Quantum Graph Transformer for NLP Sentiment Classification

Shamminuj Aktar
CCS-3 Information Sciences
Los Alamos National Laboratory
Los Alamos, NM, USA
saktar@lanl.gov

Andreas Bärtschi
CCS-3 Information Sciences
Los Alamos National Laboratory
Los Alamos, NM, USA
baertschi@lanl.gov

Abdel-Hameed A. Badawy
Klipsch School of Electrical and Computer Engineering
New Mexico State University
Las Cruces, NM, USA
badawy@nmsu.edu

Stephan Eidenbenz
CCS-3 Information Sciences
Los Alamos National Laboratory
Los Alamos, NM, USA
eidenben@lanl.gov

Abstract

Quantum machine learning is a promising direction for building more efficient and expressive models, particularly in domains where understanding complex, structured data is critical. We present the Quantum Graph Transformer (QGT), a hybrid graph-based architecture that integrates a quantum self-attention mechanism into the message-passing framework for structured language modeling. The attention mechanism is implemented using parameterized quantum circuits (PQCs), which enable the model to capture rich contextual relationships while significantly reducing the number of trainable parameters compared to classical attention mechanisms. We evaluate QGT on five sentiment classification benchmarks. Experimental results show that QGT consistently achieves higher or comparable accuracy than existing quantum natural language processing (QNLP) models, including both attention-based and non-attention-based approaches. When compared with an equivalent classical graph transformer, QGT yields an average accuracy improvement of 5.42% on real-world datasets and 4.76% on synthetic datasets. Additionally, QGT demonstrates improved sample efficiency, requiring nearly 50% fewer labeled samples to reach comparable performance on the Yelp dataset. These results highlight the potential of graph-based QNLP techniques for advancing efficient and scalable language understanding.

CCS Concepts: • **Computer systems organization** → **Quantum computing**; • **Computing methodologies** → **Natural language processing**; • **Hardware** → **Quantum technologies**.

Keywords: Quantum machine learning, Graph Transformer, Self-attention mechanism, Text classification

1 Introduction

Artificial Intelligence (AI) has become an integral part of modern technological advancement, influencing domains such as healthcare, finance, scientific discovery, and communication [28]. Natural Language Processing (NLP) plays a foundational role in AI by enabling machines to understand, interpret, and generate human language [26]. This capability allows NLP to power a variety of applications, from sentiment analysis to machine translation, conversational agents, and content moderation [22, 34]. Recent advancements in NLP models, such as GPT-4, have significantly improved machines' ability to generate coherent and context-aware text [24]. Despite their success, current NLP models still face substantial limitations. Training and inference for state-of-the-art models require large amounts of labeled data, access to high-performance computing (HPC) infrastructure, and the optimization of billions of parameters [12].

Quantum Machine Learning (QML) is a compelling framework for overcoming some of the inefficiencies associated with classical machine learning (ML) models [3, 5]. QML adopts a hybrid quantum-classical strategy, in which data is encoded into a high-dimensional Hilbert space and learning is performed using Parameterized Quantum Circuits (PQCs), which act as trainable layers analogous to neural network layers in classical architectures. These circuits leverage quantum properties such as superposition and entanglement to explore complex functions more efficiently [15, 30]. In the context of NLP, where understanding contextual relationships between tokens is essential, QML offers a promising direction for developing more efficient and expressive language models. Quantum-enhanced NLP (QNLP) models have the potential to achieve quantum advantages in sample efficiency, model accuracy, convergence speed, and generalization capability.

Recent efforts in QNLP have explored various approaches to integrate quantum principles into language modeling. Early models include the quantum bag-of-words approach, quantum support vector machines (QSVMs) and variational quantum classifiers employing amplitude encoding [1, 21]. More recent developments have focused on hybrid architectures such as quantum-enhanced LSTMs [33], self-attention based models [6, 18, 40], and ensemble based models like LEXIQL [31]. These approaches have shown promising results, indicating the potential of QNLP as a viable and scalable alternative to classical NLP models.

In classical ML, graph-based models have demonstrated strong performance in capturing the structure of complex data, including knowledge graphs, citation networks, and dependency trees in NLP [19, 38]. Graph neural networks (GNNs) represent inputs as graphs, where nodes represent entities (such as tokens in NLP) and edges capture syntactic or semantic relationships [29]. This graph-based representation enables models to effectively capture both hierarchical and contextual relationships within the data.

In this work, we propose the Quantum Graph Transformer (QGT), a novel hybrid quantum-classical model for sentiment classification. QGT represents each sentence as a fully connected graph of tokens and applies a modified quantum transformer convolution (QTransformerConv) layer to propagate information across the nodes. The self-attention mechanism in the QTransformerConv layer leverages PQCs to generate the query and key vectors necessary for computing attention scores between neighboring nodes. Specifically, each token node is encoded into a quantum state, and then PQCs parameterized by learnable weights are applied to extract the query and key representations. The resulting attention scores guide the flow of information across the graph, allowing the model to capture contextual dependencies among tokens in a quantum-enhanced feature space. We evaluate the QGT on five benchmark sentiment classification datasets and demonstrate that QGT efficiently learns contextual representations of token embeddings. We also compare the model performance on those benchmark datasets with other QNLP models and a classical graph transformer model.

Our main contribution are as follows:

1. We introduce a hybrid graph-based architecture that integrates a quantum self-attention mechanism into the message-passing framework for structured language modeling. Our design leverages PQCs to implement self-attention, requiring significantly fewer trainable parameters than traditional classical attention mechanisms while maintaining high expressibility.
2. We evaluate the proposed QGT model on five sentiment classification benchmarks, including three real-world datasets *i.e.* Yelp, IMDB, and Amazon [17] and two synthetic datasets *i.e.* MC and RP [21]. The QGT model consistently learns meaningful contextual representations and accurately predicts sentiment labels across all datasets. It outperforms or matches the accuracy of existing attention-based QNLP models and other QNLP baselines.
3. To further assess model performance, we compare QGT with an equivalent classical graph transformer model. Our results show that QGT achieves significantly higher accuracy, with an average improvement of 5.42% on three real-world datasets and 4.76% on two synthetic datasets.
4. Additionally, QGT demonstrates enhanced sample efficiency, requiring nearly 50% fewer training samples to achieve comparable accuracy on the Yelp dataset.

2 Background & Motivation

2.1 Natural Language Processing Tasks

Natural Language Processing (NLP) is a specialized domain within AI that aims to bridge the gap between human communication and machine understanding by enabling computational systems to process and generate natural language. Core NLP tasks include text classification, named entity recognition, question answering, and machine translation. These tasks are unified by the need to model complex syntactic and semantic relationships within and between sequences of text. In the sentiment classification task, the objective is to predict the underlying emotion or opinion expressed in a given sentence or document, typically as a categorical label such as positive, negative, or neutral. Sentiment classification is a widely studied problem due to its practical relevance in areas such as customer feedback analysis, user experience assessment, and automated review aggregation.

2.2 Classical NLP Models

In the early stages of natural language processing, models primarily relied on simple representations such as Bag-of-Words and N-gram features, which treated text as unordered collections of word counts without accounting for syntax or semantics [4, 14]. Subsequently, more linguistically informed approaches emerged, utilizing grammar-aware structures such as dependency trees and part-of-speech tags to capture syntactic relationships between words in a sentence. Although these structured models offered improved linguistic insight, they often struggled to generalize effectively across diverse language contexts. A major advancement came with the introduction of word embeddings such as Word2Vec [8] and GloVe [25], which represent words as continuous vectors in a dense embedding space. These embeddings capture semantic similarity based on word co-occurrence patterns, enabling models to generalize better across related linguistic constructs. Models such as recurrent neural networks (RNNs) [35], long short-term memory networks (LSTMs) [16], and gated recurrent units (GRUs) [7] leveraged these embeddings to process sequential data and learn contextual representations of text. However, these models struggled to capture long-range dependencies due to issues like vanishing gradients and limited memory capacity.

Another breakthrough in NLP came with the introduction of the attention mechanism, which enables models to dynamically focus on relevant parts of the input sequence. This

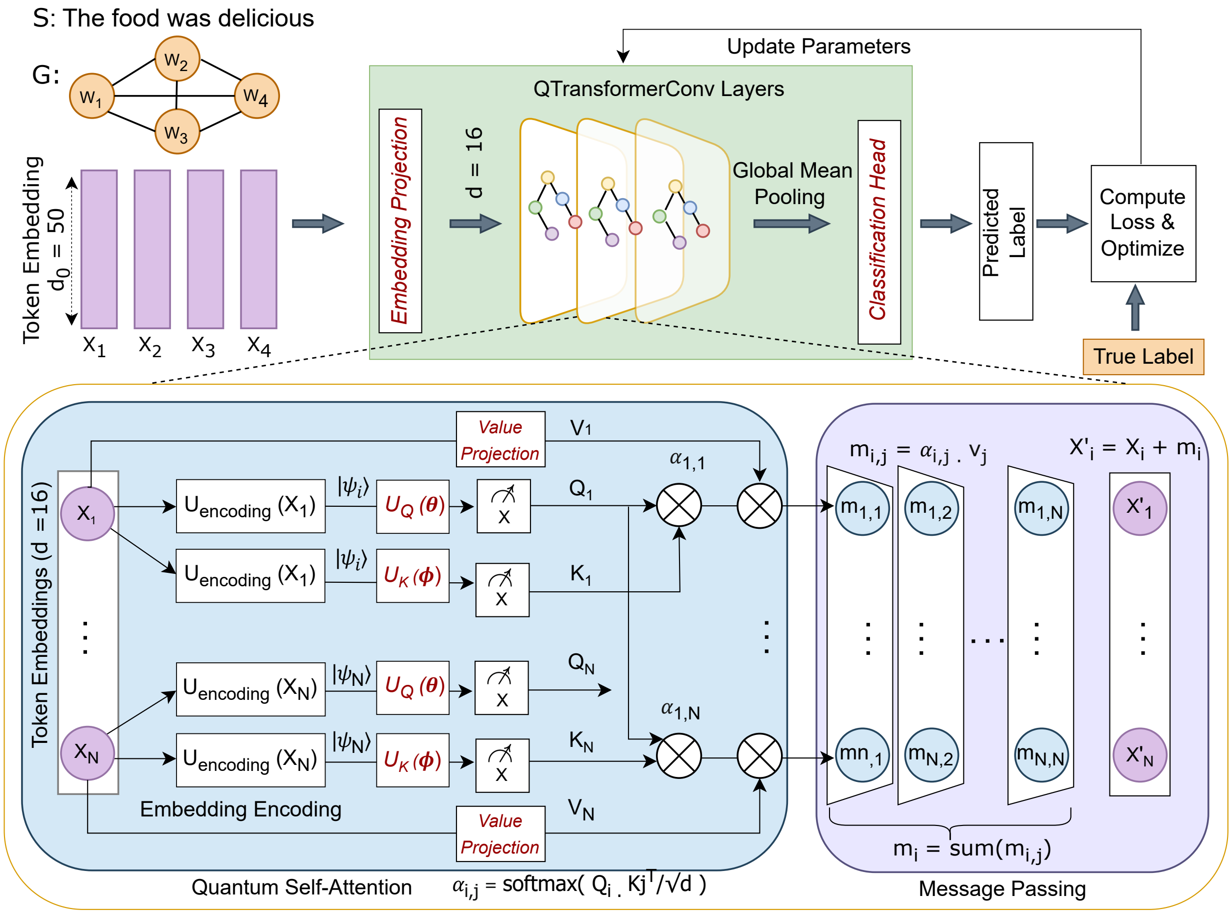

Figure 1. Overview of the Quantum Graph Transformer (QGT) architecture. Each input sentence $\mathcal{S}$ is tokenized and represented as a fully connected graph G. Tokens are initialized using GloVe embeddings ($d_0 = 50$) and projected to $\mathbf{x}_i \in \mathbb{R}^d$ with $d = 16$, then processed through stacked QTransformerConv layers and aggregated via global mean pooling. In each QTransformerConv layer, node features $\mathbf{x}_i$ are encoded into quantum states $|\psi_i\rangle$ and processed using $U_Q(\boldsymbol{\theta})$ and $U_K(\boldsymbol{\phi})$ to extract query and key vectors. These vectors are used in quantum self-attention to compute attention scores $\alpha_{i,j}$, guiding message passing to update node features. The aggregated features are passed through a classification head for label prediction, and model parameters (*marked in red*) are updated via backpropagation to minimize the loss.

concept was central to the development of the Transformer architecture [36], which entirely replaces recurrence with self-attention layers. Models such as BERT [11], GPT [13], and RoBERTa [20] achieved state-of-the-art results across a wide range of NLP tasks. These models typically require large amounts of labeled data, substantial computational resources, and still face challenges in generalizing to unseen linguistic patterns.

2.3 Existing Quantum NLP Models

Quantum Machine Learning (QML) has been extensively studied in recent years due to its potential to surpass the capabilities of classical machine learning and demonstrate quantum advantage. QML typically employs a hybrid quantum–classical approach, where PQCs are used as trainable quantum layers. A PQC is represented as a unitary transformation $U(\boldsymbol{\lambda})$ acting on an input quantum state $|\psi_{\text{in}}\rangle$, where $\boldsymbol{\lambda} = \{\lambda_1, \lambda_2, \ldots, \lambda_p\}$ are tunable real-valued parameters. The resulting output state is given by $|\psi_{\text{out}}\rangle = U(\boldsymbol{\lambda})\,|\psi_{\text{in}}\rangle$. In the QML framework, classical input data is first encoded into a quantum state, and the PQC is subsequently trained to learn the optimal parameters that minimize the loss function for the target learning task.

Recently, there has been growing interest in QNLP models as a promising alternative to classical NLP models. Hybrid quantum-enhanced NLP models aim to encode natural language into quantum states and optimize parameters to learn semantic relationships between words. Since quantum states reside in high-dimensional Hilbert space, they offer superior expressive power, even with relatively few parameters. The categorical compositional distributional (DisCoCat) model combines distributional semantics with grammatical structure and maps naturally onto quantum tensor product spaces [9]. However, the challenge of accurately extracting syntactic structure from sentence inputs limits the adaptability of the DisCoCat model, especially when handling the variability and ambiguity of natural language. Other approaches include quantum-enhanced bag-of-words (BoW) and N-gram models, which encode word occurrences or co-occurrence patterns into quantum states [1]. These models have shown promising results on synthetic datasets, but their practical implementation and performance are often constrained by the qubit requirements, limiting their scalability and real-world applicability. In addition, Quantum LSTM [33] and Quantum SVM [1] models have been proposed to address sentiment classification tasks similar to classical NLP models. Recent efforts have focused on developing quantum-enhanced self-attention mechanisms. These models demonstrated better learning on real-world benchmark datasets [6, 18, 40]. Additionally, a recent work by Silver *et al.* introduced an ensemble-based technique that employs an incremental data injection approach to improve generalization in quantum NLP models for sentiment classification tasks [31].

2.4 Classical Graph Transformers

Classical graph transformer are a class of graph neural network models that incorporate the self-attention mechanism from the original transformer architecture into graph structured data [36, 39]. Specifically, graph transformer replaces fixed graph convolution with learned attention-based message passing. This allows each node to attend to a subset or all other nodes based on learned attention scores. As a result, graph transformers can effectively capture both local and long-range dependencies in the underlying data. Given a undirected, unweighted graph $G = (V, E)$ with node features $\{\mathbf{x}_i \in \mathbb{R}^d\}_{i \in V}$, each node undergoes a feature transformation through three learned linear projections to obtain query, key and value representations:

$$\mathbf{Q}_i = \mathbf{W}_Q \cdot \mathbf{x}_i,\ \mathbf{K}_j = \mathbf{W}_K \cdot \mathbf{x}_j,\ \mathbf{V}_j = \mathbf{W}_V \cdot \mathbf{x}_j, \tag{1}$$

Here $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^d$ are shared trainable parameters. The attention scores $\alpha_{i,j}$ is computed between target node i and its neighbor j using a scaled dot-product between the query and key vectors:

$$e_{i,j} = \frac{\mathbf{Q}_i \cdot \mathbf{K}_j^T}{\sqrt{d}} \tag{2}$$

To ensure these scores are comparable across neighbors, they are normalized using the softmax function:

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{m \in \mathcal{N}(i)} \exp(e_{i,m})} \tag{3}$$

where $\mathcal{N}(i)$ denotes the set of neighbors of node i. The resulting attention coefficients $\alpha_{i,j}$ reflects the relative importance or relevance of each neighbor j when updating node i's feature vector. The final output feature for node i is computed as a weighted sum of the value vectors of its neighbors:

$$\mathbf{x}_i' = \mathbf{x}_i + \sum_{j \in \mathcal{N}(i)} \alpha_{i,j} \cdot \mathbf{V}_j \tag{4}$$

This formulation enables each node to selectively focus on different parts of its neighborhood, allowing the model to dynamically encode structural information based on relevancy score. Multiple layers of transformer-based convolution can be stacked to progressively refine node representations and capture higher-order dependencies across the graph.

In today's AI-driven world, where improving natural language understanding is becoming more crucial, quantum-enhanced self-attention models have demonstrated significant potential in advancing NLP tasks. Inspired by this, we propose the Quantum Graph Transformer (QGT) model for sentiment classification for capturing deeper context through quantum-enhanced representations.

3 Quantum Graph Transformer Model

Figure 1 illustrates the overall architecture of the proposed QGT model. The hybird QGT model intergrates quantum self-attention into a graph-based nerual architecture for sentiment classification. The architecture of the model consists of the following key components:

3.1 Sentence to Graph Construction

Each input sentence $\mathcal{S} \in \mathcal{D}$, where $\mathcal{D}$ denotes the dataset and $\mathcal{S}$ represents one sentence, is first tokenized into a sequence of tokens. This sequence is denoted as $\mathcal{S} = \{w_1, w_2, \ldots, w_N\}$, where each w_i represents a token and N denotes the total number of tokens in the sentence after tokenization. Based on this tokenized sequence, we construct a graph $G = (V, E)$, where V represents the tokens treated as nodes and E corresponds to the edge connections between tokens. In the QGT model, we consider a complete graph structure in which every node is connected to every other node, i.e., $(i, j) \in E$ for all $i \neq j$. The dense connectivity allows each token to directly exchange information with all other tokens during message passing. This is particularly essential for capturing long-range dependencies and global context required in sentiment analysis tasks. Alternatively, for efficiency and

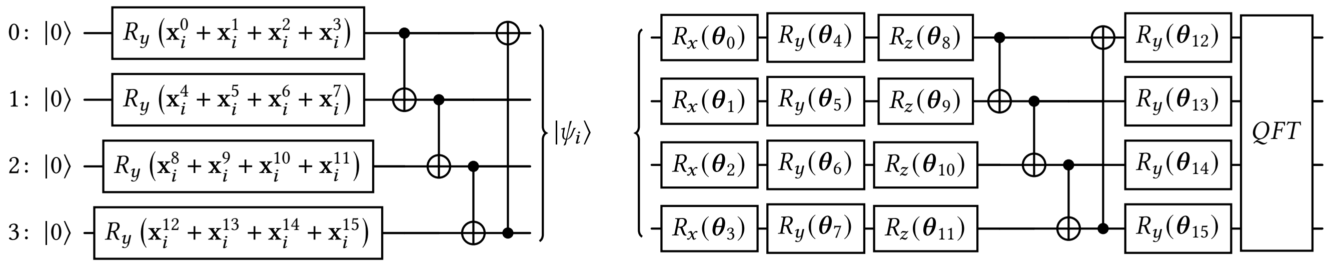
0: $|0\rangle$
1: $|0\rangle$
2: $|0\rangle$
3: $|0\rangle$
$R_y\left(\mathbf{x}_i^0+\mathbf{x}_i^1+\mathbf{x}_i^2+\mathbf{x}_i^3\right)$
$R_y\left(\mathbf{x}_i^4+\mathbf{x}_i^5+\mathbf{x}_i^6+\mathbf{x}_i^7\right)$
$R_y\left(\mathbf{x}_i^8+\mathbf{x}_i^9+\mathbf{x}_i^{10}+\mathbf{x}_i^{11}\right)$
$R_y\left(\mathbf{x}_i^{12}+\mathbf{x}_i^{13}+\mathbf{x}_i^{14}+\mathbf{x}_i^{15}\right)$
$|\psi_i\rangle$
$R_x(\boldsymbol{\theta}_0)$
$R_x(\boldsymbol{\theta}_1)$
$R_x(\boldsymbol{\theta}_2)$
$R_x(\boldsymbol{\theta}_3)$
$R_y(\boldsymbol{\theta}_4)$
$R_y(\boldsymbol{\theta}_5)$
$R_y(\boldsymbol{\theta}_6)$
$R_y(\boldsymbol{\theta}_7)$
$R_z(\boldsymbol{\theta}_8)$
$R_z(\boldsymbol{\theta}_9)$
$R_z(\boldsymbol{\theta}_{10})$
$R_z(\boldsymbol{\theta}_{11})$
$R_y(\boldsymbol{\theta}_{12})$
$R_y(\boldsymbol{\theta}_{13})$
$R_y(\boldsymbol{\theta}_{14})$
$R_y(\boldsymbol{\theta}_{15})$
QFT

Figure 2. (*Left*) Quantum embedding circuit $U_{\text{encoding}}(\mathbf{x}_i)$ maps a token embedding vector $\mathbf{x}_i \in \mathbb{R}^{d=16}$ to a 4-qubit quantum state $|\psi_i\rangle$ using grouped R_y rotations, where each rotation angle is given by $\sum_{j=4k}^{4k+3} \mathbf{x}_i^j$ for qubits $k = 0, 1, 2, 3$. The circuit is then followed by a ring-style CNOT entanglement layer. (*Right*) Parameterized quantum circuit $U_Q(\boldsymbol{\theta})$ used for query vector generation consists of R_x, R_y and R_z layers per qubit, a ring-style CNOT entanglement layer, a second R_y layer, and a Quantum Fourier Transform (QFT) [37] at the end. The combined circuit $U_Q(\boldsymbol{\theta}) \cdot U_{\text{encoding}}(\mathbf{x}_i)$ is applied, and measurement yields the query vector $\mathbf{Q}_i$. Similarly, key vector $\mathbf{K}_i$ is generated using parameters $\boldsymbol{\phi}$.

local context modeling, a k-nearest neighbor (k-NN) graph can be employed, where each node is connected only to its k immediate neighbors in the token sequence. This sparsity can significantly reduce computational overhead, particularly for shorter sentences (*e.g.* tweets, chat messages, or search queries) where full connectivity may not be necessary.

3.2 QTransformerConv: Embedding Encoding

Each node $i \in V$ (corresponding to token w_i) is initially mapped to a pretrained GloVe embedding vector [25] $em_i \in \mathbb{R}^{50}$, capturing semantic and syntactic information of the token. First, we project GloVe embeddings to a lower dimensional representation $\mathbf{x}_i \in \mathbb{R}^d$ using a linear layer. Here, $d = 16$ is the reduced dimension of each node feature vector. The dimensionality-reduced input embedding vectors can be efficiently encoded into quantum circuits, thereby requiring fewer qubits. The projected feature $\mathbf{x}_i$ is then encoded into a quantum state $|\psi_i\rangle$ using the quantum encoding circuit $U_{\text{encoding}}(\mathbf{x}_i)$ shown in Figure 2 (left). The circuit utilizes $n = \sqrt{d}$ qubits, where each qubit is initialized by applying n number of R_y gates sequentially corresponding to the components of $\mathbf{x}_i$. Following these R_y rotations, entanglement is introduced through a series of CNOT gates arranged in a ring pattern, connecting each qubit to its neighboring qubit. The encoding circuit can be represented like this

$$U_{\text{encoding}}(\mathbf{x}_i) = \left(\prod_{k=0}^{n-1} \text{CNOT}_{k,\,k+1}\right) \cdot \left(\bigotimes_{k=0}^{n-1} \left(\prod_{j=nk}^{nk+n-1} R_y(\mathbf{x}_i^j)\right)\right) \tag{5}$$

Since each qubit undergoes n sequential R_y rotations, these rotations can be merged into a single R_y gate with a cumulative rotation angle. This simplification is expressed as:

$$\prod_{j=nk}^{nk+n-1} R_y(\mathbf{x}_i^j) \equiv R_y\left(\sum_{j=nk}^{nk+n-1} \mathbf{x}_i^j\right) \tag{6}$$

Applying this encoding circuit to the initial state $|0\rangle^{\otimes n}$ yields the encoded quantum state:

$$U_{\text{encoding}}(\mathbf{x}_i) = \left(\prod_{k=0}^{n-1} \text{CNOT}_{k,\,k+1}\right) \cdot \left(\bigotimes_{k=0}^{n-1} R_y\left(\sum_{j=nk}^{nk+n-1} \mathbf{x}_i^j\right)\right) \tag{7}$$

$$|\psi_i\rangle = U_{\text{encoding}}(\mathbf{x}_i)\,|0\rangle^{\otimes n} \tag{8}$$

3.3 QTransformerConv: Quantum Self-Attention

After encoding, each node feature is represented as a quantum state $|\psi_i\rangle$. We then apply a QTransformerConv layer, which performs message passing based on a quantum-enhanced self-attention mechanism. To extract query vectors for self-attention, we apply $U_Q(\boldsymbol{\theta})$ with n qubits to each encoded state $|\psi_i\rangle$. As illustrated in Figure 2 (right), the query circuit $U_Q(\boldsymbol{\theta})$ consists of three layers of single-qubit rotations—R_x, R_y, and R_z—applied to each qubit, followed by a ring-style CNOT entanglement layer. This is succeeded by an additional layer of R_y rotations and a Quantum Fourier Transform (QFT) applied across all qubits. A similar circuit structure, $U_K(\boldsymbol{\phi})$, is used to generate key vectors, differing only in its parameter set $\boldsymbol{\phi}$. Both $U_Q(\boldsymbol{\theta})$ and $U_K(\boldsymbol{\phi})$ are fully differentiable and their parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are learned and optimized during training. The query $\mathbf{Q}_i$ and key $\mathbf{K}_i$ vectors are then obtained by repeatedly sampling from the circuit to measure the expectation values of some observables O_k:

$$\mathbf{Q}_i = \left(\langle\psi_i|\, U_Q^\dagger(\boldsymbol{\theta}) \mid O_k \mid U_Q(\boldsymbol{\theta})\,|\psi_i\rangle\right)_{k=0,\ldots,n-1} \tag{9}$$

$$\mathbf{K}_i = \left(\langle\psi_i|\, U_K^\dagger(\boldsymbol{\phi}) \mid O_k \mid U_K(\boldsymbol{\phi})\,|\psi_i\rangle\right)_{k=0,\ldots,n-1} \tag{10}$$

The observables O_k can be any combination of Pauli-X, Pauli-Y and Pauli-Z measurements. In our proposed model, we apply a Pauli-X measurement on each qubit to get query $\mathbf{Q}_i$ and key $\mathbf{K}_i$ vectors. For a graph with N nodes, this process

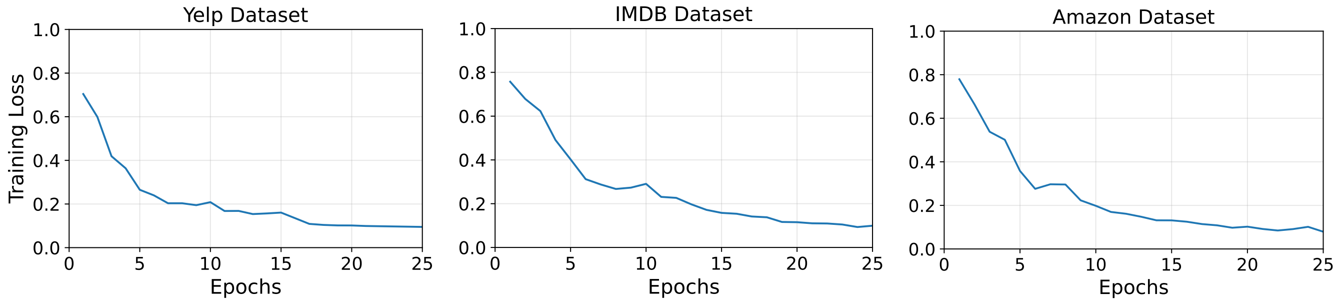

Figure 3. Training loss curves for the Yelp (*left*), IMDB (*middle*), and Amazon (*right*) datasets in the sentiment classification task using the proposed QGT model. All three datasets show consistent loss reduction /learning over training epochs.

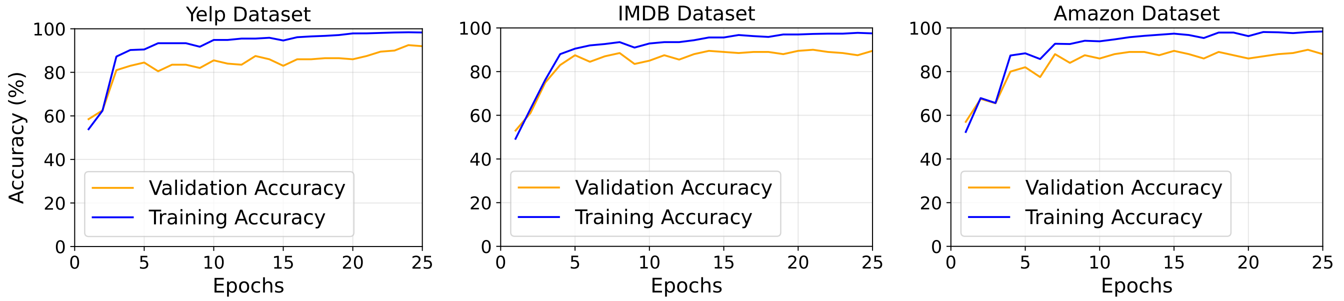

Figure 4. Training and validation accuracy curves for the Yelp (*left*), IMDB (*middle*), and Amazon (*right*) datasets in the sentiment classification task using the proposed QGT model. The consistent gap between training and validation curves indicates stable convergence and strong generalization across all three datasets.

is repeated independently for each node to obtain its corresponding query and key representations. Once the query $\mathbf{Q}_i$ and key $\mathbf{K}_i$ vectors are computed for all nodes, attention scores are calculated to determine the relevance of each neighboring node j to a given node i. Similar to classical self-attention in Equations (2) & (3), the attention score between node i and node j is computed using a scaled dot-product, and then normalized via the softmax function:

$$\alpha_{i,j} = \frac{\exp\left(\frac{\mathbf{Q}_i \cdot \mathbf{K}_j^T}{\sqrt{d}}\right)}{\sum_{m \in \mathcal{N}(i)} \exp\left(\frac{\mathbf{Q}_i \cdot \mathbf{K}_m^T}{\sqrt{d}}\right)} \tag{11}$$

This normalized score determines how much influence token j exerts on token i during the message-passing phase. Higher attention scores correspond to stronger influence from the source node.

3.4 QTransformerConv: Message Passing

Once the quantum attention scores between token nodes are computed, the model proceeds with message passing over the graph to propagate contextual information. Quantum-enhanced attention values from neighboring tokens are aggregated to update each node's embedding vector. Specifically, each token node sends messages to its directly connected neighbors (based on the edge set E), with each message weighted by the quantum attention scores $\alpha_{i,j}$ as defined in Equation 11. For each node i, the message received from a neighboring node $j \in \mathcal{N}(i)$ is computed as $m_{i,j} = \alpha_{i,j} \cdot \mathbf{V}_j$ where $\mathbf{V}_j$ is the projected feature vector (value vector) associated with node j. The total message aggregated at node i from all of its neighbors is $m_i = \sum_{j \in \mathcal{N}i} m_{i,j}$ The node's feature vector is then updated by combining the original node representation $\mathbf{x}_i$ with the aggregated message, $\mathbf{x}_i^{'} = \mathbf{x}_i + m_i$. This update scheme preserves the identity of the original node while enriching its representation with context-aware features derived from its neighborhood.

3.5 Classification Head

After the QTransformerConv layer refines the token (node) representations using quantum attention-guided message passing, the model aggregates the node-level information for sentence-level classification. A global mean pooling operation is applied across all node embeddings $\mathbf{x}_i'$. The pooled sentence representation $\mathbf{x}'_{\text{graph}}$ is computed as:

$$\mathbf{x}'_{\text{graph}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i' \tag{12}$$

where N is the number of nodes. This pooled representation is then passed through a fully connected layer to produce raw class logits $\mathbf{z}$, given by:

$$\mathbf{z} = \mathbf{W}^T \mathbf{x}'_{\text{graph}} + \mathbf{b} \tag{13}$$

where $\mathbf{W}$ is the weight matrix and $\mathbf{b}$ is the bias. The logits $\mathbf{z}$ are used in the cross-entropy loss [10] function, which

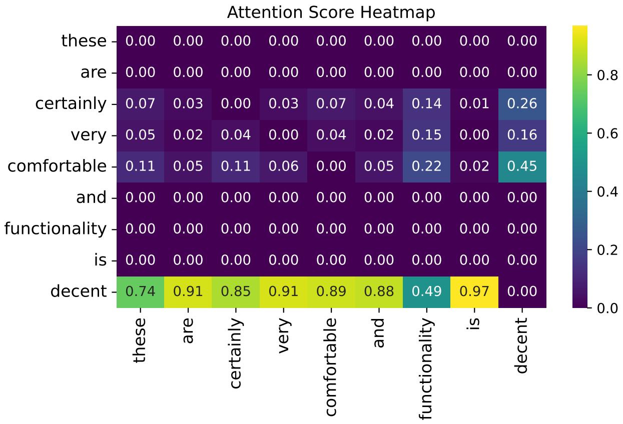

Figure 5. Attention score heatmap of a testing sample from the Amazon dataset. Lighter colors indicate higher attention given to the corresponding token from other tokens.

applies the softmax function internally to compute the probability distribution over classes. The loss is defined as:

$$\mathcal{L} = -\sum_{i=1}^{c} y_i \log\left(\frac{\exp(z_i)}{\sum_{j=1}^{c}\exp(z_j)}\right) \tag{14}$$

where c is number of classes, y_i is the true label and z_i is the logit for class i. During backpropagation, the model's trainable parameters are updated to minimize this loss. The model parameters come from two main components: (1) the quantum self-attention mechanism, which includes $U_Q(\boldsymbol{\theta})$ and $U_K(\boldsymbol{\phi})$, each with $4 \times 4 = 16$ trainable parameters; and (2) the classical layers, including the embedding projection, value projection, and the classification head. Figure 1 shows the trainable parameters in red. The detailed training setup and training parameters are discussed in Section 4.1.

4 Experimental Result & Analysis

4.1 Experimental Setup

We evaluate the QGT model on five commonly used benchmark datasets for QNLP models. Three datasets are from Yelp, IMDB, and Amazon, consisting of 1,000 samples each [17]. The Yelp dataset contains restaurant reviews with positive or negative sentiment labels, the IMDB dataset contains movie reviews with positive or negative sentiment labels, and the Amazon dataset includes user product review ratings from 1 to 5. The three datasets have maximum sentence lengths of 34, 45, and 30, respectively. The other two datasets, MC (Meaning Classification) and RP (Relative Pronoun), are synthetic and generated as described in [21]. They have relatively smaller vocabularies, and each sample has a length of 4 tokens. The MC dataset contains 130 samples, while the RP dataset has 105 samples. Each of the five datasets is split into 70%, 10%, and 20% as training, validation, and testing sets, respectively, and a separate model is trained for each dataset. Training is performed using the PennyLane Lightning device [2], and the model is run on macOS with an Apple M3 Max chip and 64 GB of RAM.

Each dataset is processed and trained using the QGT model described in Section 3. In our experiments, we use 4 qubits and a single layer of PQCs, $U_Q(\boldsymbol{\theta})$ and $U_K(\boldsymbol{\phi})$, to compute the query and key vectors, respectively. The parameters $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ are initialized from a normal distribution with mean 0 and standard deviation 0.01. We choose Adam optimizer with a learning rate of 0.01, and employ cross-entropy loss [10], which is well-suited for classification tasks involving categorical sentiment labels. We use a StepLR scheduler with step size of 5 and decay factor (gamma) of 0.7, a batch size of 32, and apply early stopping based on validation loss, with training running for up to 25 epochs. To encourage better exploration of the parameter space for $U_Q(\boldsymbol{\theta})$ and $U_K(\boldsymbol{\phi})$, we apply a reinforcement learning–based regularization strategy. The reward is defined as the negative training loss and is used to directly update the parameters of the PQCs. We also experimented with using multiple layers of $U_Q(\boldsymbol{\theta})$ and $U_K(\boldsymbol{\phi})$ as well as multi-headed attention. However, no significant improvements were observed on these datasets, likely due to their small size.

4.2 Model Learning Dynamics

We first analyze the learning dynamics of the proposed QGT model to evaluate its performance. Figure 3 presents the training loss curves for the Yelp, IMDB, and Amazon datasets. The plots show a consistent reduction in training loss over epochs, indicating effective learning across all three datasets. In each case, the loss decreases sharply during the initial epochs, followed by gradual convergence. Figure 4 demonstrates the training and validation accuracy curves for the same datasets. The validation accuracy consistently follows the training accuracy, showing the model's ability to generalize while minimizing loss over epochs. Figure 5 illustrates the attention score heatmap of a test sample from the Amazon dataset, where lighter colors indicate higher attention weights. From the plot, we observe that tokens such as *"decent"*, *"comfortable"*, and *"certainly"* receive higher attention scores from other tokens, as they contribute more to the positive sentiment label.

4.3 Comparison with existing Quantum NLP models

We compare the performance of the proposed QGT model with existing QNLP models developed for sentiment classification tasks. Figure 6 (*left*) presents the test accuracy of the QGT model alongside other attention-based quantum NLP models (QSANN [18], QSAN [40], QMSAN [6]) evaluated on all five datasets: Yelp, IMDB, Amazon, MC, and RP. The plot also includes results from a classical graph transformer model, providing a comparative perspective against non-quantum architectures. The QGT model consistently achieves higher across all five datasets, demonstrating its effectiveness in capturing semantic structure through quantum

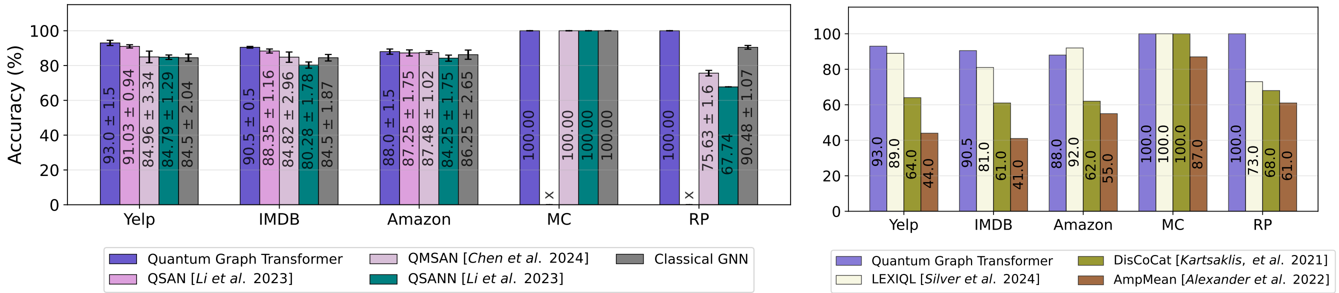

Figure 6. (*Left*) Test accuracy comparison between the proposed QGT model and existing attention-based models, including a classical graph transformer model. (*Right*) Test accuracy comparison between the QGT model and other baseline models. The QGT model consistently outperforms all baselines, with performance comparable to LEXIQL [31] on the Amazon dataset.

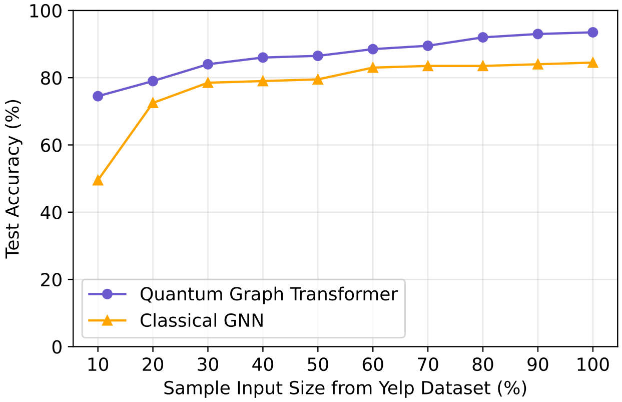

Figure 7. Demonstrating the effect of varying sample sizes from the Yelp dataset on model learning performance. The plot shows accuracy on a fixed testing set for both the QGT model and a classical graph transformer. The QGT model demonstrates better learning efficiency with fewer samples.

graph-based attention mechanisms. Figure 6 (*right*) presents a performance comparison of the QGT model against several baseline models, including the grammar-aware DisCoCat [8, 21], the ensemble-based LEXIQL model [31], and AmpMean [1]. The plot demonstrates that the QGT model outperforms other baseline Quantum NLP techniques, with the exception of LEXIQL, on the Amazon dataset. Overall, the QGT model exhibits strong generalization performance across all baseline quantum NLP models and classical graph transformer models.

4.4 Quantum NLP Advantage

In our proposed QGT model, the self-attention mechanism leverages PQCs to compute attention with significantly fewer parameters compared to the classical attention mechanism. While classical attention layers require high dimensional projections with large weight metrics, our quantum model generates query and key vectors using only 32 trainable parameters for each layers of $U_Q(\boldsymbol{\theta})$ and $U_K(\boldsymbol{\phi})$. Despite this reduced parameter count, the comparison in Figure 6 shows that the QGT model effectively learns attention and outperforms the classical graph transformer model. The key to this efficiency lies in the expressive power of PQCs, which enables the model to capture rich quantum representations within the Hilbert space.

One of the major limitations of current classical NLP models is their reliance on large corpora of data and their computationally intensive architectures. Any viable alternative model should aim to overcome these constraints. To check sample efficiency, we vary the training data from 10% to 100% of the Yelp dataset while evaluating testing accuracy on a fixed test set. Figure 7 shows the testing accuracy performance of both the quantum and classical graph transformer models across varying training sample sizes. The plot show that the QGT model requires fewer training samples to achieve comparable performance. Specifically, the QGT model reaches 82% accuracy using only 30% of the training samples from Yelp dataset, whereas the classical graph transformer requires 60% of the training samples to attain similar accuracy. This highlights the potential of quantum NLP models for data-efficient learning in NLP tasks.

5 Limitations & Future Work

The QGT model shows promising results for sentiment classification on small-scale datasets, outperforming existing QNLP techniques and classical graph transformers. However, its evaluation has been limited to small, synthetic datasets. Future work should explore QGT's performance on larger datasets such as SST-5 [32], Twitter Sentiment [27], and the Kaggle IMDb Movie Review dataset [23]. To adapt to these datasets, we may need to incorporate multiple layers of query and key PQCs and use multi-head quantum self-attention to capture richer semantic patterns. Each head in multi-head attention can be computed in parallel using separate quantum circuits, though the qubit count will grow linearly with the number of heads. Additionally, we observe improvements in parameter reduction and sample efficiency on small datasets. Future work could also investigate other quantum advantages such as generalization and resource requirements. Moreover, the performance of the QGT model could be evaluated under the impact of noise. Additionally, the graph structure can be optimized by incorporating semantic or grammar-aware connections instead of fully connected token graphs.

6 Conclusion

In this work, we propose the Quantum Graph Transformer (QGT) model for the sentiment classification task. The model introduces a novel QTransformerConv layer, which employs quantum self-attention for message passing between nodes. This architecture effectively captures the complex relationships between sentence tokens and their corresponding sentiment labels. Experimental results on benchmark datasets demonstrate the potential of graph-based quantum NLP models in learning and performing NLP tasks.

7 Acknowledgements

The research presented in this article was supported by the NNSA's Advanced Simulation and Computing Beyond Moore's Law program at Los Alamos National Laboratory and the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project number 20230049DR. This work has been assigned LANL technical report number LA-UR-25-24685.

References

[1] Aaranya Alexander and Dominic Widdows. 2022. Quantum Text Encoding for Classification Tasks. In *2022 IEEE/ACM 7th Symposium on Edge Computing (SEC)*. IEEE, 355–361. doi:10.1109/sec54971.2022.00052 arXiv:2301.03715

[2] Ville Bergholm, Josh Izaac, et al. 2022. Pennylane: Automatic differentiation of hybrid quantum-classical computations. arXiv:1811.04968 arXiv preprint.

[3] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. 2017. Quantum machine learning. *Nature* 549 (Sept. 2017), 195–202. doi:10.1038/nature23474 arXiv:1611.09347

[4] Peter F. Brown, Vincent J. Della Pietra, Peter V. Desouza, Jennifer C. Lai, and Robert L. Mercer. 1992. Class-Based n-gram Models of Natural Language. *Computational Linguistics* 18, 4 (1992), 467–480. https://aclanthology.org/J92-4003/

[5] Marco Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J. Coles. 2022. Challenges and opportunities in quantum machine learning. *Nature Computational Science* 2 (Sept. 2022), 567–576. doi:10.1038/s43588-022-00311-3 arXiv:2303.09491

[6] Fu Chen, Qinglin Zhao, Li Feng, Chuangtao Chen, Yangbin Lin, and Jianhong Lin. 2025. Quantum mixed-state self-attention network. *Neural Networks* 185 (May 2025), 107123. doi:10.1016/j.neunet.2025.107123 arXiv:2403.02871

[7] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 1724–1734. doi:10.3115/v1/d14-1179 arXiv:1406.1078

[8] Kenneth Ward Church. 2017. Word2Vec. *Natural Language Engineering* 23, 1 (Jan. 2017), 155–162. doi:10.1017/S1351324916000334

[9] Bob Coecke, Mehrnoosh Sadrzadeh, and Stephen Clark. 2010. Mathematical Foundations for a Compositional Distributional Model of Meaning. *Linguistic Analysis* 36, 1–4 (March 2010), 345–384. arXiv:1003.4394 Festschrift for Joachim Lambek.

[10] PyTorch Contributors. 2025. CrossEntropyLoss. https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html Accessed: 2025-05-07.

[11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, 4171–4186. doi:10.18653/v1/N19-1423 arXiv:1810.04805

[12] Xianzhong Ding, Le Chen, Murali Emani, Chunhua Liao, Pei-Hung Lin, Tristan Vanderbruggen, Zhen Xie, Alberto Cerpa, and Wan Du. 2023. HPC-GPT: Integrating Large Language Model for High-Performance Computing. In *SC-W'23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis*. Association for Computing Machinery, 951–960. doi:10.1145/3624062.3624172 arXiv:2311.12833

[13] Luciano Floridi and Massimo Chiriatti. 2020. GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds and Machines* 30 (Nov. 2020), 681–694. doi:10.1007/s11023-020-09548-1

[14] Zellig S. Harris. 1954. Distributional Structure. *Word* 10, 2–3 (1954), 146–162. doi:10.1080/00437956.1954.11659520

[15] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. 2019. Supervised learning with quantum-enhanced feature spaces. *Nature* 567 (March 2019), 209–212. doi:10.1038/s41586-019-0980-2 arXiv:1804.11326

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation* 9, 8 (Nov. 1997), 1735–1780. doi:10.1162/neco.1997.9.8.1735

[17] Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. 2015. From Group to Individual Labels Using Deep Features. In *21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, 597–606. doi:10.1145/2783258.2783380

[18] Guangxi Li, Xuanqiang Zhao, and Xin Wang. 2024. Quantum self-attention neural networks for text classification. *Science China Information Sciences* 67 (March 2024), 142501. doi:10.1007/s11432-023-3879-7 arXiv:2205.05625

[19] Bang Liu and Lingfei Wu. 2022. *Graph Neural Networks: Foundations, Frontiers, and Applications*. Springer, Chapter Graph Neural Networks in Natural Language Processing, 463–481. doi:10.1007/978-981-16-6054-2_21

[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv preprint* (July 2019), 1–13. arXiv:1907.11692

[21] Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, and Bob Coecke. 2023. QNLP in Practice: Running Compositional Models of Meaning on a Quantum Computer. *Journal of Artificial Intelligence Research* 76 (April 2023), 1305–1342. doi:10.1613/jair.1.14329 arXiv:2102.12846

[22] Yasmin Moslem, Rejwanul Haque, John D. Kelleher, and Andy Way. 2023. Adaptive Machine Translation with Large Language Models. In *24th Annual Conference of the European Association for Machine Translation*. European Association for Machine Translation, 227–237. arXiv:2301.13294 https://aclanthology.org/2023.eamt-1.22/

[23] Lakshmipathi Narayan. 2018. IMDb Dataset of 50K Movie Reviews. https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews Accessed: 2025-05-14.

[24] OpenAI, Josh Achiam, et al. 2024. GPT-4 Technical Report. *arXiv preprint* (March 2024), 1–100. arXiv:2303.08774

[25] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 1532–1543. doi:10.3115/v1/d14-1162

[26] Sushree Bibhuprada B. Priyadarshini, Amiya Bhusan Bagjadab, and Brojo Kishore Mishra. 2020. *Natural Language Processing in Artificial Intelligence*. Apple Academic Press, Chapter A Brief Overview of Natural Language Processing and Artificial Intelligence, 211–224. doi:10.1201/9780367808495-8

[27] Sara Rosenthal, Noura Farra, and Preslav Nakov. 2017. SemEval-2017 Task 4: Sentiment Analysis in Twitter. In *11th International Workshop on Semantic Evaluation (SemEval-2017)*. Association for Computational Linguistics, 502–518. doi:10.18653/v1/s17-2088 arXiv:1912.00741

[28] Stuart J. Russell and Peter Norvig. 2016. *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson. http://aima.cs.berkeley.edu/

[29] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2009. The Graph Neural Network Model. *IEEE Transactions on Neural Networks* 20, 1 (Jan. 2009), 61–80. doi:10.1109/TNN.2008.2005605

[30] Maria Schuld and Nathan Killoran. 2019. Quantum Machine Learning in Feature Hilbert Spaces. *Physical Review Letters* 122, 4 (Feb. 2019), 040504. doi:10.1103/physrevlett.122.040504 arXiv:1803.07128

[31] Daniel Silver, Aditya Ranjan, Rakesh Achutha, Tirthak Patel, and Devesh Tiwari. 2024. LEXIQL: Quantum Natural Language Processing on NISQ-era Machines. In *SC'24: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, IEEE, 1–15. doi:10.1109/sc41406.2024.00073

[32] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *2013 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 1631–1642. doi:10.18653/v1/d13-1170

[33] Jonas Stein, Ivo Christ, Nicolas Kraus, Maximilian Balthasar Mansky, Robert Müller, and Claudia Linnhoff-Popien. 2023. Applying QNLP to Sentiment Analysis in Finance. In *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*. IEEE, 20–25. doi:10.1109/qce57702.2023.10178 arXiv:2307.11788

[34] Kian Long Tan, Chin Poo Lee, and Kian Ming Lim. 2023. A Survey of Sentiment Analysis: Approaches, Datasets, and Future Research. *Applied Sciences* 13, 7 (April 2023), 4550. doi:10.3390/app13074550

[35] Duyu Tang, Bing Qin, and Ting Liu. 2015. Document Modeling with Gated Recurrent Neural Network for Sentiment Classification. In *2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, 1422–1432. doi:10.18653/v1/d15-1167

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems NIPS 2017*. Curran Associates, Inc., 5998–6008. arXiv:1706.03762 http://papers.nips.cc/paper/7181-attention-is-all-you-need

[37] Y. S. Weinstein, M. A. Pravia, E. M. Fortunato, S. Lloyd, and D. G. Cory. 2001. Implementation of the Quantum Fourier Transform. *Physical Review Letters* 86, 9 (Feb. 2001), 1889. doi:10.1103/physrevlett.86.1889 arXiv:quant-ph/9906059

[38] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S. Yu. 2021. A Comprehensive Survey on Graph Neural Networks. *IEEE Transactions on Neural Networks and Learning Systems* 32, 1 (Jan. 2021), 4–24. doi:10.1109/tnnls.2020.2978386 arXiv:1901.00596

[39] Seongjun Yun, Minbyul Jeong, Raehyun Kim, Jaewoo Kang, and Hyunwoo J. Kim. 2019. Graph Transformer Networks. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems NeurIPS 2019*. Curran Associates, Inc., 11960–11970. arXiv:1911.06455 http://papers.nips.cc/paper/9367-graph-transformer-networks

[40] Hui Zhang, Qinglin Zhao, and Chuangtao Chen. 2024. A light-weight quantum self-attention model for classical data classification. *Applied Intelligence* 54, 4 (Feb. 2024), 3077–3091. doi:10.1007/s10489-024-05337-w