Quantum Self-Attention Neural Networks for Text Classification

Guangxi Li[1,2], Xuanqiang Zhao[1], and Xin Wang[1]

[1]Institute for Quantum Computing, Baidu Research, Beijing 100193, China
[2]Centre for Quantum Software and Information, University of Technology Sydney, NSW 2007, Australia

Abstract

An emerging direction of quantum computing is to establish meaningful quantum applications in various fields of artificial intelligence, including natural language processing (NLP). Although some efforts based on syntactic analysis have opened the door to research in Quantum NLP (QNLP), limitations such as heavy syntactic preprocessing and syntax-dependent network architecture make them impracticable on larger and real-world data sets. In this paper, we propose a new simple network architecture, called the quantum self-attention neural network (QSANN), which can make up for these limitations. Specifically, we introduce the self-attention mechanism into quantum neural networks and then utilize a Gaussian projected quantum self-attention serving as a sensible quantum version of self-attention. As a result, QSANN is effective and scalable on larger data sets and has the desirable property of being implementable on near-term quantum devices. In particular, our QSANN outperforms the best existing QNLP model based on syntactic analysis as well as a simple classical self-attention neural network in numerical experiments of text classification tasks on public data sets. We further show that our method exhibits robustness to low-level quantum noises.

1 Introduction

Quantum computing is a promising paradigm [1] for fast computations that can provide substantial advantages in solving valuable problems [2, 3, 4, 5, 6]. With major academic and industry efforts on developing quantum algorithms and quantum hardware, it has led to an increasing number of powerful applications in areas including optimization [7], cryptography [8], chemistry [9, 10], and machine learning [6, 11, 12, 13].

Quantum devices available currently known as the *noisy intermediate-scale quantum* (NISQ) devices [14] have up to a few hundred physical qubits. They are affected by coherent and incoherent noise, making the practical implementation of many advantageous quantum algorithms less feasible. But such devices with 50-100 qubits already allow one to achieve quantum advantage against the most powerful classical supercomputers on certain carefully designed tasks [15, 16]. To explore practical applications with near-term quantum devices, plenty of NISQ algorithms [17, 18, 19] appear to be the best hope for obtaining quantum advantage in fields such as quantum chemistry [20], optimization [21], and machine learning [22, 23, 24]. In particular, those algorithms dealing with machine learning

problems, by employing *parameterized quantum circuits* (PQCs) [25] (also called *quantum neural networks* (QNNs) [26]), show great potential in the field of *quantum machine learning* (QML). However, in *artificial intelligence* (AI), the study of QML in the NISQ era is still in its infancy. Thus it is desirable to explore more QML algorithms exploiting the power that lies within the NISQ devices.

Natural language processing (NLP) is a key subfield of AI that aims to give machines the ability to understand human language. Common NLP tasks include speech recognition, machine translation, text classification, etc., many of which have greatly facilitated our life. Due to human language's high complexity and flexibility, NLP tasks are generally challenging to implement. Thus, it is natural to think about whether and how quantum computing can enhance machines' performance on NLP. Some works focus on quantum-inspired language models [27, 28, 29, 30] with borrowed ideas from quantum mechanics. Another approach, known as *quantum natural language processing* (QNLP), seeks to develop quantum-native NLP models that can be implemented on quantum devices [31, 32, 33, 34]. Most of these QNLP proposals, though at the frontier, lack scalability as they are based on syntactic analysis, which is a preprocessing task requiring significant effort, especially for large data sets. Furthermore, these syntax-based methods employ different PQCs for sentences with different syntactical structures and thus are not flexible enough to process the innumerable complex expressions possible in human language.

To overcome these drawbacks in current QNLP models, we propose the *quantum self-attention neural network* (QSANN), where the self-attention mechanism is introduced into quantum neural networks. Our motivation comes from the excellent performance of self-attention on various NLP tasks such as language modeling [35], machine translation [36], question answering [37], and text classification [38]. We also note that a recently proposed method [39] for quantum state tomography, an important task in quantum computing, adopts the self-attention mechanism and achieves decent results.

In each quantum self-attention layer of QSANN, we first encode the inputs into high-dimensional quantum states, then apply PQCs on them according to the layout of the self-attention neural networks, and finally adopt a *Gaussian projected quantum self-attention* (GPQSA) to obtain the output effectively. To evaluate the performance of our model, we conduct numerical experiments of text classification with different data sets. The results show that QSANN outperforms the currently best known QNLP model as well as a simple classical self-attention neural network on test accuracy, implying potential quantum advantages of our method. Our contributions are multi-fold:

- Our proposal is the first QNLP algorithm with a detailed circuit implementation scheme based on the self-attention mechanism. This method can be implemented on NISQ devices and is more practicable on large data sets compared with previously known QNLP methods based on syntactic analysis.

- In QSANN, we introduce the Gaussian projected quantum self-attention, which can efficiently dig out the correlations between words in high-dimensional quantum feature space. Furthermore, visualization of self-attention coefficients on text classification tasks confirms its ability to focus on the most relevant words.

- We experimentally demonstrate that QSANN outperforms existing QNLP methods based on syntactic analysis [40] and simple classical self-attention neural networks on several public data sets for text classification. Numerical results also imply that QSANN is resilient to quantum noise.

1.1 Preliminaries and Notations

1.1.1 Quantum Basis

Here, some basic concepts about quantum computing necessary for this paper are briefly introduced (for more details, see [41]). In quantum computing, quantum information is usually represented by n-qubit (pure) quantum states over Hilbert space $\mathbb{C}^{2^n}$. In particular, a pure quantum state could be represented by a unit vector $|\psi\rangle \in \mathbb{C}^{2^n}$ (or $\langle\psi|$), where the *ket* notation $|\rangle$ denotes a column vector and the *bra* notation $\langle\psi| = |\psi\rangle^\dagger$ with $\dagger$ referring to conjugate transpose denotes a row vector.

The evolution of a pure quantum state $|\psi\rangle$ is mathematically described by applying a quantum circuit (or a quantum gate), i.e., $|\psi'\rangle = U|\psi\rangle$, where U is the unitary operator (matrix) representing the quantum circuit and $|\psi'\rangle$ is the quantum state after evolution. Common single-qubit quantum gates include Hadamard gate H and Pauli operators

$$H := \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix},\ X := \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}, Y := \begin{bmatrix}0 & -i\\ i & 0\end{bmatrix}, Z := \begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix} \tag{1}$$

and their corresponding rotation gates denoted by $R_P(\theta) := \exp(-i\theta P/2) = \cos\frac{\theta}{2}I - i\sin\frac{\theta}{2}P$, where the rotation angle $\theta \in [0, 2\pi)$ and $P \in \{X, Y, Z\}$. Multiple-qubit quantum gates mainly include, in this paper, the identity gate I, the CNOT gate and the tensor product of single-qubit gates, e.g., $Z \otimes Z$, $Z \otimes I$, $Z^{\otimes n}$ and so on.

Quantum measurement is a way to extract classical information from a quantum state. For instance, given a quantum state $|\psi\rangle$ and an observable O, one could design quantum measurements to obtain the information $\langle\psi|O|\psi\rangle$. Within this work, we focus on the hardware-efficient Pauli measurements, i.e., setting O as Pauli operators or their tensor products. For instance, we could choose $Z_1 = Z \otimes I^{\otimes(n-1)}$, $X_2 = I \otimes X \otimes I^{\otimes(n-2)}$, $Z_1Z_2 = Z \otimes Z \otimes I^{\otimes(n-2)}$, etc., with n qubits in total.

1.1.2 Text Classification

As one of the central and basic tasks in NLP field, the text classification is to assign a given text sequence to one of the predefined categories. Examples of text classification tasks considered in this paper include topic classification and sentiment analysis. A commonly adopted approach in machine learning is to train a model with a set of pre-labeled sequences. When fed a new sequence, the trained model will be able to predict its category based on the experience learned from the training data set.

1.1.3 Self-Attention Mechanism

In a self-attention neural network layer [36], the input data $\{x_s \in \mathbb{R}^d\}_{s=1}^S$ are linearly mapped, via three weight matrices, i.e., query $W_q \in \mathbb{R}^{d\times d}$, key $W_k \in \mathbb{R}^{d\times d}$ and value $W_v \in \mathbb{R}^{d\times d}$, to three parts W_qx_s, W_kx_s, W_vx_s, respectively, and by applying inner product on the query and key parts, the output is computed as

$$y_s = \sum_{j=1}^{S} a_{s,j}\cdot W_v x_j \qquad \text{with} \qquad a_{s,j} = \frac{\mathrm{e}^{x_s^\top W_q^\top W_k x_j}}{\sum_{l=1}^{S}\mathrm{e}^{x_s^\top W_q^\top W_k x_l}}, \tag{2}$$

where $a_{s,j}$ denote the self-attention coefficients.

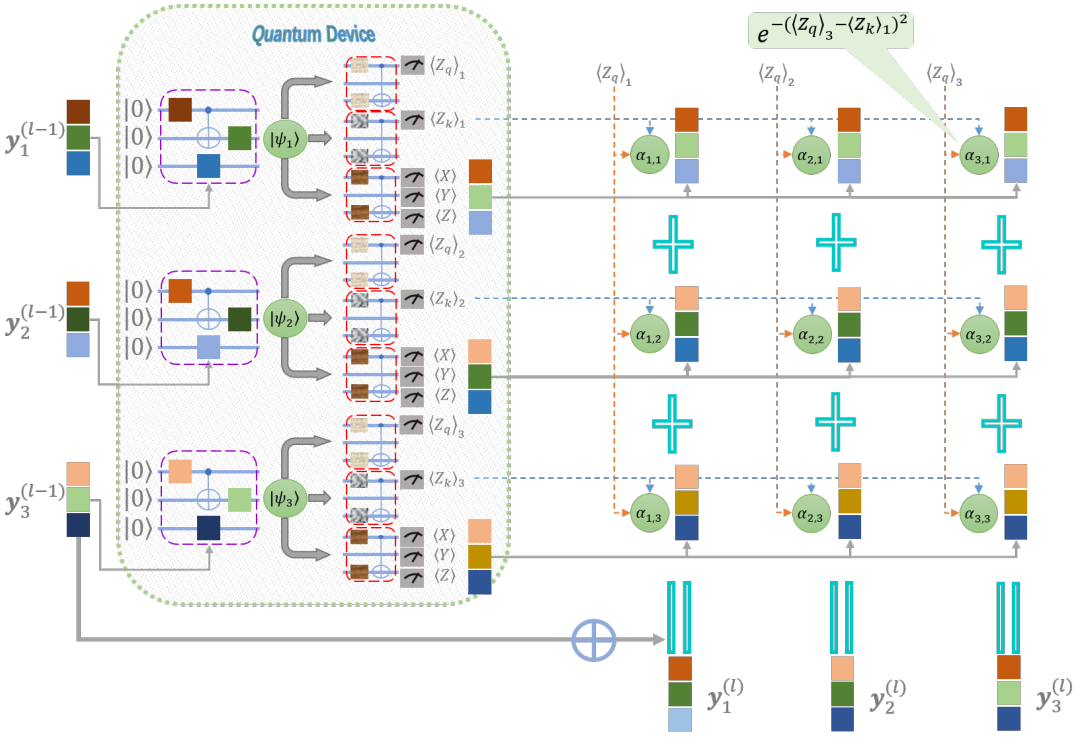


Figure 1: Sketch of a quantum self-attention layer (QSAL). On quantum devices, the classical inputs $\{\boldsymbol{y}_s^{(l-1)}\}$ are used as the rotation angles of quantum ansatzes (purple dashed boxes) to encode them into their corresponding quantum states $\{|\psi_s\rangle\}$. Then for each state, there are three different classes of ansatzes (red dashed boxes) need to be executed, where the top two classes denote the query and key parts, and the bottom one denotes the value part. On classical computers, the measurement outputs of the query part $\langle Z_q\rangle_s$ and the key part $\langle Z_k\rangle_j$ are computed through a Gaussian function to obtain the quantum self-attention coefficients $\alpha_{s,j}$ (green circles); we calculate classically weighted sums of the measurement outputs of the value part (small colored squares) and add the inputs to get the outputs $\{\boldsymbol{y}_s^{(l)}\}$, where the weights are the normalized coefficients $\tilde{\alpha}_{s,j}$, cf. Eq. (7).

2 Method

In this section, we will introduce the QSANN in detail, which mainly consists of *quantum self-attention layer* (QSAL), loss function, analytical gradients and analysis.

2.1 Quantum Self-Attention Layer

In the classical self-attention mechanism [36], there are mainly three components (vectors), i.e., queries, keys and values, where queries and keys are computed as weights assigned to corresponding values to obtain final outputs. Inspired by this mechanism, in QSAL we design the quantum analogs of these components. The overall picture of QSAL is illustrated in Fig. 1.

For the classical input data $\{\boldsymbol{y}_s^{(l-1)} \in \mathbb{R}^d\}$ of the l-th QSAL, we first use a quantum ansatz U_{enc} to encode them into an n-qubit quantum Hilbert space, i.e.,

$$|\psi_s\rangle = U_{enc}(\boldsymbol{y}_s^{(l-1)})H^{\otimes n}|0^n\rangle, \quad 1 \le s \le S, \tag{3}$$

where H denotes the Hadamard gate and S denotes the number of input vectors in a data sample.

Then we use another three quantum ansatzes, i.e., U_q, U_k, U_v with parameters $\boldsymbol{\theta}_q$, $\boldsymbol{\theta}_k$, $\boldsymbol{\theta}_v$, to represent the query, key and value parts, respectively. Concretely, for each input state $|\psi_s\rangle$, we denote by $\langle Z_q\rangle_s$ and $\langle Z_k\rangle_s$ the Pauli-Z_1 measurement outputs of the query and key parts, respectively, where

$$\begin{aligned}\langle Z_q\rangle_s &:= \langle\psi_s| U_q^\dagger(\boldsymbol{\theta}_q) Z_1 U_q(\boldsymbol{\theta}_q) |\psi_s\rangle\,,\\ \langle Z_k\rangle_s &:= \langle\psi_s| U_k^\dagger(\boldsymbol{\theta}_k) Z_1 U_k(\boldsymbol{\theta}_k) |\psi_s\rangle\,.\end{aligned} \tag{4}$$

The measurement outputs of the value part are represented by a d-dimensional vector

$$\boldsymbol{o}_s := \begin{bmatrix}\langle P_1\rangle_s & \langle P_2\rangle_s & \cdots & \langle P_d\rangle_s\end{bmatrix}^\top, \tag{5}$$

where $\langle P_j\rangle_s = \langle\psi_s| U_v^\dagger(\boldsymbol{\theta}_v) P_j U_v(\boldsymbol{\theta}_v) |\psi_s\rangle$. Here, each $P_j \in \{I, X, Y, Z\}^{\otimes n}$ denotes a Pauli observable.

Finally, by combining Eqs. (4) and (5), the classical output $\{\boldsymbol{y}_s^{(l)} \in \mathbb{R}^d\}$ of the l-th QSAL are computed as follows:

$$\boldsymbol{y}_s^{(l)} = \boldsymbol{y}_s^{(l-1)} + \sum_{j=1}^{S} \tilde{\alpha}_{s,j} \cdot \boldsymbol{o}_j, \tag{6}$$

where $\tilde{\alpha}_{s,j}$ denotes the normalized quantum self-attention coefficient between the s-th and the j-th input vectors and is calculated by the corresponding query and key parts:

$$\tilde{\alpha}_{s,j} = \frac{\alpha_{s,j}}{\sum_{m=1}^{S} \alpha_{s,m}} \qquad \text{with} \qquad \alpha_{s,j} := \mathrm{e}^{-(\langle Z_q\rangle_s - \langle Z_k\rangle_j)^2}. \tag{7}$$

Here in Eq. (6), we adopt a residual scheme when computing the output, which is analogous to [36].

2.1.1 Gaussian Projected Quantum Self-Attention

When designing a quantum version of self-attention, a natural and direct extension of the inner-product self-attention to consider is $\alpha_{s,j} := |\langle\psi_s| U_q^\dagger U_k |\psi_j\rangle|^2$. However, due to the unitary nature of quantum circuits, $\langle\psi_s| U_q^\dagger U_k$ can be regarded as rotating $|\psi_s\rangle$ by an angle, which makes it difficult for $|\psi_s\rangle$ to simultaneously correlate those $|\psi_j\rangle$ that are far away. In a word, this direct extension is not suitable or reasonable for working as the quantum self-attention. Instead, the particular quantum self-attention proposed in Eq. (7), which we call *Gaussian projected quantum self-attention* (GPQSA), could overcome the above drawback. In GPQSA, the states $U_q |\psi_s\rangle$ (and $U_k |\psi_j\rangle$) in large quantum Hilbert space are projected to classical representations $\langle Z_q\rangle_s$ (and $\langle Z_k\rangle_j$) in one-dimensional[1] classical space via quantum measurements, and a Gaussian function is applied to these classical representations. As U_q and U_k are separated, it's pretty easier to correlate $|\psi_s\rangle$ to any $|\psi_j\rangle$, making GPQSA more suitable to serve as a quantum self-attention. Here, we utilize the Gaussian function [42] mainly because it contains infinite-dimensional feature space and is well-studied in classical machine learning. Numerical experiments also verify our choice of Gaussian function. We also note that other choices of building the quantum self-attention are also worth a future study.

[1] Multi-dimension is also possible by choosing multiple measurement results, like the value part.

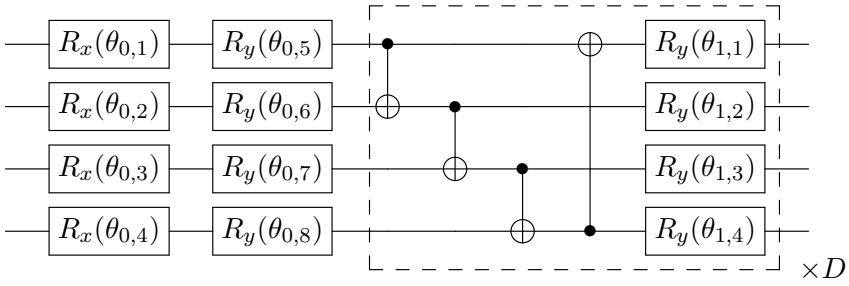


Figure 2: The ansatz used in QSANN. The first two columns denote the R_x-R_y rotations on each single-qubit subspace, then followed by repeated CNOT gates and single-qubit R_y rotations. The block circuit in the dashed box is repeated D times to enhance the expressive power of the ansatz.

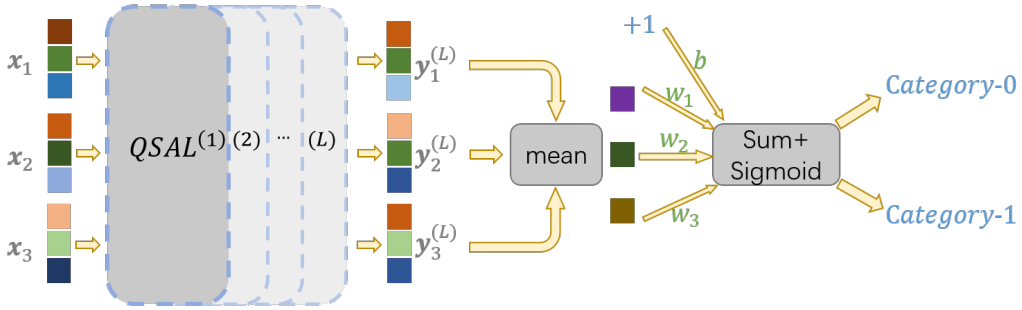


Figure 3: Sketch of QSANN, where a sequence of classical vectors $\{\boldsymbol{x}_s\}$ firstly goes through L QSALs to obtain the corresponding sequence of feature vectors $\{\boldsymbol{y}_s^{(L)}\}$, then through the average operation, and finally through the fully-connected layer for the binary prediction task.

Remark. During the preparation of this manuscript as well as after submitting our work to a peer-review conference in Sep 2021, we became aware that Ref. [43] also made initial attempts to employ the attention mechanism in QNNs. In that work, the authors mentioned a possible quantum extension towards a quantum Transformer where the straightforward inner-product self-attention is adopted. As discussed above, the inner-product self-attention may not be reasonable for dealing with quantum data. In this work, we present that GPQSA is more suitable for the quantum version of self-attention and show the validity of our method via numerical experiments on several public data sets.

2.2 Ansatz Selection

In QSAL, we employ multiple ansatzes for the various components, i.e., data encoding, query, key and value. Hence, we give a brief review about it here.

In general, an ansatz, a.k.a. parameterized quantum circuit [25], has the form $U(\boldsymbol{\theta}) = \prod_j U_j(\theta_j)V_j$, where $U_j(\theta_j) = \exp(-i\theta_j P_j/2)$ and V_j denotes a fixed operator such as Identity, CNOT and so on. Here, P_j denotes a Pauli operator. Due to the numerous choices of the form of V_j, various kinds of ansatzes can be used. In this paper, we use the strongly entangled ansatz [23] shown in Fig. 2 in QSAL. This circuit has $n(D+2)$ parameters in total for n qubits and D repeated layers.

2.3 Loss Function

Consider the data set $\mathcal{D} := \{({}^{(m)}\boldsymbol{x}_1, {}^{(m)}\boldsymbol{x}_2, \ldots, {}^{(m)}\boldsymbol{x}_{S_m}), {}^{(m)}y\}_{m=1}^{N_s}$, where there are in total N_s sequences or samples and each has S_m words with a label ${}^{(m)}y \in \{0, 1\}$. Here, we assume each word is embedded as a d-dimensional vector, i.e., ${}^{(m)}\boldsymbol{x}_s \in \mathbb{R}^d$. The whole procedure of QSANN is depicted in Fig. 3, which mainly consists of L QSALs to extract hidden features and one fully-connected layer to complete the binary prediction task. Here, the mean squared error [44] is employed as the loss function:

$$\mathcal{L}\left(\boldsymbol{\Theta}, \boldsymbol{w}, b; \mathcal{D}\right) = \frac{1}{2N_s} \sum_{m=1}^{N_s} \left({}^{(m)}\hat{y} - {}^{(m)}y\right)^2 + \text{RegTerm}, \tag{8}$$

where the predicted value ${}^{(m)}\hat{y}$ is defined as ${}^{(m)}\hat{y} := \sigma\left(\boldsymbol{w}^\top \cdot \frac{1}{S_m}\sum_{s=1}^{S_m} {}^{(m)}\boldsymbol{y}_s^{(L)} + b\right)$ with $\boldsymbol{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$ denoting the weights and bias of the final fully-connected layer, $\boldsymbol{\Theta}$ denoting all parameters in the ansatz, σ denoting the sigmoid activation function and 'RegTerm' being the regularization term to avoid overfitting in the training process.

Combining Eqs. (3) - (7), we know each output of QSAL is dependent on all its inputs, i.e.,

$$\begin{aligned} {}^{(m)}\boldsymbol{y}_s^{(l)} &:= {}^{(m)}\boldsymbol{y}_s^{(l)}\left(\boldsymbol{\theta}_q^{(l)}, \boldsymbol{\theta}_k^{(l)}, \boldsymbol{\theta}_v^{(l)}; \{{}^{(m)}\boldsymbol{y}_i^{(l-1)}\}_{i=1}^{S_m}\right) \\ &= {}^{(m)}\boldsymbol{y}_s^{(l-1)} + \sum_{j=1}^{S_m} \tilde{\alpha}_{s,j}^{(l)}\left(\boldsymbol{\theta}_q^{(l)}, \boldsymbol{\theta}_k^{(l)}; \{{}^{(m)}\boldsymbol{y}_i^{(l-1)}\}_{i=1}^{S_m}\right) \cdot \boldsymbol{o}_j^{(l)}\left(\boldsymbol{\theta}_v^{(l)}; {}^{(m)}\boldsymbol{y}_j^{(l-1)}\right), \end{aligned} \tag{9}$$

where ${}^{(m)}\boldsymbol{y}_s^{(0)} = {}^{(m)}\boldsymbol{x}_s$ and $1 \le s \le S_m$, $1 \le l \le L$. Here, the regularization term is defined as

$$\text{RegTerm} := \frac{\lambda}{2d}\|\boldsymbol{w}\|^2 + \frac{\gamma}{2d}\sum_{s=1}^{S_m} \| {}^{(m)}\boldsymbol{x}_s\|^2, \tag{10}$$

where $\lambda, \gamma \ge 0$ are two regularization coefficients.

With the loss function defined in Eq. (8), we can optimize its parameters by (stochastic) gradient-descent [45]. The analytical gradient analysis can be found in the Appendix. Finally, with the above preparation, we could train our QSANN to get the optimal (or sub-optimal) parameters. See Algorithm 1 for details on the training procedure. We remark that if the loss converges during training or the maximum number of iterations is reached, the optimization stops.

2.4 Analysis of QSANN

According to the definition of the Quantum Self-Attention Layer, for a sequence with S words, we need $S(d + 2)$ Pauli measurements to obtain the d-dimensional value vectors as well as the queries and keys for all words from the quantum device. After that, we need to compute S^2 self-attention coefficients for all S^2 pairs of words on the classical computer. In general, QSANN takes advantage of quantum devices' efficiency in processing high-dimensional data while outsourcing some calculations to classical computers. This approach keeps the quantum circuit depth low and thus makes QSANN robust to low-level noise common in near-term quantum devices. This beneficial attribute is further verified by numerical results in the next section, where we test QSANN against noise.

Algorithm 1 QSANN training for text classification

Input: The training data set $\mathcal{D} := \{({}^{(m)}\boldsymbol{x}_1, {}^{(m)}\boldsymbol{x}_2, \ldots, {}^{(m)}\boldsymbol{x}_{S_m}), {}^{(m)}y\}_{m=1}^{N_s}$, $EPOCH$, number of QSALs L and optimization procedure

Output: The final ansatz parameters Θ^*, weight $\boldsymbol{w}^*, b^*$

Initialize the ansatz parameters Θ, weight $\boldsymbol{w}$ from Gaussian distribution $\mathcal{N}(0, 0.01)$ and the bias b to 0.
for $ep = 1, \ldots, EPOCH$ **do**
  for $m = 1, \ldots, N_s$ **do**
    Apply the encoder ansatz U_{enc} to each of ${}^{(m)}\boldsymbol{x}_s$ to get the corresponding quantum state $|\psi_s\rangle$, cf. (3).
    Apply U_q and U_k to $|\psi_s\rangle$ and measure the Pauli-Z expectations to get $\langle Z_q\rangle_s, \langle Z_k\rangle_s$, cf. (4), and then calculate the quantum self-attention coefficients $\alpha_{s,j}$, cf. (7).
    Apply U_v and measure a series of Pauli expectations to get $\boldsymbol{o}_s$, cf. (5), and then compute the output $\{\boldsymbol{y}_s^{(l)}\}$ of the l-th QSAL, cf. (6).
    Repeat 4-6 L times to get the output $\{\boldsymbol{y}_s^{(L)}\}$ of the L-th QSAL.
    Average $\{\boldsymbol{y}_s^{(L)}\}$ and through a fully-connected layer to obtain the predicted value ${}^{(m)}\hat{y}$.
    Calculate the mean squared error in (8) and update the parameters through the optimization procedure.
  end for
  if the stopping condition is met **then**
    Break.
  end if
end for

In short, our QSANN first encodes words into a large quantum Hilbert space as the feature space and then projects them back to low-dimensional classical feature space by quantum measurement. Recent works have proved rigorous quantum advantages on some classification tasks by utilizing high-dimensional quantum feature space [46] and projected quantum models [47]. Thus, we expect that our QSANN might also have the potential advantage of digging out some hidden features that are classically intractable. In the following section, we carry out numerical simulations of QSANN on several data sets to evaluate its performance on binary text classification tasks.

3 Numerical Results

In order to demonstrate the performance of our proposed QSANN, we have conducted numerical experiments on public data sets, where the quantum part was accomplished via classical simulation. Concretely, we first exhibit the better performance of QSANN by comparing it with i) the syntactic analysis-based quantum model [40] on two simple tasks, i.e., MC and RP, ii) the *classical self-attention neural network* (CSANN) and the naive method on three public sentiment analysis data sets, i.e., Yelp, IMDb and Amazon [48]. Then we show the reasonableness of our particular quantum self-attention GPQSA via visualization of self-attention coefficients. Finally, the noisy experiments are performed to show the robustness of QSANN to noisy quantum channels. All the simulations and optimization

Data set	n	d	D_{enc}	$D_{q/k/v}$	λ	γ	LR
MC	2	6	1	1	0	0	0.008
RP	4	24	4	5	0.2	0.4	0.008
Yelp	4	12	1	1	0.2	0.2	0.008
IMDb	4	12	1	1	0.002	0.002	0.002
Amazon	4	12	1	2	0.2	0.2	0.008

Table 1: Overview of hyper-parameter settings. Here, 'LR' denotes learning rate, D_{enc}, D_q, D_k, D_v denote the depths of the corresponding ansatzes and $d = n(D_{enc} + 2)$.

loop are implemented via Paddle Quantum[2] on the PaddlePaddle Deep Learning Platform [49].

3.1 Data Sets

The two simple synthetic data sets we employed come directly from [40], which are named as MC and RP, respectively. MC contains 17 words and 130 sentences (70 train + 30 development + 30 test) with 3 or 4 words each; RP has 115 words and 105 sentences (74 train + 31 test) with 4 words in each one. The other three data sets we use are real-world data sets available at [50] as the Sentiment Labelled Sentences Data Set. These data sets consist of reviews of restaurants, movies and products selected from Yelp, IMDb and Amazon, respectively. Each of the three data sets contains 1000 sequences, where half are labeled as '0' (for negative) and the other half as '1' (for positive). And each sequence contains several to dozens of words. We randomly select 80% as training sequences and the rest 20% as test ones.

3.2 Experimental Setting

In the experiments, we use a single self-attention layer for both QSANN and CSANN. As a comparison, we also perform the most straightforward method, i.e., averaging directly the embedded vectors of a sequence, followed by a fully-connected layer, which we call the 'Naive' method, on the three data sets of reviews.

In QSANN, all the encoder, query, key and value ansatzes have the same qubit number and are constructed according to Fig. 2, which are easily imiplementable on the NISQ devices. Specifically, assuming the n-qubit encoder ansatz has D_{enc} layers with $n(D_{enc} + 2)$ parameters, we could just set the dimension of the input vectors as $d = n(D_{enc} + 2)$. The depths of the query, key and value ansatzes are set to the same, and are at most the polynomial size of the qubit number n. The actual hyper-parameter settings on different data sets are concluded in Table 1. In addition, we choose $Z_1, \ldots, Z_n$, $X_1, \ldots, X_n$, $Y_1, \ldots, Y_n$ as the Pauli observables P_j in Eq. (5). For example, it is just required $3n$ observables when $D_{enc} = 1$. However, if $D_{enc} > 1$, we could also choose two-qubit observables Z_{12}, Z_{23} and so on. All the ansatz parameters $\boldsymbol{\Theta}$ and weight $\boldsymbol{w}$ are initialized from a Gaussian distribution with zero mean and 0.01 standard deviation, and the bias b is initialized to zero. Here, the ansatz parameters are not initialized uniformly from $[0, 2\pi)$ is mainly due to the residual

[2] https://github.com/paddlepaddle/Quantum

Method	MC			RP		
	# Paras	TrainAcc(%)	TestAcc(%)	# Paras	TrainAcc(%)	TestAcc(%)
DisCoCat [40]	40	83.10	79.80	168	90.60	**72.30**
QSANN	25	**100.00**	**100.00**	109	**95.35±1.95**	67.74±0.00

Table 2: Training accuracy and test accuracy of QSANN as well as DisCoCat on MC and RP tasks.

Method	Yelp		IMDb		Amazon	
	# Paras	TestAcc (%)	# Paras	TestAcc (%)	# Paras	TestAcc (%)
Naive	17	82.78±0.78	17	79.33±0.67	17	80.39±0.61
CSANN	785	83.11±0.89	785	79.67±0.83	785	83.22±1.28
QSANN	49	**84.79±1.29**	49	**80.28±1.78**	61	**84.25±1.75**

Table 3: Test accuracy of QSANN compared to CSANN and the naive method on Yelp, IMDb, and Amazon data sets. The highest accuracy in each column is indicated in bold font. On all the three data sets, QSANN achieves the highest accuracies among the three methods while using much fewer parameters than CSANN.

scheme applied in Eq. (6). During the optimization iteration, we use Adam optimizer [51]. And we repeat each experiment 9 times with different parameter initializations to collect the average accuracy and the corresponding fluctuations.

In CSANN, we set $d = 16$ and the classical query, key and value matrices are also initialized from a Gaussian distribution with zero mean and 0.01 standard deviation. Except these, almost all other parameters are set the same as QSANN. These settings and initializations are the same in the naive method as well.

3.3 Results on MC and RP Tasks

The results on MC and RP tasks are summarized in Table 2. In the MC task, our method QSANN could easily achieve a 100% test accuracy while requiring only 25 parameters (18 in query-key-value part and 7 in fully-connected part). However, in DisCoCat, the authors use 40 parameters but get a test accuracy lower than 80%. This result strongly demonstrates the powerful ability of QSANN for binary text classification. Here, the parameters in the encoder part are not counted as they could be replaced by fixed representations such as pre-trained word embeddings. In the RP task, we get a higher training accuracy but a slightly lower test accuracy. However, we observe that both test accuracies are pretty low when compared with the training accuracy. It is mainly because there is a massive bias between the training set and test set, i.e., more than half of the words in the test set have not appeared in the training one. Hence, the test accuracy highly depends on random guessing.

3.4 Results on Yelp, IMDb and Amazon Data Sets

As there are no quantum algorithms for text classification on these three data sets before, we benchmark our QSANN with the classical self-attention neural network (CSANN). The naive method is

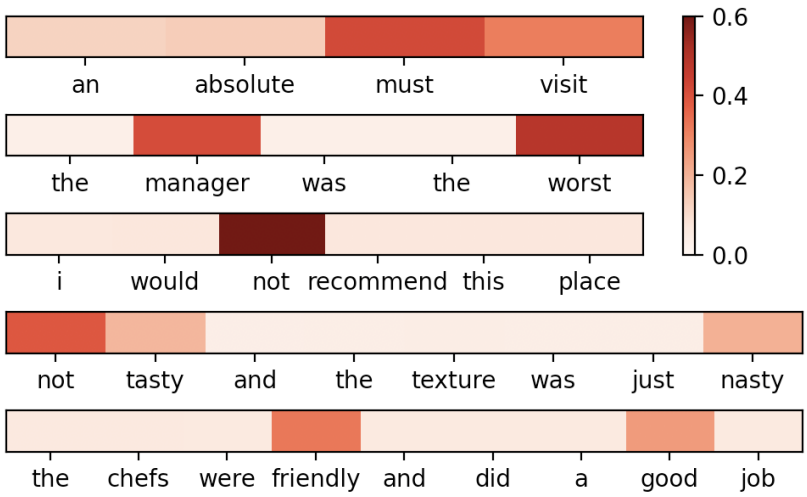


Figure 4: Heat maps of the averaged quantum self-attention coefficients for some selected test sequences from the Yelp data set, where a deeper color indicates a higher coefficient. Words that are more sentiment-related are generally assigned higher self-attention coefficients by our Gaussian projected quantum self-attention, implying the validity and interpretability of QSANN.

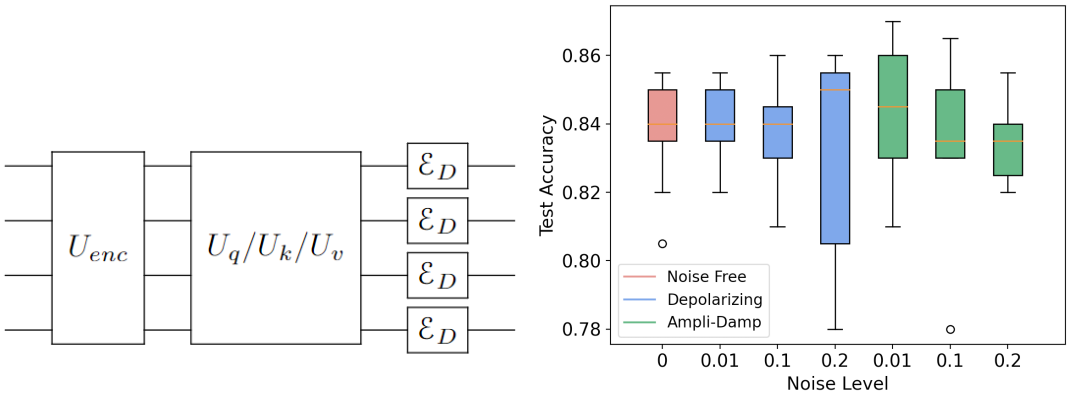
also listed for comparison. The results on Yelp, IMDb and Amazon data sets are summarized in Table 3. We can intuitively see that QSANN outperforms CSANN and the naive method on all three data sets. Specifically, CSANN has 785 parameters (768 in classical query-key-value part and 17 in fully-connected part) on all data sets. In comparison, QSANN has only 49 parameters (36 in query-key-value part and 13 in fully-connected part) on the Yelp and IMDb data sets and 61 parameters (48 in query-key-value part and 13 in fully-connected part) on the Amazon data set, improving the test accuracy by about 1% as well as saving more than 10 times the number of parameters. Therefore, QSANN could have a potential advantage for text classification.

3.5 Visualization of Self-Attention Coefficient

To intuitively demonstrate the reasonableness of the Gaussian projected quantum self-attention, in Fig. 4 we visualize the averaged quantum self-attention coefficients of some selected test sequences from the Yelp data set. Concretely, for a sequence, we calculate $\frac{1}{S}\sum_{s=1}^{S}\tilde{\alpha}_{s,j}$ for $j = 1, \ldots, S$ and visualize them via a heat map, where S is the number of words in this sequence and $\tilde{\alpha}_{s,j}$ is the quantum self-attention coefficient. As shown in the figure, words with higher quantum self-attention coefficients are indeed those that determine the emotion of a sequence, implying the power of QSANN for capturing the most relevant words in a sequence on text classification tasks.

3.6 Noisy Experimental Results on Yelp Data Set

Due to the limitations of the near-term quantum computers, we add experiments with noisy quantum circuits to demonstrate the robustness of QSANN on the Yelp data set. We consider the representative



(a) (b)

Figure 5: (a) The diagram for adding depolarizing channels in our simulated experiments. The amplitude damping channels are added in the same way. (b) Box plots of test accuracy on Yelp data set with depolarizing and amplitude damping noises. Each box contains 9 repeated experiments. The absence of a notable decrease in accuracy implies the noise-resilience attribute of QSANN.

channels [41] such as the depolarizing channel $\mathcal{E}_D(\rho)$ and the amplitude damping channel $\mathcal{E}_{AD}(\rho)$

$$\mathcal{E}_D(\rho) := (1-p)\,\rho + \frac{p}{3}\left(X\rho X + Y\rho Y + Z\rho Z\right) \tag{11}$$

$$\mathcal{E}_{AD}(\rho) := E_0\rho E_0^\dagger + E_1\rho E_1^\dagger, \tag{12}$$

with $E_0 = |0\rangle\langle 0| + \sqrt{1-p}|1\rangle\langle 1|$ and $E_1 = \sqrt{p}|0\rangle\langle 1|$ denoting the Kraus operators. Here, $\rho = |\psi\rangle\langle\psi|$ for a pure quantum state $|\psi\rangle$ and p denotes the noise level. As a regular way to analyze the effect of quantum noises, we add these single-qubit noisy channels in the final circuit layer to represent the whole system's noise, which is illustrated in Fig. 5(a).

We take the noise level p as 0.01, 0.1, 0.2 for these two noisy channels, respectively, and the box plots of test accuracies are depicted in Fig. 5(b). From the picture, we see the test accuracy of our QSANN almost does not decrease when the noise level is less than 0.1, and even when the noise level is up to 0.2, the overall test accuracy has only decreased a little, showing that QSANN is robust to these quantum noises.

4 Discussions

We have proposed a quantum self-attention neural network (QSANN) by introducing the self-attention mechanism to quantum neural networks. Specifically, the adopted Gaussian projected quantum self-attention exploits the exponentially large quantum Hilbert space as the quantum feature space, making QSANN have the potential advantage of mining some hidden correlations between words that are difficult to dig out classically. Numerical results show that QSANN outperforms the best-known QNLP method and a simple classical self-attention neural network for text classification on several public data sets. Moreover, using only shallow quantum circuits and Pauli measurements, QSANN can be easily implemented on near-term quantum devices and is noise-resilient, as implied by simulation results. We believe that this attempt to combine self-attention and quantum neural networks would

open up new avenues for QNLP as well as QML. As a future direction, more advanced techniques such as positional encoding and multi-head attention can be employed in quantum neural networks for generative models and other more complicated tasks.

Acknowledgements

We would like to thank Prof. Sanjiang Li and Prof. Yuan Feng for helpful discussions. G. L. acknowledges the support from the Baidu-UTS AI Meets Quantum project, the China Scholarship Council (No. 201806070139), and the Australian Research Council project (Grant No: DP180100691). Part of this work was done when X. Z. was a research intern at Baidu Research.

References

[1] John Preskill. Quantum computing 40 years later. *arXiv preprint arXiv:2106.10522*, 2021.

[2] Aram W Harrow and Ashley Montanaro. Quantum computational supremacy. *Nature*, 549(7671):203–209, 2017.

[3] Andrew M Childs and Wim van Dam. Quantum algorithms for algebraic problems. *Reviews of Modern Physics*, 82(1):1–52, jan 2010.

[4] Ashley Montanaro. Quantum algorithms: an overview. *npj Quantum Information*, 2(1):15023, nov 2016.

[5] Andrew M. Childs, Dmitri Maslov, Yunseong Nam, Neil J. Ross, and Yuan Su. Toward the first quantum simulation with quantum speedup. *Proceedings of the National Academy of Sciences*, 115(38):9456–9461, sep 2018.

[6] Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, Sep 2017.

[7] Fernando G.S.L. Brandao and Krysta M. Svore. Quantum Speed-Ups for Solving Semidefinite Programs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 415–426. IEEE, oct 2017.

[8] Feihu Xu, Xiongfeng Ma, Qiang Zhang, Hoi-Kwong Lo, and Jian-Wei Pan. Secure quantum key distribution with realistic devices. *Reviews of Modern Physics*, 92(2):25002, 2020.

[9] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan. Quantum computational chemistry. *Reviews of Modern Physics*, 92(1):015003, mar 2020.

[10] Yudong Cao, Jonathan Romero, Jonathan P Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, Ian D. Kivlichan, Tim Menke, Borja Peropadre, Nicolas P D Sawaya, Sukin Sim, Libor Veis, and Alán Aspuru-Guzik. Quantum Chemistry in the Age of Quantum Computing. *Chemical Reviews*, 119(19):10856–10915, oct 2019.

[11] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical Review Letters*, 113(3):130503, Sep 2014.

[12] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R. McClean. Power of data in quantum machine learning. *Nature Communications*, 12(1):2631, dec 2021.

[13] Maria Schuld and Francesco Petruccione. *Machine Learning with Quantum Computers*. 2021.

[14] John Preskill. Quantum Computing in the NISQ era and beyond. *Quantum*, 2:79, Aug 2018.

[15] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando GSL Brandao, David A Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[16] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, et al. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, 2020.

[17] Kishor Bharti, Alba Cervera-Lierta, Thi Ha Kyaw, Tobias Haug, Sumner Alperin-Lea, Abhinav Anand, Matthias Degroote, Hermanni Heimonen, Jakob S Kottmann, Tim Menke, et al. Noisy intermediate-scale quantum (nisq) algorithms. *arXiv preprint arXiv:2101.08448*, 2021.

[18] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, pages 1–29, aug 2021.

[19] Suguru Endo, Zhenyu Cai, Simon C Benjamin, and Xiao Yuan. Hybrid Quantum-Classical Algorithms and Quantum Error Mitigation. *Journal of the Physical Society of Japan*, 90(3):032001, mar 2021.

[20] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1):4213, dec 2014.

[21] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A Quantum Approximate Optimization Algorithm. *arXiv:1411.4028*, pages 1–16, Nov 2014.

[22] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, Mar 2019.

[23] Maria Schuld, Alex Bocharov, Krysta M. Svore, and Nathan Wiebe. Circuit-centric quantum classifiers. *Physical Review A*, 101(3):032308, Mar 2020.

[24] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, Sep 2018.

[25] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, Jun 2019.

[26] Edward Farhi and Hartmut Neven. Classification with Quantum Neural Networks on Near Term Processors. *arXiv:1802.06002*, pages 1–21, Feb 2018.

[27] Alessandro Sordoni, Jian-Yun Nie, and Yoshua Bengio. Modeling term dependencies with quantum language models for IR. In *Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval - SIGIR '13*, page 653, New York, New York, USA, 2013. ACM Press.

[28] Peng Zhang, Jiabin Niu, Zhan Su, Benyou Wang, Liqun Ma, and Dawei Song. End-to-End Quantum-Like Language Models with Application to Question Answering. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), 2018.

[29] Yazhou Zhang, Dawei Song, Peng Zhang, Xiang Li, and Panpan Wang. A quantum-inspired sentiment representation model for twitter sentiment analysis. *Applied Intelligence*, 49(8):3093–3108, 2019.

[30] Ivano Basile and Fabio Tamburini. Towards quantum language models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1840–1849, 2017.

[31] William Zeng and Bob Coecke. Quantum algorithms for compositional natural language processing. *arXiv preprint arXiv:1608.01406*, 2016.

[32] Konstantinos Meichanetzidis, Stefano Gogioso, Giovanni De Felice, Nicolò Chiappori, Alexis Toumi, and Bob Coecke. Quantum natural language processing on near-term quantum computers. *arXiv preprint arXiv:2005.04147*, 2020.

[33] Nathan Wiebe, Alex Bocharov, Paul Smolensky, Matthias Troyer, and Krysta M Svore. Quantum language processing. *arXiv preprint arXiv:1902.05162*, 2019.

[34] Samuel Yen-Chi Chen, Shinjae Yoo, and Yao-Lung L Fang. Quantum long short-term memory. *arXiv preprint arXiv:2009.01783*, 2020.

[35] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pages 6000–6010, 2017.

[37] Xiangpeng Li, Jingkuan Song, Lianli Gao, Xianglong Liu, Wenbing Huang, Xiangnan He, and Chuang Gan. Beyond rnns: Positional self-attention with co-attention for video question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8658–8665, 2019.

[38] Qipeng Guo, Xipeng Qiu, Pengfei Liu, Xiangyang Xue, and Zheng Zhang. Multi-scale self-attention for text classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7847–7854, 2020.

[39] Peter Cha, Paul Ginsparg, Felix Wu, Juan Carrasquilla, Peter L McMahon, and Eun-Ah Kim. Attention-based quantum tomography. *arXiv preprint arXiv:2006.12469*, 2020.

[40] Robin Lorenz, Anna Pearson, Konstantinos Meichanetzidis, Dimitri Kartsaklis, and Bob Coecke. Qnlp in practice: Running compositional models of meaning on a quantum computer. *arXiv preprint arXiv:2102.12846*, 2021.

[41] Michael A. Nielsen and Isaac Chuang. Quantum Computation and Quantum Information. *American Journal of Physics*, 70(5):558–559, May 2002.

[42] Charles A Micchelli, Yuesheng Xu, and Haizhang Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(12), 2006.

[43] Riccardo Di Sipio, Jia-Hong Huang, Samuel Yen-Chi Chen, Stefano Mangini, and Marcel Worring. The dawn of quantum natural language processing. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8612–8616. IEEE, 2022.

[44] Eric R. Ziegel, E. L. Lehmann, and George Casella. Theory of Point Estimation. *Technometrics*, 41(3):274, Aug 1999.

[45] Léon Bottou. Stochastic Learning. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 3176, pages 146–168. 2004.

[46] Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, pages 1–5, 2021.

[47] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature Communications*, 12(1):1–9, 2021.

[48] Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 597–606, 2015.

[49] Yanjun Ma, Dianhai Yu, Tian Wu, and Haifeng Wang. PaddlePaddle: An Open-Source Deep Learning Platform from Industrial Practice. *Frontiers of Data and Domputing*, 1(1):105–115, 2019.

[50] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[51] Diederik P. Kingma and Jimmy Lei Ba. Adam: A method for stochastic optimization. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, Dec 2015.

[52] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

Appendix for Quantum Self-Attention Neural Networks for Text Classification

1 Analytical Gradients

Here, we give the stochastic analytical partial gradients of the loss function with regard to its parameters as follows. We first consider the parameters in the last quantum self-attention neural network layer, i.e., $\boldsymbol{\theta}_q^{(L)}, \boldsymbol{\theta}_k^{(L)}, \boldsymbol{\theta}_v^{(L)}$, and the final fully-connected layer, i.e., $\boldsymbol{w}, b$, and then the parameters in the front layers could be evaluated in a similar way and be updated through back-propagation algorithm [52]. Given the m-th data sample $\{(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{S_m}), y\}$ (here, we omit (m) in the left superscript for writing convenience, the same below), we have

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \tilde{\sigma} \cdot \frac{1}{S_m} \sum_{s=1}^{S_m} \boldsymbol{y}_s^{(L)} + \frac{\lambda}{d} \boldsymbol{w}, \qquad \frac{\partial \mathcal{L}}{\partial b} = \tilde{\sigma}, \tag{S1}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} = \tilde{\sigma} \cdot \frac{1}{S_m} \cdot \boldsymbol{w}, \tag{S2}$$

where $\tilde{\sigma} = (\sigma - y) \cdot \sigma (1 - \sigma)$ and σ denotes the abbreviation of $\sigma \left(\boldsymbol{w}^\top \cdot \frac{1}{S_m} \sum_{s=1}^{S_m} \boldsymbol{y}_s^{(L)} + b \right)$. We also have

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_v^{(L)}} = \sum_{s=1}^{S_m} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} \right)^\top \sum_{j=1}^{S_m} \frac{\partial \boldsymbol{y}_s^{(L)}}{\partial \boldsymbol{o}_j^{(L)}} \cdot \frac{\partial \boldsymbol{o}_j^{(L)}}{\partial \boldsymbol{\theta}_v^{(L)}}, \tag{S3}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_q^{(L)}} = \sum_{s=1}^{S_m} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} \right)^\top \sum_{j=1}^{S_m} \frac{\partial \boldsymbol{y}_s^{(L)}}{\partial \alpha_{s,j}^{(L)}} \cdot \frac{\partial \alpha_{s,j}^{(L)}}{\partial \langle Z_q \rangle_s} \cdot \frac{\partial \langle Z_q \rangle_s}{\partial \boldsymbol{\theta}_q^{(L)}}, \tag{S4}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_k^{(L)}} = \sum_{s=1}^{S_m} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} \right)^\top \sum_{j=1}^{S_m} \frac{\partial \boldsymbol{y}_s^{(L)}}{\partial \alpha_{s,j}^{(L)}} \cdot \sum_{i=1}^{S_m} \frac{\partial \alpha_{s,j}^{(L)}}{\partial \langle Z_k \rangle_i} \cdot \frac{\partial \langle Z_k \rangle_i}{\partial \boldsymbol{\theta}_k^{(L)}}, \tag{S5}$$

where $\partial \boldsymbol{y}_s^{(L)} / \partial \boldsymbol{o}_j^{(L)} = \alpha_{s,j}^{(L)}$, $\partial \boldsymbol{y}_s^{(L)} / \partial \alpha_{s,j}^{(L)} = \boldsymbol{o}_j^{(L)}$, $\partial \alpha_{s,j}^{(L)} / \partial \langle Z_q \rangle_s = - \sum_{i=1}^{S_m} \partial \alpha_{s,j}^{(L)} / \partial \langle Z_k \rangle_i$ and

$$\frac{\partial \alpha_{s,j}^{(L)}}{\partial \langle Z_k \rangle_i} = -\alpha_{s,j}^{(L)} \left(\alpha_{s,i}^{(L)} - \delta_{ij} \right) \cdot 2 \left(\langle Z_q \rangle_s - \langle Z_k \rangle_i \right),$$

$$\delta_{ij} = \begin{cases} 1, & i = j \\ 0, & \text{otherwise.} \end{cases} \tag{S6}$$

Furthermore, the last three partial derivatives of Eqs. (S3), (S4) and (S5) could be evaluated directly on the quantum computers via the parameter shift rule [24]. For example,

$$\frac{\partial \langle Z_q \rangle_s}{\partial \theta_{q,j}^{(L)}} = \frac{1}{2} \left(\langle Z_q \rangle_{s,+} - \langle Z_q \rangle_{s,-} \right), \tag{S7}$$

where $\langle Z_q \rangle_{s,\pm} := \langle \psi_s | U_{q,\pm}^\dagger Z U_{q,\pm} | \psi_s \rangle$ and $U_{q,\pm} := U_q \left(\boldsymbol{\theta}_{q,-j}^{(L)}, \theta_{q,j}^{(L)} \pm \frac{\pi}{2} \right)$.

Finally, in order to derive the partial derivatives of the parameters in the front layers, we also need the following:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_i^{(L-1)}} = & \frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_i^{(L)}} + \sum_{s=1}^{S_m} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} \right)^\top \frac{\partial \boldsymbol{y}_s^{(L)}}{\partial \boldsymbol{o}_i^{(L)}} \cdot \frac{\partial \boldsymbol{o}_i^{(L)}}{\partial \boldsymbol{y}_i^{(L-1)}} \\ & + \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_i^{(L)}} \right)^\top \sum_{j=1}^{S_m} \frac{\partial \boldsymbol{y}_i^{(L)}}{\partial \alpha_{i,j}^{(L)}} \cdot \frac{\partial \alpha_{i,j}^{(L)}}{\partial \langle Z_q \rangle_i} \cdot \frac{\partial \langle Z_q \rangle_i}{\partial \boldsymbol{y}_i^{(L-1)}} \\ & + \sum_{s=1}^{S_m} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} \right)^\top \sum_{j=1}^{S_m} \frac{\partial \boldsymbol{y}_s^{(L)}}{\partial \alpha_{s,j}^{(L)}} \cdot \frac{\partial \alpha_{s,j}^{(L)}}{\partial \langle Z_k \rangle_i} \cdot \frac{\partial \langle Z_k \rangle_i}{\partial \boldsymbol{y}_i^{(L-1)}}, \end{aligned} \tag{S8}$$

where the four terms denote the residual, value, query and key parts, respectively, and each sub-term can be evaluated similarly to the above analysis. With the above preparation, we could easily calculate every parameter's gradient and update these parameters accordingly.