

Incorporating Content-Based Features into Quantum Knowledge Graph Embeddings

Jonas Hendl¹, Michael Färber²

¹Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

²ScaDS.AI, TU Dresden, Dresden, Germany

Abstract

Link prediction in relational data structures, such as knowledge graphs, plays a crucial role in maintaining up-to-date and accurate information. While classical approaches typically leverage either the graph's connectivity or associated textual descriptions (e.g., labels, definitions), recent quantum models have focused predominantly on structural aspects, leaving the potential of textual information largely unexplored. This paper presents the first quantum link prediction framework that combines both structural and textual modalities. We first generate classical text embeddings, apply dimensionality reduction, and encode them into quantum circuits using two complementary strategies: an Amplitude Encoding Model for high-dimensional fidelity and an Angle Encoding Model optimized for gate efficiency. Experimental results on standard benchmark datasets demonstrate that incorporating textual features in quantum architectures is not only feasible but also enhances the predictive performance of quantum link prediction models.

Keywords

Knowledge Graphs, Link Prediction, Quantum Machine Learning

1. Introduction

Knowledge graphs (KGs) are structured representations of entities and the relations between them (e.g., *Mount Everest*–*named after*–*George Everest*) and serve as the foundation for numerous applications, including semantic search and question answering [1, 2]. While these graphs are crucial for organizing and retrieving knowledge, real-world KGs are almost always incomplete [3], leading to an active research focus on *link prediction* – i.e., inferring missing edges (triples) to keep the graph current and accurate.

Classical knowledge graph embedding (KGE) methods have traditionally addressed link prediction by modeling the topological structure of the graph. However, advances in natural language processing have shown that incorporating textual attributes—such as labels and descriptions—into KGE models significantly boosts predictive performance [4, 3, 5, 6]. Notably, four of the top five models on the WN18RR dataset (by Hits@1) leverage such natural language descriptions (NLD) [7].

In parallel, quantum computing has emerged as a promising avenue for link prediction, with studies suggesting advantages in training and inference efficiency [8]. Several quantum-based KGE models have been proposed [9, 8], along with approaches inspired by quantum mechanics [10, 11, 12, 13]. However, these methods focus mainly on the topological aspects of KGs and have not integrated textual content—largely due to the lack of effective techniques for encoding text in the quantum setting.

In this paper, we address this gap by proposing the *first quantum link prediction framework* that fuses both the topology and the textual attributes of KGs in the context of quantum computing. Specifically, we employ a classical preprocessing pipeline to generate and dimensionally reduce text embeddings for subjects, predicates, and objects of KG triples (see Figure 3), allowing them to be seamlessly fed into two novel quantum architectures: an *Amplitude Encoding Model* for high-dimensional data representation and an *Angle Encoding Model* designed for gate efficiency. Our experiments on standard benchmark datasets (WN18RR and FB15k-237) demonstrate that incorporating textual embeddings in a quantum

AIQxQIA 2025: International Workshop on AI for Quantum and Quantum for AI | co-located with ECAI 2025, Bologna, Italy

✉ jonas.hendl@student.kit.edu (J. Hendl); michael.farber@tu-dresden.de (M. Färber)

ORCID 0000-0001-5458-8645 (M. Färber)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

setup not only yields improved predictive accuracy but also highlights the practical feasibility of quantum-enhanced, content-driven link prediction.

To summarize, our contributions are as follows:

- We introduce the first quantum link prediction framework that incorporates natural language descriptions alongside structural information, bridging a critical gap between classical and quantum approaches.
- We design a robust classical preprocessing pipeline using state-of-the-art language models and dimensionality reduction to generate entity and relation embeddings for quantum processing.
- We develop two complementary quantum architectures – amplitude encoding and angle encoding – that significantly reduce resource requirements and improve performance.
- We evaluate our models on the widely used link prediction benchmarks WN18RR and FB15k-237, showing up to 15% performance gains compared to baselines and highlighting the efficiency of our approach.

The paper is structured as follows: Sec. 2 reviews related work on link prediction and quantum KGE models. Sec. 3 outlines our methodology. Sec. 4 presents the evaluation, and Sec. 5 concludes the paper.

2. Related Work

In this section, we first present language models leveraging natural language descriptions (NLD) for link prediction (Sec. 2.1), followed by a review of quantum KGE models (Sec. 2.2), distinguishing between quantum-inspired and true quantum approaches.

2.1. Language Models for Link Prediction

Integrating natural language descriptions (NLD) into link prediction has considerably improved KGE performance. KG-BERT [14] first demonstrated that treating structured knowledge graph data as natural language allows pre-trained language models to generate contextual embeddings for link prediction, achieving state-of-the-art results by fine-tuning BERT to score triples.

Subsequent models further optimized this integration. SimKGC [6] introduced contrastive learning with diverse negative sampling strategies. This approach improved MRR on the WN18RR dataset by +19%. KERMIT [3] further refined the integration of textual features by generating more coherent descriptions using large language models, leading to further gains in Hits@1.

Other approaches, such as KEPLER [15] and BLP [16], employ independent embedding architectures. KEPLER jointly optimizes a KGE scoring function and a masked language modeling loss, while BLP encodes entity descriptions using BERT to learn inductive embeddings without fine-tuning, achieving strong generalization to unseen entities.

SimKGC and KERMIT currently provide the strongest results of language models for link prediction. Their success highlights the effectiveness of natural language description-based embeddings, motivating our exploration of integrating textual information into quantum knowledge graph embedding models.

2.2. Quantum Knowledge Graph Embedding Models

We can differentiate between quantum-inspired and true-quantum models.

2.2.1. Quantum-Inspired Models

Quantum-inspired models incorporate principles from quantum mechanics into KGE without requiring quantum hardware or claiming a quantum advantage. They primarily leverage quantum logic and algebraic structures to improve link prediction and reasoning.

Embed to Reason (E2R) [11] introduced a quantum logic-based embedding approach that maps entities and predicates into quantum-inspired vector spaces, preserving ontological hierarchies and relational

constraints. IQCE [13] extended this by improving generalization to unseen data and optimizing training efficiency. Further refinements merged quantum logic with classical embedding techniques: QLogicE [10] integrated E2R with TransE, achieving unprecedented results, particularly on the FB15k-237 dataset, while QIQE [12] combined E2R with quaternion embeddings, further improving MRR by 81.5%. Despite these gains, quantum-inspired models remain purely classical, relying on mathematical abstractions rather than quantum computation.

2.2.2. True Quantum Knowledge Graph Embedding Models

Among true quantum models, only two architectures have been proposed: Tensor Singular Value Decomposition (Tensor SVD) [9] and a variational circuit-based approach [8]. The latter is further divided into Quantum Circuit Embedding (QCE) and Fully Parameterized Quantum Circuit Embedding (FQCE). The key difference is that FQCE parameterizes all gates in the quantum circuit, which in principle allows for greater expressiveness, while QCE relies partly on classical memory writes. Both models construct quantum states for subject and object entities and measure their similarity via the SWITCH-Test (see below). In contrast, Tensor SVD applies quantum tensor singular value decomposition to utilize the graph structure.

All existing true quantum models operate in simulation, with only FQCE later tested on a quantum backend Kurokawa et al. [17]. Further refinements introduced quantum-specific training strategies, such as quantum negative sampling [18]. However, no fundamental advancements in architecture have been proposed beyond these two models. Notably, none of the implementations are publicly available, though Ma et al.[8] provided access to FQCE upon request.

Performance on FB15k-237 and WN18RR shows that true quantum models yield the lowest Hits@10 scores (32.3–37.8%), falling short of classical KGE models (40–50%) and far behind quantum-inspired models like QLogicE, which report implausibly high gains [10]. Unlike previous quantum models that encode only graph structure, our approach integrates natural language descriptions into quantum embeddings. To date, no true quantum KGE model has processed textual input—a gap already addressed in quantum GNNs [19]. This shows the need for hybrid quantum-classical KGE methods to combine relational and semantic signals.

3. Approach: Integrating Knowledge Graph Content into Quantum Models

Our approach introduces a hybrid quantum-classical framework for link prediction that effectively integrates textual information from knowledge graphs into a quantum knowledge graph embedding (KGE) model. It comprises two key components: (i) **a classical preprocessing pipeline** that extracts and encodes natural language descriptions (NLDs) into embeddings, and (ii) **a quantum module** that process and further scores these embeddings. In the following, we provide a detailed explanation of each component.

3.1. Classical Preprocessing Pipeline

For the preprocessing stage, we employed a classical preprocessing pipeline rather than a quantum alternative, primarily due to current resource limitations and the significant performance disparity between classical and quantum language models. As discussed in Section 2, classical models presently offer superior maturity, scalability, and accuracy, making them a more practical choice for generating high-quality textual embeddings. As illustrated in Figure 1, the preprocessing pipeline consists of the following steps:

1. Retrieve the natural language description for each triple.
2. Transform these descriptions into vectors using a Transformer model.
3. Reduce the dimensionality of these vectors to match the quantum model’s input constraints.

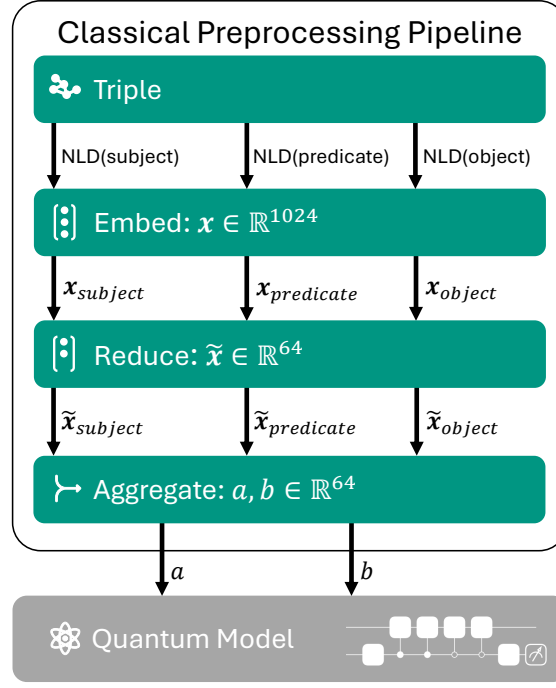


Figure 1: Preprocessing pipeline from triple to two vectors **a** and **b**, which we score in our quantum model.

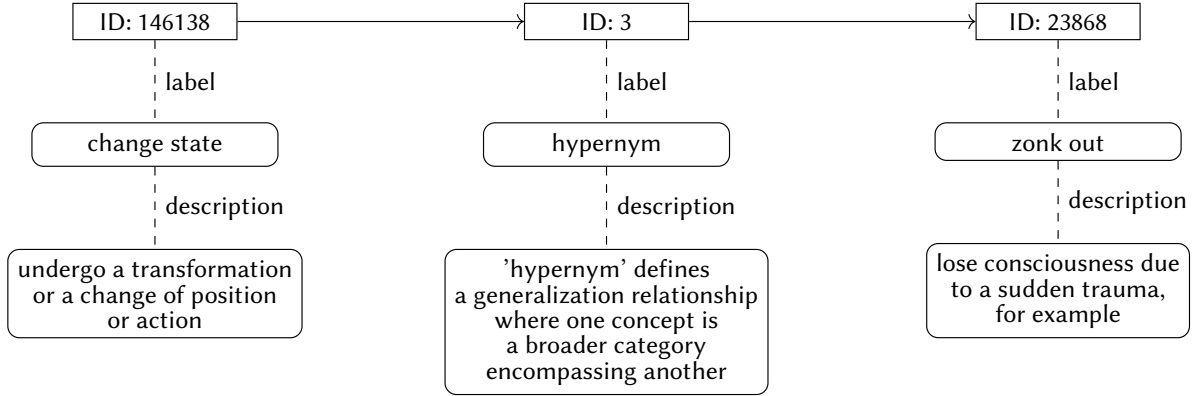


Figure 2: Example triple from the WN18RR dataset with IDs, labels, and descriptions.

4. Aggregate the subject, predicate, and object vectors into two vectors, conforming to the quantum model’s architecture.

The two resulting vectors are then passed to the quantum model, which scores their similarity to evaluate the triple. Each of the aforementioned steps is described in detail below.

3.1.1. Retrieve Natural Language Descriptions from the Knowledge Graph

For each triple, we retrieve the associated label or description from the knowledge graph. Depending on availability, we either use the label alone or concatenate the label and description into a single string for embedding. Figure 2 shows an example from the WN18RR dataset, where the ID “146138” corresponds to the label “change state” and the description “undergo a transformation or a change of position or action.”

3.1.2. Embedding Vector Generation

To capture semantic similarity, the natural language descriptions are embedded into vectors such that semantically similar texts yield similar vectors. This task of semantic text similarity is well-studied, as shown by the Massive Text Embedding Benchmark [20], where the top-performing model on the English sentence similarity leaderboard is jina-embeddings-v3 with an embedding dimension of 1,024 [21]. Therefore, we adopt jina-embeddings-v3 as our embedding model.

The model builds upon the XLM-RoBERTa architecture [22] by incorporating several key adaptations. Rotary Position Embeddings extend its input capacity from 512 to 8192 tokens, while five LoRA adapters fine-tune lightweight, low-rank parameters for task-specific performance – one of which is trained for semantic text similarity. Furthermore, the model employs Matryoshka Embeddings, which allow flexible output dimensions (ranging from 32 to 1,024).

However, due to the constraints of our quantum models, which require low-dimensional inputs (typically between 6 and 32 dimensions), we cannot use the full or simply truncated embeddings directly; instead, we next perform a dimensionality reduction step.

3.1.3. Reduction of the Vector Dimensionality

The original 1,024-dimensional embeddings must be reduced to match the input size of the quantum model. Although we are not theoretically limited to a specific dimension, simulation time doubles with each additional qubit. From preliminary runs, we found that embedding sizes (i.e., dimensions) between 6 and 64 are feasible for a comprehensive hyperparameter search.

We reduce the embeddings to the desired dimensions using UMAP, which better preserves the local and global structure than PCA or t-SNE [23] and yields denser clusters of similar points [24].

3.1.4. Aggregation Functions

Based on the steps described so far, we create three embedding vectors $\mathbf{s}, \mathbf{p}, \mathbf{o} \in \mathbb{R}^d$ for the subject, predicate, and object of a knowledge graph triple. However, our quantum model requires exactly two input vectors. To resolve this, we aggregate the three vectors into two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$. The object vector is directly used as the second input, i.e., $\mathbf{b} = \mathbf{o}$. For the first input vector \mathbf{a} , which combines the subject and predicate, we explore several techniques:

Addition (Add) Element-wise addition of the subject and predicate vectors:

$$\mathbf{a} = \mathbf{s} + \mathbf{p},$$

where $\mathbf{s}, \mathbf{p}, \mathbf{a} \in \mathbb{R}^d$.

Weighted Addition (WAdd) Here, we assign weights to control the relative importance of the subject and predicate:

$$\mathbf{a} = c_s \cdot \mathbf{s} + c_p \cdot \mathbf{p},$$

with $c_s, c_p \in \mathbb{R}$.

Average We compute the element-wise average of the subject and predicate vectors:

$$\mathbf{a} = \frac{\mathbf{s} + \mathbf{p}}{2}.$$

When using amplitude encoding, normalizing \mathbf{a} with the l^2 -norm renders averaging equivalent to addition.

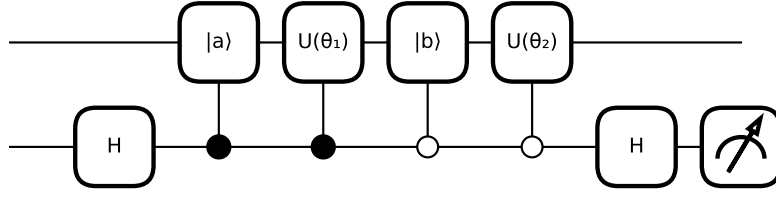


Figure 3: SWITCH-Test with ancilla bit (bottom) and main qubit (top), the latter is in practice multiple qubits, often around 6. Two states, which are superposed by the SWITCH-Test, are created by $U(\theta_1)U(\gamma_a)|\mathbf{0}\rangle$ and $U(\theta_2)U(\gamma_b)|\mathbf{0}\rangle$, where θ_1 and θ_2 are learned. The parameters γ_a and γ_b are based on the embedding vectors determined by our classical preprocessing pipeline. Combined with an encoding method, they define the states $|a\rangle$ and $|b\rangle$.

Neural Combinator This method applies a single affine transformation followed by a non-linear activation to combine the vectors:

$$\mathbf{a} = f\left(\mathbf{W} \cdot \begin{bmatrix} \mathbf{s} \\ \mathbf{p} \end{bmatrix} + \mathbf{b}\right),$$

where $\begin{bmatrix} \mathbf{s} \\ \mathbf{p} \end{bmatrix} \in \mathbb{R}^{2d}$, $\mathbf{W} \in \mathbb{R}^{d \times 2d}$, $\mathbf{b} \in \mathbb{R}^d$, and $f : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is an activation function. Since we do not optimize \mathbf{W} and \mathbf{b} (as optimization is confined to the quantum model parameters), this approach is not feasible for our use case.

Dot Product (*DotProd*) Inspired by DistMult [25], we can combine \mathbf{s} and \mathbf{p} by computing their dot product:

$$\mathbf{a} = \langle \mathbf{s}^T, \mathbf{p} \rangle,$$

where $\mathbf{s}, \mathbf{p} \in \mathbb{R}^d$.

Concatenation (*Concat*) Unlike previous methods that aggregate \mathbf{s} and \mathbf{p} , concatenation preserves their individual components. In this approach, we form:

$$\begin{aligned} \mathbf{a} &= \mathbf{s} \oplus \mathbf{p}, \\ \mathbf{b} &= \mathbf{o} \oplus \mathbf{p}, \end{aligned}$$

where $\mathbf{s}, \mathbf{o}, \mathbf{p} \in \mathbb{R}^d$, $\mathbf{a}, \mathbf{b} \in \mathbb{R}^{2d}$, and \oplus denotes concatenation. We use concatenation to ensure that both input vectors to the quantum model have the same dimensionality.

The average and concatenation methods were suggested by Garten et al. [26], while the addition and weighted addition approaches were introduced by Mitchell et al.[27]. A neural aggregator similar to ours was proposed by Maurya et al. [28].

3.2. Quantum Model

In the following, we describe the loading, development, and scoring of quantum states. Our approach connects established encoding methods with the FQCE model, enabling the integration of classical content into a quantum KGE model. The overall circuit is shown in Fig. 3.

3.2.1. Encoding a Vector as a Quantum State

To process the aggregated embedding vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^d$ in a quantum circuit, we need to encode them as quantum states in a way that preserves their geometric properties while minimizing quantum resource

requirements. Given our model constraints – ideally fewer than 6 qubits for feasible simulation and a circuit depth of approximately $\log(d)$ – the encoding needs to be efficient. Additionally, the mapping from embeddings to quantum states must be injective to ensure that semantically distinct vectors remain distinguishable.

We adopt both amplitude and angle encoding techniques. Amplitude encoding efficiently represents high-dimensional embeddings. However, it requires $2^{q+2} - 5$ rotational and $2^{q+2} - 4q - 4$ C-NOT gates, where $q = \log(d)$. For instance, a 64-dimensional vector would need 6 qubits with 251 rotational and 228 C-NOT gates. In simulation, the state is prepared directly, circumventing the explicit gate count. In contrast, angle encoding, though less qubit-efficient, uses a single gate per embedding dimension. To maintain injectivity, we scale the vectors appropriately.

3.2.2. Variational Quantum Circuit

We implement two variational circuit designs: *strongly entangling layers* for their expressiveness and a *simplified 2-design* for their robustness against barren plateaus and efficient circuit depth usage. The number of variational layers is treated as a hyperparameter during training.

3.2.3. The SWITCH-Test

We adopt the SWITCH-Test architecture from the FQCE model over alternatives such as Tensor SVD or a pure variational design with concatenated inputs. This choice is driven by how we integrate the natural language descriptions, as the selected integration method directly influences the input structure and subsequent design considerations.

Integration of Natural Language Embeddings To incorporate content efficiently, embedding vectors must be generated from the natural language descriptions ahead of time. Existing language-based knowledge graph embedding models generally fall into one of three categories: triple embedding architectures, translational embedding architectures, and independent embedding architectures.

Triple embedding architectures concatenate the natural language descriptions (NLD) of a triple into a single token sequence and embed it in one call. However, if a triple is corrupted, a new embedding must be computed, leading to significant overhead – for example, a single true triple in the WN18RR dataset can yield over 80,000 corrupted triples. Similarly, translational embedding architectures combine subject-predicate and predicate-object natural language descriptions and face the same computational burden when handling corrupted triples. In contrast, independent embedding architectures embed the subject, predicate, and object separately, allowing each entity and relation to be embedded only once. This approach minimizes computational overhead and is ideal for precomputation. We follow this paradigm for generating our embedding vectors, which seamlessly integrates with the SWITCH-Test architecture.

SWITCH-Test vs. Tensor SVD We aim for a true quantum model that can run on quantum hardware. Only three true quantum models – Quantum Circuit Embedding (QCE), FQCE, and Tensor SVD – meet this criterion. Both QCE and FQCE utilize the SWITCH-Test to compare two quantum states, and we consider them as SWITCH-Test architectures. The primary difference between QCE and FQCE lies in the encoding of the initial quantum states of the subject and object, which we adapt for integrating language embeddings. Therefore, we focus on comparing Tensor SVD and FQCE, the latter offering a quantum advantage over QCE.

Notably, Tensor SVD has not been implemented with quantum computing libraries (e.g., Qiskit or PennyLane), limiting its execution to classical simulation and lacking concrete circuit depictions – only mathematical formulations exist, which adds significant implementation overhead. In contrast, FQCE has well-documented quantum circuits that have been implemented using PennyLane, demonstrating practical feasibility on quantum hardware. Moreover, FQCE has garnered attention from the research community, resulting in additional studies that further validate its design.

Table 1

Gate Requirements for SWITCH-Test vs. Alternative Design

| Design | Rotational Gates | CNOT Gates |
|------------------------|---|--|
| SWITCH-Test | $2 \cdot (2^{\lceil \log_2(d) \rceil + 2} - 5)$ | $2 \cdot (2^{\lceil \log_2(d) \rceil + 2} - 4\lceil \log_2(d) \rceil - 4)$ |
| Variational Classifier | $2^{\lceil \log_2(3) + \log_2(d) \rceil + 2} - 5$ | $2^{\lceil \log_2(3) + \log_2(d) \rceil + 2} - 4\lceil \log_2(3) + \log_2(d) \rceil - 4$ |

In terms of performance on the FB15k-237 dataset, Tensor SVD slightly outperforms FQCE in Hits@3 (22.5 vs. 19.8) and Hits@10 (37.7 vs. 33.7), but it is significantly worse in mean rank (415 vs. 236). For these reasons, we build our model on the foundations of FQCE.

SWITCH-Test vs. a Pure Variational Quantum Classifier Design We choose the SWITCH-Test architecture, which generates the superposition of two quantum states and scores their similarity (e.g., $\Re\langle op|sp \rangle$). Although this design scores only two states – while a triple contains three elements – it remains more resource-efficient than a pure variational quantum classifier.

Table 1 shows that the pure variational classifier requires significantly more gates due to the additional factor from encoding the concatenated vector. Since $2^{\lceil \log_2(3) \rceil} > 2^1$, the gate count is substantially higher. For these reasons, the SWITCH-Test architecture is preferred for its flexibility in encoding methods and overall resource efficiency.

3.2.4. Scoring the Similarity of Quantum States

Using amplitude encoding and strongly entangling layers, we prepare two quantum states that represent the aggregated embedding vectors. Their similarity is evaluated via the SWITCH-Test [29], as illustrated in Fig. 3. This method measures the similarity by sampling a single ancilla qubit.

We begin by applying a Hadamard gate to the ancilla qubit initially in $|0\rangle_A$:

$$H|0\rangle_A = \frac{1}{\sqrt{2}}(|0\rangle_A + |1\rangle_A).$$

The remaining n qubits are initialized in the zero state $|\mathbf{0}\rangle$ and then evolved by unitary operations U_2 and U_1 conditioned on the ancilla state, yielding:

$$\frac{1}{\sqrt{2}}(|0\rangle_A U_2 |\mathbf{0}\rangle + |1\rangle_A U_1 |\mathbf{0}\rangle).$$

A second Hadamard is applied to the ancilla, transforming the state to:

$$\frac{1}{2} \left[|0\rangle_A (U_2 |\mathbf{0}\rangle + U_1 |\mathbf{0}\rangle) + |1\rangle_A (U_2 |\mathbf{0}\rangle - U_1 |\mathbf{0}\rangle) \right].$$

Substituting $|op\rangle = U_2 |\mathbf{0}\rangle$ and $|sp\rangle = U_1 |\mathbf{0}\rangle$ gives:

$$\frac{1}{2} \left[|0\rangle_A (|op\rangle + |sp\rangle) + |1\rangle_A (|op\rangle - |sp\rangle) \right].$$

The probability of measuring the ancilla in state $|0\rangle_A$ is then:

$$\begin{aligned} Pr(|0\rangle_A) &= \left\| \frac{1}{2}(|op\rangle + |sp\rangle) \right\|^2 \\ &= \frac{1}{4} (\langle op|op\rangle + \langle op|sp\rangle + \langle sp|op\rangle + \langle sp|sp\rangle) \\ &= \frac{1}{4} (2 + \langle op|sp\rangle + \overline{\langle op|sp\rangle}) \\ &= \frac{1}{2} + \frac{1}{2} \Re\langle op|sp\rangle, \end{aligned}$$

Table 2

Number of gates required for Angle Encoding and Amplitude Encoding models for a circuit with q qubits.

| | Angle Model | Amplitude Model |
|-------------------------------|-----------------|--------------------|
| Real Dimension | q | 2^q |
| Rotational Gates (Encoding) | $q + 1$ | $2^{q+2} - 5$ |
| C-NOT Gates (Encoding) | – | $2^{q+2} - 4q - 4$ |
| Rotational Gates (Processing) | $2(q - 1)L + q$ | qL |
| C-NOT Gates (Processing) | $2(q - 1)L$ | qL |

where $\Re\langle op|sp\rangle$ represents the real part of the inner product between $|op\rangle$ and $|sp\rangle$.

Defining the similarity score as $\eta_{spo} = \Re\langle op|sp\rangle$, we obtain the scoring function:

$$\eta_{spo} = 2 \Pr(|0\rangle_A) - 1.$$

This function estimates the similarity between the two quantum states by sampling the ancilla qubit.

3.2.5. Proposed Models

We propose two quantum knowledge graph embedding models: (1) the *Amplitude Encoding Model* and (2) the *Angle Encoding Model*. The Amplitude Encoding Model employs amplitude encoding to load the embedding vectors and uses strongly entangling layers in the variational circuit. Because amplitude encoding requires a high number of gates, strongly entangling layers are preferred over the simplified 2-design – which, as shown in Tab. 2, almost doubles the gate count – to keep the total number of gates manageable.

In contrast, the Angle Encoding Model requires only q gates for encoding, which allows us to use the simplified 2-design for the variational circuit. The Amplitude Encoding Model can encode a real vector from \mathbb{R}^{2^q} and utilizes more expressive gates that rotate around all available axes, making it more expressive. Meanwhile, the Angle Encoding Model encodes a vector from \mathbb{R}^q and relies solely on Y-rotational gates, which reduces its expressiveness but enhances robustness against barren plateaus and minimizes the overall gate count.

For our experiments, we combine each of these models with various aggregation functions and different levels of content integration, effectively yielding two groups of models.

4. Evaluation

In this section, we evaluate the performance of our proposed methods. We begin by introducing the baseline model and datasets in Sec. 4.1, followed by training considerations and strategies in Sec. 4.2. We then present our evaluation results in Sec. 4.3 and discuss their implications in Sec. 4.4.

4.1. Evaluation Settings

4.1.1. Baseline

As a baseline, we implement a multi-layer perceptron (MLP) that processes concatenated subject-predicate and object-predicate embeddings ($\mathbf{x} \in \mathbb{R}^{2d}$, with $d = 2^q$) through fully connected layers with batch normalization and non-linear activations, ultimately yielding a scalar output in the range $[-1, 1]$ consistent with our quantum circuit. Each layer employs variance-scaled weight initialization to maintain stable gradients. Batch normalization [30] is applied after each linear transformation. Dropout is implemented to prevent overfitting. The ReLU activation function is used in all hidden layers, while the final layer employs tanh to constrain outputs to $[-1, 1]$.

Table 3

Statistics of the FB15k-237 and WN18RR datasets.

| Dataset | # Triples | | | | # Entities | # Relations |
|-----------|-----------|--------|--------|---------|------------|-------------|
| | Train | Valid. | Test | Total | | |
| FB15k-237 | 272,115 | 17,526 | 20,438 | 310,079 | 14,505 | 237 |
| WN18RR | 86,835 | 2,824 | 2,924 | 92,583 | 40,559 | 11 |

4.1.2. Datasets

It is essential to compare our approach against the FQCE model, which was evaluated on the kinship, GDEL, WN18RR, and FB15k-237 datasets. Our goal is to determine whether the additional input from natural language descriptions improves performance. Since the kinship dataset comprises family member names with minimal exploitable natural language descriptions (NLD), and only QCE and FQCE have been evaluated on the GDEL dataset, we focus on WN18RR and FB15k-237 for their widespread use and compatibility with our approach.

The two datasets are semantically distinct and thus complementary for evaluation. WN18RR emphasizes lexical and hierarchical relations (e.g., hypernymy, meronymy), reflecting structured linguistic knowledge. In contrast, FB15k-237 comprises diverse factual relations (e.g., nationality, profession, location) across a wide schema of 237 relation types. Their combination enables a comprehensive assessment of a model’s ability to handle both abstract semantic structures and concrete real-world facts. As shown in Tab. 3, WN18RR includes 92,583 triples, distributed over 40,599 unique entities and 11 relations. Meanwhile, FB15k-237 consists of 310,079 triples spanning 14,505 unique entities and 237 relations.

We enhanced WN18RR by extracting natural language labels and descriptions from the NLTK library based on entity IDs. Since the 11 relations originally had only labels, we generated descriptions using a prompt-based approach with manual refinement. For FB15k-237, we used descriptions from [31] and retained only the original relation labels.

4.2. Model Training Settings

Training Procedure We train our models by corrupting positive triples using negative sampling. Both positive and negative samples are shuffled and grouped into mini-batches. Parameters are updated using the Adam optimizer.

Hyperparameter Optimization General hyperparameters (e.g., batch size, learning rate, dropout rate, negative samples per positive) and quantum-specific parameters (e.g., number of qubits, circuit depth) are tuned using Optuna with Tree-structured Parzen Estimator (TPE) sampling. Preliminary experiments tested batch sizes in $\{64, 128, 256, 512, 1024\}$ (with 128–256 performing best) and qubit counts in $\{3, 4, 5\}$, which were increased to 6 for the main study. Early stopping is applied with a patience of 5 epochs [32].

Initialization of Variational Quantum Circuits Rotational gates are initialized using Xavier-based sampling [33]. Additionally, paired block initialization is employed, where two consecutive blocks are initialized such that their composition equals the identity (i.e., $U^\dagger U = I$) [34].

Loss Functions and Dropout We compare Binary Cross-Entropy (BCE) and Mean Squared Error (MSE) losses. The BCE loss maps scores to probabilities via a sigmoid function, while MSE treats scores continuously in $[-1, 1]$. Rotation Dropout replaces a fraction of the rotational gates with the identity. The dropout rate is optimized as a hyperparameter.

4.3. Evaluation Results

As summarized in Tab. 4, quantum approaches exhibit a substantial performance gap compared to both the purely language-based KERMIT model and even a minimal neural baseline. On FB15k-237,

Table 4

Performance of link prediction models: Angle, Amplitude, neural baseline, and quantum models (FQCE, QCE, Tensor SVD). L indicates label only, LD indicates label plus description.

| Model | Aggreg. | NLD | FB15k-237 | | | | WN18RR | | | |
|-------------|---------|-----|------------|---------------|-------------|-------------|-------------|---------------|-------------|-------------|
| | | | MR | MRR | Hits@3 | Hits@10 | MR | MRR | Hits@3 | Hits@10 |
| Angle | Add | L | 14203 | 0.0001 | 0.00 | 0.00 | 50883 | 0.0000 | 0.00 | 0.00 |
| Angle | WAdd | L | 13891 | 0.0001 | 0.00 | 0.00 | 19003 | 0.0019 | 0.00 | 0.41 |
| Angle | Concat | L | 14171 | 0.0003 | 0.00 | 0.01 | 20757 | 0.0029 | 0.03 | 0.48 |
| Angle | DotProd | L | 14317 | 0.0001 | 0.00 | 0.00 | 40281 | 0.0000 | 0.00 | 0.00 |
| Angle | Add | LD | 14427 | 0.0001 | 0.00 | 0.00 | 37450 | 0.0000 | 0.00 | 0.00 |
| Angle | WAdd | LD | 13931 | 0.0003 | 0.00 | 0.01 | 11973 | 0.0003 | 0.00 | 0.00 |
| Angle | Concat | LD | 13915 | 0.0003 | 0.01 | 0.01 | 7016 | 0.0015 | 0.00 | 0.00 |
| Angle | DotProd | LD | 14340 | 0.0001 | 0.00 | 0.00 | 36425 | 0.0000 | 0.00 | 0.00 |
| Amplitude | Add | L | 14031 | 0.0001 | 0.00 | 0.00 | 17736 | 0.0029 | 0.07 | 0.48 |
| Amplitude | WAdd | L | 14120 | <u>0.0002</u> | 0.00 | 0.00 | 16705 | 0.0093 | 0.17 | 1.64 |
| Amplitude | Concat | L | 14327 | <u>0.0002</u> | 0.00 | 0.01 | 16638 | <u>0.009</u> | 0.27 | 1.54 |
| Amplitude | DotProd | L | 14491 | 0.0001 | 0.00 | 0.00 | 17285 | 0.0015 | 0.06 | 0.39 |
| Amplitude | Add | LD | 14200 | 0.0001 | 0.00 | 0.00 | 19387 | 0.0001 | 0.00 | 0.00 |
| Amplitude | WAdd | LD | 14464 | 0.0001 | 0.00 | 0.00 | 12979 | 0.0003 | 0.00 | 0.00 |
| Amplitude | Concat | LD | 13895 | <u>0.0002</u> | 0.00 | 0.07 | 15308 | 0.0005 | 0.00 | 0.07 |
| Amplitude | DotProd | LD | 14318 | 0.0001 | 0.00 | 0.00 | 18158 | 0.0003 | 0.00 | 0.01 |
| Neural Base | Add | L | 7582 | <u>0.0002</u> | 0.00 | 0.00 | 8670 | 0.0480 | 4.10 | 11.35 |
| Neural Base | WAdd | L | 8868 | 0.0003 | 0.00 | 0.01 | 7814 | 0.0429 | 3.42 | 10.36 |
| Neural Base | Concat | L | 9845 | <u>0.0002</u> | 0.00 | 0.00 | 7599 | 0.0234 | 1.50 | 4.96 |
| Neural Base | DotProd | L | 9642 | <u>0.0002</u> | 0.00 | 0.00 | 8284 | 0.0252 | 1.40 | 6.12 |
| Neural Base | Add | LD | 9566 | <u>0.0002</u> | 0.00 | 0.00 | 5463 | 0.0135 | 0.07 | 0.96 |
| Neural Base | WAdd | LD | 10483 | <u>0.0002</u> | 0.00 | 0.00 | 2630 | 0.0216 | 0.89 | 3.73 |
| Neural Base | Concat | LD | 8436 | 0.0003 | 0.00 | 0.00 | 2707 | 0.0340 | 2.02 | 6.91 |
| Neural Base | DotProd | LD | 11132 | <u>0.0002</u> | 0.00 | 0.00 | 1549 | 0.0457 | 2.53 | 10.91 |
| FQCE | – | – | 236 | – | <u>19.8</u> | 33.7 | <u>2160</u> | – | 27.4 | 37.8 |
| QCE | – | – | <u>259</u> | – | 22.5 | <u>35.0</u> | 3655 | – | <u>19.5</u> | <u>32.3</u> |
| Tensor SVD | – | – | 415 | – | 22.5 | 37.7 | – | – | – | – |

KERMIT achieves an MRR of 0.359 and Hits@10 of 54.7 %, whereas the best quantum variant yields only 0.003 MRR and 0.07 % Hits@10. This disparity persists on WN18RR, with KERMIT reaching 0.700 MRR and 83.2 % Hits@10, while our top quantum model delivers just 0.0093 MRR and 1.64 % Hits@10. Notably, the simple neural reference model already outperforms every quantum approach on WN18RR – attaining 0.048 MRR and 11.35 % Hits@10 – highlighting that quantum embeddings currently cannot compete with either classical embedding techniques or language-driven methods.

A closer look at each dataset reinforces this conclusion. On FB15k-237, random guessing yields an expected mean rank (MR) of approximately 14,500; our best quantum configurations – $angle_{WAdd-L}$ (MR 13,891) and $amplitude_{Concat-LD}$ (MR 13,895, Hits@10 0.07 %) – improve only marginally over chance and fall far short of the FQCE benchmark (MR 236). The neural baseline ($neural_{Add-L}$, MR 7,582) narrows the gap but remains over an order of magnitude behind. On WN18RR, quantum models achieve lower MRs (best: $angle_{Concat-LD}$, MR 7,016) yet still trail classical quantum methods – QCE (MR 3,655) and FQCE (MR 2,160) – and the neural baseline (MR 1,549). These results make clear that, despite exploring amplitude versus angle encodings and various aggregation schemes, quantum models under current designs fail to approach the effectiveness of both simple neural and advanced language-based knowledge-graph completion methods.

Comparison Between Aggregation Functions Within the *angle* models, simple addition and dot product methods performed only marginally better than random guessing, with MRs of 50,833 and 37,450 for *angle_{Add-L}* and *angle_{Add-LD}*, respectively, while *angle_{DotProd-L}* and *angle_{DotProd-LD}* scored MRs of 40,281 and 36,425. Overall, the plain addition aggregation is consistently outperformed by weighted addition and concatenation. In fact, for the neural baseline, the dot product aggregation yields the best performance (e.g., *neural_{DotProd-LD}* achieved an MR of 1,549).

Comparison Between Angle and Amplitude Architectures On average, the *amplitude* models achieve lower MRs and better Hits@3 and Hits@10 scores than the *angle* models. The best Hits@10 for the *amplitude* architecture reached 1.64% compared to 0.048% for the best *angle* model. However, despite these differences, the best quantum model by MR is *angle_{Concat-LD}* (7,016) versus 12,979 for the best *amplitude* model (*amplitude_{WAdd-LD}*). Furthermore, the performance range across different aggregation functions is wider for the *angle* models than for the more consistent *amplitude* models.

Impact of Adding Descriptions Adding natural language descriptions generally improves MR across approximately 83% of the models. For instance, *angle_{WAdd}* improved from an MR of 19,003 to 11,973, *amplitude_{Concat}* from 16,705 to 12,979, and *neural_{DotProd}* from 8,284 to 1,549. However, some outliers (e.g., *amplitude_{Add}*) exhibited an increase in MR when descriptions were added. Conversely, the impact on Hits@10 is mixed: for quantum models, Hits@10 typically decreased, while for the neural baseline, certain aggregation functions benefited (e.g., *neural_{DotProd}* improved).

In summary, while adding descriptions clearly benefits MR, its effect on Hits@10 varies across models. Overall, the neural baseline outperforms the quantum models, and among the quantum variants, weighted addition and concatenation prove to be the most effective aggregation methods.

4.4. Discussion

4.4.1. Underparameterization of the Quantum Models

The baseline model outperformed random guessing on the FB15k-237 dataset and even surpassed the QCE and FQCE models on the MR metric, indicating that our preprocessing pipeline is effective. The limited performance of our quantum models therefore stems less from the preprocessing and more from the model architecture itself and its interaction with the pipeline. In our view, this highlights a central tension in current quantum machine learning: while conceptual designs can anticipate long-term advantages, today’s hardware and simulation constraints prevent these models from competing with strong classical baselines. We see our work primarily as a step toward exploring how textual content could be represented in future quantum knowledge graph embeddings once more expressive circuits become practical.

A key difference is that our model learns only two sets of parameters (θ_1 and θ_2), whereas FQCE optimizes distinct parameters for each subject, predicate, and object. For FB15k-237, this translates to roughly 10^4 different parameter sets for subjects and objects and 237 for predicates. This disparity is particularly problematic given the higher node degree in FB15k-237.

Model Performance on the FB15k-237 Dataset The poor performance of text-based methods on FB15k-237 is well-documented in the literature. Wang et al. [35] attribute this to the dataset’s density – each subject has an average out-degree of approximately 21, compared to only 2 in WN18RR – and Cao et al. [36] note that some triples cannot be inferred solely from the training data. These factors explain the minimal learning observed in our quantum models, particularly on FB15k-237, where the neural baseline also suffers from insufficient parameter scaling relative to the dataset’s complexity.

4.4.2. Model Differences

Robustness of the Amplitude Model We observe that amplitude models perform more consistently across aggregation functions and are less sensitive to changes in input content. Their ability to handle

higher-dimensional vectors appears beneficial, though this doesn't necessarily lead to better peak performance compared to angle models.

Additional Content Improves Performance We can see that adding descriptions improves MR across all model types – angle, amplitude, and baseline. For instance, $angle_{WAdd}$ improves from 19,003 to 11,973, and $amplitude_{Concat}$ from 16,705 to 12,979. The neural $DotProd$ model drops from 8,284 to 1,549. While Hits@10 sometimes decreases for quantum models, the consistent MR improvements suggest that even small amounts of added content are useful—despite the limited input sizes (6 dim. for angle models, 64 for amplitude).

Impact of Aggregation Functions The choice of aggregation function clearly influences performance. Simple addition consistently underperforms, while weighted addition and concatenation yield better results. In weighted addition, we typically see the subject receiving more weight, resulting in minimal modification to its vector. Concatenation also performs well, likely due to preserving the original inputs. The dot product works poorly for quantum models but remains strong in the neural baseline.

Quantum vs. Baseline Models Across the board, the neural baseline outperforms the quantum models. This is likely due to its larger number of trainable parameters and non-linear activations, which help model complex, non-linear interactions between entities and relations.

4.4.3. Quantum Speed-up: Analysis and Limitations

We build on work by Ma et al. [8], who show that FQCE achieves a quantum training speed-up under two assumptions: (1) data loading via QRAM scales as $\mathcal{O}(\log_2(n))$, and (2) circuit depth does as well. While such scaling is theoretically feasible [37], its practicality remains uncertain [38], and no implementation currently exists. If such a QRAM were available, similar speed-ups would apply to our models using amplitude encoding.

FQCE circuits require a depth of approximately $3 \log_2(n)$, based on three variational layers. Our models scale similarly, with hyperparameter tuning between 2 and 10 layers. In the best cases, fewer layers suffice; in the worst, more depth is used. This raises the question of whether such quantum speed-ups are worth the hardware overhead, especially since typical KGE embeddings use 10^2 – 10^4 dimensions. However, if embedding sizes or KG complexity grow, the advantage becomes more compelling. For example, 31 qubits already support an embedding space exceeding 10^9 dimensions.

5. Conclusion and Outlook

In this paper, we integrated the textual content of a knowledge graph into a quantum embedding model by embedding text into vectors and encoding them as quantum states – introducing a new way to incorporate rich semantic information into quantum representations. We proposed two quantum models with distinct properties and developed a classical preprocessing pipeline that effectively combines labels and descriptions. Evaluation showed that while our models were under-parameterized and performed worse than established quantum models, integrating additional descriptive content positively impacted performance – demonstrating the potential of content-enriched quantum embeddings.

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT in order to: Grammar and spelling check, Paraphrase and reword. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

- [1] L. Tian, X. Zhou, Y.-P. Wu, W.-T. Zhou, J.-H. Zhang, T.-S. Zhang, Knowledge graph and knowledge reasoning: A systematic review, *Journal of Electronic Science and Technology* 20 (2022) 100159. URL: <https://www.sciencedirect.com/science/article/pii/S1674862X2200012X>. doi:10.1016/j.jnlest.2022.100159.
- [2] Z. Xu, M. J. Cruz, M. Guevara, T. Wang, M. Deshpande, X. Wang, Z. Li, Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering, in: *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2024, pp. 2905–2909. URL: <http://arxiv.org/abs/2404.17723>. doi:10.1145/3626772.3661370.
- [3] H. Li, B. Yu, Y. Wei, K. Wang, R. Y. Da Xu, B. Wang, KERMIT: Knowledge Graph Completion of Enhanced Relation Modeling with Inverse Transformation, 2023. URL: <https://arxiv.org/abs/2309.14770v2>. doi:10.48550/arXiv.2309.14770.
- [4] X. Wang, Z. Chen, H. Wang, L. Hou U, Z. Li, W. Guo, Large Language Model Enhanced Knowledge Representation Learning: A Survey, 2024. URL: <https://arxiv.org/abs/2407.00936>. doi:10.48550/arXiv.2407.00936.
- [5] Q. Li, Y. Zhong, Y. Qin, MoCoKGC: Momentum Contrast Entity Encoding for Knowledge Graph Completion, in: *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Miami, Florida, USA, 2024, pp. 14940–14952. URL: <https://aclanthology.org/2024.emnlp-main.832/>. doi:10.18653/v1/2024.emnlp-main.832.
- [6] L. Wang, W. Zhao, Z. Wei, J. Liu, SimKGC: Simple Contrastive Knowledge Graph Completion with Pre-trained Language Models, in: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, volume 1, Dublin, Ireland, 2022, pp. 4281–4294. URL: <https://aclanthology.org/2022.acl-long.295/>. doi:10.18653/v1/2022.acl-long.295.
- [7] Meta AI, Papers with Code - WN18RR Benchmark (Link Prediction), 2024. URL: <https://paperswithcode.com/sota/link-prediction-on-wn18rr>.
- [8] Y. Ma, V. Tresp, L. Zhao, Y. Wang, Variational quantum circuit model for knowledge graphs embedding, *Advanced Quantum Technologies* 2 (2019). URL: <https://arxiv.org/abs/1903.00556>. doi:10.48550/arXiv.1903.00556.
- [9] Y. Ma, V. Tresp, Quantum machine learning algorithm for knowledge graphs, *ACM Transactions on Quantum Computing* 2 (2021). URL: <https://arxiv.org/pdf/2001.01077.pdf>. doi:10.1145/3467982.
- [10] P. Chen, Y. Wang, X. Yu, R. Feng, QLogicE: Quantum Logic Empowered Embedding for Knowledge Graph Completion, *Knowledge-Based Systems* 239 (2022) 107963. URL: <https://www.sciencedirect.com/science/article/pii/S0950705121010911>. doi:10.1016/j.knosys.2021.107963.
- [11] D. Garg, S. Ikbāl, S. K. Srivastava, H. Vishwakarma, H. Karanam, L. V. Subramaniam, Quantum Embedding of Knowledge for Reasoning, in: *Advances in Neural Information Processing Systems*, volume 32, 2019. URL: <https://proceedings.neurips.cc/paper/2019/hash/cb12d7f933e7d102c52231bf62b8a678-Abstract.html>. doi:10.5555/3454287.3454789.
- [12] L. Li, X. Zhang, Z. Jin, C. Gao, R. Zhu, Y. Liang, Y. Ma, Knowledge graph completion method based on quantum embedding and quaternion interaction enhancement, *Information Sciences* 648 (2023) 119548. URL: <https://www.sciencedirect.com/science/article/pii/S0020025523011337>. doi:10.1016/j.ins.2023.119548.
- [13] S. K. Srivastava, D. Khandelwal, D. Madan, D. Garg, H. Karanam, L. V. Subramaniam, Inductive quantum embedding, in: *Advances in Neural Information Processing Systems*, volume 33, Vancouver, BC, Canada, 2020, pp. 16012–16024. URL: <https://papers.nips.cc/paper/2020/hash/b87039703fe79778e9f140b78621d7fb-Abstract.html>. doi:10.5555/3495724.3497067.
- [14] L. Yao, C. Mao, Y. Luo, KG-BERT: BERT for Knowledge Graph Completion, 2019. URL: <https://www.semanticscholar.org/paper/KG-BERT%3A-BERT-for-Knowledge-Graph-Completion-Yao-Mao/31184789ef4c3084af930b1e0dede3215b4a9240>. doi:10.48550/arXiv.1909.03193.
- [15] X. Wang, T. Gao, Z. Zhu, Z. Zhang, Z. Liu, J. Li, J. Tang, KEPLER: A unified model for knowledge embedding and pre-trained language representation, *Transactions of the Association for Computational Linguistics* 9 (2021) 176–194. doi:10.1162/tac1_a_00360.

- [16] D. Daza, M. Cochez, P. Groth, Inductive Entity Representations from Text via Link Prediction, in: *Proceedings of the Web Conference 2021*, 2021, pp. 798–808. URL: <http://arxiv.org/abs/2010.03496>. doi:10.1145/3442381.3450141.
- [17] M. Kurokawa, P. R. Giri, K. Saito, Evaluating Variational Quantum Circuit Designs for Knowledge Graph Completion, in: *2022 IEEE International Conference on Quantum Computing and Engineering (QCE)*, IEEE, Broomfield, CO, USA, 2022, pp. 777–778. URL: <https://ieeexplore.ieee.org/document/9951236/>. doi:10.1109/QCE53715.2022.00115.
- [18] P. R. Giri, M. Kurokawa, K. Saito, Quantum Negative Sampling Strategy for Knowledge Graph Embedding with Variational Circuit, in: *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, volume 02, 2023, pp. 280–281. doi:10.1109/QCE57702.2023.10242.
- [19] S. Xu, F. Wilhelm-Mauch, W. Maaß, Quantum feature embeddings for graph neural networks, in: *Hawaii international conference on system sciences (HICSS)*, USA, 2024, pp. 7633–7642. URL: <https://hdl.handle.net/10125/107303>.
- [20] N. Muennighoff, N. Tazi, L. Magne, N. Reimers, MTEB: Massive Text Embedding Benchmark, 2023. URL: <http://arxiv.org/abs/2210.07316>. doi:10.48550/arXiv.2210.07316.
- [21] N. Muennighoff, N. Tazi, L. Magne, N. Reimers, MTEB Leaderboard - a Hugging Face Space by mteb, 2024. URL: <https://huggingface.co/spaces/mteb/leaderboard>.
- [22] A. Conneau, K. Khandelwal, N. Goyal, V. Chaudhary, G. Wenzek, F. Guzmán, E. Grave, M. Ott, L. Zettlemoyer, V. Stoyanov, Unsupervised Cross-lingual Representation Learning at Scale, 2020. URL: <http://arxiv.org/abs/1911.02116>. doi:10.48550/arXiv.1911.02116.
- [23] L. McInnes, J. Healy, N. Saul, L. Großberger, UMAP: Uniform Manifold Approximation and Projection, *Journal of Open Source Software* 3 (2018) 861. URL: <https://joss.theoj.org/papers/10.21105/joss.00861>. doi:10.21105/joss.00861.
- [24] D. Kobak, G. C. Linderman, Initialization is critical for preserving global data structure in both t-SNE and UMAP, *Nature Biotechnology* 39 (2021) 156–157. URL: <https://www.nature.com/articles/s41587-020-00809-z>. doi:10.1038/s41587-020-00809-z.
- [25] B. Yang, W.-t. Yih, X. He, J. Gao, L. Deng, Embedding Entities and Relations for Learning and Inference in Knowledge Bases, 2015. URL: <http://arxiv.org/abs/1412.6575>. doi:10.48550/arXiv.1412.6575.
- [26] J. Garten, K. Sagae, V. Ustun, M. Dehghani, Combining Distributed Vector Representations for Words, in: *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, Denver, Colorado, 2015, pp. 95–101. URL: <https://aclanthology.org/W15-1513/>. doi:10.3115/v1/W15-1513.
- [27] J. Mitchell, M. Lapata, Composition in distributional models of semantics, *Cognitive Science* 34 (2010) 1388–1429. doi:10.1111/j.1551-6709.2010.01106.x.
- [28] S. K. Maurya, X. Liu, T. Murata, Simplifying approach to Node Classification in Graph Neural Networks, 2021. URL: <http://arxiv.org/abs/2111.06748>. doi:10.48550/arXiv.2111.06748, arXiv:2111.06748 [stat].
- [29] P. Chamorro-Posada, J. C. Garcia-Escartin, The SWITCH test for discriminating quantum evolutions, *Journal of Physics A: Mathematical and Theoretical* 56 (2017) 355301. URL: <https://iopscience.iop.org/article/10.1088/1751-8121/acecc5>. doi:10.1088/1751-8121/acecc5.
- [30] S. Ioffe, C. Szegedy, Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift, 2015. URL: <http://arxiv.org/abs/1502.03167>. doi:10.48550/arXiv.1502.03167.
- [31] J. Villmow, villmow/datasets_knowledge_embedding, 2024. URL: https://github.com/villmow/datasets_knowledge_embedding.
- [32] D. Ruffinelli, S. Broscheit, R. Gemulla, You CAN teach an old dog new tricks! On training knowledge graph embeddings, in: *International Conference on Learning Representations (ICLR) 2020*, 2019, pp. 1–12. URL: <https://api.openreview.net/pdf/d8532341877a4ce6e4fee643e629af2957579771.pdf>.
- [33] M. Kashif, M. Rashid, S. Al-Kuwari, M. Shafique, Alleviating Barren Plateaus in Parameterized Quantum Machine Learning Circuits: Investigating Advanced Parameter Initialization Strategies, in: *2024 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2023, pp. 1–6.

URL: <https://arxiv.org/abs/2311.13218>. doi:10.23919/DATE58400.2024.10546644.

- [34] E. Grant, L. Wossnig, M. Ostaszewski, M. Benedetti, An initialization strategy for addressing barren plateaus in parametrized quantum circuits, *Quantum* 3 (2019) 214. URL: <http://arxiv.org/abs/1903.05076>. doi:10.22331/q-2019-12-09-214.
- [35] S. Sim, P. D. Johnson, A. Aspuru-Guzik, Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms, *Advanced Quantum Technologies* 2 (2019) 1900070. URL: <http://arxiv.org/abs/1905.10876>. doi:10.1002/qute.201900070.
- [36] Y. Cao, X. Ji, X. Lv, J. Li, Y. Wen, H. Zhang, Are Missing Links Predictable? An Inferential Benchmark for Knowledge Graph Completion, in: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, Online, 2021, pp. 6855–6865. URL: <https://aclanthology.org/2021.acl-long.534/>. doi:10.18653/v1/2021.acl-long.534.
- [37] A. Prakash, Quantum algorithms for linear algebra and machine learning., Ph.D. thesis, UC Berkeley, 2014. URL: <https://escholarship.org/uc/item/5v9535q4>.
- [38] S. Jaques, A. G. Rattew, QRAM: A Survey and Critique, 2023. URL: <http://arxiv.org/abs/2305.10310>. doi:10.48550/arXiv.2305.10310, arXiv:2305.10310.