Quantum linear algebra is all you need for Transformer architectures

Naixu Guo,[1,*] Zhan Yu,[1,†] Aman Agrawal,[2] and Patrick Rebentrost[1,3,‡]

[1] *Centre for Quantum Technologies, National University of Singapore, 117543, Singapore*
[2] *Department of Mathematics, National University of Singapore, 117543, Singapore*
[3] *School of Computing, National University of Singapore, 117543, Singapore*

(Dated: February 27, 2024)

Generative machine learning methods such as large-language models are revolutionizing the creation of text and images. While these models are powerful they also harness a large amount of computational resources. The transformer is a key component in large language models that aims to generate a suitable completion of a given partial sequence. In this work, we investigate transformer architectures under the lens of fault-tolerant quantum computing. The input model is one where pre-trained weight matrices are given as block encodings to construct the query, key, and value matrices for the transformer. As a first step, we show how to prepare a block encoding of the self-attention matrix, with a row-wise application of the softmax function using the Hadamard product. In addition, we combine quantum subroutines to construct important building blocks in the transformer, the residual connection, layer normalization, and the feed-forward neural network. Our subroutines prepare an amplitude encoding of the transformer output, which can be measured to obtain a prediction. We discuss the potential and challenges for obtaining a quantum advantage.

I. INTRODUCTION

Large language models (LLMs) such as GPT4 recently arrived in the public consciousness and continue to make headlines [1, 2]. These models are called pretrained foundation models (PFMs) in the sense that once pretrained on large-scale data, they can be applied across a wide range of applications [3, 4]. The new mainstream model architecture of these PFMs is the transformer [5]. One of the problems the transformer solves is that given an input sequence, i.e., a sentence of English words, produce an output sequence based on training data of given input-output pairs. The transformer was developed "to learn what to pay attention to", where the self-attention block can capture correlations among different parts of the sequence via inner products [5, 6]. It contains two parts called the encoder and decoder separately. The encoder maps a sequence of symbolic representations to a sequence of continuous (real) representations. The decoder maps from a sequence in continuous representation to a sequential output in symbolic representation, where the output sequence is given one at a time. The frameworks of these two parts are mostly the same, constructed with self-attention blocks and intermediate feed-forward neural networks. It has become common to use the decoder-only structure, which corresponds to an auto-regressive model and can be used for the next token prediction [7–10].

Quantum computing has been investigated for linear algebra-based tasks for the last decades, showing the potential for quantum advantages for linear systems and other matrix linear algebra operations [11, 12]. Quantum singular value transformation is a unified framework for quantum algorithms [13] based on block-encodings and polynomial transformations [14]. It has recently been generalized to non-normal matrix cases [15]. Hardware progress has significantly improved the quantity and quality of quantum bits [16, 17], where recent experiments have designs for 10s of logical qubits [18]. While the hardware is still in comparable infancy, it is worthwhile to discuss if quantum computers in principle could provide any advantage for large-language models. It is hence a valuable goal to discuss the usage of advanced quantum algorithms to construct a state-of-the-art machine learning algorithm.

A few problems emerge, of which we list here some of the main ones. First, LLMs are based on large data sets of terabytes of input data. Quantum computers so far are not good at big classical data applications,

* naixug@u.nus.edu
† yu.zhan@u.nus.edu
‡ cqtfpr@nus.edu.sg

as the proposals of quantum Random Access Memory (qRAM) are hard to realize in practice [19, 20]. Second, modern LLMs contain billions of training parameters. Current quantum computers are of the size of around 100 qubits and even if every qubit carries several training parameters, overall the number of training parameters is vanishingly small compared to the advanced classical models. Third, the no-cloning principle for quantum states holds in general. In classical information processing, it is a natural step to save computed data to memory for later access. With quantum computing, such a step should in general be avoided as it incurs the cost of full state tomography, which often destroys possibilities of quantum advantage.

In this work, we investigate the use of quantum computing for building transformer architectures. We show progress towards an end-to-end architecture, which includes all the key building blocks and a discussion of their quantum complexity. Our computational framework is the framework of quantum signal processing and quantum singular value transformation [13, 14, 21]. The modularity of this framework makes it a natural candidate for transformer architectures. Our first simplification is that we assume that the transformer is already pretrained, i.e., we are given the pretrained weight matrices for query, key, and value parts as quantum circuits. The second simplification is that we focus on the inference part of predicting a single next token. We develop quantum subroutines for self-attention, residual connection and layer normalization, and feed-forward neural networks with the GELU activation function. One of our technical contributions is a subroutine for the elementwise-application of the softmax function in the attention computation. Our subroutines are efficient in their use of qubits and circuit depth, which allows for the potential for a variety of quantum speedups. Our output is twofold. We output a quantum circuit which is a block encoding of the transformer architecture. Depending on the input quantities, this block encoding can then be used for subsequent layers of a neural network architecture. In addition, we show how to output the index of a next predicted token according to the probabilities modeled by the transformer architecture.

The paper is organized as follows. The preliminary is provided in Section II, where we introduce the notations, descriptions about the transformer, and quantum subroutines. In Section III, we formulate the classical transformer building blocks into quantum problems. Following this, we show our main results in Section IV. In Section V, we discuss the input assumptions, potential quantum speedups, and future research directions. The technical contributions in Section IV are combined into the following theorem about a single transformer layer, here stated informally.

Theorem 1 (Quantum single-layer Transformer, informal)**.** *For a transformer, see Fig. 1, with embedding dimension d and an input sequence S of length $N = 2^n$, assume block-encoded inputs with encoding normalization factors at most α and certain normalization factors being constants, see below in Definition 4 for the formal statement of the input assumption. If all encoding errors are at most $\epsilon_{\text{block}} = o(\epsilon^8/d^4N)$, then for the index $j \in [N]$, one can construct an ϵ-accurate state preparation quantum circuit for the quantum state*

$$\sum_{k=1}^{d} \text{Transformer}(S, j)_k |k\rangle, \tag{1}$$

by using the input block encodings for $\mathcal{O}(dn^2\alpha^2 \log^2(1/\epsilon_{\text{block}}))$ times.

We discuss the formal result and detailed polynomial dependencies on all of the normalization factors in Theorem 12. In Section V, we show that a conservative regime of the parameters exhibits the potential for a quadratic advantage over the classical algorithm.

II. PRELIMINARY

A. Notation

We use the Dirac notation $|\psi\rangle$ to represent a vector with $\|\psi\|_2 = 1$ (pure quantum state). Denote by $\mathbb{N}$ the natural numbers $\{1, 2, \cdots\}$. For $N \in \mathbb{N}$, we use the notation $[N]$ to represent the set $\{1, \ldots, N\}$. For a n-qubit state $|0\rangle^{\otimes n}$, we write $|0^n\rangle$ for simplicity. When there is no ambiguity, we may further ignore the superscript n of $|0^n\rangle$. For a matrix or an operator A, we use $A_{jk} \coloneqq \langle j|A|k\rangle$ to represent its (j, k)-th element, where $\{|k\rangle\}$ are the standard basis. We use $A_{j\star}$ to represent its j-th row and $A_{\star k}$ to represent its k-th column. The

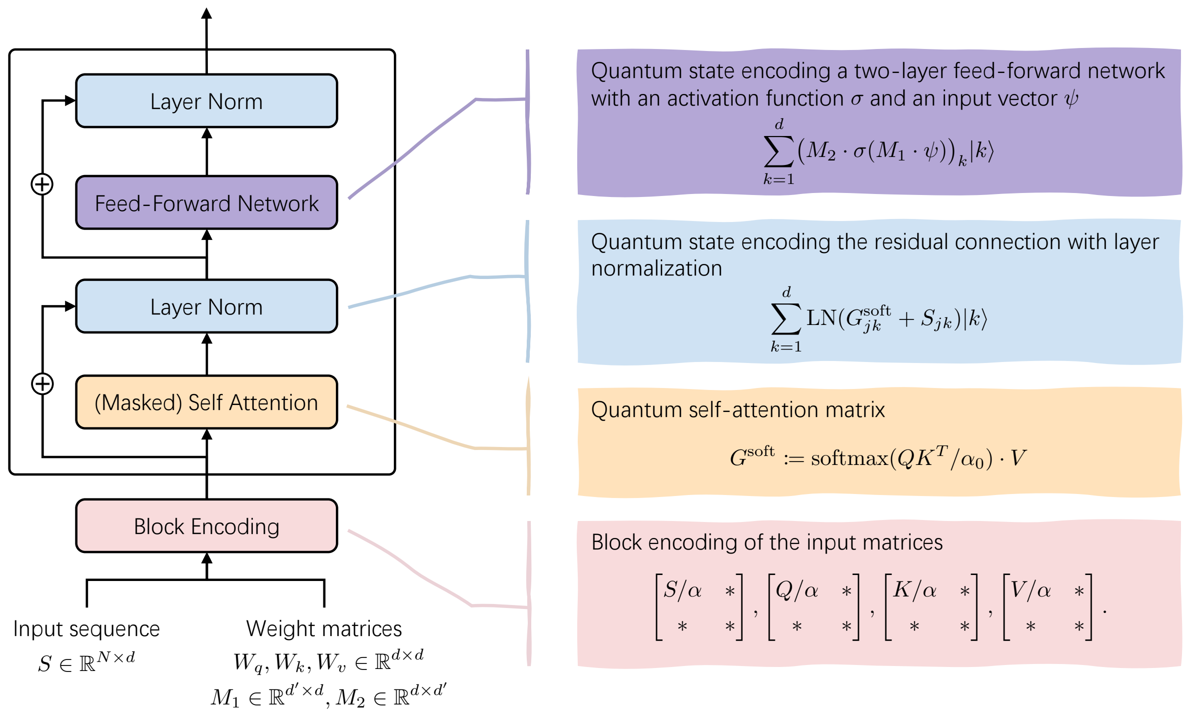

FIG. 1. **Overview of the quantum transformer architecture.** Here, we visualize a single-layer transformer architecture and highlight the parts relevant to this work. We construct and discuss the corresponding quantum subroutines and combine them to our final theorem on obtaining a sample from the first layer output. The inputs are matrices for the input sequence and pretrained weights, from which the relevant matrices for the transformer are constructed (query Q, key K, and value V). These inputs are given as block encodings, where the block encoding of a matrix is a unitary matrix that contains the desired normalized matrix in a diagonal block. The buiding blocks and symbols are further explained in Section II B.

spectral norm, i.e., the largest singular value, is denoted by $\|A\|$. For a normal matrix $A := \sum_k \lambda_k(A)|\psi_k\rangle\langle\psi_k|$ and a function f, we write $f(A) := \sum_k f(\lambda_k(A))|\psi_k\rangle\langle\psi_k|$ to represent the eigenvalue transformation of A with f. For a matrix A and a function f, we use $f \circ (A)$ to represent the element-wise application of the function to the matrix, i.e., $(f\circ(A))_{jk} = f(A_{jk})$. We define a vector-input function called softmax, which is widely used in machine learning. For $z \in \mathbb{R}^N$, it is defined as $\mathrm{softmax}(z)_j := e^{z_j}/(\sum_{k\in[N]} e^{z_k})$ for $j \in [N]$. For a matrix $M \in \mathbb{R}^{N\times N}$, it is defined as a row-wise application of the softmax function, i.e., $\mathrm{softmax}(M)_{ij} := e^{M_{ij}}/(\sum_{k\in[N]} e^{M_{ik}})$ for $i, j \in [N]$.

B. Brief description about transformer

The transformer is a key component of pretrained foundation models. It has many applications and one of the main ones is the next token prediction, achieving great success in natural language processing. Given a part of a sequence, the transformer aims to predict the next object of the sequence. The transformer is constructed by three main building blocks: self-attention, residual connection with layer normalization, and feed-forward networks (FFN). These building blocks will be described in this section. The original paper [5] contains both the encoder and decoder parts. Later many practically significant models only use one part, especially the decoder-only structure, which is shown in Fig. 1.

A key aspect of large-language models is *tokenization*. A token is the basic unit of the transformer process. Concepts like words, codes, and even images can be converted to tokens with the so-called tokenization method [22–24]. For the transformer, tokens are further mapped to real vectors via *embedding* [5]. Let d_{token} be the number of tokens in our language and d_{model} be the dimension of the vectors of the embedding. Let $\mathcal{W} := \{\omega_j \in \mathbb{R}^{d_{\text{model}}} : \omega_j \text{ is the embedding of token } j \in [d_{\text{token}}]\}$ be the set of the embedding vectors of all tokens. For simplicity, when we mention tokens in this paper, we directly mean their vector representations. An N-length sentence is a sequence of vectors $\{S_j\}_{j=1}^N$, where $S_j \in \mathcal{W}$. Due to the vector embeddings of the tokens, a sentence can also be understood as a real matrix $S \in \mathbb{R}^{N \times d_{\text{model}}}$.

Self-attention — The correlations of the original concepts, such as words in natural languages, imply correlations of the corresponding tokens in the set of tokens. Self-attention is the building block to encode such correlation information among tokens (vectors) into a new vector, which is the input vector for the next block. The correlation is computed via estimating inner products. The block is also called "scaled dot-product attention".

There are three real parameterized (weight) matrices $W_q, W_k \in \mathbb{R}^{d_{\text{model}} \times d_k}$ and $W_v \in \mathbb{R}^{d_{\text{model}} \times d_v}$ arising in the self-attention block. In practical cases, $d_{\text{model}} = d_k = d_v$ is widely used, e.g., in the original paper [5]. In our discussion, we will keep this condition and write $d := d_{\text{model}}$ for simplicity. Given the sentence S, the convention is to call $Q := SW_q$, $K := SW_k$, and $V := SW_v$ the query, key, and value matrices respectively. The attention block computes the matrix $G^{\text{soft}} \in \mathbb{R}^{N \times d}$ such that

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T/\alpha_0)V =: G^{\text{soft}}, \tag{2}$$

where $\alpha_0 > 0$ is scaling factor.

In the attention block, the softmax is implemented for each row of the matrix QK^T/α_0. The factor α_0 controls that the exponentiated values are not too large. The value $\alpha_0 = \sqrt{d}$ has been discovered to be a good choice in practice. To see this, assume that each row of Q and K has zero mean and unit standard deviation. Then for each element of $(QK^T)_{jk} = \sum_{m=1}^d Q_{jm}K_{km}$, the standard deviation will be bounded by $\sqrt{d}$. The coefficient rescales the standard deviation to 1. Depending on the architecture and embeddings other scaling factors may also be employed. Inspired from the block-encoding discussion in this work, there is a natural choice for this scaling as we discuss in Section IV C.

For $j \in [N]$, if the current query token is the j-th token S_j, the corresponding output vector is the j-th row of the self-attention matrix in Eq. (2). More explicitly, the output vector of the self-attention layer for the j-th token is

$$z_j = \sum_{k=1}^d G^{\text{soft}}_{jk} \hat{e}_k \equiv (G^{\text{soft}})^T \hat{e}_j, \tag{3}$$

where $\{\hat{e}_j\}_{j=1}^N$ is the standard basis. For the decoder-only structure which achieves the best practical performance, the so-called *masked* self-attention is used, which has the effect to mask or hide the tokens after the current query token. This is achieved by adding a masked matrix $QK^T \to QK^T + M$, where

$$M_{jk} = \begin{cases} 0 & k \leq j, \\ -\infty & k > j. \end{cases} \tag{4}$$

Since $\exp(-\infty) = 0$, tokens larger than j receive no attention. A further generalization called *multi-head* self-attention is based on computing several smaller attention matrices and concatenating them together. The h-head self attention can be achieved with linear transformations $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d \times \lceil \frac{d}{h} \rceil}$, and $W^O \in \mathbb{R}^{d \times d}$ for $i \in [h]$:

$$\text{Multihead}(Q, K, V) = [\text{head}_1, \ldots, \text{head}_h]W^O \in \mathbb{R}^{N \times d},$$

where $\text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V) \in \mathbb{R}^{N \times \lceil \frac{d}{h} \rceil}$.

Residual connection — Using the self-attention block, a residual connection with subsequent layer normalization is employed. This layer provides the ability to skip the self-attention block. Note that $z_j \equiv z_j(S)$ is the S dependent output vector of the self-attention block. The residual connection is the output vector

$z_j(S) + S_j$[1]. The next step is to normalize (or standardize) this output. Let $\bar{s}_j := \frac{1}{d}\sum_{k=1}^{d}((z_j)_k + S_{jk})$ be the average value and $\varsigma := \sqrt{\sum_{k=1}^{d}((z_j)_k + S_{jk} - \bar{s}_j)^2}$ be the standard deviation. The complete residual connection with the normalization layer can be expressed as

$$f(S,j) = \frac{z_j + S_j - \bar{s}_j}{\varsigma}. \tag{5}$$

The role of this part is to improve the trainability, which has been found essential for training deep neural networks in practice [25, 26].

Feed-forward network — Finally, a two-layer fully-connected feed-forward network is implemented, i.e.,

$$\mathrm{FFN}(f(S,j)) = \sigma(f(S,j)W_1 + b_1)W_2 + b_2, \tag{6}$$

where σ is an activation function, such as $\tanh(x)$ and $\mathrm{ReLU}(x) = \max(0,x)$. Another activation function that may not be widely known, yet has been used in many transformer-based models like GPT series and BERT, is the Gaussian Error Linear Units function [27]. Formally, we have $\mathrm{GELU}(x) := x \cdot \frac{1}{2}(1 + \mathrm{erf}(\frac{x}{\sqrt{2}}))$, where $\mathrm{erf}(x) := \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}\,dt$ is the error function. The function can be understood as a smoother ReLU activation function and will be our focus in the paper. In addition, $W_1 \in \mathbb{R}^{d\times d_{\mathrm{ff}}}, W_2 \in \mathbb{R}^{d_{\mathrm{ff}}\times d}$ are linear transformation matrices, and b_1, b_2 are vectors. In most practical cases, $d_{\mathrm{ff}} = 4d$.

In currently employed transformer architectures, several of these building blocks are iterated for a constant number of times. The output, i.e., the next token, is sampled from the distribution by further implementing a linear transformation and softmax function. Considering the run time, recall that the length of the input sentence is N and the dimension of the embedded vectors is d. We summarize the time complexity as Table I.

Block	**Time complexity**
Preparation of Q, K, V	Nd^2
Preparation of QK^T	$N^2 d$
Preparation of $\mathrm{softmax}(QK^T/\sqrt{d})V$	$N^2 + Nd^2$
Residual connection $f(z_j, S_j)$	$3d$
Feed-forward NN $\mathrm{FFN}(f(z_j, S_j))$	$O(Nd^2)$

TABLE I. Time complexity of transformer steps.

The time complexity of a constant number of iterations of the three main blocks is $\mathcal{O}(N^2 d + Nd^2)$, which mainly comes from the self-attention matrix computation. If we only consider the 1-layer transformer, the time complexity is $\mathcal{O}(Nd + d^2)$, as we do not need to compute all N vectors that are needed for the second layer self-attention block.

C. Quantum subroutines

For the input assumption, we use an assumption called the *block encoding* as follows, which is standard in some of the quantum algorithms literature [13]. Note that the encoding can be generalized to that of a non-square matrix of arbitrary size by padding the matrix with zeros.

Definition 1 (Block encoding [13, 28])**.** *We say a unitary U_A is an (α, a, ϵ)-encoding of matrix $A \in \mathbb{C}^{2^n\times 2^n}$ if*

$$\|A - \alpha(\langle 0^a| \otimes I_n)U_A(|0^a\rangle \otimes I_n)\| \le \epsilon. \tag{7}$$

[1] We note that this output vector can also be written as $z_j(S) + \mathrm{Attention}(0,0,S)^T \hat{e}_j$.

When we say we can construct or are given a block encoding unitary, it means we have access to the corresponding quantum circuit. When given a quantum circuit, it implies that we can also implement the controlled, self-adjoint, and controlled self-adjoint of the circuit. Further discussions about achieving a block encoding are provided in Section V. We introduce some results on "arithmetic computations" for block-encoded matrices such as addition and multiplication. The first result is to achieve a linear combination of block-encoded matrices, which requires the so-called state preparation pair.

Definition 2 (State preparation pair [13, 28]). *Let $y \in \mathbb{C}^m$ and $\|\gamma\| = 1 \leq \beta$, the pair of unitaries (P_L, P_R) is called a (β, b, ϵ)-state-preparation-pair if $P_L|0^b\rangle = \sum_{k=1}^{2^b} c_k|k\rangle$ and $P_R|0^b\rangle = \sum_{k=1}^{2^b} d_k|k\rangle$ such that $\sum_{k=1}^{m} |\beta(c_k^* d_k) - y_k| \leq \epsilon$ and for all $k \in m+1, \ldots, 2^b$ we have $c_k^* d_k = 0$.*

This pair of circuits allows to create a linear combination of matrices with given coefficients as the next lemma shows.

Lemma 1 (Linear combination of block-encoded matrices [13, 28]). *Let $A = \sum_{k=1}^{m} y_k A_k$ be an s-qubit operator and $\epsilon > 0$. Suppose that (P_L, P_R) is a (β, b, ϵ_1)-state-preparation-pair for y, and that $W = \sum_{k=1}^{m} |k\rangle\langle k| \otimes U_k + ((I - \sum_{k=1}^{m} |k\rangle\langle k|) \otimes I_a \otimes I_s)$ is an $s+a+b$ qubit unitary such that for all $k \in [m]$, the unitary U_k is an (α, a, ϵ_2)-encoding of A_k. Then we can implement an $(\alpha\beta, a+b, \alpha\epsilon_1 + \alpha\beta\epsilon_2)$-encoding of A, with a single use of W, P_R and $P_L^\dagger$.*

The second result is to achieve a multiplication of block-encoded matrices.

Lemma 2 (Product of block-encoded matrices [13, 28]). *If U is an (α, a, δ)-encoding of an s-qubit operator A, and V is a (β, b, ϵ)-encoding of an s-qubit operator B, then $(I_b \otimes U)(I_a \otimes V)$ is an $(\alpha\beta, a+b, \alpha\epsilon + \beta\delta)$-encoding of AB.*

Once given the block-encoding, one can implement polynomial functions on singular values of block-encoded matrices (or eigenvalues for Hermitian matrices) by using the quantum singular value transformation (QSVT) method.

Theorem 2 (Polynomial eigenvalue transformation [13]). *Let $\delta > 0$. Given U that is an (α, a, ϵ)-encoding of a Hermitian matrix A, and a real ℓ-degree function $f(x)$ with $|f(x)| \leq \frac{1}{2}$ for $x \in [-1, 1]$, one can prepare a $(1, a+n+4, 4\ell\sqrt{\epsilon/\alpha} + \delta)$-encoding of $f(A/\alpha)$ by using $\mathcal{O}(\ell)$ queries to U and $\mathcal{O}(\ell(a+1))$ one- and two-qubit quantum gates. The description of the quantum circuit can be computed classically in time $\mathcal{O}(\mathrm{poly}(d, \log(1/\delta)))$.*

Since the outputs from each block of the transformer are vectors, we construct quantum circuits whose output quantum states correspond to these vectors. We use the format of state preparation encoding introduced in Ref. [29], yet change from L_2 norm to L_∞ norm.

Definition 3 (State preparation encoding). *We say a unitary U_ψ is an (α, a, ϵ)-state-encoding of an n-qubit quantum state $|\psi\rangle$ if*

$$\||\psi\rangle - \alpha(\langle 0^a| \otimes I)U_\psi|0^{a+n}\rangle\|_\infty \leq \epsilon. \tag{8}$$

More straightforwardly, the (α, a, ϵ)-state-encoding U_ψ prepares the state

$$U_\psi|0\rangle|0\rangle = \frac{1}{\alpha}|0\rangle|\psi'\rangle + \sqrt{1-\alpha^2}|1\rangle|\mathrm{bad}\rangle,$$

where $\||\psi'\rangle - |\psi\rangle\|_\infty \leq \epsilon$ and $|\mathrm{bad}\rangle$ is an arbitrary quantum state. One can further prepare the state $|\psi'\rangle$ by using $\mathcal{O}(\alpha)$ times of amplitude amplification [30]. To encode the classical coefficients into quantum states which will be used multiple times, we follow the results in Ref. [31, 32].

Theorem 3 (Quantum state preparation [31]). *For a given vector $v \in \mathbb{C}^N$ with $\|v\|_2 = 1$, one can prepare a $(1, 0, 0)$-state-encoding U_v of state $|v\rangle = \sum_{i=1}^{N} v_i|i\rangle$ with depth $\mathcal{O}(N/\log N)$ without using ancilla qubits. One can also achieve this with depth $\mathcal{O}(\log N)$ with $\mathcal{O}(N)$ ancilla qubits.*

An additional point to note is that for the classical case, they consider the row vector as described previously. However, for the quantum case, we consider the column vector, i.e., quantum state, so we may implement the self-adjoint of the unitary to achieve it.

III. PROBLEM FORMULATIONS

Here, we describe our assumptions and the problem statements that are considered for the solving on quantum computers. Recall that in this paper, we focus on the inference and assume the training process has already been achieved. The classical problems assume memory access to the inputs such as the sentence and the query, key, and value matrices. The quantum algorithms change this input assumption to a block encoding input assumption. The dimensions of N and d can be achieved by padding with zeros.

Definition 4 (Input assumption)**.** *We assume $N = 2^n$ and $d = 2^{\log d}$ for $n, \log d \in \mathbb{N}^+$. For the input sequence $S \in \mathbb{R}^{N\times d}$, we assume given access to a quantum circuit U_S which is an $(\alpha_s, a_s, \epsilon_s)$-encoding of S. For matrices $W_q, W_k, W_v \in \mathbb{R}^{d\times d}$, assume given access to quantum circuits U_{W_q}, U_{W_k}, and U_{W_v} that are $(\alpha_w, a_w, \epsilon_w)$-encodings of W_q, W_k and W_v respectively. For the feed-forward neural network, we assume $(\alpha_m, a_m, \epsilon_m)$-encodings U_{M_1} and U_{M_2} of two weight matrices $M_1 \in \mathbb{R}^{N_1\times N}$ and $M_2 \in \mathbb{R}^{N_2\times N_1}$.*

We reformulate the classical problems to the quantum version based on this input assumption.

Problem 1 (Quantum self-attention)**.** *Assume the input assumption as in Definition 4. Define $Q := SW_q$, $K := SW_k$, and $V := SW_v$. Let the current focused token be $j \in [N]$, the task is to construct a block-encoding of the matrix G such that*

$$G_{j\star} = G^{\text{soft}}_{j\star} := \big(\text{softmax}(QK^T/\alpha_0)V\big)_{j\star}, \tag{9}$$

where $\alpha_0 = \alpha_s^2\alpha_w^2$. For the masked self-attention, change G^{soft} to $\text{softmax}(QK^T/\alpha_0 + M)V$, where M is the masked matrix as Eq. (4).

Note that we change the scaling coefficient α_0 for the quantum case. Details of the explanation can be found in Section IV C.

Problem 2 (Quantum residual connection)**.** *Let $c > 0$ and $g(x)$ be a real k-degree polynomial function. Given an (α, a, ϵ)-state-encoding U of a quantum state $\sum_{j=1}^d x_j|j\rangle$, where $\{x_j\}$ are real and $\|x\|_2 = 1$, prepare a state-encoding of the state*

$$\frac{1}{\sqrt{\sum_{j=1}^d (c\cdot g(x)_j + x_j)^2}}\sum_{j=1}^d (c\cdot g(x)_j + x_j)|j\rangle. \tag{10}$$

Problem 3 (Quantum residual connection with layer normalization)**.** *Assume the input assumption as in Definition 4. Assume given access to an $(\alpha_g, a_g, \epsilon_g)$-encoding of the self-attention G^{soft} as Eq. (9). Let the current query token be the j-th token. Construct a state preparation encoding of the state*

$$\sum_{k=1}^d \text{LN}(G^{\text{soft}}_{jk} + S_{jk})|k\rangle, \tag{11}$$

where LN is to standardize the vector.

Note that standardization rescales the ℓ_2-norm of the vector to be 1, so there is no additional normalization factor in the quantum state.

Problem 4 (Two-layer feedforward network)**.** *Assume the input assumption as in Definition 4. Given an (α, a, ϵ)-state-encoding U_ψ of an n-qubit state $|\psi\rangle = \sum_{k=1}^N \psi_k|k\rangle$, where $\{\psi_k\}$ are real and $\|\psi\|_2 = 1$, and an activation function σ, prepare a state encoding of the quantum state $|\phi\rangle$*

$$|\phi\rangle = \frac{1}{C}\sum_{k=1}^{N_2}\big(M_2\cdot\sigma(M_1\cdot\psi)\big)_k|k\rangle, \tag{12}$$

where C is the normalization factor.

IV. MAIN RESULTS

In this section, we present our main technical contributions. Our contributions are element-wise application of polynomials to a matrix, attention computation, ResNet and normalization, and FNN with the GELU activation function.

A. Element-wise function of block-encoded matrices

In this section, we show an essential building block for our algorithm. For a function $f:\mathbb{R}\to\mathbb{R}$ and a matrix $A\in\mathbb{C}^{2^n\times 2^n}$, the task is to apply the element-wise operation $f\circ(A)$. In a classical or quantum query model, the solution is to apply the function after each particular element is queried. However, here we do not work in the query model. The matrix A is accessed via a block encoding, which includes the query model, but also includes the use of other input models such as input from a preceding subroutine.

The key idea of our subroutines is as follows, see below for the formal results. Assume that f in some range admits a polynomial approximation g with some degree d_{poly} and some point-wise error, i.e, $f(x)\approx g(x)=\sum_{k=0}^{d_{\text{poly}}} c_k x^k$. For each entry of the matrix inside the range, it holds that $f(A_{ij})\approx g(A_{ij})$ and thus $[f\circ(A)]_{ij}\approx[g\circ(A)]_{ij}$. We can express the entry as $[g\circ(A)]_{ij}=\sum_{k=0}^{d_{\text{poly}}} c_k A_{ij}^k=\sum_{k=0}^{d_{\text{poly}}} c_k (A^{\circ k})_{ij}$, using the k-th Hadamard product of the matrix with itself, $A^{\circ k}$. Furthermore, there exist a matrix P such that $A^{\circ k}=[PA^{\otimes_k}P^T]_{\text{block}}$, where the subscript "block" indicates that we choose the correct block of the matrix $PA^{\otimes_k}P^T$. Hence, we find that

$$[f\circ(A)]_{ij}\approx\sum_{k=0}^{d_{\text{poly}}} c_k\left[[PA^{\otimes_k}P^T]_{\text{block}}\right]_{ij}. \tag{13}$$

In summary, the quantum algorithm uses a tensor-product of matrices, permutation matrices, linear combination of matrices, and polynomial approximation to construct an elementwise application of a function to the entries of a matrix.

We start with a lemma about the max-norm of a block-encoding.

Lemma 3. *If U is an (α,a,ϵ)-encoding of matrix $A\in\mathbb{C}^{2^n\times 2^n}$, we have*

$$\max_{i,j\in[2^n]}|\alpha(\langle 0^a|\otimes\langle i|)U(|0^a\rangle\otimes|j\rangle)-A_{ij}|\le\epsilon. \tag{14}$$

Proof. Let $B=A-\alpha(\langle 0^a|\otimes I)U(|0^a\rangle\otimes I)$, which is a complex matrix. By definition,

$$\|B\|=\|A-\alpha(\langle 0^a|\otimes I)U(|0^a\rangle\otimes I)\|\le\epsilon.$$

By Lemma S1, we have $\max_{i,j}|B_{ij}|\le\epsilon$. □

As seen from the qualitative discussion above, we have to be able to construct the Hadamard product between matrices. Here, we consider the general case of two different matrices.

Theorem 4 (Hadamard product of block-encoded matrices)**.** *With $n\in\mathbb{N}$ and $N=2^n$, consider matrices $A_1,A_2\in\mathbb{C}^{N\times N}$, and assume that we have an (α,a,δ)-encoding of matrix A_1 and (β,b,ϵ)-encoding of matrix A_2. We can construct an $(\alpha\beta,a+b+n,\alpha\epsilon+\beta\delta)$-encoding of matrix $A_1\circ A_2$.*

Proof. For simplicity, we first consider the perfect case without input block-encoding errors. Let U_1 and U_2 be the $(\alpha,a,0)$- or $(\beta,b,0)$-encoding unitary of A_1 and A_2, respectively. Note that

$$(\langle 0^{a+b}|)U_1\otimes U_2(|0^{a+b}\rangle)=\frac{1}{\alpha\beta}A_1\otimes A_2. \tag{15}$$

Let $P'=\sum_{i=0}^{N-1}|i\rangle\langle i|\otimes|0\rangle\langle i|$. As shown in Ref. [33], $P'(A_1\otimes A_2)P'^{\dagger}=(A_1\circ A_2)\otimes|0\rangle\langle 0|$. However, note that P' is not a unitary. Instead, we consider $P=\sum_{i,j=0}^{N-1}|i\rangle\langle i|\otimes|i\oplus j\rangle\langle j|$, which can be easily constructed by using n CNOT gates, i.e., one CNOT gate between each pair of qubits consisting of one qubit from the first register and the corresponding qubit from the second register. By direct computation, we have

$$(I_n \otimes \langle 0^n|)P(A_1 \otimes A_2)P^\dagger(I_n \otimes |0^n\rangle) = A_1 \circ A_2. \tag{16}$$

Therefore,

$$(I_n \otimes \langle 0^{n+a+b}|)\big((P \otimes I_{a+b})U_1 \otimes U_2(P^\dagger \otimes I_{a+b})\big)(I_n \otimes |0^{n+a+b}\rangle) = \frac{1}{\alpha\beta}A_1 \circ A_2. \tag{17}$$

Now we consider the error from the input block encodings. Write $\bar{A}_1 := \alpha\langle 0^a|U_1|0^a\rangle$ and $\bar{A}_2 := \beta\langle 0^b|U_2|0^b\rangle$. Let $B_1 = A_1 - \bar{A}_1$ and $B_2 = A_2 - \bar{A}_2$. By definition, $\|B_1\| \le \delta$, $\|B_2\| \le \epsilon$. The error can be bounded by

$$\begin{aligned}
&\big\|A_1 \circ A_2 - \alpha\beta(\langle 0^{n+a+b}|)\big((P \otimes I_{a+b})U_1 \otimes U_2(P^\dagger \otimes I_{a+b})\big)(|0^{n+a+b}\rangle)\big\| \\
&\le \big\|A_1 \circ A_2 - \alpha\beta\langle 0^n|\big(P(\langle 0^{a+b}|U_1 \otimes U_2|0^{a+b}\rangle)P^\dagger\big)|0^n\rangle\big\| \\
&\le \big\|A_1 \circ A_2 - \langle 0^n|\big(P\bar{A}_1 \otimes \bar{A}_2P^\dagger\big)|0^n\rangle\big\| \\
&\le \big\|A_1 \circ A_2 + \langle 0^n|\big(PA_1 \otimes \bar{A}_2P^\dagger\big)|0^n\rangle - \langle 0^n|\big(PA_1 \otimes \bar{A}_2P^\dagger\big)|0^n\rangle - \langle 0^n|\big(P\bar{A}_1 \otimes \bar{A}_2P^\dagger\big)|0^n\rangle\big\| \\
&\le \big\|A_1 \circ A_2 - \langle 0^n|\big(PA_1 \otimes \bar{A}_2P^\dagger\big)|0^n\rangle\big\| + \big\|\langle 0^n|\big(PA_1 \otimes \bar{A}_2P^\dagger\big)|0^n\rangle - \langle 0^n|\big(P\bar{A}_1 \otimes \bar{A}_2P^\dagger\big)|0^n\rangle\big\| \\
&\le \big\|\langle 0^n|\big(PA_1 \otimes B_2P^\dagger\big)|0^n\rangle\big\| + \big\|\langle 0^n|\big(PB_1 \otimes \bar{A}_2P^\dagger\big)|0^n\rangle\big\| \\
&\le \alpha\epsilon + \beta\delta.
\end{aligned} \tag{18}$$

□

The previous lemma can be implemented iteratively. Given an (α, a, ϵ)-encoding of matrix A, for $j \in \mathbb{N}$, one can construct an $(1, ja + (j-1)n, j\epsilon/\alpha)$-encoding of matrix $(A/\alpha)^{\circ j} := (A/\alpha) \circ (A/\alpha) \circ \cdots \circ (A/\alpha)$ containing $j-1$ Hadamard products among j copies of matrix A/α. Hence, we can implement polynomials on the entries of A/α.

Theorem 5 (Element-wise polynomial function of block-encoded matrix)**.** *Let $n, k \in \mathbb{N}$. Given access to an (α, a, ϵ) block-encoding of a matrix $A \in \mathbb{C}^{2^n \times 2^n}$ and an ℓ-degree polynomial function $f_\ell(x) = \sum_{j=0}^{\ell} c_j x^j$, $c_0, c_j \in \mathbb{C}$ for $j \in [l]$, one can construct a (C, b, γ)-encoding of $f_\ell \circ (A/\alpha)$ by using $\mathcal{O}(\ell^2)$ times the input unitary, where $C := \sum_{j=0}^{\ell} |c_j|$, $b := \ell a + (\ell-1)n + 2\log\ell$, and $\gamma := \frac{\epsilon}{\alpha} \cdot (\sum_{j=0}^{\ell} |c_j| j)$.*

Proof. We first consider the perfect case, i.e., $\epsilon = 0$. To achieve this implementation, we construct two state-preparation unitaries, which act on $\lceil \log(\ell+1) \rceil$ qubits such that

$$U_1 : \ |0^{\lceil \log(\ell+1) \rceil}\rangle \to \frac{1}{\sqrt{C}} \sum_{j=0}^{\ell} \sqrt{|c_j|}|j\rangle, \tag{19}$$

$$U_2 : \ |0^{\lceil \log(\ell+1) \rceil}\rangle \to \frac{1}{\sqrt{C}} \sum_{j=0}^{\ell} \sqrt{|c_j|}e^{i\theta_j}|j\rangle, \tag{20}$$

where $C = \sum_{j=0}^{\ell} |c_j|$ and $|c_j|e^{i\theta_j} = c_j$. By Theorem 3, U_1 and U_2 can be prepared with depth $\mathcal{O}(\ell)$ using only elementary quantum gates. Therefore, (U_1, U_2) is a $(C, 2\log\ell, 0)$ state-preparation pair of $(c_0, c_1, \ldots, c_\ell)$.

Let U_j be the $(1, ja + (j-1)n, 0)$-encoding of $(A/\alpha)^{\circ j}$, which we construct by iteratively applying Theorem 4. Then, we construct a $(\ell a + \ell n + 2\log\ell)$-qubit unitary $W = \sum_{j=0}^{\ell} |j\rangle\langle j| \otimes V_j + (I_{2\log\ell} - \sum_{j=0}^{\ell} |j\rangle\langle j|) \otimes I_{\ell a + \ell n}$. We are in the setting of linear combination of block encodings and by Lemma 1, we can implement a $(C, \ell a + (\ell-1)n + 2\log\ell, 0)$-encoding of $f_\ell \circ (A/\alpha)$.

Now we perform the error analysis. Let the exact block-encoded matrix be $\bar{A}$. As mentioned, for each $(A/\alpha)^{\circ j}$, the error is bounded by $j\epsilon/\alpha$. Summing up these errors, the error of $f_\ell \circ (A/\alpha)$ can be bounded by $\frac{\epsilon}{\alpha} \cdot (\sum_{j=0}^{\ell} |c_j| j) =: \gamma$. □

How to use polynomial functions to approximate many useful functions has been well studied in the field of approximation theory. Those results have also been utilized in the quantum computing field for QSVT-based quantum algorithms via *quantum signal processing* [14].

B. Conversion between state preparation encoding and matrix block encoding

Typically for each block in the transformer, the input is a vector ψ and the output is another vector $f(\psi)$ in the same dimension with some nonlinear transformations. As the quantum analog, the question becomes given a state-preparation unitary of some input state $|\psi\rangle$, output a state-preparation unitary of the state $|f(\psi)\rangle$. Hence, as one of the main subroutines, we discuss conversions between matrix block encoding and state preparation encoding.

To achieve this, we use the diagonal block encoding developed in the context of the nonlinear amplitude transformation method, which has been introduced in Ref. [29, 34]. The key insight of the nonlinear amplitude transformation is that it can convert a state preparation encoding as in Definition 3 to a matrix block encoding as Definition 1. Then, by Theorem 2 one can implement polynomial functions onto these amplitudes. For our discussion, we directly describe the robust version, which is a straightforward generalization of previous works. The proof is provided in Appendix A.

Theorem 6 (Robust amplitude encoding [29, 34])**.** *Given an (α, a, ϵ)-state-encoding U_ψ of an n-qubit state $|\psi\rangle = \sum_{j=1}^{N} \psi_j |j\rangle$, where $\{\psi_j\}$ are real and $\|\psi\|_2 = 1$, one can construct an $(\alpha, 2a + n + 2, \epsilon)$-encoding of the diagonal matrix $A = \mathrm{diag}(\psi_1, \dots, \psi_N)$ with $\mathcal{O}(n)$ circuit depth and $\mathcal{O}(1)$ queries to controlled-U and controlled-$U^\dagger$. One can also construct an $(\alpha^2, 3a + 2n + 2, 3\epsilon)$-encoding of diagonal matrix $A_{abs} = \mathrm{diag}(\psi_1^2, \dots, \psi_N^2)$.*

The reason why we slightly changed the definition of state preparation encoding compared to Ref. [29], i.e., from L_2 norm to L_∞ norm is that after robust amplitude encoding, the L_∞ distance between the target state $|\psi\rangle$ and exact preparable state $|\psi'\rangle$ is directly the upper bound of $\|\mathrm{diag}(\psi_1, \dots, \psi_N) - \mathrm{diag}(\psi'_1, \dots, \psi'_N)\|$.

After implementing functions with QSVT, one needs to convert the block-encoding back to the state-encoding. This can be achieved by either the uniform-weighted [34] or the importance-weighted [29] method. The first one is more general, yet the latter one can achieve a much better, i.e., up to exponentially better, dependency on the state dimension. A point to note is about the error analysis. We have the error bound in matrix norm for block-encoding, which is also an upper bound for each matrix element difference, as Lemma 3. However, in general, the column/row of the block-encoded matrix is not normalized in L_2 norm, so we also need to consider the influence of the normalization factor. We prove the following lemmas, which correspond to a general case and the specific case where we know the upper bound. The proofs are provided in C.

Lemma 4. *For two d-dimensional vectors $\psi = (\psi_1, \dots, \psi_d)$ and $\psi' = (\psi'_1, \dots, \psi'_d)$, if $|\psi_j - \psi'_j| \le \epsilon$ for each $j \in [d]$, we have*

$$\left\| \frac{1}{C}\psi - \frac{1}{C'}\psi' \right\|_\infty \le \frac{(\sqrt{d}+1)\epsilon}{C} + \sqrt{\frac{2\epsilon\sqrt{d}}{C}} = \mathcal{O}\left(\sqrt{\frac{2\epsilon\sqrt{d}}{C}}\right), \tag{21}$$

where $C = \|\psi\|_2$ and $C' = \|\psi'\|_2$.

Lemma 5. *For two d-dimensional vectors $\psi = (\psi_1, \dots, \psi_d)$ and $\psi' = (\psi'_1, \dots, \psi'_d)$, if $|\psi_j - \psi'_j| \le \epsilon$ and $\psi_j, \psi'_j \le \Gamma \in \mathcal{O}(1)$ for each $j \in [d]$, we have*

$$\left\| \frac{1}{C}\psi - \frac{1}{C'}\psi' \right\|_\infty \le \frac{\epsilon}{C} + \frac{\Gamma\sqrt{d}\epsilon}{CC'} + \frac{\Gamma}{C'}\sqrt{\frac{2\epsilon\sqrt{d}}{C}}, \tag{22}$$

where $C = \|\psi\|_2$ and $C' = \|\psi'\|_2$.

C. Quantum self-attention

In this section, we describe how to achieve the quantum self-attention block. We are given the block encoding of matrices as input and let j-th token be the current query vector, the output is a block encoding unitary of a matrix whose j-th row is the same as the output of the classical transformer. We divide it into two parts: the first part is to achieve the softmax function, where we use the element-wise function method as Theorem 5; the second part is to achieve the remaining procedures, where we use the amplitude encoding as Theorem 6. The key insight for achieving the softmax function is that it can also be understood that we first implement $\exp\circ(QK^T/\alpha_0)$, then multiply with different coefficients (normalization) for each row.

For quantum self-attention, we set the scaling factor $\alpha_0 = \alpha_s^2\alpha_w^2$ for the following reasons. The first is that the $1/\sqrt{d}$ is chosen somehow in a heuristic sense, and there are some classical works considering different coefficients [35]. The second, which is more important, is that the quantum input assumption using the block encoding format naturally contains the normalization factor α which plays a similar role to $1/\sqrt{d}$. Therefore, for the quantum case in the context of our work, it suffices to use α directly.

Theorem 7 (Quantum softmax for self-attention)**.** *Given an (α, a, ϵ)-encoding U_A of a matrix $A \in \mathbb{R}^{N\times N}$, a positive integer $d \in \mathbb{N}^+$, and an index $j \in [N]$, one can prepare a $\big(1, \mathcal{O}(\ell(a+n)), \mathcal{O}\big(\sqrt[4]{\frac{N}{Z_j}}\sqrt{\epsilon}\big)\big)$-encoding of the matrix*

$$\mathrm{diag}(\mathrm{softmax}(A/\alpha)_{j1}, \dots, \mathrm{softmax}(A/\alpha)_{jN}),$$

by using U_A for $\mathcal{O}\big(\frac{\ell^2}{\sqrt{Z}}\big)$ times, where $Z_j = \sum_{k=1}^N \exp\circ(A/\alpha)_{jk}$, and $\ell = \mathcal{O}\big(n\log(\frac{1}{\epsilon})\big)$.

Proof. We first construct the block encoding of $\exp\circ(\frac{A}{2\alpha})$. This can be achieved with Theorem 5 and Lemma S5. There are two error terms in this step. Note that by the definition of Definition 1, $|A/\alpha|_{jk} \le 1$ for $j, k \in [N]$. The first term comes from the intrinsic error of block encodings, and the second is from the polynomial approximation. Denote $U_{f\circ(A)}$ as the constructed block encoding unitary. By Theorem 5, $U_{f\circ(A)}$ is a (C_f, b_f, γ_f)-encoding of $f_\ell\circ(A)$, where $C_f = \sum_{j=0}^{\ell} 1/j!$, $b_f = \ell a + (\ell-1)n + 2\log\ell$, and $\gamma_f = \frac{\epsilon}{\alpha}\cdot\sum_{j=1}^{\ell} 1/(j-1)!$. By triangle inequality, we have

$$\begin{aligned}
&\left\|\exp\circ\Big(\frac{A}{2\alpha}\Big) - C_f\langle 0^{b_f}|U_{f\circ(A)}|0^{b_f}\rangle\right\| \\
=&\left\|\exp\circ\Big(\frac{A}{2\alpha}\Big) - f_\ell\circ(A) + f_\ell\circ(A) - C_f\langle 0^{b_f}|U_{f\circ(A)}|0^{b_f}\rangle\right\| \\
\le&\left\|\exp\circ\Big(\frac{A}{2\alpha}\Big) - f_\ell\circ(A)\right\| + \left\|f_\ell\circ(A) - C_f\langle 0^{b_f}|U_{f\circ(A)}|0^{b_f}\rangle\right\| \\
\le&\left\|\exp\circ\Big(\frac{A}{2\alpha}\Big) - f_\ell\circ(A)\right\| + \gamma_f.
\end{aligned} \tag{23}$$

Note that we can bound for each element between $\exp\circ(\frac{A}{2\alpha})$ and $f_\ell\circ(A)$ with error δ, which comes from the polynomial approximation. By the norm inequality between spectral and Frobenius norm, we have

$$\begin{aligned}
\left\|\exp\circ\Big(\frac{A}{2\alpha}\Big) - f_k\circ(A)\right\| &\le \left\|\exp\circ\Big(\frac{A}{2\alpha}\Big) - f_k\circ(A)\right\|_F \\
&= \left(\sum_{j,k}\left|\exp\circ\Big(\frac{A}{2\alpha}\Big)_{jk} - f_\ell\circ(A)_{jk}\right|^2\right)^{\frac{1}{2}} \\
&\le \big(N^2\delta^2\big)^{\frac{1}{2}} \le N\delta.
\end{aligned} \tag{24}$$

To make the error bounded ϵ, we set $\ell = \mathcal{O}\big(\log(\frac{N}{\epsilon})\big) = \mathcal{O}\big(n\log(\frac{1}{\epsilon})\big)$. By Lemma S1, we have

$$\begin{aligned}
\max_{j,k\in[N]}\left|\exp\circ\Big(\frac{A}{2\alpha}\Big)_{jk} - C_f(\langle 0^{b_f}|\langle i|)U_{f\circ(A)}(|0^{b_f}\rangle|j\rangle)\right| &\le \left\|\exp\circ\Big(\frac{A}{2\alpha}\Big) - C_f\langle 0^{b_f}|U_{f\circ(A)}|0^{b_f}\rangle\right\| \\
&\le \epsilon + \gamma_f = \mathcal{O}(\epsilon).
\end{aligned} \tag{25}$$

Note that $\exp\circ(\frac{A}{2\alpha})_{jk} = \exp\circ(\frac{A}{2\alpha})^T_{kj}$. For index $j \in [N]$, let $U_j : |0\rangle \to |j\rangle$. Unitary $U^\dagger_{f\circ(A)}(I \otimes U_j)$ and amplitude amplification prepare a state that is close to the target state

$$|A_j\rangle \coloneqq \frac{1}{\sqrt{Z_j}} \sum_{k=1}^{N} \exp\circ\Big(\frac{A}{2\alpha}\Big)_{jk} |k\rangle, \tag{26}$$

where $Z_j = \sum_{k=1}^N \exp\circ(A/\alpha)_{jk}$ is the normalization factor of softmax function for the j-th row. By Lemma 4, the L_∞ distance between the prepared state and the target state is $\mathcal{O}\big((\epsilon\sqrt{N/Z_j})^{\frac{1}{2}}\big)$. Therefore, $U^\dagger_{f\circ(A)}(I \otimes U_j)$ is an $\big(\mathcal{O}(C_f/\sqrt{Z_j}), b_f, \mathcal{O}\big((\epsilon\sqrt{N/Z_j})^{\frac{1}{2}}\big)\big)$-state-encoding of state $|A_j\rangle$. By amplitude amplification [30], one can prepare a $(1, b_f, \mathcal{O}\big((\epsilon\sqrt{N/Z})^{\frac{1}{2}}\big)$-state-encoding of state $|A_i\rangle$ using $\mathcal{O}(C_f/\sqrt{Z})$ times of $U^\dagger_{f\circ(A)}(I \otimes U_i)$. By Theorem 6, this can be converted to a $(1, 2n+3b_f+2, \mathcal{O}\big((\epsilon\sqrt{N/Z})^{\frac{1}{2}}\big))$-encoding of $\mathrm{diag}(\mathrm{softmax}(A/\alpha)_{j1}, \ldots, \mathrm{softmax}(A/\alpha)_{jN})$. □

Then we use the quantum softmax function to implement the block encoding of self-attention matrix, as shown in the following theorem.

Theorem 8 (Quantum self-attention). *Consider the setting as in Problem 1. Let $\alpha_0 = \alpha_s^2\alpha_w^2$. For the index $j \in [N]$, one can construct an $\big(\alpha_s\alpha_w\sqrt{N}, \mathcal{O}(\ell(n+a_s+a_w)), \mathcal{O}\big(\sqrt{N}\sqrt[4]{\frac{N}{Z_j}}\sqrt{\epsilon_s+\epsilon_w}\big)\big)$-encoding of a matrix G such that $G_{j\star} = G^{\mathrm{soft}}_{j\star} \coloneqq (\mathrm{softmax}\big(\frac{QK^T}{\alpha_0^2}\big)V)_{j\star}$, by using $\mathcal{O}(\frac{\ell^2}{\sqrt{Z_j}})$ times of U_S, U_{W_q}, U_{W_k} and U_{W_v}, where $Z_j = \sum_{k=1}^N \exp\circ(QK^T/\alpha_0)_{jk}$, and $\ell = \mathcal{O}(n\log(\frac{1}{\epsilon_s+\epsilon_w}))$.*

Proof. In the first step, we construct the block encoding of matrix QK^T and V. Note that for a real matrix M and its block encoding unitary U_M, $U_M^\dagger$ is the block encoding of M^T. By Lemma 2, one can construct an $(\alpha_0, a_0, \epsilon_0)$-encoding U_{QK^T} of QK^T, where $\alpha_0 \coloneqq \alpha_s^2\alpha_w^2$, $a_0 = 2a_s + 2a_w$, and $\epsilon_0 = 2\alpha_s\alpha_w^2\epsilon_s + 2\alpha_s^2\alpha_w\epsilon_w$. One can also construct an $(\alpha_v, a_v, \epsilon_v)$-encoding U_V of V, where $\alpha_v = \alpha_s\alpha_w$, $a_v = a_s + a_w$, and $\epsilon_v = \alpha_s\epsilon_w + \alpha_w\epsilon_s$.

By Theorem 7, using U_{QK^T} for $\mathcal{O}(\frac{C_f\ell^2}{\sqrt{Z_j}})$ times, where $Z_j = \sum_{k=1}^N \exp\circ(QK^T/\alpha_0)_{jk}$, $\ell = \mathcal{O}\big(n\log(\frac{1}{\epsilon_s+\epsilon_w})\big)$, $C_f = \sum_{j=0}^{\ell}\frac{1}{j!}$, $b_f = \ell a_0 + (\ell-1)n + 2\log\ell$, and $\gamma_f = \frac{\epsilon_0}{\alpha_0}\cdot\sum_{j=1}^{\ell}\frac{1}{(j-1)!} = \mathcal{O}(\epsilon_s+\epsilon_w)$, one can prepare a $(1, 2n+3b_f+2, \mathcal{O}\big(((\epsilon_s+\epsilon_w)\sqrt{N/Z})^{\frac{1}{2}}\big))$-encoding of the matrix

$$\mathrm{diag}\big(\mathrm{softmax}(QK^T/\alpha_0)_{j1}, \ldots, \mathrm{softmax}(QK^T/\alpha_0)_{jN}\big),$$

whose diagonal elements correspond to the j-th row of $\mathrm{softmax}(QK^T/\alpha_0)$. By Lemma 3, the absolute difference for each element is also bounded by $\mathcal{O}\big(((\epsilon_s+\epsilon_w)\sqrt{N/Z})^{\frac{1}{2}}\big)$. Let this block-encoding unitary be $U_{f(QK^T)}$.

Finally, we need to do the matrix multiplication with V. To achieve this, we first need a projection operator $\sum_{k=1}^N |j\rangle\langle k|$ to shift the diagonal elements back to the j-th row. For index $j \in [N]$, let $U_j : |0\rangle \to |j\rangle$. Consider unitary $H_n U_j^\dagger : |j\rangle \to \frac{1}{\sqrt{N}}\sum_{k=1}^N |k\rangle$ and identity I_n. By Lemma 47 in Ref. [13], $(H_n U_j^\dagger)^\dagger I_n = U_j H_n$ is a $(1,0,0)$-encoding of a gram matrix, whose j-th row is $\frac{1}{\sqrt{N}}(1,\ldots,1)$. In other words, $\langle j|U_j H_n|k\rangle = \frac{1}{\sqrt{N}}$ for $k \in [N]$. By the block encoding multiplication between $U_j H_n$ and $U_{f(QK^T)}$ as Lemma 2, we can move the diagonal elements to the j-th row by paying the price of a coefficient $\frac{1}{\sqrt{N}}$. Then the constructed unitary is a $\big(\sqrt{N}, 2n+3b_f+2, \mathcal{O}\big(\sqrt{N}((\epsilon_s+\epsilon_w)\sqrt{N/Z})^{\frac{1}{2}}\big)\big)$-encoding of a matrix G, whose j-th row is the same as the j-th row of $\frac{1}{Z_j}\exp\circ\Big(\frac{QK^T}{\alpha_0}\Big)$. By Lemma 2 again, one can construct an $\big(\alpha_v\sqrt{N}, 2n+3b_f+a_v+2, \mathcal{O}\big(\sqrt{N}((\alpha_v(\epsilon_s+\epsilon_w)\sqrt{N/Z})^{\frac{1}{2}}+\epsilon_v)\big)\big)$-encoding of the matrix GV whose j-th row is the same as j-th row of $G^{\mathrm{soft}} \coloneqq \mathrm{softmax}\Big(\frac{QK^T}{\alpha_0}\Big)V$. In total this needs $\mathcal{O}(C_f\ell^2\sqrt{\frac{1}{Z_j}}) = \mathcal{O}(\ell^2/\sqrt{Z_j})$ times of U_s, U_q, U_k and U_v.

□

Now we consider how to implement the *masked* self-attention, which is essential for the decoder-only structure. This can be achieved by slightly changing some steps as introduced in previous theorems.

Corollary 1 (Quantum masked self-attention). *Consider the same as Problem 1. Let $\alpha_0 = \alpha_s^2\alpha_w^2$. For the index $j \in [N]$, one can construct an $\big(\alpha_s\alpha_w\sqrt{j}, \mathcal{O}(\ell(n+a_s+a_w)), \mathcal{O}\big(\sqrt{j}\cdot\sqrt[4]{\frac{j}{Z_j}}\sqrt{\epsilon_s+\epsilon_w}\big)\big)$-encoding of a matrix G^{mask} such that $G^{\text{mask}}_{j\star} = (\text{softmax}(\frac{QK^T}{\alpha_0}+M)V)_{j\star}$, by using $\mathcal{O}(\frac{\ell^2}{\sqrt{Z_j}})$ times of U_S, U_{W_q}, U_{W_k} and U_{W_v}, where M is the masked matrix as Eq. (4), $Z_j = \sum_{k=1}^N \exp\circ(\frac{QK^T}{\alpha_0}+M)_{jk}$, and $\ell = \mathcal{O}(n\log(\frac{1}{\epsilon_s+\epsilon_w}))$.*

Proof. To achieve the masked self-attention, we change two places of the previous proof in Theorem 7 and Theorem 8. First, in the proof of Theorem 7, we add one more step after constructing a block-encoding of matrix $\exp\circ(\frac{A}{2\alpha})$. For the index $j \in [N]$, we multiply $\exp\circ(\frac{A}{2\alpha})$ with a projector $\sum_{k:k\le j}|k\rangle\langle k|$ to mask the elements. Though the projector $\sum_{k\in\mathcal{S}}|k\rangle\langle k|$ for $S \subseteq [N]$ is not unitary in general, one can construct a block encoding of the projector by noticing that it can be written by the linear combination of two unitaries:

$$\sum_{k\in\mathcal{S}}|k\rangle\langle k| = \frac{1}{2}I + \frac{1}{2}\Big(2\sum_{k\in\mathcal{S}}|k\rangle\langle k| - I\Big). \tag{27}$$

Define $U_{\text{proj}} := |0\rangle\langle 0|\otimes I + |1\rangle\langle 1|\otimes(2\sum_{k\in\mathcal{S}}|k\rangle\langle k| - I)$. One can easily verify that $(H\otimes I)U_{\text{proj}}(H\otimes I)$ is a $(1,1,0)$-encoding of $\sum_{k\in\mathcal{S}}|k\rangle\langle k|$, where H is the Hadamard gate. Let $\mathcal{S} = [j]$, by Lemma 2, one can construct a (C_f, b_f+1, γ_f)-encoding of $\exp\circ(\frac{A}{2\alpha})\sum_{k:k\le j}|k\rangle\langle k|$, i.e., only add an ancilla qubit in the final result. Note that $Z_j = \sum_{k=1}^N \exp\circ(QK^T/\alpha_{qk}+M)_{jk} = \sum_{k=1}^j \exp\circ\big(QK^T/\alpha_0\big)_{jk}$.

Second, we change the projection operator prepared in Theorem 8. As there are at most j non-zero elements in the j-th row for masked self-attention case, it suffices to prepare the isometry $\sum_{k=1}^j|k\rangle\langle k|$ instead of $\sum_{k=1}^N|k\rangle\langle k|$. As a result, the coefficient changes from $\frac{1}{\sqrt{N}}$ to $\frac{1}{\sqrt{j}}$. □

One may further achieve the multi-head self-attention case by using the linear combination of unitaries. We do not describe further details on multi-head attention in this work. For simplicity, in the following, we will directly say we have a $(\alpha_g, a_g, \epsilon_g)$-encoding of G, e.g., $\alpha_g = \alpha_s\alpha_w\sqrt{N}$, $a_g = \mathcal{O}(\ell(n+a_s+a_w))$ and $\epsilon_g = \mathcal{O}(\sqrt{N}\sqrt[4]{\frac{N}{Z_j}}\sqrt{\epsilon_s+\epsilon_w})$.

D. Quantum residual connection

Here, we first show how to achieve the task as Problem 2 basically following the nonlinear amplitude transformation method [29, 34]. Then, we discuss how to implement the residual connection with layer normalization as Problem 3.

Theorem 9 (Quantum residual connection). *Consider the setting of Problem 2. For the polynomial $g(x)$, let $g_{\max} := \max_{x\in[-1,1]}|g(\alpha x)|$, one can prepare an $\big(\mathcal{O}\big(\sqrt{N}(\alpha+2cg_{\max})/C\big), a+n+4, \mathcal{O}\big((cg_{\max}(4\ell\sqrt{\epsilon}+\delta)+\alpha\epsilon)/C\big)\big)$-state-encoding of the state $\frac{1}{C}\sum_{k=1}^N(c\cdot g(x_k)+x_k)|k\rangle$, where $C^2 := \sum_{k=1}^N(c\cdot g(x_k)+x_k)^2$. Further, if $g(x)/x$ is bounded with $\eta := \max_{x\in[-1,1]}|g(\alpha x)/x|$, one can prepare an $\big(\mathcal{O}(\alpha(1+2c\eta)/C), a+n+4, \mathcal{O}(c\eta(4\ell\sqrt{\epsilon}+\delta)/C)\big)$-state-encoding instead. The preparation uses $\mathcal{O}(\ell)$ times of U_x and $U_x^\dagger$.*

Proof. We first discuss the general case. Given the state-encoding U_x, by Theorem 6, one can construct an $(\alpha, a+n+2, \epsilon)$-encoding of $A = \text{diag}(x_1,\dots,x_N)$. Let $g_{\max} := \max_{x\in[-1,1]}|g(\alpha x)|$, then by Theorem 2 with function $g(x)/(2g_{\max})$, one can construct a $(2g_{\max}, a+n+4, 2g_{\max}(4\ell\sqrt{\epsilon}+\delta))$-encoding of the matrix $\text{diag}(g(x_1),\dots,g(x_N))$. Note that the normalization factor $2g_{\max}$ is to satisfy the requirements of Theorem 2.

By using the linear combination of block-encoded matrices as Lemma 1 with state preparation pair (P,P), where $P: |0\rangle \to 1/\sqrt{\alpha+2cg_{\max}}(\sqrt{\alpha}|0\rangle + \sqrt{2cg_{\max}}|1\rangle)$, one can construct an $(\alpha+2cg_{\max}, a+n+$

$5, 2cg_{\max}(4\ell\sqrt{\epsilon}+\delta)+\alpha\epsilon)$-encoding U_g of the matrix $\mathrm{diag}(c\cdot g(x_1)+x_1,\ldots,c\cdot g(x_N)+x_N)$. One can easily verify that $U_g(I\otimes H_n)$ is a state-encoding of the target state $\frac{1}{C}\sum_{k=1}^{N}(c\cdot g(x_k)+x_k)|k\rangle$. We have

$$\begin{aligned} U_g(I\otimes H_n)|0\rangle|0\rangle &= \frac{1}{\sqrt{N}(\alpha+2cg_{\max})}|0\rangle\sum_{k=1}^{N}\psi_k|k\rangle+|\widetilde{\perp}\rangle \\ &= \frac{C'}{\sqrt{N}(\alpha+2cg_{\max})}|0\rangle\frac{1}{C'}\sum_{k=1}^{N}\psi_k|k\rangle+|\widetilde{\perp}\rangle, \end{aligned} \tag{28}$$

where $C'=\|\psi\|_2$, $\|\psi-(c\cdot g(x)+x)\|_\infty\le 2cg_{\max}(4\ell\sqrt{\epsilon}+\delta)+\alpha\epsilon$, and $|\widetilde{\perp}\rangle$ is a unnormalized orthogonal state. For simplicity, let $\epsilon_g:=2cg_{\max}(4\ell\sqrt{\epsilon}+\delta)+\alpha\epsilon$. By Lemma 4, the final error bound is

$$\frac{\epsilon_g}{C}+\frac{(cg_{\max}+1)}{C'}\left(\frac{\sqrt{N}\epsilon_g}{C}+\sqrt{\frac{2\sqrt{N}\epsilon_g}{C}}\right)=\mathcal{O}\big((cg_{\max}(4\ell\sqrt{\epsilon}+\delta)+\alpha\epsilon)/C\big).$$

Now we consider the specific case, i.e., when the polynomial $g(x)$ has no constant term. Note that for a polynomial $g(x)$, if $g(x)/x$ is bounded on the interval across $x=0$, it cannot have constant term. Instead of implementing function $g(x)/(2g_{\max})$ with quantum singular value transformation, here we implement $g'(A)/2\eta$ instead, where $g'(x):=g(\alpha x)/x$ and $\eta:=\max_{x\in[-1,1]}|g'(x)|$. By Lemma 1 with state preparation pair (P',P'), where $P':|0\rangle\to 1/(\sqrt{1+2c\eta})(|0\rangle+\sqrt{2c\eta}|1\rangle)$ to construct a $(1+2c\eta, a+n+4, 2c\eta(4\ell\sqrt{\epsilon}+\delta))$-encoding of diagonal matrix $I+c\cdot g'(A)$. Let this block-encoding unitary be $U_{g'}$ and $\epsilon_{g'}:=2c\eta(4\ell\sqrt{\epsilon}+\delta)$. We have $U_{g'}(I\otimes U_x)$ is the $\left(\frac{\alpha(1+2c\eta)}{C''}, a+n+4, \frac{\epsilon_{g'}}{C}+\frac{(c\eta+1)}{C''}\left(\frac{\sqrt{N}\epsilon_{g'}}{C}+\sqrt{\frac{2\sqrt{N}\epsilon_{g'}}{C}}\right)\right)$-state-encoding of the target state, where C'' is the L_2 norm for the exact prepared state. ☐

In the following, we explicitly consider the residual connection with layer normalization in the transformer framework.

Theorem 10 (Quantum residual connection with layer normalization)**.** *Consider the setting of Problem 3. One is able to construct an $(\mathcal{O}(\sqrt{d}(\alpha_g+\alpha_s)/\varsigma), 2a_g+n+4, \mathcal{O}((\epsilon_g+\epsilon_s)/\varsigma))$-state-encoding of the state*

$$\sum_{k=1}^{d}\mathrm{LN}(G^{\mathrm{soft}}_{jk}+S_{jk})|k\rangle=\frac{1}{\varsigma}\sum_{k=1}^{d}(G^{\mathrm{soft}}_{jk}+S_{jk}-\bar{s}_j)|k\rangle,$$

where $\bar{s}_j:=\frac{1}{d}\sum_{k=1}^{d}(G^{\mathrm{soft}}_{jk}+S_{jk})$ and $\varsigma:=\sqrt{\sum_{k=1}^{d}(G^{\mathrm{soft}}_{jk}+S_{jk}-\bar{s}_j)^2}$.

Proof. As shown in Theorem 8, we can construct an $(\alpha_g, a_g, \epsilon_g)$-encoding of a matrix whose j-th row is the same row as that of G^{soft}. By assumption, we are given U_s which is an $(\alpha_s, a_s, \epsilon_s)$-encoding of S. By Lemma 1 with state preparation pair (P,P) such that

$$P|0\rangle=\frac{1}{\sqrt{\alpha_g+\alpha_s}}(\sqrt{\alpha_g}|0\rangle+\sqrt{\alpha_s}|1\rangle), \tag{29}$$

one can construct a quantum circuit U_{res} which is an $(\alpha_g+\alpha_s, a_g+1, \epsilon_g+\epsilon_s)$-encoding of an $N\times d$ matrix whose j-th row is the same as that of $G^{\mathrm{soft}}+S$.

Now we consider how to create a block encoding of a diagonal matrix $\bar{s}_j\cdot I$, where $\bar{s}_j:=\frac{1}{d}\sum_{k=1}^{d}(G^{\mathrm{soft}}_{jk}+S_{jk})$. Let us define a unitary $H_{\log d}:=H^{\otimes\log d}$. Note that $H_{\log d}$ is a $(1,0,0)$-encoding of itself, and the first column of $H_{\log d}$ is $\frac{1}{\sqrt{d}}(1,\ldots,1)^T$. By Lemma 2, one can multiply $G^{\mathrm{soft}}+S$ with $H_{\log d}$ to construct an $(\alpha_g+\alpha_s, a_g+1, \epsilon_g+\epsilon_s)$-encoding of an $N\times d$ matrix, whose $(i,1)$-element is $\sqrt{d}\bar{s}_i$. One can further move this element to $(1,1)$ by switching the first row with the i-th row. By tensoring with the identity I of $\log d$ qubits, one can construct an $(\alpha_g+\alpha_s, a_g+n+1, \epsilon_g+\epsilon_s)$-encoding of $\sqrt{d}\bar{s}_i\cdot I$.

With $U_j : |0\rangle \to |j\rangle$, one can prepare the state

$$U_{\text{res}}^\dagger (I \otimes U_j)|0\rangle|0\rangle = \frac{1}{\alpha_g + \alpha_s}|0\rangle \sum_{k=1}^{d} \psi'_k |k\rangle + \sqrt{1 - \frac{\sum_k \psi_k'^2}{(\alpha_g + \alpha_s)^2}}|1\rangle|\text{bad}\rangle, \tag{30}$$

where $|\psi'_k - (G_{jk}^{\text{soft}} + S_{jk})| \leq \epsilon_g + \epsilon_s$ for $k \in [d]$. By Theorem 6, this can be converted to an $(\alpha_g + \alpha_s, 2a_g + n + 3, \epsilon_g + \epsilon_s)$-encoding of the diagonal matrix $\text{diag}(G_{j1} + S_{j1}, \dots, G_{jd} + S_{jd})$.

By Lemma 1 with state preparation pair (P_1, P_2), where

$$P_1|0\rangle = \frac{1}{\sqrt{1 + 1/\sqrt{d}}}(|0\rangle + \frac{1}{\sqrt{d}}|1\rangle) \tag{31}$$

and

$$P_2|0\rangle = \frac{1}{\sqrt{1 + 1/\sqrt{d}}}(|0\rangle - \frac{1}{\sqrt{d}}|1\rangle), \tag{32}$$

one can construct an $((\alpha_g + \alpha_s)(1 + 1/\sqrt{d}), 2a_g + n + 4, (\epsilon_g + \epsilon_s)(1 + 1/\sqrt{d}))$-encoding of $\text{diag}(G_{j1} + S_{j1} - \bar{s}_j, \dots, G_{jd} + S_{jd} - \bar{s}_j)$.

Let this unitary be U_{LN}. Then the unitary $U_{\text{LN}}(I \otimes H_{\log d})$ is an $(\mathcal{O}(\sqrt{d}(\alpha_g + \alpha_s)/\varsigma), 2a_g + n + 4, \mathcal{O}((\epsilon_g + \epsilon_s)/\varsigma))$-state-encoding of the state

$$\frac{1}{\varsigma}\sum_{k=1}^{d}(G_{jk}^{\text{soft}} + S_{jk} - \bar{s}_j)|k\rangle,$$

where $\varsigma := \sqrt{\sum_{k=1}^d (G_{jk}^{\text{soft}} + S_{jk} - \bar{s}_j)^2}$. □

E. Quantum feedforward network

For the activation function, we consider the well-known *sigmoid functions*. To satisfy the requirement of quantum singular value transformation, we further consider activation functions that can be well approximated by polynomial functions. Functions like $\text{ReLU}(x) = \max(0, x)$ do not satisfy this condition, hence we do not consider them here. We list some of the implementable functions as follows.

Lemma 6 (Polynomial approximation of tanh function [34])**.** *Let $\epsilon > 0$. For $x \in [-\frac{\pi}{2}, \frac{\pi}{2}]$, the function $\tanh(x)$ can be approximated with error up to ϵ by a polynomial function with degree $\mathcal{O}(\log(\frac{1}{\epsilon}))$.*

Lemma 7 (Polynomial approximation of error function [36])**.** *Let $\epsilon > 0$. For every $k > 0$, the error function $\text{erf}(kx) := \frac{2}{\sqrt{\pi}}\int_0^{kx} e^{-t^2}\, dt$ can be approximated with error up to ϵ by a polynomial function with degree $\mathcal{O}(k\log(\frac{1}{\epsilon}))$.*

Based on this lemma, one can easily approximate the GELU function with polynomials.

Corollary 2 (Polynomial approximation of GELU function)**.** *Let $\epsilon > 0$ and $\lambda \in \mathcal{O}(1)$. For every $k > 0$ and $x \in [-\lambda, \lambda]$, the* GELU *function* $\text{GELU}(kx) := kx \cdot \frac{1}{2}(1 + \text{erf}(\frac{kx}{\sqrt{2}}))$ *can be approximated with error up to ϵ by a polynomial function with degree $\mathcal{O}(k\log(\frac{k\lambda}{\epsilon}))$.*

Proof. It suffices to approximate the error function with precision $\frac{\epsilon}{k\lambda}$ by Lemma 7. □

In the following theorem, we consider how to implement the two-layer feedforward network. As mentioned in Section II B, GELU function is widely used in transformer-based models and we explicitly consider it as the activation function in the theorem. Cases for other functions follow the same computation.

Theorem 11 (Two-layer feedforward network with GELU function)**.** *Consider the setting as in Problem 4. Let the activation function be* $\mathrm{GELU}(x) \coloneqq x \cdot \frac{1}{2}(1+\mathrm{erf}(\frac{x}{\sqrt{2}}))$*. One can prepare an* $(\mathcal{O}(\alpha\alpha_m^2/C), 2a+n+2a_m+4, \mathcal{O}((\frac{\sqrt{N_2}}{C}\alpha_m^2\ell'\sqrt{\alpha_m\epsilon+\epsilon_m})^{\frac{1}{2}}))$*-state-encoding of the state*

$$|\phi\rangle = \frac{1}{C}\sum_{k=1}^{N_2}\Big(M_2\cdot\mathrm{GELU}(M_1\cdot\psi)\Big)_k|k\rangle, \tag{33}$$

by using ℓ' *times of* U_ψ *and* $U_\psi^\dagger$*, where* C *is the normalization factor and* $\ell' = \tilde{\mathcal{O}}(\alpha\alpha_m\log(1/\epsilon_m))$*.*

Proof. Let the erroneous block-encoded matrices be M_1' and M_2'. We have

$$(I_a\otimes U_{M_1})(I_{a_m}\otimes U_\psi)|0^{a+a_m+n}\rangle = \frac{1}{\alpha\alpha_m}|0^{a+a_m}\rangle M_1'|\psi'\rangle + |\widetilde{\perp}\rangle, \tag{34}$$

where $|\widetilde{\perp}\rangle$ is an unnormalized orthogonal state. For the case $N_1 \geq N$, this can be achieved by padding ancilla qubits to the initial state. By direct computation, we have

$$\begin{aligned}
&\|M_1|\psi\rangle - M_1'|\psi'\rangle\|_\infty\\
\leq&\|M_1|\psi\rangle - M_1|\psi'\rangle + M_1|\psi'\rangle - M_1'|\psi'\rangle\|_\infty\\
\leq&\|M_1|\psi\rangle - M_1|\psi'\rangle\|_\infty + \|M_1|\psi'\rangle - M_1'|\psi'\rangle\|_\infty\\
\leq&\|M_1\|\||\psi\rangle - |\psi'\rangle\|_\infty + \|M_1 - M_1'\|\||\psi'\rangle\|_\infty\\
\leq&\alpha_m\epsilon + \epsilon_m.
\end{aligned} \tag{35}$$

By Theorem 6, one can construct an $(\alpha\alpha_m, a+n+2, \alpha_m\epsilon+\epsilon_m)$-encoding of matrix $\mathrm{diag}((M_1\psi)_1,\ldots,(M_1\psi)_{N_1})$. Note that the GELU function does not have a constant term, and is suitable to use the importance-weighted amplitude transformation as in Ref. [29]. Instead of directly implementing the GELU function, we first implement the function $f(x) = \frac{1}{2}(1+\mathrm{erf}(\frac{x}{\sqrt{2}}))$. Note that the value of $|\mathrm{erf}(x)|$ is upper bounded by 1. By Theorem 6 with function $\frac{1}{4}(1+\mathrm{erf}(\alpha\alpha_m\frac{x}{\sqrt{2}}))$, one can construct a $(2, a+n+4, 4\ell\sqrt{\alpha_m\epsilon+\epsilon_m}+\gamma+\delta)$-encoding of matrix $\mathrm{diag}(f(M_1\psi)_1,\ldots,f(M_1\psi)_{N_1})$, where $\ell = \tilde{\mathcal{O}}(\alpha\alpha_m\log(1/\gamma))$.

Let the previously constructed block-encoding unitary be $U_{f(x)}$. We have

$$U_{f(x)}(I\otimes U_{M_1})(I\otimes U_\psi)|0\rangle|0\rangle = \frac{1}{2\alpha\alpha_m}|0\rangle\sum_k \mathrm{GELU}'(M_1'\psi')_k|k\rangle + |\widetilde{\perp'}\rangle, \tag{36}$$

where $|\widetilde{\perp'}\rangle$ is an unnormalized orthogonal state. Setting $\gamma,\delta = \mathcal{O}(\epsilon_m)$, by direct computation, we have

$$\begin{aligned}
&\|\mathrm{GELU}'(M_1'\psi') - \mathrm{GELU}(M_1\psi)\|_\infty\\
=&\|M_1'\psi' f'(M_1'\psi') - M_1\psi f(M_1\psi)\|_\infty\\
\leq&\|M_1'\psi' f'(M_1'\psi') - M_1'\psi' f(M_1\psi)\|_\infty + \|M_1'\psi' f(M_1\psi) - M_1\psi f(M_1\psi)\|_\infty\\
\leq&\alpha_m(4\ell\sqrt{\alpha_m\epsilon+\epsilon_m}+\gamma+\delta) + \alpha_m\epsilon+\epsilon_m = \mathcal{O}(\alpha_m\ell\sqrt{\alpha_m\epsilon+\epsilon_m}).
\end{aligned} \tag{37}$$

Finally, by implementing the block-encoding unitary U_{M_2}, we have

$$\begin{aligned}
&(I\otimes U_{M_2})(I\otimes U_{f(x)})(I\otimes U_{M_1})(I\otimes U_\psi)|0\rangle|0\rangle\\
=&\frac{C'}{2\alpha\alpha_m^2}|0\rangle\frac{1}{C'}\sum_j \psi_{\mathrm{fin}}|j\rangle + |\widetilde{\perp''}\rangle,
\end{aligned} \tag{38}$$

where C' is the exact normalization factor, $\|\psi_{\mathrm{inf}} - M_2\mathrm{GELU}(M_1\psi)\|_\infty = \mathcal{O}(\alpha_m^2\ell'\sqrt{\alpha_m\epsilon+\epsilon_m}+\epsilon_m) = \mathcal{O}(\alpha_m^2\ell'\sqrt{\alpha_m\epsilon+\epsilon_m})$, and $|\widetilde{\perp}''\rangle$ is an unnormalized orthogonal state. By Lemma 4, we have

$$\Big\|\frac{1}{C'}\psi_{\mathrm{inf}} - \frac{1}{C}M_2\mathrm{GELU}(M_1\psi)\Big\|_\infty = \mathcal{O}\left(\left(\frac{\sqrt{N_2}}{C}\alpha_m^2\ell'\sqrt{\alpha_m\epsilon+\epsilon_m}\right)^{\frac{1}{2}}\right). \tag{39}$$

□

F. Quantum one-layer transformer

Combining the previous results, one can obtain the following result. Note that for a one-layer transformer, we mean the same as Fig. 1, i.e., combined with a self-attention block, a two-layer feedforward network, and two residual connection with layer normalization blocks. The proof is provided in Appendix E.

Theorem 12 (Quantum one-layer Transformer). *Let the input assumptions be as in Definition 4. If $\epsilon_s, \epsilon_w, \epsilon_m = o(\epsilon^8 \alpha_m^{-26} d^{-4} \varsigma^2 \varsigma'^8 \sqrt{\frac{Z_j}{N}} \frac{1}{N})$, then for the index $j \in [N]$, one can construct a $(1, \mathcal{O}(\ell(n + a_s + a_w) + a_M), \epsilon)$-state-encoding of the quantum state*

$$\sum_{k=1}^{d} \mathrm{Transformer}(S, j)_k |k\rangle, \tag{40}$$

by using $\mathcal{O}(d\alpha_s \alpha_w \alpha_m^3 \ell^2 \sqrt{\frac{N}{Z_j}} \frac{1}{\varsigma\varsigma'} \log(\frac{1}{\epsilon_m}))$ times of U_S, U_{W_q}, U_{W_k}, U_{W_v} and U_M, where $\ell = \mathcal{O}(n \log(\frac{1}{\epsilon_s + \epsilon_w}))$, $Z_j = \sum_{k=1}^{N} \exp \circ (QK^T / \alpha_s^2 \alpha_w^2)_{jk}$, and ς, ς' are standard deviations from two layer normalization blocks.

In the next section, we discuss the simplification of this theorem to arrive at our informal version (Theorem 1), and a statement about the potential of quantum advantages.

V. DISCUSSION

We show in this work progress towards implementing transformer architectures on fault-tolerant quantum computers. We show how to formulate and achieve each block of the transformer as quantum subroutines, which can be further combined in a modular fashion. We discuss here several aspects and open directions.

Related works — We note previous works on quantum algorithms for (part of) the transformer architecture [37–40]. They consider the quantum analogue version in the variational quantum circuit setting, or focus on the self-attention matrix computation based on the Grover algorithm.

Input assumption — There are multiple ways to construct the block encoding as given in the Definition 4. In general, these block encodings can be achieved with a QRAM and the data structure mentioned in Ref. [41]. However, we note that the price to pay for this step is typically measured by the Frobenius norm, where the optimal dependency is on the spectral norm. To obtain a better intuition, we compute the spectral and Frobenius norms of weight matrices (W_q, W_k, W_v) for several open-source large language models[2]. The result can be seen in Fig. 2.

Many of the LLMs below a dimension d of 10^3 that we have checked have substantially different norms. We observe that for larger models such as *Llama2-7b* and *Mistral-7b*, the norms do not change dramatically. Hence, it appears that the spectral norm and the Frobenius norm of the weight matrices as a function of d scale roughly constant or $\mathcal{O}(\log d)$. We leave characterization of the input sequence matrix S for future work, as it depends on the specific instances. Under the label of efficient transformer [42], many classical works utilize ideas like sparsification and low rank approximation to make the matrix computation more efficient. These results may also benefit the quantum side, e.g., being able to use the standard sparse oracle for block encoding. Other methods may also be possible to achieve the input assumption based on more understanding of the input sequence and weight matrices.

[2] Parameters are obtained from the Hugging Face website, which is an open-source platform for machine learning models.

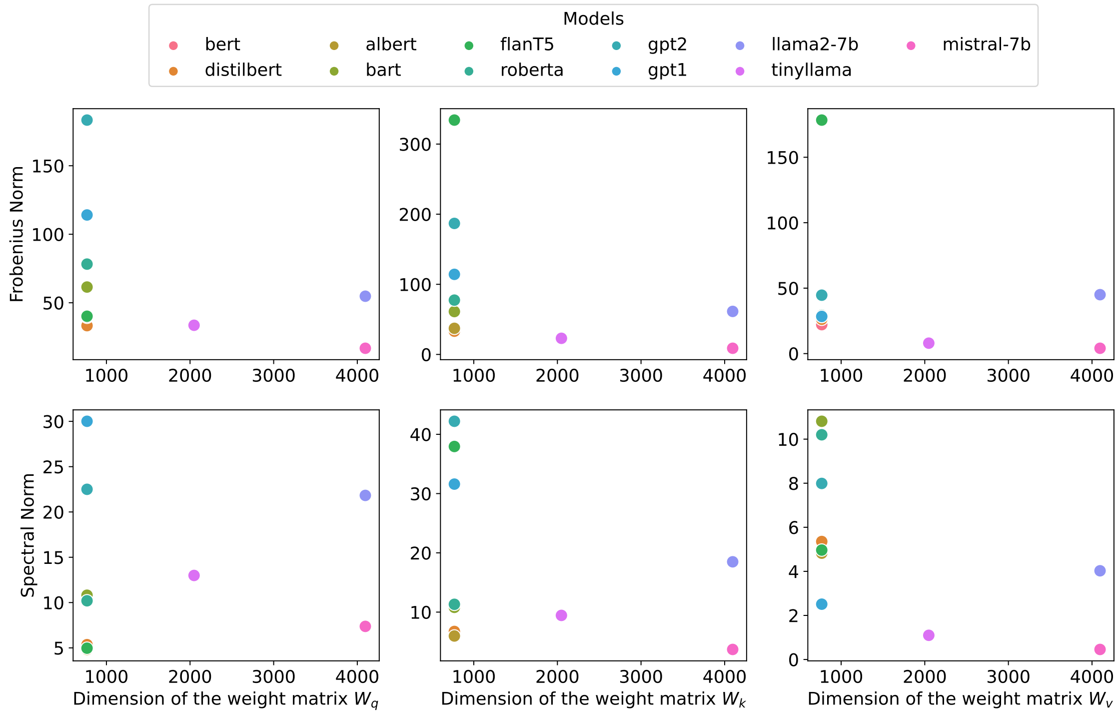

FIG. 2. Norms of weight matrices in popular open-source LLMs. We compute the spectral and Frobenius norms of the weight matrices W_q, W_k and W_v in the first layer. Note that for the multi-head self-attention, matrices have been concatenated to achieve the square matrix.

Possible quantum advantage — The ability to obtain a quantum advantage hinges on how the input is given and the particular problem. We do not provide a provable end-to-end advantage here, but rather develop the pertinent quantum subroutines and combine them into a transformer architecture. Given the input, our subroutines are efficient in several aspects. They use a number of working plus ancilla qubits that is logarithmic in the problem size specified by the sequence length N and the embedding size d. In addition, due to the use of block-encoding and polynomial approximation techniques, we use a near-optimal circuit depth in terms of the final accuracy. The use of an amplification step and its cost depends on the final task at hand. A regime for a possible quantum advantage is summarized in the Table II. If we use the bounds in this table, we obtain a number of queries to the input of $\widetilde{\mathcal{O}}(d^{3/2}\sqrt{N})$. The classical run time is $\mathcal{O}(Nd + d^2)$. The efficiency of the subroutines allows for the potential for larger speedups in other regimes.

Quantity	**Symbol**	**Regime**
Softmax normalization factor	Z_j	$\Omega(N)$
Sequence matrix normalization	α_s	$O(\sqrt{N})$
Attention weight matrix normalization	α_w	$O(\sqrt{d})$
Layer normalization factors	ς, ς'	$\Omega(1)$
FNN matrix normalization	α_m	$O(1)$
Final output error	ϵ	$\Omega(1/N)$

TABLE II. A possible regime for the transformer where a quantum advantage could be exhibited, based on our result in Theorem 12.

Future research directions and open problems — We conclude with several remaining questions and research directions. First, we leave open the discussion about the complexities of multilayer architectures. In many cases with naive concatenation of our subroutines, the complexity will be exponential in the number of layers. Are there situations where this dependence on the number of layers can be avoided? One possible direction is to utilize the results from quantum state learning. The output of each block is usually the quantum states without any complex phases, i.e., they can be understood as distributions. Further, the dimension of the quantum state is d instead of N, where N is the dominant factor. Therefore, using methods like classical shadows [43–45] or machine learning-based methods [46, 47] may enable the efficient extension to multi-layers.

Second, we leave open a more detailed analysis of the required quantum resources. Our asymptotic analysis does not explicitly specify multiplicative constants in the complexity. Also, as mentioned above, concrete problem settings with specific parameters for the input should be investigated to see if there exists the possibility for an end-to-end quantum advantage.

In addition, we have not considered the training step of the transformer architecture on the quantum computer. The weight matrices are determined from a large data set of training data and the optimization of a loss function. Embedding large data into quantum computers is difficult in the absence of the availability of functioning quantum RAMs. The iterative update of the weights would potentially also incur significant overheads in terms of measurements. Hence, it could be an interesting direction to train the weights on *quantum data*, which may allow for a more direct construction of the weight-matrix block encodings. It may also be interesting to explore whether the residual connection, described as one key block here, may improve the trainability of parameterized quantum circuits similar to how it improves the trainability of classical neural networks.

Acknowledgement This research is supported by the National Research Foundation, Singapore, and A*STAR under its CQT Bridging Grant and its Quantum Engineering Programme under grant NRF2021-QEP2-02-P05. NG thanks Fuzhao Xue and Zhi Li for their insightful discussions about classical transformer architectures.

[1] OpenAI. GPT-4 technical report. *arXiv:2303.08774*, 2023.

[2] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv:2303.12712*, 2023.

[3] Ce Zhou, Qian Li, Chen Li, Jun Yu, Yixin Liu, Guangjing Wang, Kai Zhang, Cheng Ji, Qiben Yan, Lifang He, Hao Peng, Jianxin Li, Jia Wu, Ziwei Liu, Pengtao Xie, Caiming Xiong, Jian Pei, Philip S. Yu, and Lichao Sun. A Comprehensive Survey on Pretrained Foundation Models: A History from BERT to ChatGPT, 2023.

[4] Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, and et al. On the opportunities and risks of foundation models. *arXiv:2108.07258*, 2022.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[6] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*, 2016.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*, 2019.

[8] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. *https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language_understanding_paper.pdf*, 2018.

[9] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf*, 2019.

[10] Tom B. Brown, Benjamin Mann, Nick Ryder, and et al. Language models are few-shot learners. *arXiv:2005.14165*, 2020.

[11] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum Algorithm for Linear Systems of Equations. *Physical Review Letters*, 103(15):150502, October 2009.

[12] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*,

10(9):631–633, September 2014.
[13] András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 193–204, June 2019.
[14] Guang Hao Low and Isaac L. Chuang. Optimal Hamiltonian Simulation by Quantum Signal Processing. *Physical Review Letters*, 118(1):010501, January 2017.
[15] Guang Hao Low and Yuan Su. Quantum eigenvalue processing. *arXiv:2401.06240*, 2024.
[16] Laird Egan, Dripto M. Debroy, Crystal Noel, Andrew Risinger, Daiwei Zhu, Debopriyo Biswas, Michael Newman, Muyuan Li, Kenneth R. Brown, Marko Cetina, and Christopher Monroe. Fault-tolerant control of an error-corrected qubit. *Nature*, 598(7880):281–286, October 2021.
[17] Google Quantum AI. Suppressing quantum errors by scaling a surface code logical qubit. *Nature*, 614(7949):676–681, February 2023.
[18] Dolev Bluvstein, Simon J. Evered, Alexandra A. Geim, and Et al. Logical quantum processor based on reconfigurable atom arrays. *Nature*, 626(7997):58–65, 2024.
[19] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone. Quantum random access memory. *Phys. Rev. Lett.*, 100:160501, Apr 2008.
[20] Samuel Jaques and Arthur G. Rattew. Qram: A survey and critique. *arXiv:2305.10310*, 2023.
[21] Guang Hao Low and Isaac L. Chuang. Hamiltonian Simulation by Qubitization. *Quantum*, 3:163, July 2019.
[22] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In Katrin Erk and Noah A. Smith, editors, *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany, August 2016. Association for Computational Linguistics.
[23] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In Eduardo Blanco and Wei Lu, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium, November 2018. Association for Computational Linguistics.
[24] Sabrina J. Mielke, Zaid Alyafeai, Elizabeth Salesky, Colin Raffel, Manan Dey, Matthias Gallé, Arun Raja, Chenglei Si, Wilson Y. Lee, Benoît Sagot, and Samson Tan. Between words and characters: A brief history of open-vocabulary modeling and tokenization in nlp. *arXiv:2112.10508*, 2021.
[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv:1512.03385*, 2015.
[26] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *arXiv:1607.06450*, 2016.
[27] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *https://openreview.net/forum?id=Bk0MRI5lg*, 2017.
[28] Shantanav Chakraborty, András Gilyén, and Stacey Jeffery. The Power of Block-Encoded Matrix Powers: Improved Regression Techniques via Faster Hamiltonian Simulation. In Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi, editors, *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
[29] Arthur G. Rattew and Patrick Rebentrost. Non-linear transformations of quantum amplitudes: Exponential improvement, generalization, and applications. *arXiv:2309.09839*, 2023.
[30] Gilles Brassard, Peter Høyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Information (Washington, DC, 2000)*, volume 305 of *Contemporary Mathematics*, pages 53–74. American Mathematical Society, Providence, RI, 2002.
[31] Xiaoming Sun, Guojing Tian, Shuai Yang, Pei Yuan, and Shengyu Zhang. Asymptotically Optimal Circuit Depth for Quantum State Preparation and General Unitary Synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(10):3301–3314, October 2023.
[32] Xiao-Ming Zhang, Tongyang Li, and Xiao Yuan. Quantum State Preparation with Optimal Circuit Depth: Implementations and Applications. *Physical Review Letters*, 129(23):230504, November 2022.
[33] Liming Zhao, Zhikuan Zhao, Patrick Rebentrost, and Joseph Fitzsimons. Compiling basic linear algebra subroutines for quantum computers. *Quantum Machine Intelligence*, 3(2):21, June 2021.
[34] Naixu Guo, Kosuke Mitarai, and Keisuke Fujii. Nonlinear transformation of complex amplitudes via quantum singular value transformation. *arXiv:2107.10764*, 2021.
[35] Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer. *arXiv:2203.03466*, 2022.
[36] Guang Hao Low. *Quantum Signal Processing by Single-Qubit Dynamics*. Thesis, Massachusetts Institute of Technology, 2017.
[37] El Amine Cherrat, Iordanis Kerenidis, Natansh Mathur, Jonas Landman, Martin Strahm, and Yun Yvonna Li.

Quantum vision transformers. *arXiv:2209.08167*, 2024.

[38] Guangxi Li, Xuanqiang Zhao, and Xin Wang. Quantum self-attention neural networks for text classification. *arXiv:2205.05625*, 2023.

[39] Yeqi Gao, Zhao Song, Xin Yang, and Ruizhe Zhang. Fast quantum algorithm for attention computation. *arXiv:2307.08045*, 2023.

[40] 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, 2022.

[41] Iordanis Kerenidis and Anupam Prakash. Quantum Recommendation Systems. *arXiv:1603.08675*, September 2016.

[42] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *arXiv:2009.06732*, 2022.

[43] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting Many Properties of a Quantum System from Very Few Measurements. *Nature Physics*, 16(10):1050–1057, October 2020.

[44] Hsin-Yuan Huang, Yunchao Liu, Michael Broughton, Isaac Kim, Anurag Anshu, Zeph Landau, and Jarrod R. McClean. Learning shallow quantum circuits. *arXiv:2401.10095*, 2024.

[45] Christian Bertoni, Jonas Haferkamp, Marcel Hinsche, Marios Ioannou, Jens Eisert, and Hakop Pashayan. Shallow shadows: Expectation estimation using low-depth random clifford circuits. *arXiv:2209.12924*, 2023.

[46] Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nature Physics*, 14(5):447–450, May 2018.

[47] Liming Zhao, Naixu Guo, Ming-Xing Luo, and Patrick Rebentrost. Provable learning of quantum states with graphical models. *arXiv:2309.09235*, 2023.

[48] Kosuke Mitarai, Masahiro Kitagawa, and Keisuke Fujii. Quantum analog-digital conversion. *Phys. Rev. A*, 99:012301, Jan 2019.

Supplementary Material

Appendix A: Robust nonlinear amplitude transformation

Theorem S13 (Robust amplitude encoding [29, 34]). *Given an (α, a, ϵ)-state-encoding U_ψ of an n-qubit state $|\psi\rangle = \sum_{j=1}^{N} \psi_j |j\rangle$, where $\{\psi_j\}$ are real and $\|\psi\|_2 = 1$, one can construct an $(\alpha, 2a+n+2, \epsilon)$-encoding of the diagonal matrix $A = \mathrm{diag}(\psi_1, \ldots, \psi_N)$ with $\mathcal{O}(n)$ circuit depth and $\mathcal{O}(1)$ queries to controlled-U and controlled-$U^\dagger$. One can also construct an $(\alpha^2, 3a+2n+2, 3\epsilon)$-encoding of diagonal matrix $A_{abs} = \mathrm{diag}(\psi_1^2, \ldots, \psi_N^2)$.*

Proof. The construction is the same as Ref. [29, 34] and our focus is on the error analysis. The (α, a, ϵ)-state-encoding U_ψ approximately prepares the state

$$U|0\rangle|0\rangle = \frac{1}{\alpha}|0\rangle|\psi\rangle + \sqrt{1-\alpha^2}|1\rangle|\mathrm{bad}\rangle, \tag{A.1}$$

where $|\mathrm{bad}\rangle$ is a quantum state we are not interested. By the diagonal amplitude block-encoding introduced in Ref. [29, 34], one can approximately construct a block-encoding of $A = \mathrm{diag}(\psi_1, \ldots, \psi_N)$. By direct computation, one can see it is an $(\alpha, 2a+n+2, \epsilon)$-encoding, where α is directly from the state-encoding, and the error can be obtained from the L_∞-norm. Let the exact block-encoded diagonal matrix be A'. Note that $\|A-A'\| = \max_j |\psi_j - \psi_j'| = \||\psi\rangle - |\psi'\rangle\|_\infty \leq \epsilon$. Block-encoding of A_{abs} can be constructed following Theorem 2 in Ref. [48] and Ref. [29, 34]. The error analysis follows $\max_j |\psi_j^2 - \psi_j'|^2 \leq \max_j |\psi_j^2 - (\psi_j+\epsilon)^2| \leq 3\epsilon$. Query complexity analysis follows the previous results. □

Appendix B: Matrix maximum entry norm

The standard block encoding assumption only tells us about the matrix norm of the block-encoded matrix, i.e., $\|A\| \leq 1$. With the following lemma, the condition also tells us that $\max_{i,j} |A_{i,j}| \leq 1$, i.e., the absolute value of each element is also bounded by 1.

Lemma S1. *For a complex matrix $A \in \mathbb{C}^{n\times m}$, $\max_{i,j} |A_{i,j}| \leq \|A\|$.*

Proof. Let $\sigma_{\max}(A)$ be the largest singular value of A. By definition, we have $\|A\| = \sigma_{\max}(A)$. Consider the singular value decomposition $A = U\Sigma V^\dagger$, where U and V are unitaries and Σ is a diagonal matrix. Let $\{f_i\}_i$ and $\{g_j\}_j$ be the basis of $\mathbb{C}^n$ and $\mathbb{C}^m$ respectively. Since U and V are unitaries, we have

$$\|U^\dagger f_i\| = \|V^\dagger g_j\| = 1. \tag{B.1}$$

Write $v = V^\dagger g_j$. We have

$$\begin{aligned}
\|A_{i,j}\| &= |\langle f_i, A g_j\rangle| = |\langle f_i, U\Sigma V^\dagger g_j\rangle| = |\langle U^\dagger f_i, \Sigma V^\dagger g_j\rangle| \leq \|U^\dagger f_i\| \|\Sigma V^\dagger g_j\| \\
&= \|\ \Sigma V^\dagger g_j\| = \left(\sum_k (\Sigma v)_k^2\right)^{\frac{1}{2}} = \left(\sum_k \left(\sum_j \Sigma_{kj} v_j\right)^2\right)^{\frac{1}{2}} = \left(\sum_k \Sigma_{kk}^2 v_k^2\right)^{\frac{1}{2}} \\
&\leq \left(\sum_k \sigma_{\max}^2 v_k^2\right)^{\frac{1}{2}} = \sigma_{\max}\|v\|^2 = \|A\|.
\end{aligned} \tag{B.2}$$

□

Appendix C: Normalized error bound

Lemma S2. *For two d-dimensional vectors $\psi = (\psi_1, \dots, \psi_d)$ and $\psi' = (\psi'_1, \dots, \psi'_d)$, if $|\psi_j - \psi'_j| \le \epsilon$ for each $j \in [d]$, we have*

$$\left\| \frac{1}{C}\psi - \frac{1}{C'}\psi' \right\|_2 \le \frac{2\sqrt{d}\epsilon}{C} + \sqrt{\frac{2\epsilon\sqrt{d}}{C}}, \tag{C.1}$$

where $C = \|\psi\|_2$ and $C' = \|\psi'\|_2$.

Proof. By direct computation, we have

$$\begin{aligned}
\left\| \frac{1}{C}\sum_{j\in\mathcal{S}} \psi_j |j\rangle - \frac{1}{C'}\sum_{j\in\mathcal{S}} \psi'_j |j\rangle \right\|_2 &= \frac{1}{CC'} \left\| C'\sum_{j\in\mathcal{S}} \psi_j|j\rangle - C\sum_{j\in\mathcal{S}} \psi'_j|j\rangle \right\|_2 \\
&= \frac{1}{CC'} \left\| C'\Big(\sum_{j\in\mathcal{S}} \psi_j|j\rangle - \sum_{j\in\mathcal{S}} \psi'_j|j\rangle\Big) + (C'-C)\sum_{j\in\mathcal{S}} \psi'_j|j\rangle \right\|_2 \\
&\le \frac{1}{CC'} \left(\left\| C'\Big(\sum_{j\in\mathcal{S}} \psi_j|j\rangle - \sum_{j\in\mathcal{S}} \psi'_j|j\rangle\Big) \right\|_2 + \left\| (C'-C)\sum_{j\in\mathcal{S}} \psi'_j|j\rangle \right\|_2 \right),
\end{aligned} \tag{C.2}$$

where the inequality comes from the triangle inequality. The first term can be easily bounded by $\sqrt{d}\epsilon/C$ since for each $j \in [d]$, we have $|\psi_j - \psi'_j| \le \epsilon$. Note that for nonnegative real numbers a and b, we have $|a-b| = |(\sqrt{a}-\sqrt{b})(\sqrt{a}+\sqrt{b})| = |\sqrt{a}-\sqrt{b}||\sqrt{a}+\sqrt{b}| \ge |\sqrt{a}-\sqrt{b}|^2$, hence $|\sqrt{a}-\sqrt{b}| \le \sqrt{|a-b|}$. The second term can be bounded with the following computation:

$$\begin{aligned}
\frac{1}{C}|C - C'| &\le \frac{\sqrt{|C^2 - C'^2|}}{C} \\
&\le \frac{\sqrt{|\sum_{j\in\mathcal{S}}(\psi_j^2 - \psi_j'^2)|}}{C} \\
&\le \frac{\sqrt{\sum_{j\in\mathcal{S}}|(\psi_j - \psi'_j)(\psi_j + \psi'_j)|}}{C} \\
&\le \frac{\sqrt{\epsilon\sum_{j\in\mathcal{S}}|\psi_j + \psi'_j|}}{C} \\
&\le \frac{\sqrt{\epsilon\sum_{j\in\mathcal{S}}(2|\psi_j| + \epsilon)}}{C} \\
&\le \frac{\sqrt{d\epsilon^2 + 2\epsilon\sum_{j\in\mathcal{S}}|\psi_j|}}{C} \\
&\le \frac{\sqrt{d}\epsilon}{C} + \frac{\sqrt{2\epsilon\sum_{j\in\mathcal{S}}|\psi_j|}}{C} \\
&\le \frac{\sqrt{d}\epsilon}{C} + \sqrt{\frac{2\epsilon\sqrt{d}}{C}},
\end{aligned} \tag{C.3}$$

where the last inequality is from the inequality between L_1 and L_2 norm. Combining these two terms together, we achieve our final result. □

Lemma S3. *For two d-dimensional vectors* $\psi = (\psi_1, \ldots, \psi_d)$ *and* $\psi' = (\psi'_1, \ldots, \psi'_d)$*, if* $|\psi_j - \psi'_j| \le \epsilon$ *for each* $j \in [d]$*, we have*

$$\Big\|\frac{1}{C}\psi - \frac{1}{C'}\psi'\Big\|_\infty \le \frac{(\sqrt{d}+1)\epsilon}{C} + \sqrt{\frac{2\epsilon\sqrt{d}}{C}}, \tag{C.4}$$

where $C = \|\psi\|_2$ *and* $C' = \|\psi'\|_2$*.*

Proof. Note that the L_∞ distance can be written as

$$\Big\|\frac{1}{C}\psi - \frac{1}{C'}\psi'\Big\|_\infty = \max_{j\in[d]}\Big|\frac{\psi_j}{C} - \frac{\psi'_j}{C'}\Big|. \tag{C.5}$$

We consider each element individually as

$$\Big|\frac{\psi_j}{C} - \frac{\psi'_j}{C'}\Big| = \frac{1}{CC'}|C'\psi_j - C\psi'_j|. \tag{C.6}$$

Having $\max_{j\in[d]}|\psi_j - \psi'_j| \le \epsilon$, we can write $\psi_j = \psi'_j + \Delta_j$ where $|\Delta_j| \le \epsilon$. Substituting ψ_j in $|C'\psi_j - C\psi'_j|$ we have

$$\begin{aligned}
|C'\psi_j - C\psi'_j| &= |C'\psi'_j + C'\Delta_j - C\psi'_j| &\text{(C.7)}\\
&= |(C'-C)\psi'_j + C'\Delta_j| &\text{(C.8)}\\
&\le |(C'-C)\psi'_j| + C'\epsilon. &\text{(C.9)}
\end{aligned}$$

Then we can write

$$\begin{aligned}
\max_{j\in[d]}\Big|\frac{\psi_j}{C} - \frac{\psi'_j}{C'}\Big| &= \max_{j\in[d]}\frac{1}{CC'}|C'\psi_j - C\psi'_j| &\text{(C.10)}\\
&\le \frac{C'\epsilon + \max_{j\in[d]}|(C'-C)\psi'_j|}{CC'} &\text{(C.11)}\\
&\le \frac{\epsilon}{C} + \frac{|C'-C|C'}{CC'} &\text{(C.12)}\\
&= \frac{\epsilon}{C} + \frac{|C'-C|}{C} &\text{(C.13)}\\
&\le \frac{(\sqrt{d}+1)\epsilon}{C} + \sqrt{\frac{2\epsilon\sqrt{d}}{C}}. &\text{(C.14)}
\end{aligned}$$

The bound of $\frac{|C'-C|}{C}$ directly follows from the proof of Lemma S2. ☐

Lemma S4. *For two d-dimensional vectors* $\psi = (\psi_1, \ldots, \psi_d)$ *and* $\psi' = (\psi'_1, \ldots, \psi'_d)$*, if* $|\psi_j - \psi'_j| \le \epsilon$ *and* $\psi_j, \psi'_j \le \Gamma \in \mathcal{O}(1)$ *for each* $j \in [d]$*, we have*

$$\Big\|\frac{1}{C}\psi - \frac{1}{C'}\psi'\Big\|_\infty \le \frac{\epsilon}{C} + \frac{\Gamma\sqrt{d}\epsilon}{CC'} + \frac{\Gamma}{C'}\sqrt{\frac{2\epsilon\sqrt{d}}{C}}, \tag{C.15}$$

where $C = \|\psi\|_2$ *and* $C' = \|\psi'\|_2$*.*

Proof. Note that the L_∞ distance can be written as

$$\Big\|\frac{1}{C}\psi - \frac{1}{C'}\psi'\Big\|_\infty = \max_{j\in[d]}\Big|\frac{\psi_j}{C} - \frac{\psi'_j}{C'}\Big|. \tag{C.16}$$

We consider each element individually as

$$\left|\frac{\psi_j}{C}-\frac{\psi_j'}{C'}\right|=\frac{1}{CC'}|C'\psi_j-C\psi_j'|. \tag{C.17}$$

Having $\max_{j\in[d]}|\psi_j-\psi_j'|\leq\epsilon$, we can write $\psi_j=\psi_j'+\Delta_j$ where $|\Delta_j|\leq\epsilon$. Substituting ψ_j in $|C'\psi_j-C\psi_j'|$ we have

$$\begin{aligned}|C'\psi_j-C\psi_j'| &= |C'\psi_j'+C'\Delta_j-C\psi_j'| && \text{(C.18)}\\ &= |(C'-C)\psi_j'+C'\Delta_j| && \text{(C.19)}\\ &\leq |(C'-C)\psi_j'|+C'\epsilon. && \text{(C.20)}\end{aligned}$$

Then we can write

$$\begin{aligned}\max_{j\in[d]}\left|\frac{\psi_j}{C}-\frac{\psi_j'}{C'}\right| &= \max_{j\in[d]}\frac{1}{CC'}|C'\psi_j-C\psi_j'| && \text{(C.21)}\\ &\leq \frac{C'\epsilon+\max_{j\in[d]}|(C'-C)\psi_j'|}{CC'} && \text{(C.22)}\\ &\leq \frac{\epsilon}{C}+\frac{\Gamma|C'-C|}{CC'} && \text{(C.23)}\\ &= \frac{\epsilon}{C}+\frac{\Gamma\sqrt{d}\epsilon}{CC'}+\frac{\Gamma}{C'}\sqrt{\frac{2\epsilon\sqrt{d}}{C}}. && \text{(C.24)}\end{aligned}$$

□

Appendix D: Polynomial approximation of exponential function

Lemma S5. *For $x\in[-1,1]$, the function $f(x):=e^x$ can be approximated with error bound ϵ with an $\mathcal{O}(\log(1/\epsilon))$-degree polynomial function.*

Proof. Consider the Taylor expansion of $f(x)=\sum_{j=0}^{\infty}\frac{x^j}{j!}$. Let $f_k(x):=\sum_{j=0}^{k}\frac{x^j}{j!}$. To achieve $|f_k(x)-f(x)|\leq\epsilon$ for $|x|\leq1$,

$$\begin{aligned}|f_k(x)-f(x)| &= \left|\sum_{j=k+1}^{\infty}\frac{x^j}{j!}\right| \leq \left|\sum_{j=k+1}^{\infty}\frac{1}{j!}\right| = \left|\sum_{j=1}^{\infty}\frac{1}{(j+k)!}\right|\\ \text{(Assume } k>2\text{)}\quad &\leq \frac{1}{k!}\left|\sum_{j=1}^{\infty}\frac{1}{2^j}\right| \leq \frac{1}{k!}\leq\epsilon.\end{aligned}$$

It suffices to set $k=\mathcal{O}(\log(\frac{1}{\epsilon}))$, which can be seen by the Stirling's approximation. □

Appendix E: Quantum single-layer transformer

In this section, we combine the previous theorems together to obtain the final main theorem.

Proof of Theorem 12. As shown in Fig. 1, a single-layer transformer contains the self-attention, residual connection and layer normalization, and the feedforward network. In Theorem 8, 10 and 11, we have considered each block in detail. Here, we complete the analysis for the second residual connection after the feedforward network.

As described in Problem 4 and Theorem 11, we have access to (α, a, ϵ)-state-encoding of $|\psi\rangle$ and $(2\alpha\alpha_m^2, \mathcal{O}(a+n+a_m), \mathcal{O}((\sqrt{d}\alpha_m^2\ell\sqrt{\alpha_m\epsilon+\epsilon_m})^{\frac{1}{2}}))$-encoding of matrix B such that $B_{\star 1} = (\tilde{\phi}_1, \cdots, \tilde{\phi}_d)$, where $\tilde{\phi} \coloneqq M_2 \cdot \mathrm{GELU}(M_1 \cdot \psi)$. Note that here, the dimension of vector ψ is d and $N_2 = d$. The target is to construct a state encoding of

$$\sum_{k=1}^{d} \mathrm{LN}(\tilde{\phi}_k + \psi_k)|k\rangle. \tag{E.1}$$

The state encoding can be understood as a block encoding of a matrix whose first column corresponds to the quantum state. By Lemma 1 and taking the self-adjoint, one can construct a $(2\alpha\alpha_m^2 + \alpha, \mathcal{O}(a+n+a_m), \mathcal{O}((\sqrt{d}\alpha_m^2\ell\sqrt{\alpha_m\epsilon+\epsilon_m})^{\frac{1}{2}}))$-encoding of a matrix whose first row is $(\psi_1 + \tilde{\phi}_1, \ldots, \psi_d + \tilde{\phi}_d)$.

The following steps are the same as in Theorem 10. One can construct an $(\mathcal{O}((\sqrt{d}+1)\alpha\alpha_m^2/\varsigma', \mathcal{O}(a+n+a_m), \mathcal{O}((\sqrt{d}\alpha_m^2\ell\sqrt{\alpha_m\epsilon+\epsilon_m})^{\frac{1}{2}}/\varsigma'))$-state-encoding of the state

$$\sum_{k=1}^{d} \mathrm{LN}(\tilde{\phi}_k + \psi_k)|k\rangle, \tag{E.2}$$

where $\varsigma' \coloneqq \sqrt{\sum_{k=1}^{d}(\tilde{\phi}_k + \psi_k - \bar{\psi})^2}$ and $\bar{\psi} \coloneqq \frac{1}{d}\sum_{k=1}^{d}(\tilde{\phi}_k + \psi_k)$.

Combining the results with Theorem 8, 10, and 11, one can achieve the final result. $\square$