2.

Proof. For any given entity h and relation r, we have proved that they can be mapped to unit quaternions naturally (See Proposition 1). For any unit quaternions, we also prove that there exists a surjection that maps to complex numbers (See Proposition 7). Let $\mathbf{z}_e = a'_e + 0\mathbf{i} + b'_e\mathbf{j} + 0\mathbf{k}$ where e represents qubit states, $\mathbf{z}_e$ is the projected quaternion format of e. Therefore, we obtain the following equation:

$$\begin{aligned} f(h, r, t) &= Re(\langle \mathbf{h}_r, \bar{\mathbf{t}} \rangle) \\ &= Re(\langle \mathbf{z}_{h_r}, \overline{\mathbf{z}_t} \rangle) \\ &= \sum_{i=1}^{d} Re(\langle \mathbf{z}_{h_{ri}}, \overline{\mathbf{z}_{ti}} \rangle) \\ &= \sum_{i=1}^{d} Re(\langle \mathbf{z}_{h_i}, \mathbf{z}_{r_i}, \overline{\mathbf{z}_{ti}} \rangle) \\ &= f_{\text{ComplEx}}(h, r, t) \end{aligned} \tag{22}$$

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

$$\begin{aligned} f(h,r,t) &= -\|\mathbf{h}_r - \mathbf{t}\| \\ &= -\|\mathbf{z}_{h_r} - \mathbf{z}_t\| \\ &= -\|\mathbf{z}_h \circ \mathbf{z}_r - \mathbf{z}_t\| \\ &= f_{\text{RotatE}}(h,r,t) \end{aligned} \tag{23}$$

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. □

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 $\mathcal{A} \in \mathbb{R}^{n_e \times n_e \times n_r}$ where n_e is the number of entities and n_r is number of relations in KG. The ikj-th element of $\mathcal{A}$, denoted α_{ikj}, 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 $\mathcal{B} \in \{0,1\}^{n_e \times n_e \times n_r}$, QubitE obtains a ranking tensor which is consistent with $\mathcal{B}$. That is, for $\beta_{ikj} = 1$ where the triple (h_i, r_k, t_j) is positive and $\beta_{i'k'j'} = 0$ where the triple (h'_i, r'_k, t'_j) is negative, we have $\alpha ikj > \alpha_{i'k'j'}$ to correctly separate the triples.

For the first step, Wang et al. (2018) proved that the ComplEx model can obtain score tensor $\mathcal{M}^{n_e \times n_e \times n_r}$ that fulfills the ranking rules. The model gives score $\mu_{ikj} = f(h_i, r_k, t_j)$ for triple (h_i, r_k, t_j), such that $\mu_{ikj} < \mu_{i'k'j'}$ holds for the definition of ranking tensor $\mathcal{A}$. 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 $\mathcal{B}$, there exists a ranking matrix consistent with $\mathcal{B}$. Therefore, it is also true for QubitE to obtain a ranking matrix consistent with $\mathcal{B}$.

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. □

A.1.3 Inference of Patterns

Symmetry/Antisymmetry

Definition 3. *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 $\mathfrak{U}_{r,i} = \mathfrak{U}_{r,i}^{-1}$ 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 3, 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

$$g_{r,i}(\mathbf{x}_i) = \mathbf{y}_i \tag{24}$$

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

$$\mathbf{y}_i = g_{r,i}^{-1}(\mathbf{x}_i) \tag{25}$$

which equals to $\mathbf{x}_i = g_{r,i}(\mathbf{y}_i)$. 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 $\mathfrak{U}_{r,i} \neq \mathfrak{U}_{r,i}^{-1}$ to get $\mathbf{x}_i \neq g_{r,i}(\mathbf{y}_i)$, which means that the triple (y,r,x) is inferred as negative. □

Inversion

Definition 4. *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 $\mathfrak{U}_{r_2,i} = \mathfrak{U}_{r_1,i}^{-1}$ for $i = 1, 2, \cdots, d$ where d is relation embedding dimension.*

Proof. According to Definition 4, 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

$$g_{r_1,i}(\mathbf{x}_i) = \mathbf{y}_i \tag{26}$$

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

$$\mathbf{y}_i = g_{r_2,i}^{-1}(\mathbf{x}_i) \tag{27}$$

which equals to $\mathbf{x}_i = g_{r_2,i}(\mathbf{y}_i)$. This means that the triple (y, r_2, x) must be positive, *i.e.* inferred as positive. □

Commutative/Non-commutative Composition

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

$$\begin{aligned}
&\forall x, y \in \mathcal{E}, (x, r_1 \circ r_2, y) \in \mathcal{T} \\
&\qquad \Rightarrow (x, r_2 \circ r_1, y) \in \mathcal{T} \\
&(\exists x, y \in \mathcal{E}, (x, r_1 \circ r_2, y) \in \mathcal{T} \\
&\qquad \Rightarrow (x, r_2 \circ r_1, y) \notin \mathcal{T})
\end{aligned}$$

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*

$$\begin{aligned}
&\forall x, y, z \in \mathcal{E}, (x, r_1, y) \in \mathcal{T} \wedge (y, r_2, z) \in \mathcal{T} \\
&\qquad \Rightarrow (x, r_3, z) \in \mathcal{T}
\end{aligned}$$

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 $\mathfrak{U}_{r_2,i}\mathfrak{U}_{r_1,i} = \mathfrak{U}_{r_3,i}$. If r_1 and r_2 are commutative, then $\mathfrak{U}_{r_2,i}\mathfrak{U}_{r_1,i} = \mathfrak{U}_{r_1,i}\mathfrak{U}_{r_2,i}$. If r_1 and r_2 are non-commutative, then $\mathfrak{U}_{r_2,i}\mathfrak{U}_{r_1,i} \neq \mathfrak{U}_{r_1,i}\mathfrak{U}_{r_2,i}$ for $i = 1, 2, \cdots, d$ where d is relation embedding dimension.*

Proof. According to Definition **??**, 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

$$\begin{aligned}
g_{r_1,i}(\mathbf{x}_i) &= \mathbf{y}_i \\
g_{r_2,i}(\mathbf{y}_i) &= \mathbf{z}_i
\end{aligned} \tag{28}$$

then it holds $g_{r_3,i}(\mathbf{x}_i) = \mathbf{z}_i$ for $i = 1, 2, \cdots, d$ where

$$\begin{aligned}
&g_{r_j,i}(\mathbf{h}_i) = \mathfrak{U}_{r_j,i}\mathbf{h}_i, \\
&j = 1, 2, 3;\ i = 1, 2, \cdots, d
\end{aligned} \tag{29}$$

From Equation 28, we insert $\mathbf{y}_i = g_{r_1,i}(\mathbf{x}_i)$ into $g_{r_2,i}(\mathbf{y}_i) = \mathbf{z}_i$, which gives $g_{r_2,i}(g_{r_1,i}(\mathbf{x}_i)) = \mathbf{z}_i$. Therefore, we have

$$g_{r_2,i} \circ g_{r_1,i}(\mathbf{x}_i) = \mathfrak{U}_{r_2,i}\mathfrak{U}_{r_1,i}\mathbf{x}_i = \mathbf{z}_i. \tag{30}$$

Considering the Proposition 6 and assuming $\mathfrak{U}_{r_2,i}\mathfrak{U}_{r_1,i} = \mathfrak{U}_{r_3,i}$, we have $g_{r_2,i} \circ g_{r_1,i}(\mathbf{x}_i) = g_{r_3,i}(\mathbf{x}_i) = \mathbf{z}_i$. 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 $\mathfrak{U}_{r_2,i}\mathfrak{U}_{r_1,i} = \mathfrak{U}_{r_1,i}\mathfrak{U}_{r_2,i}$. If r_1 and r_2 are non-commutative, then $\mathfrak{U}_{r_2,i}\mathfrak{U}_{r_1,i} \neq \mathfrak{U}_{r_1,i}\mathfrak{U}_{r_2,i}$. □

A.2 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 many 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.* $awardnominee/\ldots/nominatedfor$), symmetry (*e.g.* $derivationally\_related\_form$ in WN18RR), and anti-symmetry (*e.g.* has_part in WN18RR).

A.3 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. In addition, we adopt the same type constraint from QuatE (Zhang et al.,

Dataset	lr	drop	d_e	d_r	bs
FB15k	0.00005	0.1	600	600	512
FB15k-237	0.0005	0.2	200	200	512
WN18	0.0001	0.1	400	400	512
WN18RR	0.00005	0.2	400	400	512

Table 6: Hyper-parameter values for QubitE across all datasets.

Dataset	MRR	Hits@10	Hits@3	Hits@1
FB15k-237	.366 $\pm 3 * 10^{-7}$	.554 $\pm 2 * 10^{-6}$	.400 $\pm 2 * 10^{-6}$	.273 $\pm 3 * 10^{-7}$
WN18RR	.467 $\pm 9 * 10^{-7}$	.525 $\pm 2 * 10^{-2}$	.478 $\pm 3 * 10^{-2}$	.437 $\pm 1 * 10^{-2}$
FB15k	.807 $\pm 2 * 10^{-6}$	.894 $\pm 1 * 10^{-3}$	.838 $\pm 3 * 10^{-2}$	.758 $\pm 2 * 10^{-2}$
WN18	.950 $\pm 5 * 10^{-7}$	.957 $\pm 3 * 10^{-6}$	.952 $\pm 5 * 10^{-6}$	.945 $\pm 8 * 10^{-7}$

Table 7: The mean values and variances of QubitE's results across all datasets.

2019). More clearly, type constraint is to constraint the type (head or tail) of indicate entities in evaluation. Besides, we perform grid search to obtain the best hyperparameters according to MRR on the validation set. The hyperparameters are selected as follows: embedding dimension $n \in \{100, 200, 300, 400, 500, 600, 800, 1000\}$, dropout rate $drop \in \{0.1, 0.2, 0.3\}$, batch size $B \in \{256, 512, 1024\}$. To clarify, we take 1-N scoring (Dettmers et al., 2018) to speed up training.

Table 6 shows the hyperparameter values reported for QubitE across all datasets, where lr denotes (learning rate), drop (dropout rate), d_e (entity embedding dimension), d_r (relation embedding dimension), bs (batch size).

A.4 Baselines

In this section, we introduce the baseline models in our experiments.

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 dependently; **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 convolutional neural network (CNN) to predict tails and define the scoring function; **CoPER** (Stoica et al., 2020) generates contextual parameters into neural network to predict links.

A.5 Error Bars of Main Results

To evaluate the link prediction performance of QubitE, we run the model five times with random seeds $1, 10, 100, 1000, 10000$. In this section, we report the error bars of these results. Table 7 shows the error bar of QubitE's results on the four datasets. Overall, the variances are small, which demonstrate that the performence of QubitE is stable.

Source Entity	Source Entity Type	Negation Entity	Negation Entity Type	Score
Hermann Hesse	/music/artist	Dannii Minogue	/tv/tv_actor	0.8221
Norman Stiles	/award/award_winner	The Verdict	/award/award_winning_work	0.9402
Edward G. Robinson	/award/award_winner	Snow White and the Huntsman	/award/award_winning_work	0.8526
Martin Scorsese	/tv/tv_producer	Liza Minnelli	/film/actor	0.8513
Ellie Kemper	/tv/tv_actor	Amy Winehouse	/music/artist	0.6913

Table 8: The negation entities for source entities, generated by quantum gate NOT.

A.6 Semantic Logic Computing

Benifit from quantum logic computing which relies on quantum advantages, we can apply classical quantum gates (not relation ones) to entity embeddings to create new entities. Take NOT gate into consideration. Mathematically, NOT gate can be written as following:

$$\mathrm{NOT} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{31}$$

Here we can create new entity NOT(h), the semantic negation of entity $\mathbf{h}$, via the following equation:

$$\mathrm{NOT}(\mathbf{h}) = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{h}_a \\ \mathbf{h}_b \end{pmatrix} = \begin{pmatrix} \mathbf{h}_b \\ \mathbf{h}_a \end{pmatrix} \tag{32}$$

Then we score NOT$(\mathbf{h})$ to all entities. The closest entity is regarded as the best interpretation of NOT$(\mathbf{h})$. We randomly select 5 entities in FB15k and list their negations in Table 8. From the result we observe that the negation create a connection between "artist" and "tv_actor", "award_winner" and "award_winning_work", "tv_producer" and "film_actor". Overall, from the type of entities, it makes sense that the target entity is the negation of the source entity.

A.7 Limitation

In our model, one entity is only represented by one qubit. However, there exists multi qubits system, that represents entities as multi qubits and brings more favorable features, though the theoretical analysis becomes difficult.

A.8 Potential Societal Impacts

Since our method learn quantum embeddings of entities and preserve quantum advantages, the model can capture deep semantic information of entities without the involvement of relations. If we use public data on the Internet to construct a knowledge graph, personal information may be exposed unexceptedly.

A.9 Supplementary Material

We also provide our experiment logs online[2].

[2]https://timecat.notion.site/QubitE-Exp-Logs-63c9ff16f03d49468131b5475849fc1e