SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

April 2024, Vol. 67, Iss. 4, 142501:1–142501:13
https://doi.org/10.1007/s11432-023-3879-7

Quantum self-attention neural networks for text classification

Guangxi LI[1,2], Xuanqiang ZHAO[1,3] & Xin WANG[1,4*]

[1]*Institute for Quantum Computing, Baidu Research, Beijing* 100193, *China;*
[2]*Centre for Quantum Software and Information, University of Technology Sydney, Sydney* NSW 2007, *Australia;*
[3]*QICI Quantum Information and Computation Initiative, Department of Computer Science, The University of Hong Kong, Hong Kong* 999077, *China;*
[4]*Thrust of Artificial Intelligence, Information Hub, Hong Kong University of Science and Technology (Guangzhou), Guangzhou* 511453, *China*

Received 28 February 2023/Revised 18 April 2023/Accepted 28 September 2023/Published online 27 March 2024

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 compensate 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 and showcases resilience to quantum neural network architectures.

Keywords quantum neural networks, self-attention, natural language processing, text classification, parameterized quantum circuits

1 Introduction

Quantum computing is a promising paradigm [1] for fast computations that can provide substantial advantages in solving valuable problems [2–6]. 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 [6,11–13].

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 [17–19] appear to be the best hope for obtaining a quantum advantage in fields such as quantum chemistry [20], optimization [21], and machine learning [22–27]. In particular, those algorithms dealing with machine learning problems, by employing parameterized quantum circuits (PQCs) [28] (also called quantum neural networks (QNNs) [29]), show great potential in the field of quantum machine learning (QML). See [12,30–40] for some recent progress on the theory and applications of QNNs in many directions. 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.

* Corresponding author (email: felixxinwang@hkust-gz.edu.cn)

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, and text classification, many of which have greatly facilitated our lives. 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 studies focus on quantum-inspired language models [41–44] 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 [45–48]. 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 QNNs. Our motivation comes from the excellent performance of self-attention on various NLP tasks such as language modeling [49], machine translation [50], question answering [51], and text classification [52]. We also note that a recently proposed method [53] 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 (QSAL) 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 (CSANN) on test accuracy, implying the 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 GPQSA, 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 [54] and simple CSANNs on several public data sets for text classification. Numerical results also imply that QSANN is resilient to both quantum noises and QNN architectures.

Quantum basics. Here, some basic concepts about quantum computing necessary for this paper are briefly introduced (for more details, see [55]). 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 $|\cdot\rangle$ 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 $R_P(\theta) := \exp(-\mathrm{i}\theta P/2) = \cos\frac{\theta}{2}I - \mathrm{i}\sin\frac{\theta}{2}P$, where the rotation angle $\theta \in [0, 2\pi)$ and $P \in \{X, Y, Z\}$. In this paper, multiple-qubit quantum gates mainly include the identity gate I, the CNOT gate and the tensor product of single-qubit gates, e.g., $Z \otimes Z$, $Z \otimes I$, and $Z^{\otimes n}$.

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 $\langle\psi|O|\psi\rangle$. This work focuses on the hardware-efficient Pauli measurements, i.e., setting O as Pauli operators or their tensor products. For instance, we could choose $Z_1 = Z \otimes I^{\otimes(n-1)}$, $X_2 = I \otimes X \otimes I^{\otimes(n-2)}$, and $Z_1Z_2 = Z \otimes Z \otimes I^{\otimes(n-2)}$, with n qubits in total.

Text classification. As one of the central and basic tasks in the NLP field, 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.

Self-attention mechanism. In a self-attention neural network layer [50], 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 $W_q x_s$, $W_k x_s$, $W_v x_s$, respectively, and by applying the inner product on the query and key parts, the output is computed as

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

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

2 Method

In this section, we will introduce the QSANN in detail, which mainly consists of QSAL, loss function, analytical gradients and analysis.

2.1 QSAL

In the classical self-attention mechanism [50], 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 Figure 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.,

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

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

$$\boldsymbol{o}_s := \Big[\langle P_1\rangle_s \ \langle P_2\rangle_s \ \cdots \ \langle P_d\rangle_s\Big]^{\mathrm{T}}, \tag{5}$$

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 (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:

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

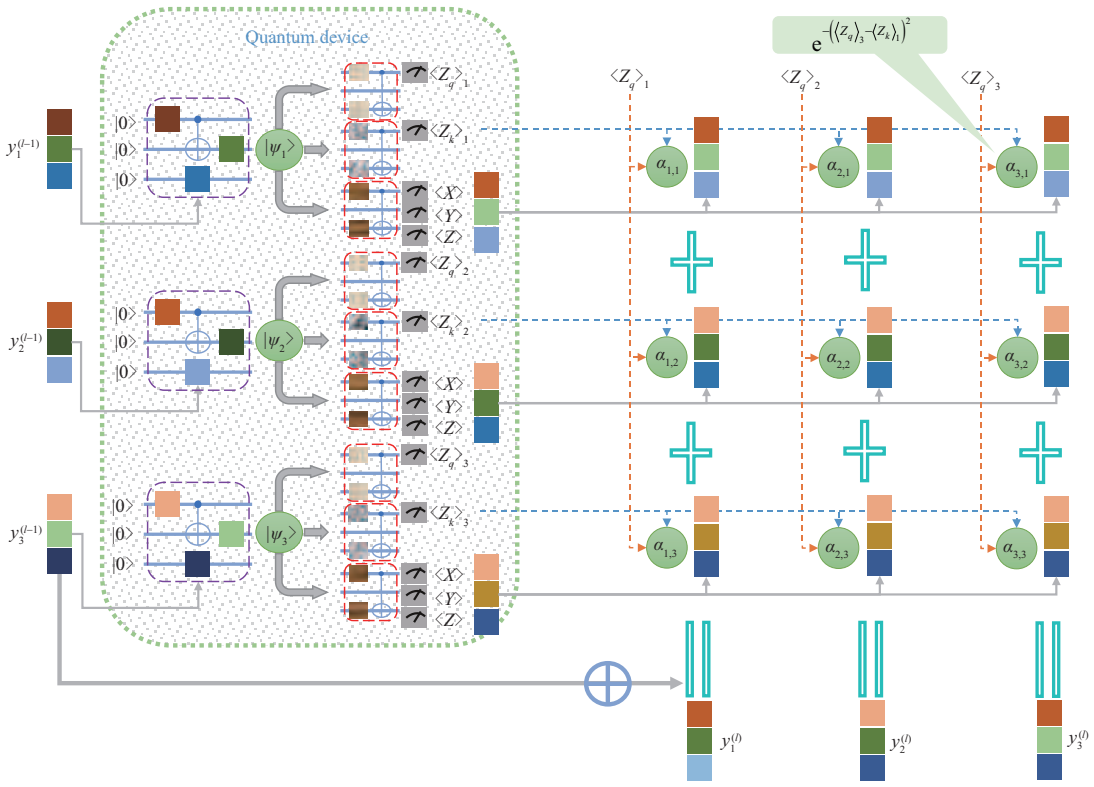

Figure 1 (Color online) Sketch of a QSAL. On quantum devices, the classical inputs $\{\boldsymbol{y}_s^{(l-1)}\}$ are used as the rotation angles of quantum ansatzes (purple dashed boxes) to encode them into their corresponding quantum states $\{|\psi_s\rangle\}$. Then, a set of three ansatzes (in red dashed boxes) representing query, key, and value is applied to each state. Note that it is the same set of ansatzes applied to all the input states. 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).

Here in (6), we adopt a residual scheme when computing the output, which is analogous to [50].

GPQSA. 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 (7), which we call 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 is pretty easy 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 [27,56] 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 for building quantum self-attention are also worth future study.

Remark. During the preparation of this manuscript, we became aware that Ref. [57] 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.

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 of it here.

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

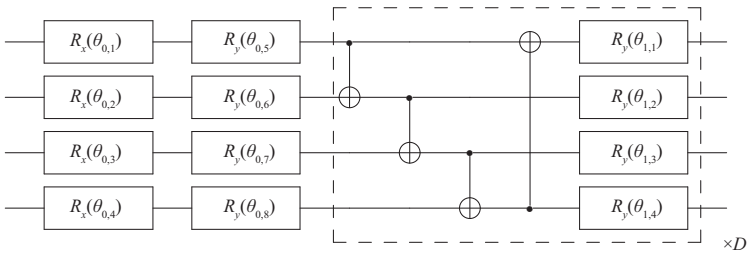

Figure 2 Ansatz used in QSAL. The first two columns denote the R_x-R_y rotations on each single-qubit subspace, followed by repeated CNOT gates and single-qubit R_y rotations. The block circuit in the dashed box is repeated D times to enhance the expressive power of the ansatz.

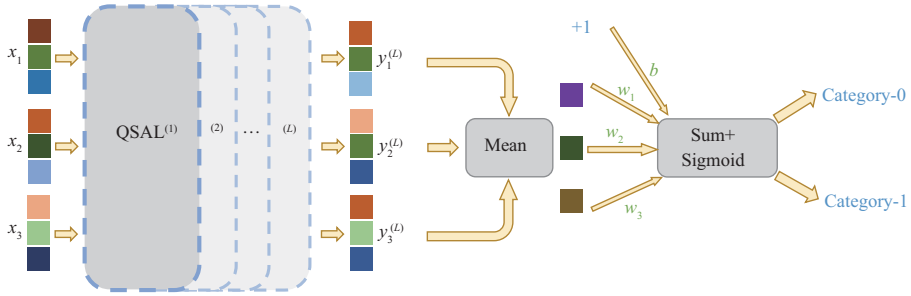

Figure 3 (Color online) Sketch of QSANN, where a sequence of classical vectors $\{\boldsymbol{x}_s\}$ first goes through L QSALs to obtain the corresponding sequence of feature vectors $\{\boldsymbol{y}_s^{(L)}\}$, then through the average operation, and finally through the fully-connected layer for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

In general, an ansatz, a.k.a. PQC [28], has the form $U(\boldsymbol{\theta}) = \prod_j U_j(\theta_j)V_j$, where $U_j(\theta_j) = \exp(-\mathrm{i}\theta_j P_j/2)$ and V_j denotes a fixed operator such as Identity and CNOT. 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 Figure 2 in QSAL. This circuit has $n(D+2)$ parameters in total for n qubits and D repeated layers.

2.2 Loss function

Consider the data set $\mathcal{D} := \{(\boldsymbol{x}_{m;1}, \boldsymbol{x}_{m;2}, \ldots, \boldsymbol{x}_{m;S_m}), y_m\}_{m=1}^{N_s}$, where there are in total N_s sequences or samples and each has S_m words with a label $y_m \in \{0, 1\}$. Here, we assume each word is embedded as a d-dimensional vector, i.e., $\boldsymbol{x}_{m;s} \in \mathbb{R}^d$. The whole procedure of QSANN is depicted in Figure 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 [58] is employed as the loss function:

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

where the predicted value $\hat{y}_m$ is defined as $\hat{y}_m := \sigma(\boldsymbol{w}^{\mathrm{T}} \cdot \frac{1}{S_m}\sum_{s=1}^{S_m}\boldsymbol{y}_{m;s}^{(L)} + b)$ 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 (3)–(7), we know each output of QSAL is dependent on all its inputs, i.e.,

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

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

$$\mathrm{RegTerm} := \frac{\lambda}{2d}\|\boldsymbol{w}\|^2 + \frac{\gamma}{2d}\sum_{s=1}^{S_m}\|\boldsymbol{x}_{m;s}\|^2, \tag{10}$$

Algorithm 1 QSANN training for text classification

Input: The training data set $\mathcal{D} := \{(\boldsymbol{x}_{m;1}, \boldsymbol{x}_{m;2}, \ldots, \boldsymbol{x}_{m;S_m}), y_m\}_{m=1}^{N_s}$, EPOCH, number of QSALs L and optimization procedure.
Output: The final ansatz parameters Θ^*, weight $\boldsymbol{w}^*, b^*$.

```
Initialize the ansatz parameters Θ, weight w from Gaussian distribution N(0, 0.01) and the bias b to 0;
for ep = 1, ..., EPOCH do
    for m = 1, ..., N_s do
        Apply the encoder ansatz U_enc to each of x_{m;s} to get the corresponding quantum state |ψ_s⟩, cf. (3);
        Apply U_q and U_k to |ψ_s⟩ and measure the Pauli-Z expectations to get ⟨Z_q⟩_s, ⟨Z_k⟩_s, cf. (4), and then calculate the
        quantum self-attention coefficients α_{s,j}, cf. (7);
        Apply U_v and measure a series of Pauli expectations to get o_s, cf. (5), and then compute the output {y_s^(l)} of the l-th
        QSAL, cf. (6);
        Repeat 4–6 L times to get the output {y_s^(L)} of the L-th QSAL;
        Average {y_s^(L)} and through a fully-connected layer to obtain the predicted value ŷ_m;
        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
```

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

With the loss function defined in (8), we can optimize its parameters by (stochastic) gradient-descent [59]. The analytical gradient analysis can be found in Subsection 2.3. 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.

2.3 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., $\boldsymbol{w}, b$. Then the parameters in the front layers could be evaluated similarly and be updated through the back-propagation algorithm [60]. Given the m-th data sample $\{(\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_{S_m}), y\}$ (here, we omit m in the subscript for writing convenience, the same below), we have

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \tilde{\sigma} \cdot \frac{1}{S_m} \sum_{s=1}^{S_m} \boldsymbol{y}_s^{(L)} + \frac{\lambda}{d} \boldsymbol{w}, \qquad \frac{\partial \mathcal{L}}{\partial b} = \tilde{\sigma}, \tag{11}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} = \tilde{\sigma} \cdot \frac{1}{S_m} \cdot \boldsymbol{w}, \tag{12}$$

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

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_v^{(L)}} = \sum_{s=1}^{S_m} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} \right)^{\mathrm{T}} \sum_{j=1}^{S_m} \frac{\partial \boldsymbol{y}_s^{(L)}}{\partial \boldsymbol{o}_j^{(L)}} \cdot \frac{\partial \boldsymbol{o}_j^{(L)}}{\partial \boldsymbol{\theta}_v^{(L)}}, \tag{13}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_q^{(L)}} = \sum_{s=1}^{S_m} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} \right)^{\mathrm{T}} \sum_{j=1}^{S_m} \frac{\partial \boldsymbol{y}_s^{(L)}}{\partial \alpha_{s,j}^{(L)}} \cdot \frac{\partial \alpha_{s,j}^{(L)}}{\partial \langle Z_q \rangle_s} \cdot \frac{\partial \langle Z_q \rangle_s}{\partial \boldsymbol{\theta}_q^{(L)}}, \tag{14}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\theta}_k^{(L)}} = \sum_{s=1}^{S_m} \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{y}_s^{(L)}} \right)^{\mathrm{T}} \sum_{j=1}^{S_m} \frac{\partial \boldsymbol{y}_s^{(L)}}{\partial \alpha_{s,j}^{(L)}} \cdot \sum_{i=1}^{S_m} \frac{\partial \alpha_{s,j}^{(L)}}{\partial \langle Z_k \rangle_i} \cdot \frac{\partial \langle Z_k \rangle_i}{\partial \boldsymbol{\theta}_k^{(L)}}, \tag{15}$$

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

$$\begin{aligned} \frac{\partial \alpha_{s,j}^{(L)}}{\partial \langle Z_k \rangle_i} &= -\alpha_{s,j}^{(L)} \left(\alpha_{s,i}^{(L)} - \delta_{ij} \right) \cdot 2 \left(\langle Z_q \rangle_s - \langle Z_k \rangle_i \right), \\ \delta_{ij} &= \begin{cases} 1, & i = j, \\ 0, & \text{otherwise.} \end{cases} \end{aligned} \tag{16}$$

Furthermore, the last three partial derivatives of (13)–(15) could be evaluated directly on the quantum computers via the parameter shift rule [24]. For example,

$$\frac{\partial\langle Z_q\rangle_s}{\partial\theta_{q,j}^{(L)}}=\frac{1}{2}\left(\langle Z_q\rangle_{s,+}-\langle Z_q\rangle_{s,-}\right), \tag{17}$$

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

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

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}_i^{(L-1)}}=&\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}_i^{(L)}}+\sum_{s=1}^{S_m}\left(\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}_s^{(L)}}\right)^{\mathrm{T}}\frac{\partial\boldsymbol{y}_s^{(L)}}{\partial\boldsymbol{o}_i^{(L)}}\cdot\frac{\partial\boldsymbol{o}_i^{(L)}}{\partial\boldsymbol{y}_i^{(L-1)}}\\&+\left(\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}_i^{(L)}}\right)^{\mathrm{T}}\sum_{j=1}^{S_m}\frac{\partial\boldsymbol{y}_i^{(L)}}{\partial\alpha_{i,j}^{(L)}}\cdot\frac{\partial\alpha_{i,j}^{(L)}}{\partial\langle Z_q\rangle_i}\cdot\frac{\partial\langle Z_q\rangle_i}{\partial\boldsymbol{y}_i^{(L-1)}}\\&+\sum_{s=1}^{S_m}\left(\frac{\partial\mathcal{L}}{\partial\boldsymbol{y}_s^{(L)}}\right)^{\mathrm{T}}\sum_{j=1}^{S_m}\frac{\partial\boldsymbol{y}_s^{(L)}}{\partial\alpha_{s,j}^{(L)}}\cdot\frac{\partial\alpha_{s,j}^{(L)}}{\partial\langle Z_k\rangle_i}\cdot\frac{\partial\langle Z_k\rangle_i}{\partial\boldsymbol{y}_i^{(L-1)}},\end{aligned} \tag{18}$$

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.

2.4 Analysis of QSANN

According to the definition of the QSAL, 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 Section 3, where we test QSANN against noise.

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 studies have proved rigorous quantum advantages on some classification tasks by utilizing high-dimensional quantum feature space [61] and projected quantum models [12]. Thus, we expect that our QSANN might also have the potential advantage of digging out some hidden features that are classically intractable. Furthermore, the low-parameter variational quantum circuit exhibits the ability to achieve low generalization error [62] with few training data [31], providing further evidence for the effectiveness of our QSANN method. In Section 3, 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 [54] on two simple tasks, i.e., MC and RP, (ii) the CSANN and the naive method on three public sentiment analysis data sets, i.e., Yelp, IMDb, and Amazon [63]. Then we show the reasonableness of our particular quantum self-attention GPQSA via visualization of self-attention coefficients. Next, we perform noisy experiments to show the robustness of QSANN to noisy quantum channels. Finally, we perform noisy experiments with different ansatzes to demonstrate the resilience of QSANN to the architectures of QNNs. All the simulations and optimization loops are implemented via Paddle quantum[2)] on the PaddlePaddle deep learning platform [64].

2) https://github.com/paddlepaddle/Quantum.

Table 1 Overview of hyper-parameter settings[a)]

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

a) Here, 'LR' denotes learning rate, $D_{\text{enc}}, D_q, D_k, D_v$ denote the depths of the corresponding ansatzes and $d = n(D_{\text{enc}} + 2)$.

Table 2 Training accuracy and test accuracy of QSANN as well as DisCoCat on MC and RP tasks[a)]

Method	MC			RP		
	# Paras	TrainAcc (%)	TestAcc (%)	# Paras	TrainAcc (%)	TestAcc (%)
DisCoCat [54]	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

a) The highest accuracy in each column is indicated in bold font. In the MC task, QSANN could easily achieve a 100% test accuracy while requiring only 25 parameters. In the RP task, QSANN gets a higher training accuracy and a slightly lower test accuracy, because of the data set bias.

Data sets. The two simple synthetic data sets we employed come directly from [54], which are named 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 [65] 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.

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., directly averaging 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. Here, we note that only comparing these simple classical models is because there are still significant restrictions on current quantum hardware. It is pretty unfair to compare with the most potent classical models.

Remark. We note that due to the current limitations of quantum hardware, using mini- or small-scale tasks for benchmarking is a common practice in current QNLP research. Additionally, the quantum transformer is still in its infancy, and it may not be fair to directly compare it with the most advanced classical transformers or hybrid transformers [25] currently available. Despite all this, we believe QSANN provides a good starting point for demonstrating the potential advantages and applications of quantum computing in NLP, providing valuable experience and insights for more in-depth research in the future.

In QSANN, all the encoder, query, key, and value ansatzes have the same qubit number and are constructed according to Figure 2, which are easily implementable on the NISQ devices. Specifically, assuming the n-qubit encoder ansatz has D_{enc} layers with $n(D_{\text{enc}} + 2)$ parameters, we just set the dimension of the input vectors as $d = n(D_{\text{enc}} + 2)$. 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 (5). For example, it just required $3n$ observables when $D_{\text{enc}} = 1$. However, if $D_{\text{enc}} > 1$, we could also choose two-qubit observables Z_{12}, Z_{23}. 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 scheme applied in (6). During the optimization iteration, we use Adam optimizer [66]. 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 for these, almost all other parameters are set the same as QSANN. These settings and initializations are the same in the naive method as well.

Results on MC and RP tasks. The results on MC and RP tasks are summarized in Table 2. In

Table 3 Test accuracy of QSANN compared to CSANN and the naive method on Yelp, IMDb, and Amazon data sets[a)]

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**

a) The highest accuracy in each column is indicated in bold font. On all three data sets, QSANN achieves the highest accuracies among the three methods while using much fewer parameters than CSANN.

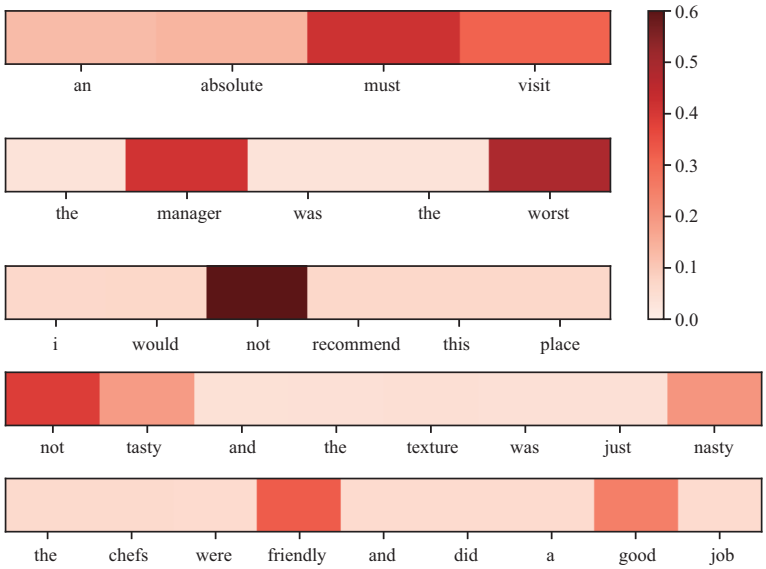

Figure 4 (Color online) Heat maps of the averaged quantum self-attention coefficients for some selected test sequences from the Yelp data set, where a deeper color indicates a higher coefficient. Words that are more sentiment-related are generally assigned higher self-attention coefficients by our GPQSA,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

the MC task, our method QSANN could easily achieve a 100% test accuracy while requiring only 25 parameters (18 in the query-key-value part and 7 in the 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 the 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.

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 CSANN. The naive method is 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 the classical query-key-value part and 17 in the fully-connected part) on all data sets. In comparison, QSANN has only 49 parameters (36 in the query-key-value part and 13 in the fully-connected part) on the Yelp and IMDb data sets and 61 parameters (48 in the query-key-value part and 13 in the 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.

Visualization of self-attention coefficient. To intuitively demonstrate the reasonableness of the GPQSA, in Figure 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 Figure 4, words with higher quantum self-attention

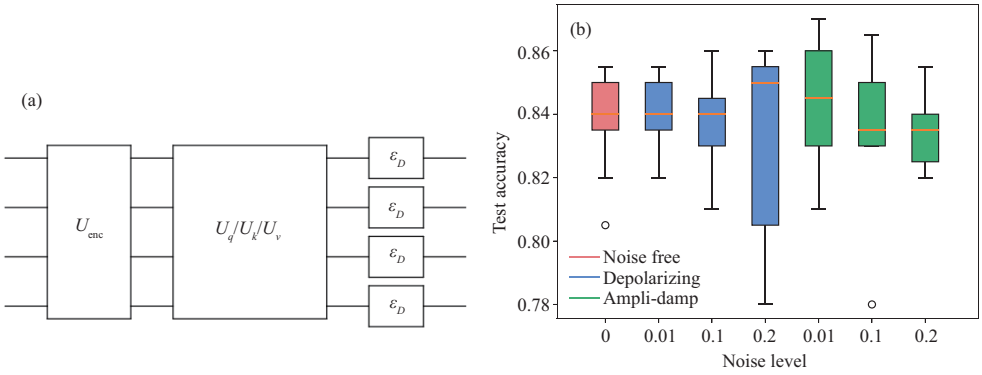

Figure 5 (Color online) (a) Diagram for adding depolarizing channels in our simulated experiments. The amplitude-damping channels are added in the same way. (b) Box plots of test accuracy on Yelp data set with depolarizing and amplitude damping noises. Each box contains nine repeated experiments. The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

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.

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 channels [55] such as the depolarizing channel $\mathcal{E}_D(\rho)$ and the amplitude-damping channel $\mathcal{E}_{AD}(\rho)$:

$$\mathcal{E}_D(\rho) := (1-p)\,\rho + \frac{p}{3}\left(X\rho X + Y\rho Y + Z\rho Z\right), \tag{19}$$

$$\mathcal{E}_{AD}(\rho) := E_0\rho E_0^\dagger + E_1\rho E_1^\dagger, \tag{20}$$

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 Figure 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 Figure 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.

Noisy experimental results with different ansatzes. Given the recent limitations of quantum hardware topology, some ansatzes are easier to implement than others. As such, exploring the performance of QSANN under different ansatzes is crucial to determining the difficulty level in deploying QSANN on current quantum hardware. Additionally, it is worth investigating which ansatz can most easily achieve optimal performance of QSANN for specific practical tasks.

In this subsection, we test QSANN using different ansatzes on both MC and RP data sets. As depicted in Figure 6, these ansatzes utilize different entanglement layers while keeping the single-qubit gates and the total number of parameters unchanged. Furthermore, a depolarizing channel with $p = 0.1$ is added to each ansatz, as shown in (19). Other settings remain the same as in Table 1. The final results are shown in Table 4, where we see that the performance of the four ansatz types is virtually identical. This directly indicates that QSANN is resilient to ansatz architectures.

4 Discussions

We have proposed a QSANN by introducing the self-attention mechanism to QNNs. Specifically, the adopted GPQSA 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 CSANN 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

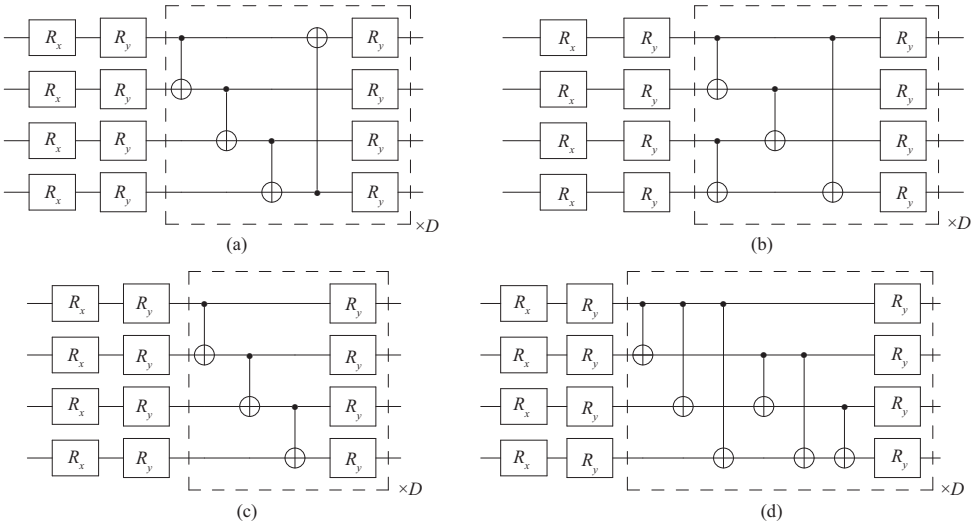

Figure 6 Four types of ansatzes used in QSANN. Each has a different entangled layer. (a) Ansatz-0; (b) Ansatz-1; (c) Ansatz-2; (d) Ansatz-3.

Table 4 Training accuracy and test accuracy of QSANN with four different ansatzes on MC and RP tasks

Method	MC		RP	
	TrainAcc (%)	TestAcc (%)	TrainAcc (%)	TestAcc (%)
Ansatz-0	100.00	100.00	94.74±1.20	67.74±0.00
Ansatz-1	100.00	100.00	94.71±0.11	67.03±0.72
Ansatz-2	100.00	100.00	94.71±0.93	67.38±0.36
Ansatz-3	100.00	100.00	94.74±0.15	67.74±0.00

quantum devices and is noise-resilient, as implied by simulation results. We believe that this attempt to combine self-attention and QNNs 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 QNNs for generative models and other more complicated tasks. Another exciting future research direction is to move toward large language models. However, we must realize that there are still many challenges and limitations to overcome, particularly in the NISQ era. Despite these challenges, our work represents a promising step towards this goal, and we are optimistic about the potential of quantum computing in NLP. As quantum hardware continues to evolve and improve, we anticipate that our methods can be gradually extended to more complex algorithms and tasks, unlocking new possibilities for QNLP research.

Acknowledgements This work was partially supported by Guangdong Provincial Quantum Science Strategic Initiative (Grant No. GDZX2303007). Guangxi LI acknowledges the support from Quantum Science Center of Guangdong-Hong Kong-Macao Greater Bay Area, Baidu-UTS AI Meets Quantum project, the China Scholarship Council (Grant No. 201806070139), and Australian Research Council Project (Grant No. DP180100691). Xin WANG was partially supported by Start-up Fund (Grant No. G0101000151) from The Hong Kong University of Science and Technology (Guangzhou), Innovation Program for Quantum Science and Technology (Grant No. 2021ZD0302901), and Education Bureau of Guangzhou Municipality. We would like to thank Prof. Sanjiang LI and Prof. Yuan FENG for their helpful discussions. We also thank Zihe WANG and Chenghong ZHU for their help related to the experiments. Part of this work was done when all of the authors were at Baidu Research.

References

1 Preskill J. Quantum computing 40 years later. 2021. ArXiv:2106.10522
2 Harrow A W, Montanaro A. Quantum computational supremacy. Nature, 2017, 549: 203–209
3 Childs A M, van Dam W. Quantum algorithms for algebraic problems. Rev Mod Phys, 2010, 82: 1–52
4 Montanaro A. Quantum algorithms: an overview. npj Quantum Inf, 2016, 2: 15023
5 Childs A M, Maslov D, Nam Y, et al. Toward the first quantum simulation with quantum speedup. Proc Natl Acad Sci USA, 2018, 115: 9456–9461
6 Biamonte J, Wittek P, Pancotti N, et al. Quantum machine learning. Nature, 2017, 549: 195–202
7 Brandao F G S L, Svore K M. Quantum speed-ups for solving semidefinite programs. In: Proceedingsof the 58th Annual Symposium on Foundations of Computer Science (FOCS), 2017. 415–426
8 Xu F, Ma X, Zhang Q, et al. Secure quantum key distribution with realistic devices. Rev Mod Phys, 2020, 92: 25002
9 McArdle S, Endo S, Aspuru-Guzik A, et al. Quantum computational chemistry. Rev Mod Phys, 2020, 92: 015003
10 Cao Y, Romero J, Olson J P, et al. Quantum chemistry in the age of quantum computing. Chem Rev, 2019, 119: 10856–10915

11 Rebentrost P, Mohseni M, Lloyd S. Quantum support vector machine for big data classification. Phys Rev Lett, 2014, 113: 130503
12 Huang H Y, Broughton M, Mohseni M, et al. Power of data in quantum machine learning. Nat Commun, 2021, 12: 2631
13 Schuld M, Petruccione F. Machine Learning with Quantum Computers. Berlin: Springer, 2021
14 Preskill J. Quantum computing in the NISQ era and beyond. 2018. ArXiv:1801.00862
15 Arute F, Arya K, Babbush R, et al. Quantum supremacy using a programmable superconducting processor. Nature, 2019, 574: 505–510
16 Zhong H S, Wang H, Deng Y H, et al. Quantum computational advantage using photons. Science, 2020, 370: 1460–1463
17 Bharti K, Cervera-Lierta A, Kyaw T H, et al. Noisy intermediate-scale quantum (NISQ) algorithms. 2021. ArXiv:2101.08448
18 Cerezo M, Arrasmith A, Babbush R, et al. Variational quantum algorithms. Nat Rev Phys, 2021, 3: 625–644
19 Endo S, Cai Z, Benjamin S C, et al. Hybrid quantum-classical algorithms and quantum error mitigation. J Phys Soc Jpn, 2021, 90: 032001
20 Peruzzo A, McClean J, Shadbolt P, et al. A variational eigenvalue solver on a photonic quantum processor. Nat Commun, 2014, 5: 4213
21 Farhi E, Goldstone J, Gutmann S. A quantum approximate optimization algorithm. 2014. ArXiv:1411.4028
22 Havlíček V, Córcoles A D, Temme K, et al. Supervised learning with quantum-enhanced feature spaces. Nature, 2019, 567: 209–212
23 Schuld M, Bocharov A, Svore K M, et al. Circuit-centric quantum classifiers. Phys Rev A, 2020, 101: 032308
24 Mitarai K, Negoro M, Kitagawa M, et al. Quantum circuit learning. Phys Rev A, 2018, 98: 032309
25 Yang C H H, Qi J, Chen S Y C, et al. When bert meets quantum temporal convolution learning for text classification in heterogeneous computing. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022. 8602–8606
26 Qi J, Tejedor J. Classical-to-quantum transfer learning for spoken command recognition based on quantum neural networks. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022. 8627–8631
27 Yang C H H, Li B, Zhang Y, et al. A quantum kernel learning approach to acoustic modeling for spoken command recognition. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2023. 1–5
28 Benedetti M, Lloyd E, Sack S, et al. Parameterized quantum circuits as machine learning models. Quantum Sci Technol, 2019, 4: 043001
29 Farhi E, Neven H. Classification with quantum neural networks on near term processors. 2018. ArXiv:1802.06002
30 Yu Z, Yao H S, Li M J, et al. Power and limitations of single-qubit native quantum neural networks. In: Proceedings of Advances in Neural Information Processing Systems, 2022. 27810–27823
31 Caro M C, Huang H Y, Cerezo M, et al. Generalization in quantum machine learning from few training data. Nat Commun, 2022, 13: 4919
32 Li G X, Ye R L, Zhao X Q, et al. Concentration of data encoding in parameterized quantum circuits. In: Proceedings of the 36th Conference on Neural Information Processing Systems (NeurIPS 2022), 2022
33 Du Y, Tu Z, Yuan X, et al. Efficient measure for the expressivity of variational quantum algorithms. Phys Rev Lett, 2022, 128: 80506
34 Jerbi S, Fiderer L J, Nautrup H P, et al. Quantum machine learning beyond kernel methods. Nat Commun, 2023, 14: 517
35 Yu Z, Zhao X, Zhao B, et al. Optimal quantum dataset for learning a unitary transformation. Phys Rev Appl, 2023, 19: 034017
36 Wang K, Song Z, Zhao X, et al. Detecting and quantifying entanglement on near-term quantum devices. npj Quantum Inf, 2022, 8: 52
37 Zhao X, Zhao B, Wang Z, et al. Practical distributed quantum information processing with LOCCNet. npj Quantum Inf, 2021, 7: 159
38 Tian J K, Sun X Y, Du Y X, et al. Recent advances for quantum neural networks in generative learning. 2022. ArXiv:2206.03066
39 Wang Y, Li G, Wang X. A hybrid quantum-classical Hamiltonian learning algorithm. Sci China Inf Sci, 2023, 66: 129502
40 Abbas A, Sutter D, Zoufal C, et al. The power of quantum neural networks. Nat Comput Sci, 2021, 1: 403–409
41 Sordoni A, Nie J T, Bengio Y. 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, New York, 2013
42 Zhang P, Niu J B, Su Z, et al. End-to-end quantum-like language models with application to question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2018
43 Zhang Y, Song D, Zhang P, et al. A quantum-inspired sentiment representation model for twitter sentiment analysis. Appl Intell, 2019, 49: 3093–3108
44 Basile I, Tamburini F. Towards quantum language models. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2017. 1840–1849
45 Zeng W, Coecke B. Quantum algorithms for compositional natural language processing. 2016. ArXiv:1608.01406
46 Meichanetzidis K, Gogioso S, de Felice G, et al. Quantum natural language processing on near-term quantum computers. 2020. ArXiv:2005.04147
47 Wiebe N, Bocharov A, Smolensky P, et al. Quantum language processing. 2019. ArXiv:1902.05162
48 Chen S Y C, Yoo S, Fang Y L L. Quantum long short-term memory. 2020. ArXiv:2009.01783
49 Devlin J, Chang M W, Lee K, et al. Bert: pre-training of deep bidirectional transformers for language understanding. 2018. ArXiv:1810.04805
50 Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017. 6000–6010
51 Li X P, Song J K, Gao L L, et al. Beyond RNNs: positional self-attention with co-attention for video question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2019. 8658–8665
52 Guo Q P, Qiu X P, Liu P F, et al. Multi-scale self-attention for text classification. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020. 7847–7854
53 Cha P, Ginsparg P, Wu F, et al. Attention-based quantum tomography. 2020. ArXiv:2006.12469
54 Lorenz R, Pearson A, Meichanetzidis K, et al. QNLP in practice: running compositional models of meaning on a quantum computer. 2021. ArXiv:2102.12846
55 Nielsen M A, Chuang I. Quantum Computation and Quantum Information. Cambridge: Cambridge University Press, 2011
56 Micchelli C A, Xu Y S, Zhang H Z. Universal kernels. J Mach Learn Res, 2006, 7: 2651–2667

57 Di Sipio R, Huang J H, Chen S Y C, et al. The dawn of quantum natural language processing. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2022. 8612–8616
58 Ziegel E R, Lehmann E L, Casella G. Theory of point estimation. Technometrics, 1999, 41: 274
59 Bottou L. Stochastic learning. In: Proceedings of Advanced Lectures on Machine Learning, 2004. 146–168
60 Goodfellow I, Bengio Y, Courville A. Deep Learning. Cambridge: MIT Press, 2016
61 Liu Y, Arunachalam S, Temme K. A rigorous and robust quantum speed-up in supervised machine learning. Nat Phys, 2021, 17: 1013–1017
62 Qi J, Yang C H H, Chen P Y, et al. Theoretical error performance analysis for variational quantum circuit based functional regression. npj Quantum Inf, 2023, 9: 4
63 Kotzias D, Denil N, de Freitas N, et al. From group to individual labels using deep features. In: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2015. 597–606
64 Ma Y J, Yu D H, Wu T, et al. PaddlePaddle: an open-source deep learning platform from industrial practice. Front Data Domput, 2019, 1: 105–115
65 Dua D, Graff C. UCI machine learning repository. 2017. http://archive.ics.uci.edu/ml
66 Kingma D P, Ba J L. Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations, 2015