Variational Quantum Circuit Model for Knowledge Graph Embedding

Yunpu Ma, Volker Tresp, Liming Zhao, and Yuyi Wang*

In this work, the first quantum Ansätze for the statistical relational learning on knowledge graphs using parametric quantum circuits are proposed. Two types of variational quantum circuits for knowledge graph embedding are introduced. Inspired by the classical representation learning, latent features for entities are first considered as coefficients of quantum states, while predicates are characterized by parametric gates acting on the quantum states. For the first model, the quantum advantages disappear when it comes to the optimization of this model. Therefore, a second quantum circuit model is introduced where embeddings of entities are generated from parameterized quantum gates acting on the pure quantum state. The benefit of the second method is that the quantum embeddings can be trained efficiently meanwhile preserving the quantum advantages. It is shown that the proposed methods can achieve comparable results to the state-of-the-art classical models, for example, RESCAL, DistMult. Furthermore, after optimizing the models, the complexity of inductive inference on the knowledge graphs might be reduced with respect to the number of entities.

1. Introduction

Over the last few years, some large-scale triple-oriented knowledge databases have been generated. These databases are principled approaches to knowledge representation and reasoning. They are widely used in large-scale artificial intelligence systems, such as question answering engines, human-computer interaction platforms, and decision-making support systems. One well-known example is the IBM's cognitive computing platform, the IBM Watson, where knowledge graphs are at the core of it. The other example is the largest universally accessible knowledge graph (KG) maintained by Google.

Nowadays, knowledge graphs proliferate with increasing numbers of semantic triples and distinct entities. The reason is that knowledge graphs collect and merge information from various unstructured data, such as publications and internet. The increasing number of semantic triples and distinct entities leads to a slow training of knowledge graphs, as well as a sluggish response to the inductive inference tasks on knowledge graphs after training. Therefore, in order to accelerate the learning and inference on knowledge graphs, we propose statistical relational learning using quantum Ansätze.

In this work, we propose the first quantum Ansätze for modeling and learning large-scale relational databases using parametric quantum circuits. We simulate our quantum learning algorithms on graphics processing units (GPUs) and demonstrate the model performance on multiple state-of-the-art relational databases. We will also discuss how these quantum Ansätze could speed up the inference.

2. Representation Learnings on Knowledge Graphs

Various statistical relational models for large-scale KGs have been proposed in the literature, such as the bilinear model (RESCAL[1]), the bilinear diagonal model (DistMult[2]), the complex embedding model (ComplEx[3]). In this section, we first introduce knowledge graphs, and provide a succinct introduction to representation learning in KGs. We adapt the notation of ref. [4] for convenience.

2.1. Knowledge Graphs

Knowledge graphs are triple-oriented knowledge representations. The core components of KGs are semantic triples *(subject, predicate, object)* where subject and object are entities represented as nodes in the graph and where predicate indicates the labeled link from the subject to the object. One example of a semantic triple in **Figure 1** could be *(Angela_Merkel, Chancellor_of, Germany)*. Observed semantic triples (marked as solid lines in Figure 1) are elements of the training dataset, while unobserved triples (marked as dashed lines) will be inferred during the test.

Y. Ma, Prof. V. Tresp
Ludwig Maximilian University of Munich
Geschwister-Scholl-Platz 1, 80539 Munich, Germany
E-mail: yunpu.ma@siemens.com

Y. Ma, Prof. V. Tresp
Siemens AG
Otto-Hahn-Ring 6, 81739 Munich, Germany

Dr. L. Zhao
Singapore University of Technology and Design
Singapore

Dr. Y. Wang
ETH Zurich
Gloriastrasse 35, 8092 Zurich, Switzerland

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/qute.201800078

DOI: 10.1002/qute.201800078

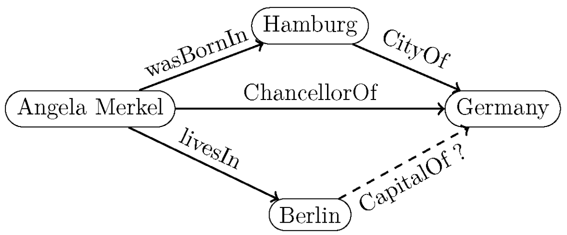

Figure 1. A knowledge graph fragment: observed semantic triples are marked with solid arrows, while unobserved semantic triples are marked with dashed arrows.

2.2. Representation Learning

Let $\mathcal{E}$ denote the set of entities, and $\mathcal{P}$ the set of predicates. Let N_e be the number of entities in $\mathcal{E}$, and N_p the number of predicates in $\mathcal{P}$. Given a predicate $p \in \mathcal{P}$, the indicator function $\phi_p : \mathcal{E} \times \mathcal{E} \to \{1, 0\}$ indicates whether a triple $(\cdot, p, \cdot)$ is true or false. Furthermore, $\mathcal{R}_p$ indicates the set of all subject–object pairs, such that $\phi_p = 1$. The entire KG can be written as $\chi = \{(i, j, k)\}$, with $i = 1, \dots, N_e$, $j = 1, \dots, N_p$, and $k = 1, \dots, N_e$. A knowledge graph can be equivalently treated as a $N_e \times N_p \times N_e$-dimensional three-order tensor, and an entry indicates whether a semantic triple is true, false or unobserved.

We assume that each entity and predicate has a unique latent representation. Let $\mathbf{a}_{e_i}$, $i = 1, \dots, N_e$, be the representations of entities, and $\mathbf{a}_{p_i}$, $i = 1, \dots, N_p$, be the representations of predicates. Note that $\mathbf{a}_{e_i}$ and $\mathbf{a}_{p_i}$ could be real- or complex-valued vectors/matrices. Moreover, when we consider a concrete example, say (s, p, o), we use $\mathbf{a}_\mathrm{s}$, $\mathbf{a}_\mathrm{p}$, and $\mathbf{a}_\mathrm{o}$ to represent the latent representation of the subject s, the predicate p, and the object o, respectively.

A probabilistic model for the knowledge graph χ is defined as $\Pr(\phi_p(s, o) = 1|\mathcal{A}) = \sigma(\eta_{spo})$ for all (s, p, o)-triples in χ, where $\mathcal{A} = \{\mathbf{a}_{e_i}\}_{i=1}^{N_e} \cup \{\mathbf{a}_{p_i}\}_{i=1}^{N_p}$ denotes the collection of all embeddings; $\sigma(\cdot)$ denotes the sigmoid function; η_{spo} is the value function of latent representations $\mathbf{a}_s$, $\mathbf{a}_p$, and $\mathbf{a}_o$. Given a labeled dataset containing both false and true triples $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$, with $x_i \in \chi$, and $y_i \in \{-1, 1\}$, latent representations can be learned from a loss function. Commonly, one minimizes the regularized logistic loss function

$$\min_{\mathcal{A}} \sum_{i=1}^{m} \log(1 + \exp(-y_i \eta_{x_i})) + \lambda ||\mathcal{A}||_2^2 \tag{1}$$

where m is the number of training samples, η_{x_i} is the value function for the semantic triple x_i, and λ is the regularization hyperparameter. Another commonly used loss function is the regularized mean squared error (MSE) loss

$$\frac{1}{m} \sum_{i=1}^{m} (y_i - \eta_{x_i})^2 + \lambda ||\mathcal{A}||_2^2 \tag{2}$$

Note that the value function η_{spo} can be defined differently in different models. For instance, for the RESCAL[1] model, entities are represented as unique R-dimensional real-valued vectors, $\mathbf{a}_{e_i} \in \mathbb{R}^R$, with $i = 1, \dots, N_e$, and predicates are represented as $R \times R$ matrices, $\mathbf{a}_{p_i} \in \mathbb{R}^{R \times R}$, with $i = 1, \dots, N_p$. Moreover, the value function is defined as

$$\eta_{spo} = \mathbf{a}_s^\intercal \mathbf{a}_p \mathbf{a}_o \tag{3}$$

For DistMult,[2] $\mathbf{a}_{e_i}, \mathbf{a}_{p_j} \in \mathbb{R}^R$, with $i = 1, \dots, N_e$, $j = 1, \dots, N_p$. The value function is defined as

$$\eta_{spo} = \langle \mathbf{a}_s, \mathbf{a}_p, \mathbf{a}_o \rangle \tag{4}$$

where $\langle \cdot, \cdot, \cdot \rangle$ denotes the tri-linear dot product.

For ComplEx,[3] entities and predicates are complex-valued vectors $\mathbf{a}_{e_i}, \mathbf{a}_{p_j} \in \mathbb{C}^R$, with $i = 1, \dots, N_e$, $j = 1, \dots, N_p$. The value function for the ComplEx model reads

$$\eta_{spo} = \Re(\langle \mathbf{a}_s, \mathbf{a}_p, \bar{\mathbf{a}}_o \rangle) \tag{5}$$

where the bar denotes complex conjugate, and $\Re$ denotes the real part of a complex number.

For the Tucker[5] tensor decomposition model, entities and predicates are real-valued vectors, $\mathbf{a}_{e_i} \in \mathbb{R}^R$, with $i = 1, \dots, N_e$, and $\mathbf{a}_{p_j} \in \mathbb{R}^R$, with $j = 1, \dots, N_p$. Additionally, a global core tensor $\mathbf{W} \in \mathbb{R}^{R \times R \times R}$ is introduced. The value function for the Tucker model reads

$$\eta_{spo} = \mathbf{W} \times_1 \mathbf{a}_s \times_2 \mathbf{a}_p \times_3 \mathbf{a}_o \tag{6}$$

Tensor models and compositional models are principled approaches for modeling large-scale relational data. The global relational patterns are encoded in the latent features of entities and predicates after optimizing the models. Thus, it is beneficial to analyze how the dimensionality of latent features influences the expressiveness and the generalization ability of the models. These questions have been studied in ref. [6]. In order to interpret the results in ref. [6], we first introduce the following notations.

Definition 1. *Let $\mathbf{X} \in \mathbb{R}^{\prod_{i=1}^m n_i}$ be an m-order tensor with dimensions $\mathbf{n} = (n_1, \dots, n_m)$. Suppose that it can be written as a (Tucker) tensor product $\mathbf{X} = \mathbf{W} \times_1 U^{(1)} \times_2 \cdots \times_m U^{(m)}$ with n-rank $\mathbf{R} = (R_1, \dots, R_m)$, where $\mathbf{W} \in \mathbb{R}^{\prod_{i=1}^m R_i}$ is the core tensor, and $U^{(i)} \in \mathbb{R}^{n_i \times R_i}$ are the latent factor matrices. Each entry of the tensor $\mathbf{X}$ can be written as a polynomial*

$$x_{i_1,\dots,i_m} = \sum_{j_1=1}^{R_1} \cdots \sum_{j_m=1}^{R_m} w_{j_1,\dots,j_m} \prod_{k=1}^{m} u_{i_k, j_k}^{(k)}$$

The set of different sign patterns which can be expressed by the tensor $\mathbf{X}$ is defined as

$$\mathcal{S}_{\mathbf{n},\mathbf{R}} := \{\mathrm{sgn}(\mathbf{X}) \in \{-1, 0, +1\}^{\prod \mathbf{n}} | n\text{-}rank(\mathbf{X}) \leq \mathbf{R}\} \tag{7}$$

Note the cardinality $|\mathcal{S}_{\mathbf{n},\mathbf{R}}|$ indicates how expressive and flexible the Tucker tensor decomposition could be. For the KGs modeling with tensor decomposition, we focus on the case of three-order tensors. The upper bound of $|\mathcal{S}_{\mathbf{n},\mathbf{R}}|$ is given in the following Lemma.

Lemma 1 (Upper Bound for Sign Patterns[6]). *Consider a three-order tensor $\mathbf{X} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ which can be written as a tensor product $\mathbf{X} = \mathbf{W} \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}$ with rank $\mathbf{R} = (R_1, R_2, R_3)$. The* *number of different sign patterns of the tensor* $\mathbf{X}$ *is upper bounded by the following number*

$$|\mathcal{S}_{\mathbf{n},\mathbf{R}}| \leq \left(\frac{16\mathrm{e}\; n_1 n_2 n_3}{\mathrm{var}(\mathbf{X})}\right)^{\mathrm{var}(\mathbf{X})} \tag{8}$$

where $\mathrm{var}(\mathbf{X})$ *is defined as* $\mathrm{var}(\mathbf{X}) := \prod_{i=1}^{3} R_i + \sum_{i=1}^{3} n_i R_i$.

Given observed entries of a KG, the above Lemma indicates that the ranks should be large enough to fit the observed entries via the tensor decomposition. Therefore, in order to model an ever-increasing KG with increasingly complex relational structures, the dimension of latent features also needs to grow with the KG. As a reminder, the complexity of value functions grows at least linearly with the dimension of the latent features for entities. For example, the computational complexity of the value function for the DistMult model is $\mathcal{O}(R)$ (see Equation (4)), while for the Tucker model it becomes $\mathcal{O}(R^3)$ (see Equation (6)). One goal of this work is to learn a probabilistic model for relational databases by making a quantum Ansatz for the value function. We will discuss how the evaluation of value functions can be accelerated via low-depth quantum circuits.

3. Quantum Circuit Models

In this section, we focus on variational unitary circuits. Algorithms of quantum classifiers using variational unitary circuits with parameterized and trainable gates have been proposed in ref. [7]. A quantum circuit U composed of L unitary operations can be decomposed into a product of unitary matrices

$$U = U_L \cdots U_l \cdots U_1$$

where each U_l indicates either a unitary operation on one qubit or a controlled gate acting on two qubits. Since a single qubit gate is a 2×2 unitary matrix in $SU(2)$, we apply the following parameterization

$$G(\alpha, \beta, \gamma) = \begin{pmatrix} \mathrm{e}^{i\beta}\cos\alpha & \mathrm{e}^{i\gamma}\sin\alpha \\ -\mathrm{e}^{-i\gamma}\sin\alpha & \mathrm{e}^{-i\beta}\cos\alpha \end{pmatrix} \tag{9}$$

where $\{\alpha, \beta, \gamma\}$ are the tunable parameters of the single qubit gate. Note that a global phase factor is neglected.

In the following, we introduce the parameterization of controlled gates. The controlled gate $C_i(G_j)$ which acts on the j-th qubit conditioned on the state of the i-th qubit can be defined as

$$C_i(G_j)\,|x\rangle_i \otimes |y\rangle_j = |x\rangle_i \otimes G_j^x\,|y\rangle_j$$

where $|x\rangle_i, |y\rangle_j$ denotes the state of the i-th and the j-th qubit, respectively.

Using the parametric gates G and $C(G)$, we are capable to describe the quantum circuit model U_θ with parameterization θ in more details. Let us consider a quantum state with n entangled qubits. Suppose that the l-th unitary operation U_l is a single qubit gate acting on the k-th qubit, then it can be written as

$$U_l = \mathbb{1}_1 \otimes \cdots \otimes G_k \otimes \cdots \otimes \mathbb{1}_n$$

If the l-th unitary operation acts on the j-th qubit and conditioned on the state of the i-th qubit, U_l will have the following matrix representation

$$\begin{aligned} U_l &= \mathbb{1}_1 \otimes \cdots \otimes \underbrace{\mathbb{P}_0}_{i\text{-th}} \otimes \cdots \otimes \underbrace{\mathbb{1}_j}_{j\text{-th}} \otimes \cdots \otimes \mathbb{1}_n \\ &\quad + \mathbb{1}_1 \otimes \cdots \otimes \underbrace{\mathbb{P}_1}_{i\text{-th}} \otimes \cdots \otimes \underbrace{G_j}_{j\text{-th}} \otimes \cdots \otimes \mathbb{1}_n, \end{aligned}$$

where $\mathbb{P}_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$ and $\mathbb{P}_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$.

4. Circuit Models for Knowledge Graphs

In this section, we introduce two quantum Ansätze for the value function and compare their computational complexities.

4.1. Quantum Circuit Embedding

We first introduce the *Quantum Circuit Embedding* (QCE) model, which can be considered as a generalization of the classical RESCAL model to the quantum regime. Similar to the RESCAL model, in QCE entities are represented by R-dimensional latent features. Without loss of generality, we assume that $R = 2^r$. In this way, an R-dimensional latent vector corresponds to a state of an r-qubit system.

One significant barrier to quantum learning algorithms is an efficient preparation of quantum states from classical data. In this work, we only consider real-valued representations for entities stored in a classical data structure $\mathcal{T}$. Then, a technique developed in ref. [8] can be utilized now, which can efficiently prepare the quantum states corresponding to latent features by accessing the classical data structure $\mathcal{T}$. In this way, the complexity of quantum state preparation can be reduced to the logarithm of R. Details related to the classical data structure $\mathcal{T}$ and the preparation of quantum states via $\mathcal{T}$ are relegated to the appendix. In summary, in the QCE model, entities are defined as $\mathbf{a}_{e_i} \in \mathbb{R}^R$, with normalized $l2$-norm $||\mathbf{a}_{e_i}||_2 = 1$, for $i = 1, \ldots, N_e$.

Furthermore, in QCE each predicate p is associated with a specific quantum circuit composed of sequential implementations of variational gates. Therefore, each predicate has an $R \times R$ unitary matrix representation $U_p(\theta_p)$, where θ_p are the predicate-specific trainable parameters stemming from the variational quantum gates. Moreover, we fix the circuit architecture of implementing predicates such that each predicate is uniquely determined by the circuit parameterizations θ_p.

Given a semantic triple (s, p, o), how the value function η_{spo} is defined in the quantum model? As a reminder, in The RESCAL model, the value function η_{spo} can be seen as the dot product of two vectors $\mathbf{a}_{\mathrm{sp}}$ and $\mathbf{a}_{\mathrm{o}}$, where $\mathbf{a}_{\mathrm{sp}} := \mathbf{a}_{\mathrm{s}}^{\intercal}\mathbf{a}_{\mathrm{p}}$. The loss function encourages the two vectors $\mathbf{a}_{\mathrm{sp}}$ and $\mathbf{a}_{\mathrm{o}}$ to point in the same direction if the given semantic triple is genuine, otherwise in opposite directions.

Inspired by the classical model ComplEx, we define the quantity $\eta_{\mathrm{spo}}^{\mathrm{QCE}} := \Re\,\langle \mathrm{o}|\, U_{\mathrm{p}}(\theta_{\mathrm{p}})\,|\mathrm{s}\rangle$. This quantity is the real part of the

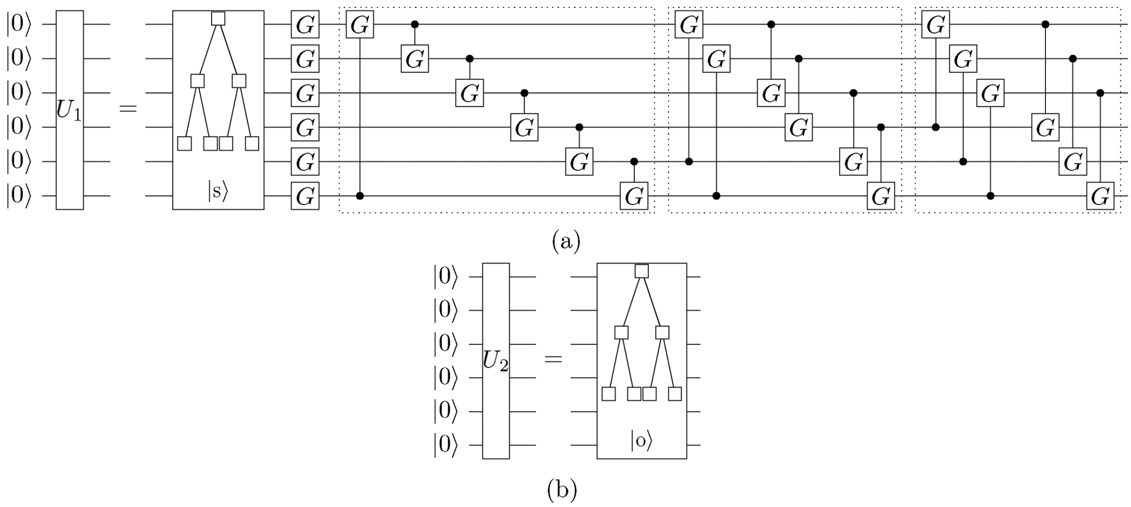

Figure 2. Building blocks of the QCE model. a) The U_1 module in the QCE model is displayed. U_1 encodes the latent feature of the subject s as the state $|\text{s}\rangle$. The quantum circuit associated to the predicate p maps the ket state $|\text{s}\rangle$ to another ket state $U_\text{p}(\theta_\text{p})\,|\text{s}\rangle$. For all the following experiments, we set the dimension of entity latent features as $R = 64$, which corresponds to a six-qubit system. In addition, the circuit architecture for all predicates is fixed, and it can be decomposed in four blocks: single qubit gates, two-qubit controlled gates with control range 1, 2, and 3 (dashed blocks). b) The U_2 module in the QCE model, which prepares the quantum state $|\text{o}\rangle$ is displayed. In both (a) and (b), the tree structure represents the quantum access to the classical data structure $\mathcal{T}$.

inner product of two quantum states $|\text{o}\rangle$ and $|\text{sp}\rangle := U_\text{p}(\theta_\text{p})\,|\text{s}\rangle$ generated by separate unitary circuits. The model parameters can be optimized by maximizing the inner product given genuine triples and minimizing the inner product given false or unobserved semantic triples. A relation between $\eta^{\text{QCE}}_{\text{spo}}$ and the label of the triple (s, p, o) will be specified later.

We explain the circuit architecture in more details. Latent features $\mathbf{a}_\text{s}$ for the subject and $\mathbf{a}_\text{o}$ for the object are first encoded in quantum states $|\text{s}\rangle$ and $|\text{o}\rangle$ through a quantum access to the memory structure $\mathcal{T}$. The dimension of features is set to $R = 64$ in the following experiments, which corresponds to a six-qubit system. Next, a unitary circuit $U_\text{p}(\theta_\text{p})$ corresponding to the predicate p evolves $|\text{s}\rangle$ to the state $U_\text{p}(\theta_\text{p})\,|\text{s}\rangle$. Note that both the latent features of entities and the parametric circuits of predicates need to be optimized during the training.

We develop the circuit for predicates out of four building blocks, and each block consists of variational gates or controlled gates operating on each of the six qubits. To be more specific, the first block consists of single qubit rotations, and the rest of the blocks consist of two-qubit controlled gates with control range 1, 2, 3, respectively. So, the unitary circuit associated with the predicates can be written as $U_{p_i}(\theta_{p_i}) = U_4\ U_3\ U_2\ U_1$, with $i = 1, \ldots, N_p$, where

$$U_1 = G_6\ G_5\ G_4\ G_3\ G_2\ G_1$$
$$U_2 = C_6(G_1)\ C_1(G_2)\ C_2(G_3)\ C_3(G_4)\ C_4(G_5)\ C_5(G_6)$$
$$U_3 = C_5(G_1)\ C_6(G_2)\ C_1(G_3)\ C_2(G_4)\ C_3(G_5)\ C_4(G_6)$$
$$U_4 = C_4(G_1)\ C_5(G_2)\ C_6(G_3)\ C_1(G_4)\ C_2(G_5)\ C_3(G_6) \tag{10}$$

Note that the index for the predicate was neglected since we assume that the circuit architecture is fixed for all the predicates. **Figure 2** illustrates the circuits for preparing the states $|\text{o}\rangle$ and $|\text{sp}\rangle$. In the following, we show that the value $\eta^{\text{QCE}}_{\text{spo}}$ can be measured physically. We adopt a similar idea to SWAP test for discriminating two quantum states. The SWAP test was initially proposed for quantum fingerprinting,[9] and it was further developed within refs. [10,11] for discriminating quantum evolution operators.

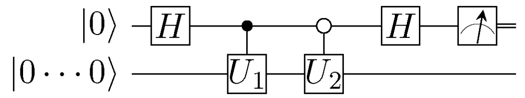

Figure 3. Quantum circuit for estimating the value $\Re\,\langle \text{o}|\,U_\text{p}(\theta_\text{p})\,|\text{s}\rangle$. The detailed architectures of unitary evolutions U_1 and U_2 can be found in Figure 2 for the QCE model and Figure 4 for the fQCE model.

The basic idea is illustrated in **Figure 3**. This architecture is inspired by ref. [11] and Observation 3 in ref. [7]. Consider two unitary operations U_1 and U_2 which operate on a pure state $|\mathbf{0}\rangle := |0\cdots 0\rangle$ conditioned on the state of the ancilla qubit. Particularly, the quantum state becomes $U_1\,|\mathbf{0}\rangle$ if the ancilla qubit is $|1\rangle_\text{A}$ and $U_2\,|\mathbf{0}\rangle$ if it is in the state $|0\rangle_\text{A}$. Before measuring the ancilla qubit, the underlying quantum state of the entire system reads

$$\frac{1}{\sqrt{2}}(|0\rangle_\text{A}\ U_2\,|\mathbf{0}\rangle + |1\rangle_\text{A}\ U_1\,|\mathbf{0}\rangle)$$

The second Hadamard gate acting on the ancilla qubit brings the state to

$$\frac{1}{2}[|0\rangle_\text{A}\,(U_2\,|\mathbf{0}\rangle + U_1\,|\mathbf{0}\rangle) + |1\rangle_\text{A}\,(U_2\,|\mathbf{0}\rangle - U_1\,|\mathbf{0}\rangle)]$$

For the QCE model, the unitary modules U_1 and U_2 are illustrated in Figure 2. Considering a concrete semantic triplet (s, p, o), with an access to the quantum gates parameters we can prepare the following quantum state according to the second Hadamard formula above:

$$\frac{1}{2}\left[|0\rangle_{\mathrm{A}} (|\mathrm{o}\rangle + |\mathrm{sp}\rangle) + |1\rangle_{\mathrm{A}} (|\mathrm{o}\rangle - |\mathrm{sp}\rangle)\right]$$

Therefore, the probability of sampling the ancilla qubit in the state $|0\rangle_{\mathrm{A}}$ is

$$\Pr(|0\rangle_{\mathrm{A}}) = \frac{1}{2} + \frac{1}{2}\Re\,\langle \mathrm{o}|\mathrm{sp}\rangle = \frac{1}{2} + \frac{1}{2}\eta_{\mathrm{spo}} \tag{11}$$

while the probability of measuring it in the state $|1\rangle_{\mathrm{A}}$ is

$$\Pr(|1\rangle_{\mathrm{A}}) = \frac{1}{2} - \frac{1}{2}\Re\,\langle \mathrm{o}|\mathrm{sp}\rangle = \frac{1}{2} - \frac{1}{2}\eta_{\mathrm{spo}} \tag{12}$$

In the upper equation, we temporarily neglect the superscript of the value function. As we can see, the value η_{spo} is related to the statistics of sampled quantum states of the ancillary qubit via

$$\eta_{\mathrm{spo}} = 2\Pr(|0\rangle_{\mathrm{A}}) - 1 = 1 - 2\Pr(|1\rangle_{\mathrm{A}}) \tag{13}$$

Similar to the classical models, this quantity defines the loss function jointly with the labels of the triplets.

4.2. Loss Function and Training

Details of the loss function and the optimization method are provided in this section. We focus on the QCE model. Given a training dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^m$ with $x_i \in \chi$, the loss function of the quantum circuit model is defined as the following mean error

$$\mathcal{L} = \frac{1}{m}\sum_{i=1}^{m}\left(y_i - \eta_{x_i}^{\mathrm{QCE}}\right)^{2\kappa} \tag{14}$$

where $y_i \in \{-1, 1\}$ are labels, and $\kappa \in \mathbb{Z}^+$ is a hyperparameter. The reason for this choice of the labels will be clarified later. One can also notice that for the quantum model, the loss function is not regularized by the norm of parameters. Because of the unitary constraint on the evolution of quantum circuits, hidden quantum states are automatically normalized. Therefore, the $l2$ norm cannot either effect the norms of embedding vectors or improve the generalization ability of the quantum circuit model.

With the loss function, the model is optimized by updating the parameters via gradient descent. Parameters of the variational gates can be efficiently estimated using a hybrid gradient descent scheme introduced in ref. [7]. The partial derivative of Equation (14) with respect to the gate parameters reads

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{2\kappa}{m}\sum_{i=1}^{m}(\eta_{x_i}^{\mathrm{QCE}} - y_i)^{2\kappa-1}\frac{\partial}{\partial\theta}\eta_{x_i}^{\mathrm{QCE}} \tag{15}$$

where $\theta \in \{\alpha_{p_i}, \beta_{p_i}, \gamma_{p_i}\}$, with $i = 1, \ldots, N_p$.

The techniques developed within refs. [7,12] allow the above partial derivate to be estimated from the states' statistics of the ancilla qubit since the partial derivate can be written as a linear combination of gates with shifted parameters. To be specific, we have the following derivatives for a single qubit gate

$$\frac{\partial}{\partial\alpha}G(\alpha,\beta,\gamma) = G\left(\alpha + \frac{\pi}{2}, \beta, \gamma\right)$$

$$\frac{\partial}{\partial\beta}G(\alpha,\beta,\gamma) = \frac{1}{2}G\left(\alpha, \beta + \frac{\pi}{2}, 0\right) + \frac{1}{2}G\left(\alpha, \beta + \frac{\pi}{2}, \pi\right)$$

$$\frac{\partial}{\partial\gamma}G(\alpha,\beta,\gamma) = \frac{1}{2}G\left(\alpha, 0, \gamma + \frac{\pi}{2}\right) + \frac{1}{2}G\left(\alpha, \pi, \gamma + \frac{\pi}{2}\right)$$

Moreover, partial derivatives of two-qubit gates can be written as a combination of control gates with shifted gates' parameters. More details of the hybrid gradient descent approach can be found in Section 4 of ref. [7].

However, this technique cannot be applied to the estimation of the gradients with respect to the latent features of entities. Another problem is that even if we could efficiently estimate the gradients with respect to the latent features, the entire classical data structure $\mathcal{T}$ needs to be updated after each step of optimization due to the normalization constraints. It leads to a computational overhead of $\mathcal{O}(R^2)$ for just one update of $\mathbf{a}_{e_i}$, with $i = 1, \ldots, N_e$.

4.3. Fully Parameterized Quantum Circuit Embedding

To overcome the disadvantages of the QCE, at this place, we introduce another *fully parameterized Quantum Circuit Embedding* (fQCE) model. The idea behind fQCE is reasonably simple. Instead of storing and reading entity features as normalized R-dimensional vectors, they are obtained by applying parameterized quantum circuit to initial quantum states which can be easily prepared. In this way, each entity is uniquely identified by the circuit architecture and the gates parameters similar to the circuits definition of predicates in the QCE model.

Compared to the QCE model, the advantages of this approach are twofolds. First, latent features do not need to be loaded from the classical data structure $\mathcal{T}$ and encoded as the coefficients of quantum states. Alternatively, they are generated from the quantum evolution of initial quantum states. Second, fQCE can be optimized efficiently since the only trainable parameters are in the variational quantum gates. Therefore, techniques explained in the last subsection can be applied to accelerate the optimization.

The circuit architecture for generating quantum representations of entities is given in **Figure 4** and overall we fix the circuit architecture for all entities. The six-qubit quantum system is initialized as a pure quantum state $|\mathbf{0}\rangle$. Hadamard gates act on each qubit to create a superposition $H_{6,\ldots,1}|\mathbf{0}\rangle := H_6 H_5 \cdots H_1 |\mathbf{0}\rangle$. Subsequently, an entity-specific unitary circuit develops the quantum representation from the superposition,

$$|e_i\rangle = U_{e_i} H_{6,\ldots,1}|\mathbf{0}\rangle, \quad \text{with} \quad i = 1, \ldots, N_e \tag{16}$$

where U_{e_i} have the same circuit architecture design as U_{p_i} in Equation (10).

To harvest the quantum advantages, the circuit depth should be low and in the order of $\log(R)$. In this way, we only need to replicate the experiments $\mathcal{O}(\log^2 R/\epsilon^2)$ times to update the model parameters given one training example, where ϵ is the accuracy

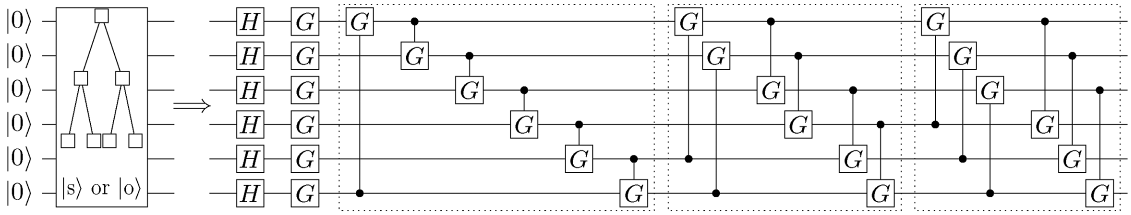

Figure 4. In the fQCE model, the classical data structure $\mathcal{T}$ is replaced by variational unitary circuits. Therefore, the quantum states $|\text{s}\rangle$ or $|\text{o}\rangle$ can be prepared by applying unitary circuits to the initial quantum states $|00\cdots0\rangle$, instead of loading data from the classical data structure $\mathcal{T}$. Note that the circuit architecture is fixed for all entities (subjects and objects). The unitary circuit contains five blocks. The first block consists of Hadamard gates which can develop superposition from the initial quantum state $|0\cdots0\rangle$. The rest of the blocks consist of single qubit gates and two-qubit controlled gates with control range 1, 2, and 3 (dashed blocks).

required. Moreover, the overall model architecture for estimating the value function $\eta_{\text{spo}}^{\text{fQCE}}$ using ancilla qubit remains the same as in Figure 3.

Before simulating the proposed quantum Ansätze, we compare computational complexities of them. We first consider the time complexity of evaluating the value function. In the QCE model, loading the entity features from the classical data structure $\mathcal{T}$ costs time $\mathcal{O}(\log R)$. Since we use shallow circuits with depth $\mathcal{O}(\log R)$ to specify the predicates, the unitary evolution of quantum states for entities requires $\mathcal{O}(\log^2 R)$ unitary operations. The value function is estimated from the Bernoulli distribution of the ancilla qubit. Therefore, one has to perform $\mathcal{O}(\frac{1}{\epsilon^2})$ repetitions of the experiment in Figure 3 to resolve the statistics of the ancilla qubit up to a predefined error ϵ. To summarize, the entire procedure of evaluating η_{spo}^{QCE} can be completed in runtime $\mathcal{O}(\text{poly}(\log R, \frac{1}{\epsilon}))$. Similarly, the evaluation of η_{spo}^{fQCE} requires a runtime $\mathcal{O}(\text{poly}(\log R, \frac{1}{\epsilon}))$.

A notable complexity difference between two quantum circuit models appears when it comes to the training. Let us first consider the fQCE model. Given one training sample (s, p, o), it requires $\mathcal{O}(\log^2 R/\epsilon^2)$ repetitions of the experiments to estimate the gradients and update the parameters in U_s, U_p, and U_o. Let D indicate the total number of semantic triples in the training dataset, then the runtime of one epoch is $\mathcal{O}(D\ \text{poly}(\log R, \frac{1}{\epsilon}))$. However, for QCE, the runtime of one training epoch becomes $\mathcal{O}(D\ \text{poly}(R, \log R, \frac{1}{\epsilon}))$ since after each step of optimization, renormalizing the entity latent features and updating the classical memory structure $\mathcal{T}$ require additional $\mathcal{O}(R)$ operations. As one can see, the quantum advantages disappear when we optimize the QCE model.

5. Experiments

5.1. Datasets and Evaluation

To evaluate proposed quantum models for knowledge graph embedding, we use four link prediction datasets of different sizes: Kinship,[13] FB15k-237,[14] WN18RR,[15] and GDELT.[16]

Kinship contains relations between family members. An example of the triple is *(Max, husband_of, Mary)*

FB15k-237 is a subset of Freebase with only 237 predicates. Most of the semantic triples in the FB15k-237 are related to the facts of cities, movies, sports, and musics, for example, *(California, located_in, U.S.)*.

Table 1. Statistics of different knowledge graphs.

	$\#\mathcal{D}$	N_e	N_p	N_a
Kinship	10 790	104	26	≈ 104
WN18RR	79 043	39 462	18	≈ 2
FB15k-237	310 079	14 505	237	≈ 21
GDELT	497 603	6785	230	≈ 73

GDELT The Global Database of Events, Language and Tone (GDELT) monitors evens between different countries and organizations. An example could be *(ESA, collaborates with, NASA)*.

WN18RR This hierarchical knowledge base is a subset of WordNet which consists of hyponym and hypernym relations between words, for example, *(pigeon, hyponym_of, bird)*.

The exact statistics of datasets are listed in **Table 1**, including the total number of triplets in the dataset $\#\mathcal{D}$, the number of entities N_e, the number of predicates N_p, and the average number of labeled links connecting to a node N_a.

Since the above-mentioned datasets only consist of positive (genuine) semantic triples, we generate negative (false) instances according to the method of corrupting semantic triples proposed in ref. [17]. Given a genuine semantic triple (s, p, o), negative triples are drawn by corrupting the object o to a different entity o', and similarly corrupting subject s to s'. This corruption method makes a local-closed world assumption, meaning that the knowledge graph is assumed to be only locally connected. Therefore, corrupted and unobserved semantic triples are treated as negative examples during the training.

The model performance is evaluated using the following metrics on the test dataset. Let us consider a semantic triple (s, p, o) in the test dataset. To evaluate the retrieval of the object o, given the subject s and the predicate p, we first replace the object o with every object o' and compute the values of $\eta_{\text{spo}'}$. Following that, we sort these values in a decreasing order and locate the target object o. This position is referred to as the rank of the target object. We provide the filtered ranking scores as suggested in ref. [17] by removing all semantic triples (s, p, o') with $y_{\text{spo}'} = 1$ and $o' \neq \text{o}$. Filtered ranking scores eliminate the ambiguity of retrieved information and provide a clearer performance evaluation of different models. In the same way, we also evaluated the retrieval of the subject s by ranking $\eta_{s'\text{po}}$ for all possible subjects s'.

To quantify the performance of the classical and quantum models on missing links predication, we report three metrics: filtered mean rank (MR), filtered Hits@3, and filtered Hits@10

Table 2. Filtered recall scores evaluated on four different datasets. Four metrics are compared: filtered Mean Rank (MR), filtered Hits@3 (@3), and filtered Hits@10 (@10).

	Kinship			WN18RR			FB15k-237			GDELT		
Methods	MR	@3	@10	MR	@3	@10	MR	@3	@10	MR	@3	@10
RESCAL	3.2	88.8	95.5	12036	21.3	25.0	291.3	20.7	35.1	185.0	10.4	22.2
DistMult	4.5	61.0	87.7	10903	21.0	24.8	305.4	23.4	39.1	130.4	12.1	24.5
Tucker	2.9	89.8	95.0	11997	19.1	23.9	276.1	20.9	35.7	144.0	14.5	27.3
ComplEx	**2.2**	**90.0**	**97.7**	11895	24.6	26.1	242.7	25.2	39.7	137.6	12.9	26.4
Best Known	–	–	–	4187[15]	**44.0**	**52.0**	244.0[15]	**35.6**	**50.1**	**102.0**[20]	**31.5**	**47.1**
QCE	3.6	73.8	93.8	3655	19.5	32.3	258.7	22.5	35.0	128.8	12.5	23.8
fQCE	3.6	73.1	94.0	**2160**	27.4	37.8	**236.0**	19.8	33.7	131.0	10.8	24.1

evaluated on the test dataset. Filtered mean rank is the average filtered ranking scores, and filtered Hits@n indicates the probability of finding the correct retrieval within the top-n filtered ranking.

The dimension of latent representations for all classical baselines is chosen as $R = 64$. For comparison, circuits algorithms for knowledge graph embedding are evaluated via six-qubit Ansätze. Overall, we fix the quantum circuit architecture depicted in Figure 2 for QCE and Figure 4 for fQCE model. Experiments show the recall scores on the test dataset are not sensitive to the order of implementing different blocks. Thus, for a simple implementation, we only consider four different blocks without repetitions. Exploration of various quantum circuit architectures to achieve better results could be an interesting research direction, and we leave it for future work.

During the training, the datasets are randomly split into training, validation, and test datasets. Early stopping on the validation set is used to avoid overfitting by monitoring the filtered His@3 entity recall scores every 20 epochs. Before training, all model parameters, including the entity embeddings and the gates parameters, are randomly initialized. In particular, we found that for the quantum Ansätze the model performance is relatively sensitive to the initialization of the gates parameters. After hyperparameter search, the gates parameters are initialized uniformly in the interval $[-\frac{\pi}{10}, \frac{\pi}{10}]$.

Here, we give more details on how quantum Ansätze are simulated. As discussed in Section 3, unitary evolution of a quantum state is equivalent to the unitary matrix-vector product. Therefore, we can simulate the quantum Ansätze on a single Tesla K80 GPU without exploiting real quantum devices.[18] For the QCE model, each entity embedding is randomly initialized from a multivariate normal distribution and normalized after initialization. Embeddings for entities are stored as NumPy arrays instead of in the classical data structure $\mathcal{T}$. All the parameters, including entity embeddings and gate parameters, are updated via stochastic gradient descent. Moreover, after each step of the update, entity embeddings will be normalized again. Differently, for the fQCE model, each entity is initialized as $\frac{1}{8}\,|00\cdots 0\rangle \equiv \frac{1}{8}(0, 0, \ldots, 0)^\intercal$ since all the trainable parameters are in the variational circuit. The codes are based on Tensorflow,[19] and they will be available online.

Table 2 reports the performance of classical models and quantum Ansätze evaluated on different datasets. In addition, the row *best known* in Table 2 shows the best results reported in the literatures.[21] From the table, we can read that both quantum circuit models can achieve comparable results to the classical models using the dimension $R = 64$. In some cases, for example, the filtered Mean Rank recall scores on the WN18RR, FB15k-237, and GDELT datasets, the quantum models can outperform all classical models.

Another interesting observation is that the Mean Rank score on the WN18RR dataset returned by the fQCE model is even better than the best-known models. We have to emphasize that among the four datasets, WN18RR contains the largest number of distinct entities (see Table 1). Therefore, fQCE is the desired quantum Ansatz of relational learning which shows both quantum advantages and superior performance on a vast database. However, one has to note that WN18RR possesses the smallest number of average links per node. Thus, questions are: Whether the quantum circuit models are only practical for modeling large and sparse datasets due to the intrinsic linearity of the circuit models; and whether applying nonlinearity activation functions on the circuit models[22,23] can further improve the performance on other dense datasets? We leave these questions for future research.

5.2. Regularizations

As mentioned before, the quantum circuit models cannot be regularized using the $l2$-norm due to the unitarity constraints. Hence, what regularization methods can be applied to improve the generalization ability of the circuit models? We examine two techniques that are widely used in classical learning: dropout and Gaussian noise of model parameters. Generally speaking, dropout reduces the overfitting on the training dataset and noise encourages the model to land on flat minima of the loss surface. These two methods can be efficiently applied without destroying the unitarity restrictions.

We first apply the dropout technique. During the training, each quantum gate has a nonzero probability to be switched off. From the perspective of a classical neural network, this dropout is equivalent to randomly removing some weight matrices and replacing them with identity matrices. Similar regularization methods have been used to train very deep neural networks with vanishing gradients.[24] However, all the gates will be

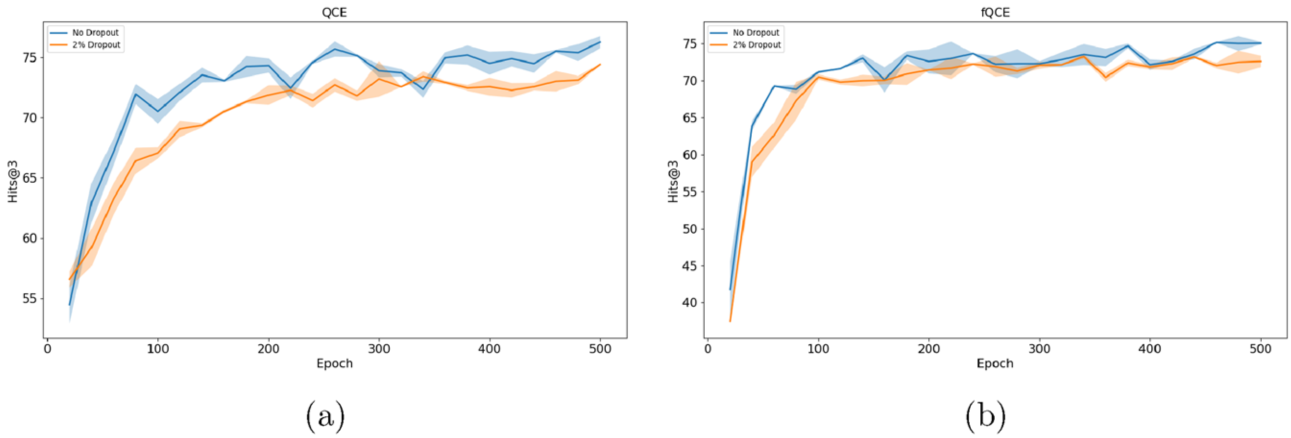

Figure 5. Comparison of the filtered Hits@3 recall scores on the Kinship dataset for a) QCE and b) fQCE. Blue line: without employing the dropout; orange line: 2% probability of dropping out a quantum gate randomly.

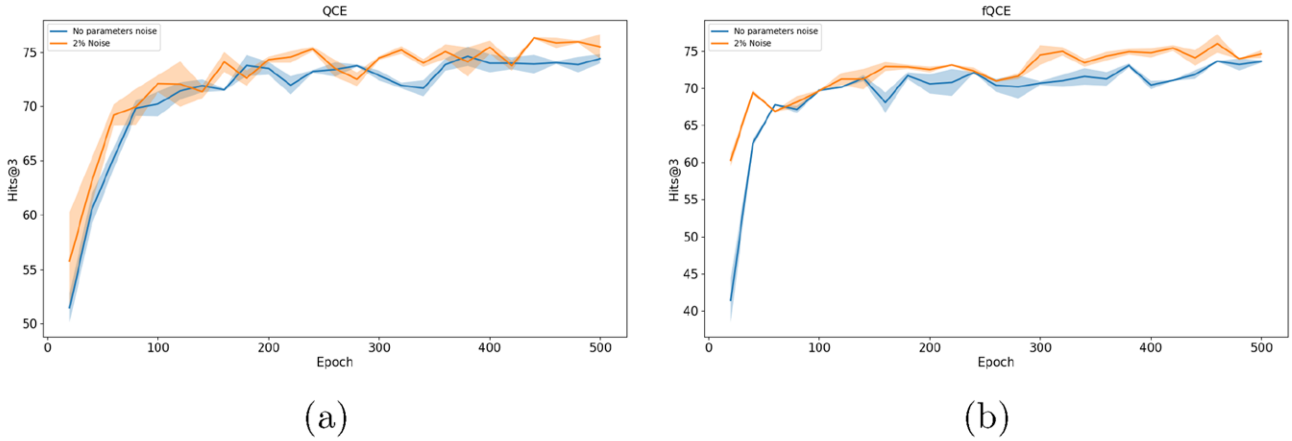

Figure 6. Filtered Hits@3 scores on the Kinship dataset for a) QCE and b) fQCE. Blue line: without introducing random noise; orange line: adding 2% noise to the model parameters both during the training and test.

implemented during the test phase. Because of the imperfections of quantum gates, quantum circuit models inherently possess this regularization property.

Simulations are performed for both circuit models using the Kinship dataset, and the dropout probability is chosen from $\{0.02, 0.05, 0.1, 0.2\}$. However, we could not observe any improvement in the performance, even using the smallest dropout probability. Recall scores for no dropout and 2% dropout probability are compared in **Figure 5**. Even though the dropout regularization cannot augment the performance of both models, we still learn that the fQCE model is more robust and resistant to imperfect quantum circuits, making it a potential candidate for the future test on real quantum devices.

Now we turn to study another regularization method which adds Gaussian noise to the model parameters. System noises are quite common in quantum computational devices, for example, they can stem from the disentanglement, flips of the qubits, or random phase rotations. However, in this work, we focus on noises stemming from inaccuracies. For example, the inevitably inaccurate loading of the classical data into quantum devices, the inaccurate parametric gates, or the statistical uncertainty about the state of ancilla qubit. To simulate quantum system imprecision, we add Gaussian perturbations to the model parameters as follows

$$\theta' = \theta + \mu\mathcal{N}(0, |\theta|) \tag{17}$$

where θ could be a gate parameter or an element of an entity latent feature defined in the QCE model, and μ indicates the noise level. We further assume the amplitude of perturbation added to a model parameter is proportional to this parameter's absolute value.

To be more realistic, perturbations are introduced not only during the training but also in the *test phase*. **Figures 6** and **7** compare the recall scores, the filtered Mean Rank and filtered Hits@3, on the Kinship dataset. Performance improvement can be observed in both quantum models which indicates that system imprecision brings the models to flat minima of the loss functions. As first pointed out in ref. [25], the flatness of the minimum on the loss surface found by an optimization algorithm is a good indicator of the generalization ability. Improved performance by adding noise also suggests that the computational complexity can be reduced by controlling the accuracy ϵ.

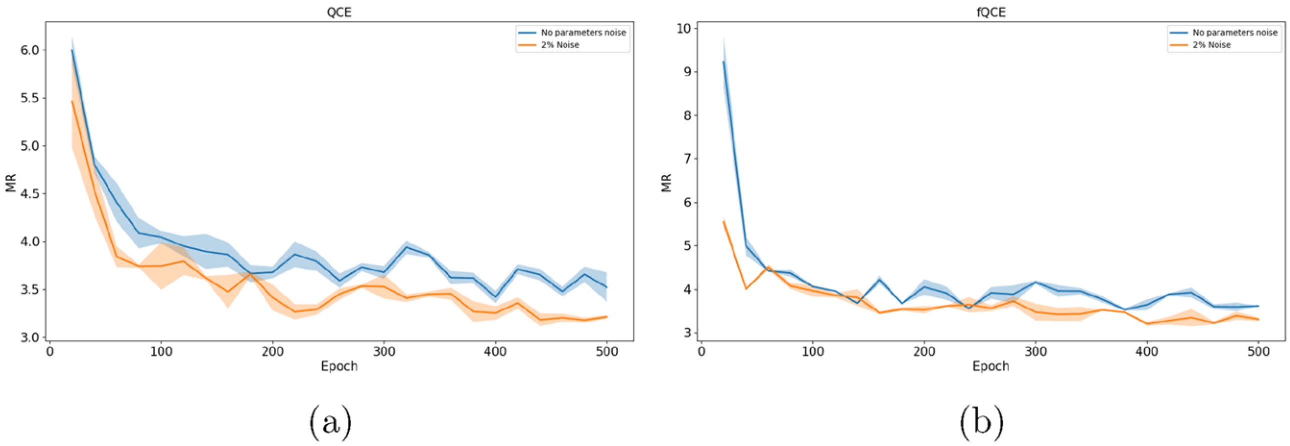

Figure 7. Filtered Mean Rank recall scores on the KINSHIP dataset for a) QCE and b) FQCE. Blue line: without introducing random noise; orange line: adding 2% noise to the model parameters both during the training and test.

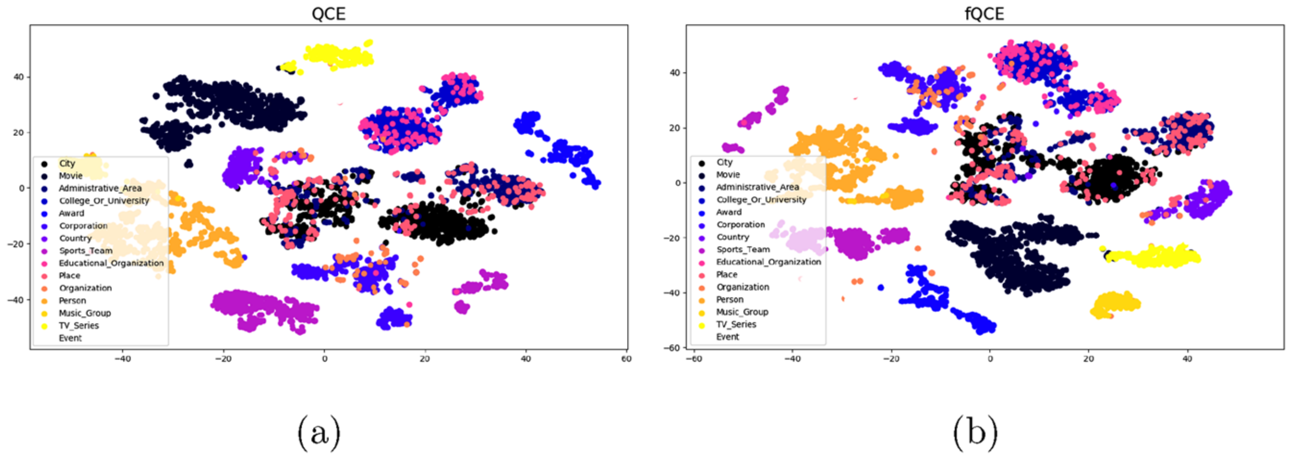

Figure 8. T-SNE visualizations of entity representations learned by a) QCE and b) FQCE.

5.3. T-SNE

We perform a qualitative analysis to visualize and understand the learned representations from the quantum Ansätze. Particularly, we focus on the latent features of entities. It has been reported that classical embeddings of entities show clustering effects. Entities with similar semantic meaning tend to group in the vector space. Here, we use t-SNE to analyze whether entity representations in the quantum models render this property. T-SNE[26] is a powerful method for visualizing high-dimensional data on a 2D plane.

In order to visualize the semantic clustering effect, we focus on the FB15K-237 dataset, since it contains categorical information about the entities. We extract the top-15 most frequent categories, for example, *Movie*, *Administrative_Area*, *Organization*, and display them using different colors on the t-SNE plot. We still need to clarify how the quantum features are defined. In QCE, entity representations are normalized vectors $\mathbf{a}_{e_i} \in \mathbb{R}^R$, with $i = 1, \ldots, N_e$. Besides, in the FQCE model, we define the hidden quantum states $|e_i\rangle = U_{e_i} H_{6,\ldots,1} |\mathbf{0}\rangle$, with $i = 1, \ldots, N_e$ (see Equation (16)), as entity representations.

The t-SNE visualizations of learned quantum representations are displayed in **Figure 8**. One can clearly recognize the clustering effect based on the categories of entities. It is intriguing to point out that in Figure 8, the pink nodes representing the category *Educational_Organization* overlap with the blue nodes which represent the category *College_Or_University*.

Quantum circuit models reveal better semantic clustering effects of the learned latent features than classical models. **Figure 9** displays the t-SNE visualization of entity latent representations learned by DISTMULT. Particularly, one can notice that the learned latent features of the semantic categories *City*, *Administrative_Area*, and *Place* strongly overlap without revealing more detailed structures. The better semantic clustering effect might explain why FQCE performs consistently well when comparing with the Mean Rank metric.

6. Accelerated Inference

In previous sections, we have shown that the value functions can be evaluated with reduced complexity using the quantum

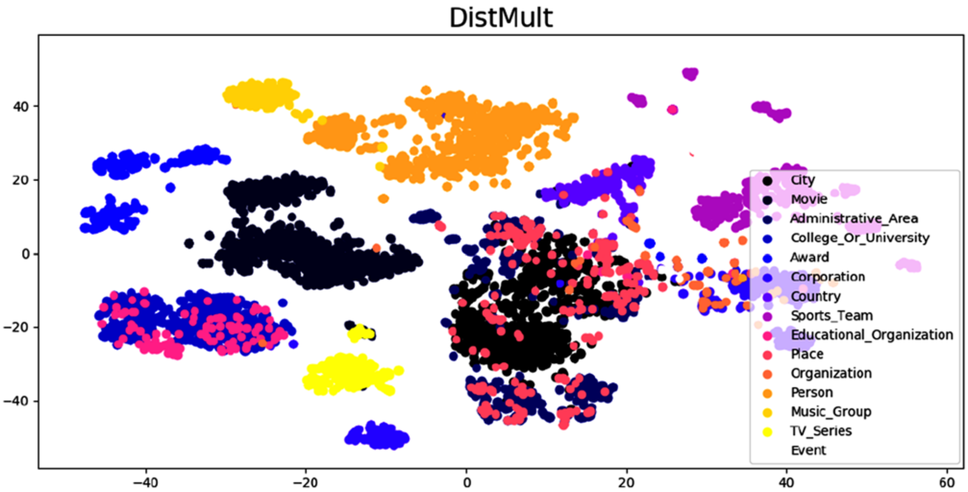

Figure 9. T-SNE visualization of entity representations learned by DistMult.

Ansätze. However, there is another quantum advantage we have not mentioned yet, namely the acceleration on the inference task. To be more specific, given an incomplete semantic triple (s, p, ·), we attempt to find a quantum algorithm which can accelerate the search for the best (or the most possible) candidates for the unknown object.

What makes this task very challenging? As mentioned before, we are dealing with ever-increasing knowledge graphs with the consistently increasing number of distinct entities. Inference using classical models, for example, RESCAL and Tucker requires many computation resources. The reason is we need to calculate all the value functions η_{spe_i}, with $i = 1, \ldots, N_e$. Then, the entity e_i that corresponds to the maximum η_{spe_i} will be located, and the algorithm returns e_i as the best candidate for the unknown object. It could be extremely time and resource consuming since the same algorithm has to be repeated at least N_e times and each time of evaluation requires $\mathcal{O}(\text{poly}\, R)$ classical operations.

We are motivated to find a quantum algorithm showing quantum acceleration on the inference task. Here, we describe an *idealistic* and *heuristic* quantum algorithm for the inference. First, we prepare the following quantum state

$$\frac{1}{\sqrt{2N_e}}\sum_{i=1}^{N_e}\left(|0\rangle_{\text{A}}\,|i\rangle_{\text{I}}\,|\mathbf{0}\rangle_{\text{L}} + |1\rangle_{\text{A}}\,|i\rangle_{\text{I}}\,|\mathbf{0}\rangle_{\text{L}}\right) \tag{18}$$

The first qubit with the subscript A is an ancilla qubit. The second index register with the subscript I consists of $n_e := \lceil \log_2 N_e \rceil$ qubits, and the state $|i\rangle_{\text{I}}$ is the binary representation for the index i of the entity e_i. Furthermore, the third register with $r = \log_2 R$ qubits is prepared in the pure state $|\mathbf{0}\rangle_{\text{R}}$ which will be used to generate the quantum representations of the entities.

Afterward, we use unitary circuit evolutions to prepare the states $|\text{sp}\rangle$ and $|e_i\rangle$. To be more specific, the U_1 circuit brings $|\mathbf{0}\rangle_{\text{L}}$ to the state $|\text{sp}\rangle$ conditioned on the ancilla qubit being $|1\rangle_{\text{A}}$. Moreover, an entity-dependent circuit $U_2(e_i)$ brings $|\mathbf{0}\rangle_{\text{L}}$ into the quantum state $|e_i\rangle$ conditioned on the ancilla being $|0\rangle_{\text{A}}$ and the index register being $|i\rangle_{\text{I}}$. Recall that the circuits U_1 and U_2 are defined in Figures 2 and 4.

To summarize, the unitary circuits will generate the following quantum state

$$\frac{1}{\sqrt{2N_e}}\sum_{i=1}^{N_e}\left(|0\rangle_{\text{A}}\,|i\rangle_{\text{I}}\,U_2(e_i)\,|\mathbf{0}\rangle_{\text{L}} + |1\rangle_{\text{A}}\,|i\rangle_{\text{I}}\,U_1\,|\mathbf{0}\rangle_{\text{L}}\right)$$
$$= \frac{1}{\sqrt{2N_e}}\sum_{i=1}^{N_e}\left(|0\rangle_{\text{A}}\,|i\rangle_{\text{I}}\,|e_i\rangle_{\text{L}} + |1\rangle_{\text{A}}\,|i\rangle_{\text{I}}\,|\text{sp}\rangle_{\text{L}}\right) \tag{19}$$

Performing the Hadamard gate on the ancilla qubit gives

$$\frac{1}{2\sqrt{N_e}}\sum_{i=1}^{N_e}\left(|0\rangle_{\text{A}}\,|i\rangle_{\text{I}}\,(|e_i\rangle_{\text{L}} + |\text{sp}\rangle_{\text{L}}) + |1\rangle_{\text{A}}\,|i\rangle_{\text{I}}\,(|e_i\rangle_{\text{L}} - |\text{sp}\rangle_{\text{L}})\right) \tag{20}$$

Note that the values η_{spe_i} are encoded in the probability amplitudes of the above quantum state Equation (20). For example, the probability of measuring the ancilla qubit and index register being in the quantum state $|0\rangle_{\text{A}}\,|i\rangle_{\text{I}}$ is given by

$$\Pr(|0\rangle_{\text{A}}\,|i\rangle_{\text{I}}) = \frac{1}{2N_e}(1 + \Re\,\langle e_i|\text{sp}\rangle_{\text{L}}) = \frac{1}{2N_e}(1 + \eta_{\text{spe}_i}) \tag{21}$$

Let us consider an idealistic case for the inference: The negative semantic triples have value functions −1, while the positive triples have value functions +1. In this case, the probability in Equation (21) takes value $\Pr(|0\rangle_{\text{A}}\,|i\rangle_{\text{I}}) = 0$ if the entity e_i is not a correct return to the query (s, p, ?), while $\Pr(|0\rangle_{\text{A}}\,|i\rangle_{\text{I}}) = \frac{1}{N_e}$ if the entity e_i is correct.

Since the index register is sampled conditioned on the ancilla qubit, we need to discuss the probability of post-selection on the ancilla qubit. The marginalized probabilities of measuring the ancilla qubit being $|0\rangle_{\text{A}}$ and $|1\rangle_{\text{A}}$ read

$$\Pr(|0\rangle_{\text{A}}) = \frac{1}{2} + \frac{1}{2N_e}\sum_{i=1}^{N_e}\eta_{\text{spe}_i}$$
$$\Pr(|1\rangle_{\text{A}}) = \frac{1}{2} - \frac{1}{2N_e}\sum_{i=1}^{N_e}\eta_{\text{spe}_i} \tag{22}$$

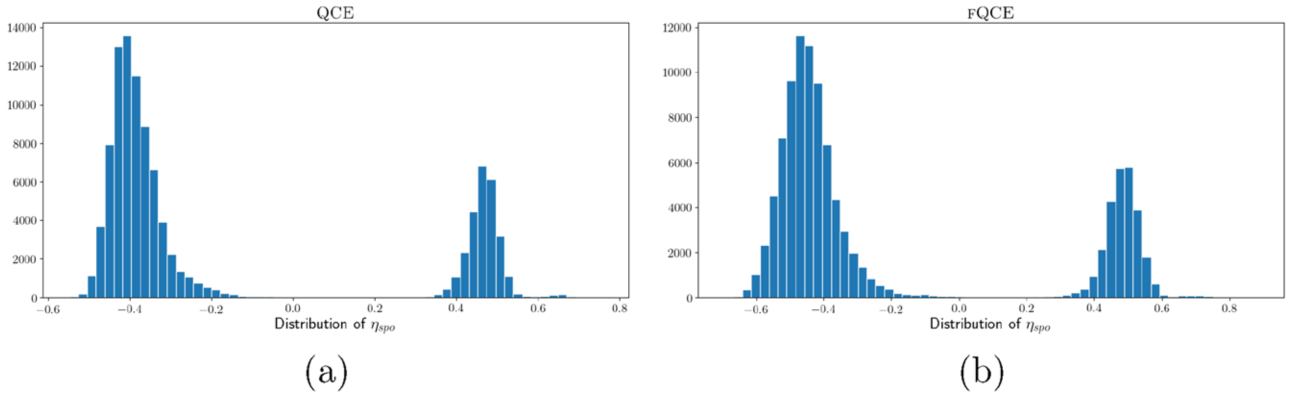

Figure 10. Empirical distributions of the value functions a) η_{spo}^{QCE} and b) η_{spo}^{fQCE} evaluated on the test dataset of Kinship. The targets are set as $y_i \in \{-1, 1\}$ during the training. Note that for each triple in the test dataset, say (s, p, o), the value functions η_{spe_i} and $\eta_{e_i\text{sp}}$, with $i = 1, \ldots, N_e$ are evaluated and accumulated for the plotting.

Assume that the cardinality of the solution set to the query (s, p, ?) is $H \in \mathcal{O}(1)$. In the *idealistic* situation, we have the marginalized probability $\Pr(|0\rangle_{\text{A}}) = \frac{H}{2N_e}$, and $\Pr(|1\rangle_{\text{A}}) = 1 - \frac{H}{2N_e}$. To read out the indices that correspond to the entities in the solution set, we can perform amplitude amplification[27] on the subspace $|0\rangle_{\text{A}}$ of the ancilla qubit. The number of required iterations is approximately $\lfloor \frac{\pi}{4}\sqrt{\frac{2N_e}{H}} \rfloor = \mathcal{O}(\sqrt{N_e})$. The resulting quantum state after the amplitude amplification reads

$$\frac{1}{\sqrt{H}} \sum_{i \in \{i | \phi_{\text{p}}(\text{s}, e_i) = 1\}} |0\rangle_{\text{A}} \, |i\rangle_{\text{I}} \tag{23}$$

It is unnecessary to perform quantum state tomography and read out all the probability amplitudes. We can sample the states of the index register conditioned on $|0\rangle_{\text{A}}$ and determine the most frequent states that are related to the indices of the entities giving the highest scores. Since the cardinality of the solution set is assumed to be $H \in \mathcal{O}(1)$, the same experiment needs to be replicated at least $\mathcal{O}(H\sqrt{N_e})$ times. Hence, this heuristic quantum algorithm realizes a quadratic acceleration with respect to the number of entities N_e.

Our idealistic quantum algorithm provides a quadratic acceleration during the inductive inference on the database. Even a quadratic speedup is desirable when the number of entities N_e is large. Note that another well-known quantum algorithm, Grover's algorithm,[28] which was designed for searching in a database, also provides a quadratic speedup. More specifically, Grover's algorithm can identify the input to an unknown function in $\mathcal{O}(\sqrt{N})$ steps from a N-item database. At the same time as Grover's publication, it is proved in ref. [29] that Grover's algorithm is an almost optimal solution. Different from this quantum algorithm for the database search, our algorithm is learning-based, adaptive, and inference-oriented.

Note that the above described quantum algorithm is merely *idealistic* and *heuristic*, since the scores of semantic triples in the test dataset take values from the interval $[-1, 1]$ instead of the discrete set $\{-1, 1\}$. **Figure 10** shows the empirical distribution of value functions on the Kinship test dataset.[30]

As one can observe that the empirical value functions concentrate around −0.5 and 0.5. The quantum advantage on inference might disappear in these cases since $\Pr(|0\rangle_{\text{A}} \, |i\rangle_{\text{I}}) \approx \Pr(|0\rangle_{\text{A}} \, |j\rangle_{\text{I}})$, $\forall i \in \{i | \phi_{\text{p}}(\text{s}, e_i) = 1\}$, and $j \notin \{i | \phi_{\text{p}}(\text{s}, e_i) = 1\}$. In other words, the probability of sampling correct solutions is approximately equal to the probability of sampling incorrect solutions. Thus, one promising future research direction is to study whether performing nonlinear functions on quantum representations can separate the positive and negative triples in an inference task.

7. Conclusion and Outlook

In this work, we study the quantum Ansätze for the statistical relational learning on knowledge graphs as well as latent quantum representations. Two different quantum models QCE and fQCE are proposed and compared by their complexity and performance. To be specific, QCE assumes that entity representations are stored in a classical data structure, while in the fQCE model quantum entity representations are generated from pure quantum states through unitary circuit evolution. The experiments show that both quantum Ansätze can achieve comparable results to the state-of-the-art classical models on several benchmark datasets.

This work can be further explored in several directions. The quantum circuit architecture could be fine-tuned using reinforcement learning or evolutionary algorithms. It is necessary to understand why quantum circuit models show superior performance on the WN18RR dataset which contains the most entities and the smallest average number of links. Whether this observation indicates that quantum circuit models are only suitable for modeling large but simple relational dataset due to the inherent linearity? Thus, a reasonable question is whether acting nonlinear operations on the quantum representations can improve the inductive inference on complex relational datasets and make the *idealistic* and *heuristic* quantum algorithm for the accelerated inference more realizable?

Appendix A: Preparation of Quantum States

Theorem A1.[31] *Let* $\mathbf{x} \in \mathbb{R}^R$ *be a real-valued vector. The quantum state* $|x\rangle = \frac{1}{\|\mathbf{x}\|_2} \sum_{i=1}^{R} x_i \, |i\rangle$ *can be prepared using* $\lceil \log_2 R \rceil$ *qubits in time* $\mathcal{O}(\log_2 R)$.

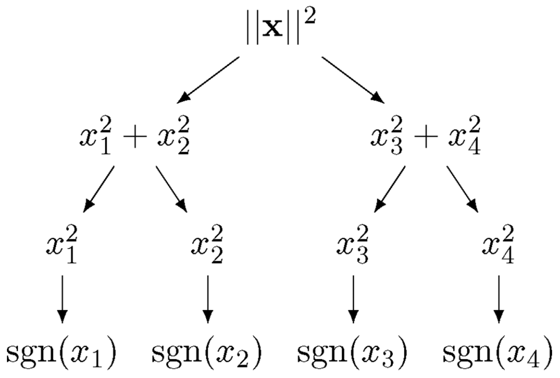

Figure A1. Classical memory structure with quantum access for creating the quantum state $|x\rangle = x_1|00\rangle + x_2|01\rangle + x_3|10\rangle + x_4|11\rangle$.

Theorem A1 claims that there exist a classical memory structure and a quantum algorithm which can load classical data into a quantum state with exponential acceleration. **Figure A1** illustrates a simple example. Given an $R = 4$D real-valued vector, the quantum state $|x\rangle = x_1|00\rangle + x_2|01\rangle + x_3|10\rangle + x_4|11\rangle$ can be prepared by querying the classical memory structure and applying three controlled rotations.

Let us assume that $\mathbf{x}$ is normalized, namely $||\mathbf{x}||_2 = 1$. The quantum state $|x\rangle$ is created from the initial state $|0\rangle|0\rangle$ by querying the memory structure from the root to the leaf. The first rotation is applied on the first qubit, giving

$$(\cos\theta_1|0\rangle + \sin\theta_1|1\rangle)|0\rangle = \left(\sqrt{x_1^2+x_2^2}\,|0\rangle + \sqrt{x_3^2+x_4^2}\,|1\rangle\right)|0\rangle$$

where $\theta_1 := \tan^{-1}\sqrt{\frac{x_3^2+x_4^2}{x_1^2+x_2^2}}$. The second rotation is applied on the second qubit conditioned on the state of qubit 1. It gives

$$\sqrt{x_1^2+x_2^2}\,|0\rangle\,\frac{1}{\sqrt{x_1^2+x_2^2}}(|x_1|\,|0\rangle + |x_2|\,|1\rangle)+$$
$$\sqrt{x_3^2+x_4^2}\,|1\rangle\,\frac{1}{\sqrt{x_3^2+x_4^2}}(|x_3|\,|0\rangle + |x_4|\,|1\rangle)$$

The last rotation loads the signs of coefficients conditioned on qubits 1 and 2. In general, an R-dimensional real-valued vector needs to be stored in a classical memory structure with $\lceil\log_2 R\rceil + 1$ layers. The data vector can be loaded into a quantum state using $\mathcal{O}(\log_2 R)$ nontrivial controlled rotations.

Acknowledgements

This research was supported by the BMBF funded project Machine Learning with Knowledge Graphs, and Siemens Corporate Technology.

Conflict of Interest

The authors declare no conflict of interest.

Keywords

inference on relational database, knowledge graphs, quantum acceleration, representation learning, variational quantum circuit

Received: September 5, 2018
Revised: December 12, 2018
Published online:

[1] M. Nickel, V. Tresp, H. P. Kriegel, presented at *Proc. of the 28th Int. Conf. on Machine Learning*, Bellevue, WA, June 28–July 02, **2011**.

[2] B. Yang, W. t. Yih, X. He, J. Gao, L. Deng, *arXiv preprint arXiv:1412.6575*, **2014**.

[3] T. Trouillon, J. Welbl, S. Riedel, É. Gaussier, G. Bouchard, in *Proc. of the 33rd Int. Conf. on Machine Learning (ICML 2016)*, New York, **2016**, pp. 2071–2080.

[4] M. Nickel, L. Rosasco, T. A. Poggio, in *Proc. of the 30th AAAI Conf. on Artificial Intelligence*, Phoenix, AZ, **2016**, pp. 1955–1961.

[5] L. R. Tucker, *Psychometrika* **1966**, *31*, 279.

[6] M. Nickel, V. Tresp, in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, Springer, **2013**, pp. 272–287.

[7] M. Schuld, A. Bocharov, K. Svore, N. Wiebe, *arXiv preprint arXiv:1804.00633*, **2018**.

[8] I. Kerenidis, A. Prakash, *arXiv preprint arXiv:1603.08675*, **2016**.

[9] H. Buhrman, R. Cleve, J. Watrous, R. De Wolf, *Phys. Rev. Lett.* **2001**, *87*, 167902.

[10] J. C. Garcia-Escartin, P. Chamorro-Posada, *Phys. Rev. A* **2013**, *87*, 052330.

[11] P. Chamorro-Posada, J. C. Garcia-Escartin, *arXiv preprint arXiv:1706.06564*, **2017**.

[12] A. M. Childs, N. Wiebe, *arXiv preprint arXiv:1202.5822*, **2012**.

[13] A. Asuncion, D. Newman, UCI Machine Learning Repository, **2007**.

[14] K. Toutanova, D. Chen, in *Proc. of the 3rd Workshop on Continuous Vector Space Models and Their Compositionality*, Association for Computational Linguistics, **2015**, pp. 57–66.

[15] T. Dettmers, P. Minervini, P. Stenetorp, S. Riedel, *arXiv preprint arXiv:1707.01476*, **2017**.

[16] K. Leetaru, P. A. Schrodt, presented at *ISA Annual Convention*, San Francisco, CA, April 3–6, **2013**.

[17] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, *Advances in Neural Information Processing Systems 26 (NIPS 2013)*, **2013**, pp. 2787–2795.

[18] By the time of finishing this project, none of the quantum computing cloud platforms provide fully tunable entangled qubits.

[19] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, *Proc. of 12th USENIX Conf. on Operating Systems Design and Implementation (OSDI '16)*, **2016**, pp. 265–283.

[20] Y. Ma, M. Hildebrandt, S. Baier, V. Tresp, Holistic Representations for Memorization and Inference, *Proc. of 34th Conf. on Uncertainty in Artificial Intelligence (UAI 2018)*, Monterey, CA, **2018**, pp. 403–413.

[21] Note that *best known* models might employ arbitrarily complex structures, for example, convolutional or recurrent neural networks. For comparison, we provide the number of trainable parameters in different models. Note that the number of trainable parameters depends not only on the model structure but also the number of different entities and predicates in the dataset. In the case of the FB15k-237 dataset, the state-of-the-art model described in ref. [15] contains

5.05 parameters, and the RESCAL model contains 1.89M parameters, while the fQCE model contains 1.06M parameters.

[22] M. Schuld, I. Sinayskiy, F. Petruccione, *Quantum Inf. Process.* **2014**, *13*, 2567.

[23] E. Torrontegui, J. J. Garcia-Ripoll, *arXiv preprint arXiv:1801.00934*, **2018**.

[24] G. Huang, Y. Sun, Z. Liu, D. Sedra, K. Q. Weinberger, in *European Conference on Computer Vision*, Springer, **2016**, pp. 646–661.

[25] S. Hochreiter, J. Schmidhuber, *Neural Comput.* **1997**, *9*, 1.

[26] L. van der Maaten, G. Hinton, *J. Mach. Learn Res.* **2008**, *9*, 2579.

[27] G. Brassard, P. Hoyer, M. Mosca, A. Tapp, *Contemp. Math.* **2002**, *305*, 53.

[28] L. K. Grover, in *Proceedings of the 28th Annual ACM Symp. on Theory of Computing*, ACM, **1996**, pp. 212–219.

[29] C. H. Bennett, E. Bernstein, G. Brassard, U. Vazirani, *SIAM J. Comput.* **1997**, *26*, 1510.

[30] The empirical distributions are obtained in the following way: Given a semantic triple (s, p, o) in the test dataset, we calculate the value functions η_{spe_i} and $\eta_{e_i\mathrm{po}}$, with $i = 1, \ldots, N_e$.

[31] A. Prakash, *Ph.D. Thesis*, UC Berkeley, **2014**.