Quantum Transformer for Molecular Learning: Multi-Configuration Ground-State Energy Prediction

Yuichi Kamata, Quoc Hoan Tran*, Yasuhiro Endo, Hirotaka Oshima

Quantum Laboratory, Fujitsu Research, Fujitsu Limited,
Kawasaki, Kanagawa 211-8588, Japan
{kamata.yuichi, tran.quochoan, endo.yasuhiro, oshimah}@fujitsu.com

Abstract

The Transformer model, renowned for its powerful attention mechanism, has achieved state-of-the-art performance in various artificial intelligence tasks but faces challenges with quantum data. With a growing focus on leveraging quantum machine learning for quantum data, particularly in quantum chemistry, we propose the Molecular Quantum Transformer (MQT) for modeling interactions in molecular quantum systems. By utilizing quantum circuits to implement the attention mechanism on the molecular configurations, MQT can efficiently calculate ground-state energies for all configurations. Numerical demonstrations show that in calculating ground-state energies for H_2, LiH, BeH_2, and H_4, MQT outperforms the classical Transformer, highlighting the promise of quantum effects in Transformer structures. Furthermore, its pretraining capability on diverse molecular data facilitates the efficient learning of new molecules, extending its applicability to complex molecular systems with minimal additional effort. Our method offers an alternative to existing quantum algorithms for estimating ground-state energies, opening new avenues in quantum chemistry and materials science.

Code — https://github.com/FujitsuResearch/molecular_quantum_transformer

Introduction

The marriage of quantum computing and machine learning (ML) has given rise to the field of quantum machine learning (QML) (Dunjko, Taylor, and Briegel 2016; Biamonte et al. 2017; Havlíček et al. 2019), which aims to leverage quantum computers to tackle problems that are infeasible for classical computers. In recent years, there has been a growing recognition within the research community that classical data, such as text and images, do not inherently require quantum effects for processing. There is a shift towards employing QML methods that exploit quantum effects on data originating from quantum systems. One promising yet underexplored area is the application of QML methods in quantum chemistry, where QML holds the potential to determine molecular and material properties more efficiently than traditional quantum computational chemistry algorithms.

*Corresponding author.

A central focus in quantum computational chemistry is the electronic structure problem, which involves calculating the electronic Hamiltonian's ground-state energy while assuming the molecule's nuclei remain fixed. Among the most studied algorithms for this problem are the Variational Quantum Eigensolver (VQE) (Peruzzo et al. 2014; Kandala et al. 2017) for Noisy Intermediate-Scale Quantum (NISQ) devices and Quantum Phase Estimation (QPE) (Kitaev 1995; Nielsen and Chuang 2010) for fault-tolerant quantum computers. While these approaches show promise, they face practical limitations. For instance, estimating the ground-state energy of FeMoCo, a well-known and practical benchmark in quantum chemistry, would require a fault-tolerant quantum computer with millions of physical qubits operating for nearly four days using QPE (Lee et al. 2021). In contrast, VQE can utilize fewer, noisier qubits, but its scalability is hindered by the extensive measurement demands during optimization, particularly for large-scale molecular systems (Tilly et al. 2022; Gonthier et al. 2022).

Practical Motivation. In practical quantum chemistry, estimating the ground-state energy for a single molecular configuration is often insufficient. Determining dynamic and structural properties, such as reaction barriers and optimal geometries, necessitates exploring multiple configurations. This requires knowledge of a family of ground states for a series of Hamiltonians parameterized by variables like nuclear coordinates and electron-nucleus distances. Consequently, one must compute numerous ground states with corresponding energies over a potential energy surface. However, a straightforward approach, such as independently running QPE or VQE for each configuration, incurs prohibitive computational costs. To address this, meta-based approaches leverage classical ML to optimize circuit parameters across multiple configurations simultaneously, as seen in Meta-VQE (Cervera-Lierta, Kottmann, and Aspuru-Guzik 2021). Another intriguing approach is the use of generative QML to produce ground states learning from quantum data (Ceroni et al. 2023). Nevertheless, these methods lack adaptability to varying molecule types and Hamiltonian forms and fail to capture correlations effectively.

Contributions. We propose the Molecular Quantum Transformer (MQT), a Quantum Transformer (QT) model designed to calculate molecular ground-state energies [Fig. 1(a)]. The MQT leverages attention mechanisms im-

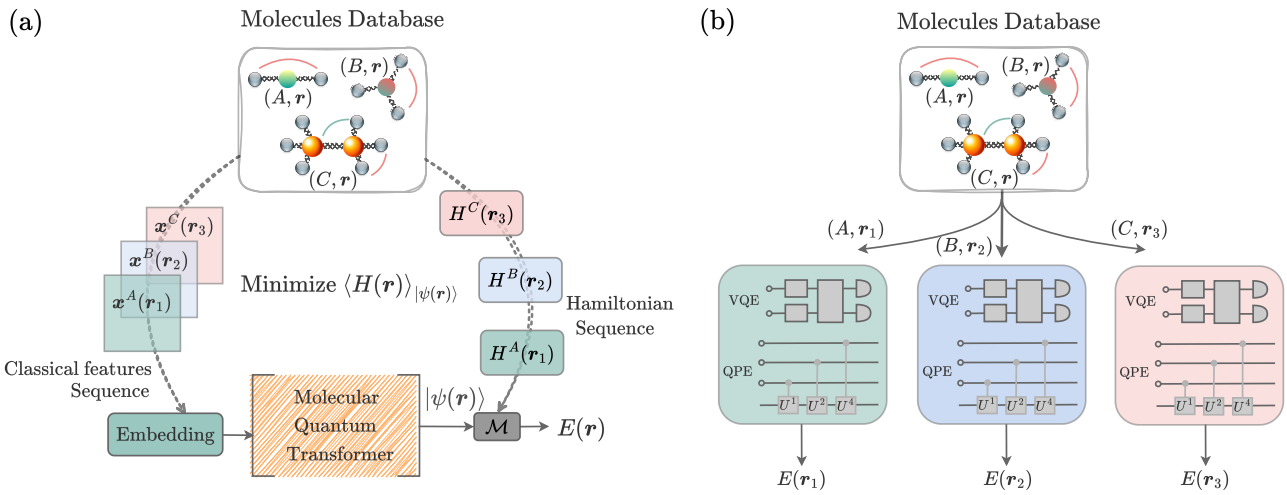

Figure 1: (a) Overview of the Molecular Quantum Transformer (MQT) model for the ground-state energy calculation across various molecules and their configurations, and the comparison with traditional methods. (a) For each molecule A, B, C,... and its associated configuration $\boldsymbol{r}_1, \boldsymbol{r}_2, \boldsymbol{r}_3, \ldots$, the MQT receives a corresponding classical features sequence $\boldsymbol{x}^A(\boldsymbol{r}_1), \boldsymbol{x}^B(\boldsymbol{r}_2), \boldsymbol{x}^C(\boldsymbol{r}_3), \ldots$ through an embedding process. Leveraging a quantum attention mechanism, the MQT captures the complex interactions and correlations within the molecular system. The output of the MQT is a sequence of quantum states $|\psi^A(\boldsymbol{r}_1)\rangle, |\psi^B(\boldsymbol{r}_2)\rangle, |\psi^C(\boldsymbol{r}_3)\rangle, \ldots$, which reflects these correlations in a variational representation of the estimated ground states for $(A, \boldsymbol{r}_1), (B, \boldsymbol{r}_2), (C, \boldsymbol{r}_3), \ldots$, respectively. The Hamiltonian $H^A(\boldsymbol{r}_1), H^B(\boldsymbol{r}_2), H^C(\boldsymbol{r}_3), \ldots$, derived from quantum mechanics are transformed into measurable operators to be measured on $|\psi^A(\boldsymbol{r}_1)\rangle, |\psi^B(\boldsymbol{r}_2)\rangle, |\psi^C(\boldsymbol{r}_3)\rangle, \ldots$. During training, the optimization process adjusts the variational parameters in both the MQT and the embedding process to minimize the expectation value $\langle H(\boldsymbol{r})\rangle_{|\psi(\boldsymbol{r})\rangle}$ across various molecules and a range of $\boldsymbol{r}$ values. In the evaluation phase, given a molecule, the MQT can provide an estimator of ground-state energy $E(\boldsymbol{r})$ for any configuration $\boldsymbol{r}$. (b) In contrast, traditional methods such as VQE or QPE require an independent and computationally expensive solver for each molecule and configuration $\boldsymbol{r}$.

plemented through variational quantum circuits (VQCs), enabling efficient modeling of complex interactions and correlations within molecular quantum systems. By training on random molecular configurations at each iteration, the MQT captures these interactions, allowing it to generalize and obtain ground-state energies across diverse configurations. The MQT enables the simultaneous learning of ground-state energies across a range of bond lengths within a single model, providing greater resource efficiency compared to independently executing QPE or VQE for each molecular configuration [Fig. 1(b)]. We compare MQT to a classical Transformer model under identical conditions of model dimensionality, demonstrating the superior performance of MQT over its classical counterpart. Furthermore, the MQT can be utilized to learn new molecules efficiently through pretraining with diverse molecular data. This capability enables the application of MQT in complex molecules with minimal effort, leveraging well-established molecular data.

Background

Variational Quantum Circuits (VQCs)

VQCs represent a pivotal advancement in QML, bridging classical and quantum computing paradigms to address complex ML tasks. At their core, quantum circuits consist of sequences of quantum gates applied to qubits, generating entangled superposition states that enable quantum systems to perform computations unattainable by classical means. VQCs extend this framework by incorporating tunable parameters, allowing optimization for specific objectives. Here, VQCs are constructed from a combination of fixed gates, which encode classical input data $\boldsymbol{x}$ (e.g., features for ML tasks) via rotation angles, and variable gates parameterized by $\boldsymbol{\theta}$. The measurement process on qubits captures objective function details, such as the discrepancy between the predicted output and the target. The function $\Psi(\boldsymbol{x}, \boldsymbol{\theta}) = U(\boldsymbol{x}, \boldsymbol{\theta})\,|0\rangle$ maps these parameters to a quantum state through a series of unitary operations applied to an initial $|0\rangle$ state, defining a unique circuit architecture (Havlíček et al. 2019). This hybrid approach leverages classical optimization techniques to tune parameters, rendering VQCs fully differentiable and compatible with standard deep learning algorithms (Mitarai et al. 2018).

Quantum Transformer

The Transformer model (Vaswani et al. 2017) has been recognized as a remarkable advancement in artificial intelligence. Its key power lies in its "attention mechanism", which discerns the relative importance of different parts of its input and the connection strengths between them. Despite these successes, the current implementation of the Transformer faces several challenges, including high computational costs, substantial memory requirements, the necessity for large datasets, and a vast number of training parameters.

These limitations have prompted researchers to explore improved Transformer designs.

Efforts have been made to develop quantum versions of the Transformer model. A significant advancement in quantum neural networks (QNNs) involves integrating the self-attention mechanism by encoding query and key vectors as quantum states using VQCs. This adaptation enables quantum analogs of the classical self-attention framework, though various approaches differ in their implementation. One straightforward extension replaces the classical inner-product self-attention with the overlap of quantum states (Sipio et al. 2021), but this method struggles to scale effectively for capturing correlations in large datasets due to its computational complexity. An alternative method employs Gaussian projections of query and key quantum states, enhancing scalability while retaining essential features (Li, Zhao, and Wang 2024). Other quantum self-attention variants include quantum vision Transformers (Xu et al. 2025) as an end-to-end approach that leverages analog encoding with quantum random access memory (qRAM) (Giovannetti, Lloyd, and Maccone 2008), and hybrid classical-quantum methods (Cherrat et al. 2024; Smaldone et al. 2025) to reduce the time complexity of computing query-key dot products. Additionally, some proposals diverge from inner-product-based attention entirely, opting instead to mix tokens directly in Hilbert space to model correlations without explicitly calculating query-key dot products (Zheng, Gao, and Miao 2023; Evans et al. 2025). Despite innovative adaptations, these quantum self-attention implementations have demonstrated limited performance on tasks involving classical data, such as text and image classification, highlighting challenges in translating the advantages to practical applications.

The Electronic Structure Problem

We consider a complex quantum many-body system in which multiple electrons and nuclei interact with each other through Coulomb interactions. Since the nuclei are thousands of times heavier than the electrons, they hardly move under the attraction from the electrons and can be regarded as fixed at coordinates $\boldsymbol{R}_m$ (Born-Oppenheimer approximation). The wave function where N electrons move around M (fixed) nuclei (with atomic numbers $Z_1, \ldots, Z_M$) can be written as $\psi(\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_N)$, and the Hamiltonian H for the electrons can be expressed in Hartree atomic units as follows in the following simplified form (first quantization):

$$H(\boldsymbol{R}) = -\sum_i \frac{\nabla_i^2}{2} + \sum_{i<j} \frac{1}{|\boldsymbol{r}_i - \boldsymbol{r}_j|} - \sum_{i,m} \frac{Z_m}{|\boldsymbol{r}_i - \boldsymbol{R}_m|}. \tag{1}$$

Solving the Schrödinger equation $H\psi = E\psi$ to determine the electronic state of a quantum many-body system is known as the molecular electronic problem. This problem is crucial for materials design and drug discovery but remains a central challenge in quantum chemistry.

Classical methods such as density functional theory (DFT) (Kohn, Becke, and Parr 1996), quantum monte carlo (QMC) (Hammond, Lester, and Reynolds 1994), and density matrix renormalization group (DMRG) (White 1992) have significantly advanced the approximation of molecular electronic structure, each balancing accuracy and computational cost in different ways. In parallel, quantum computing offers a promising alternative by natively handling quantum states, with algorithms like QPE (Kitaev 1995; Nielsen and Chuang 2010) and VQE (Peruzzo et al. 2014; Kandala et al. 2017) emerging as leading approaches. While QPE provides exponential precision, it requires fault-tolerant hardware. In contrast, VQE is more practical for near-term devices but relies heavily on ansatz design and optimization strategies.

To be implemented on quantum computers, the Hamiltonian in Eq. (1) is transformed as (the second quantization):

$$H(\boldsymbol{R}) = \sum_{p,q} h_{pq}(\boldsymbol{R}) c_p^\dagger c_q + \frac{1}{2} \sum_{p,q,r,s} h_{pqrs}(\boldsymbol{R}) c_p^\dagger c_q^\dagger c_r c_s. \tag{2}$$

This operation is equivalent to the basis function expansion of the wave function. Here, $c_p^\dagger$ and c_p are the fermionic creation and annihilation operators acting on the p-th orbital. Therefore, $c_p^\dagger c_q$ is the operator to transit the state on the q-th orbital to the p-th orbital, and $c_p^\dagger c_q^\dagger c_r c_s$ is the operator to transit a pair of state $(r, s) \rightarrow (p, q)$. The one and two-electron integrals $h_{pq}(\boldsymbol{R})$ and $h_{pqrs}(\boldsymbol{R})$ in the molecular orbital basis $\phi_p(\boldsymbol{r})$ (yielded from the Hartree-Fock optimization procedure) depend implicitly on $\boldsymbol{R}$, which is the general nuclear coordinate associated with the fixed nuclear configuration in 3-dimensional space. These integral coefficients are calculated on the classical computer as follows:

$$h_{pq}(\boldsymbol{R}) = \int d\boldsymbol{r}\, \phi_p^*(\boldsymbol{r}) \left(-\frac{\nabla^2}{2} - \sum_m \frac{Z_m}{|\boldsymbol{r} - \boldsymbol{R}_m|} \right) \phi_q(\boldsymbol{r}), \tag{3}$$

$$h_{pqrs}(\boldsymbol{R}) = \int d\boldsymbol{r}_1 d\boldsymbol{r}_2\, \frac{\phi_p^*(\boldsymbol{r}_1)\phi_q^*(\boldsymbol{r}_2)\phi_r(\boldsymbol{r}_2)\phi_s(\boldsymbol{r}_1)}{|\boldsymbol{r}_1 - \boldsymbol{r}_2|}. \tag{4}$$

The next step in implementing the Hamiltonian form in Eq. (2) using quantum circuits is to map the fermionic creation and annihilation operators to Pauli operators using methods such as the Jordan-Wigner or Bravyi-Kitaev transforms (Seeley, Richard, and Love 2012).

Proposal: Molecular Quantum Transformer

We propose the Molecular Quantum Transformer (MQT) model, which replaces the VQE ansatz with a QT structure to determine the electron wave function that minimizes the Hamiltonian's energy. The fundamental concept of the attention mechanism involves capturing correlations among all tokens simultaneously. In the context of the MQT, these tokens represent features derived from the positions and distances between atomic nuclei. As configurations change, the model adapts its features, such as electron configurations, based on these relationships. This adaptation is generalized through training the model across a variety of conditions. In contrast, running QPE or VQE demands independent resources for each molecular configuration, necessitating multiple models. The MQT, however, can learn the ground-state energies for various molecules and various configurations simultaneously within a single model.

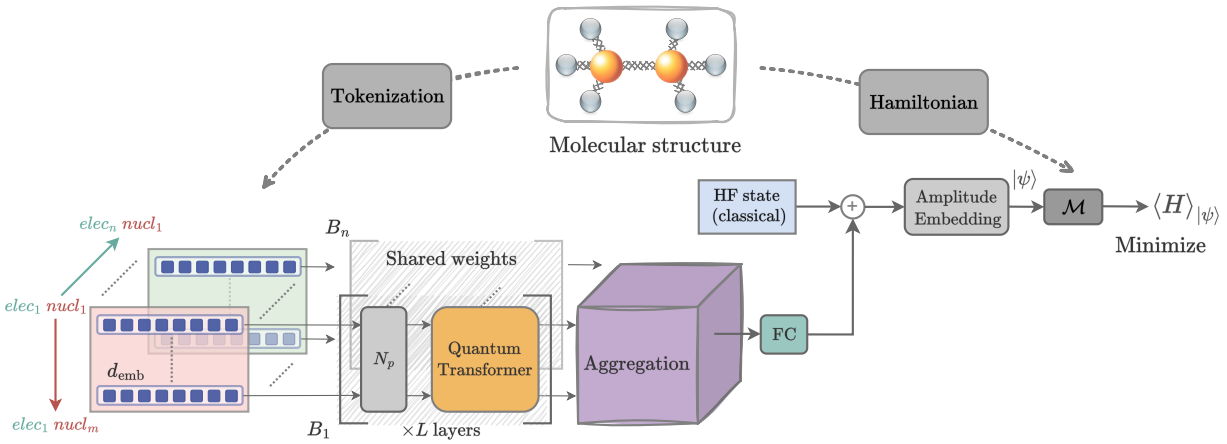

Figure 2: Structure of Molecular Quantum Transformer (MQT).

The structure of our MQT model is illustrated in Fig. 2. Given a molecular structure defined by atomic symbols and the nuclear coordinates of its constituent atoms, we construct the molecular Hamiltonian H in a qubit-based representation using n_q qubits as a preprocessing step. Here, we utilize the PennyLane molecules dataset (Azad 2023) and its built-in functions (Bergholm and et al. 2022) to generate the molecular Hamiltonians. The MQT begins with a tokenization module that creates input tokens $elec_i$-$nucl_j$ ($i = 1, \ldots, n$; $j = 1, \ldots, m$), representing the electronic state as an $n \times m$ two-dimensional array of d_{emb}-dimensional features. Here, d_{emb} is the embedding dimension, n and m denote the number of electrons and nuclei, respectively, resulting in an $n \times m \times d_{\text{emb}}$ feature matrix. For each electron index i, the m tokens $elec_i$-$nucl_j$ ($j = 1, \ldots, m$) are processed by a block B_i, which consists of L layers. Each layer includes an amplification module N_p and a QT module. The amplification module N_p scales the feature values by the proton number N_p to reflect the differences among nuclear species before passing them to the QT module. The QT module contains trainable parameters, which are shared across all blocks $B_1, \ldots, B_n$. The outputs of all QT modules are combined into a single feature representation via an aggregation module. A fully connected (FC) module then maps this aggregated feature into a vector with the same dimensionality as the n_q-qubit state vector. This transformed vector is subsequently added to the Hartree-Fock (HF) state vector, and the resulting representation is used to generate the final electronic state through an amplitude embedding module. The expectation value of the Hamiltonian H is computed from this amplitude-embedded state via a measurement process. This expectation value is minimized via the trainable parameters in the QT, aggregation, and FC modules.

Tokenization Figure 3(a) depicts the tokenization module. Each d_{emb}-dimensional feature input vector $\boldsymbol{x}$ in the token matrix is derived from a vector concatenating the relative positions of electrons to each nucleus ($p_{\text{en}}^{\text{in}}$) and their initial distances ($r_{\text{en}}^{\text{in}}$), processed through a FC layer with trainable weights W_{en} (von Glehn, Spencer, and Pfau 2023). The computational steps to obtain $p_{\text{en}}^{\text{in}}$ and $r_{\text{en}}^{\text{in}}$ are outlined as follows:

- *Input*: The input consists of the positions of individual atoms (nuclear coordinates) and electron identifiers (atomic orbital assignments). For example, in BeH_2, the 3D coordinates of the Hydrogen (H) atoms are symmetrically positioned along the z-axis at $(0, 0, -2.5)$ and $(0, 0, 2.5)$, with the Beryllium (Be) atom at the origin $(0, 0, 0)$. The nucleus position is an $m \times 3$ matrix with $m = 3$, where the first and last rows correspond to the two H atoms. Electron identifiers are assigned as follows: [1] (1s) for the first H, [1] (1s), [2] (1s), [3] (2s), and [4] (2s) for Be, and [1] (1s) for the second H.
- *Embedding (1)*: Each electron identifier is converted into a one-hot vector of dimension equal to the maximum identifier (4 in the BeH_2 example). Thus, all n electron identifiers are represented as an $n \times 4$ binary matrix. This matrix is then multiplied by a 4×3 trainable weight matrix to produce an $n \times 3$ matrix of electron relative positions. These positions are relative to their respective atoms, and the trainable weights ensure that electron positions adapt to changes in the molecular structure.
- *Adder (2)*: The absolute position of each electron is computed by adding its relative position to the position of its associated atom, resulting in an $n \times 3$ matrix.
- *Subtraction (3)*: The relative positions between all electrons and all nuclei are calculated. For each nucleus, an $n \times 3$ matrix is obtained by subtracting the absolute positions of all electrons from that nucleus's position, yielding an output matrix $p_{\text{en}}^{\text{in}}$ of shape $n \times m \times 3$.
- *Concatenation (4)*: The relative distance matrix $r_{\text{en}}^{\text{in}}$, with shape $n \times m \times 1$, is derived from $p_{\text{en}}^{\text{in}}$ where each element in $r_{\text{en}}^{\text{in}}$ is the Euclidean norm of each position vector. Concatenating $p_{\text{en}}^{\text{in}}$ with $r_{\text{en}}^{\text{in}}$ produces an $n \times m \times 4$ matrix.

Quantum Transformer (QT) The architecture of the QT module is depicted in the lower panel of Fig. 3(a), consisting of L layers. In each layer, the input features derived from the classical representation of the molecule are processed through the arccos, tanh, and amplification N_p modules. For

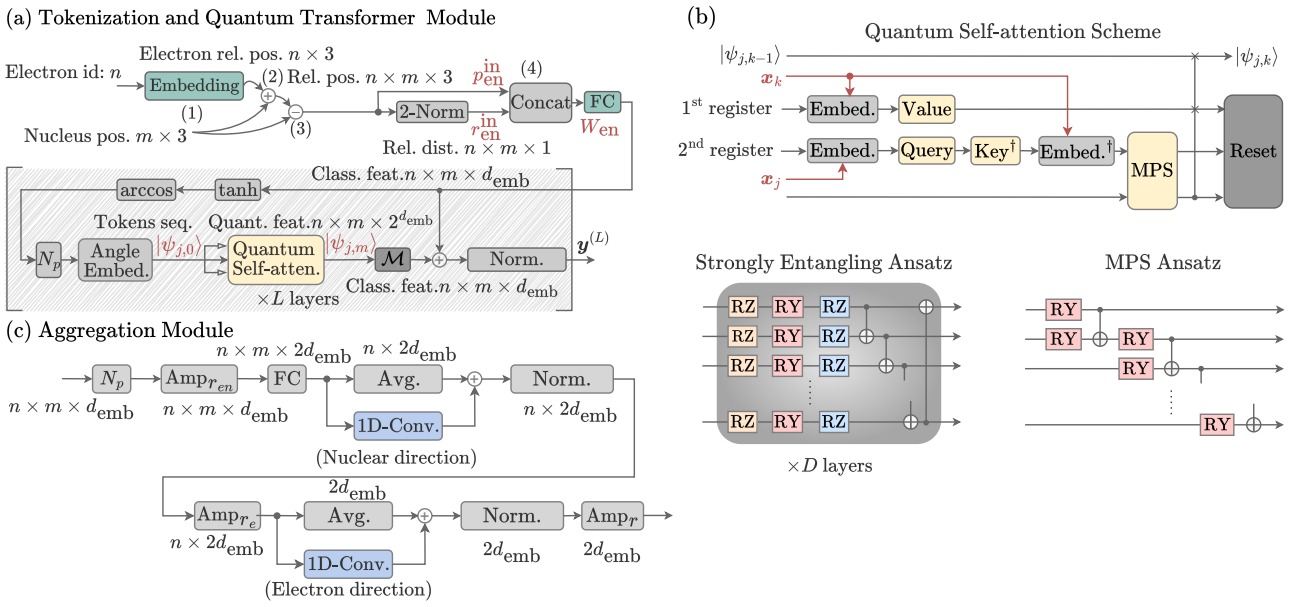

Figure 3: (a) Tokenization and Quantum Transformer module in the MQT. (b) Quantum self-attention update ($|\psi_{j,k-1}\rangle \to |\psi_{j,k}\rangle$) with the StrEnt and MPS ansatzes are depicted in the lower panel with the trainable rotation gates. (c) The aggregation module transforming the Quantum Transformer output matrix $\boldsymbol{y}^{(L)}$ ($n \times m \times d_{\text{emb}}$) into a $2d_{\text{emb}}$-dimensional feature vector.

each electron, this generates an $m \times d_{\text{emb}}$ features matrix corresponding to m tokens $\boldsymbol{x}_j$. Each d_{emb}-dimensional feature vector of token $\boldsymbol{x}_j$ is embedded into a quantum state $|\psi_{j,0}\rangle$ via an angle embedding layer using d_{emb} qubits. Query, Key, and Value transformations, implemented with quantum ansatzes, are applied to update the states $|\psi_{j,k-1}\rangle \to |\psi_{j,k}\rangle$ ($k = 1, \ldots, m$) through a quantum self-attention mechanism. The final state $|\psi_{j,m}\rangle$ is measured in the computational basis and converted back into classical features of dimension $m \times d_{\text{emb}}$, yielding an $n \times m \times d_{\text{emb}}$ feature matrix for n blocks of n electrons. These features are added to the input features via a residual connection and normalization, producing input for the next layer.

The detailed of the update $|\psi_{j,k-1}\rangle \to |\psi_{j,k}\rangle$ is shown in Fig. 3(b). For an input token $\boldsymbol{x}_j$, angle embedding is performed using a R_Y rotation gate (Embed.) on a second auxiliary token register. This is followed by a Query transformation and the adjoint of Key transformation, both constructed with a single layer of strongly entangling (StrEnt) ansatz. The adjoint matrix of the angle embedding (Embed.†) for token $\boldsymbol{x}_k$ is applied to compute the Hadamard product of the Query and Key, which resembles the single-head attention. This product is then transformed into a 1-qubit attention representation using a 2-bond matrix product state (MPS) ansatz implemented with trainable R_Y and an additional ancilla qubit. Next, a Value transformation for $\boldsymbol{x}_k$ is applied using the first auxiliary token register, involving the angle embedding and a six-layer StrEnt ansatz. Finally, a controlled SWAP gate updates the features based on the attention.

Aggregation The outputs of all QT modules are summarized into a feature matrix $\boldsymbol{y}^{(L)}$ of dimension $n \times m \times d_{\text{emb}}$, which is then transformed into a $2d_{\text{emb}}$-dimensional feature vector via an aggregation module [Fig. 3(c)]. Initially, the amplification modules N_p and $\text{Amp}_{r_{\text{en}}}$ are applied to $\boldsymbol{y}^{(L)}$ without altering its dimension. Each element y_{ijk} ($1 \le i \le n, 1 \le j \le m, 1 \le k \le d_{\text{emb}}$) in $\boldsymbol{y}^{(L)}$ is transformed as:

$$y_{ijk} \to e^{-r_{\text{en}}(i,j,0)} N_p y_{ijk}, \tag{5}$$

where N_p is the proton number, r_{en} is the $n \times m \times 1$ electron-nucleus relative distance matrix with elements $r_{\text{en}}(i,j,0)$ ($1 \le i \le n, 1 \le j \le m$). The relative distance r_{en} is updated from the initial distance $r_{\text{en}}^{\text{in}}$ using the attention scheme through the following inversion:

$$[p_{\text{en}}^{\text{out}}, r_{\text{en}}^{\text{out}}] = \left(W_{\text{en}}^\top \times W_{\text{en}}\right)^{-1} \times W_{\text{en}}^\top \times \boldsymbol{y}^{(L)}, \tag{6}$$

$$r_{\text{en}} = \text{std}(r_{\text{en}}^{\text{in}}) \times \frac{r_{\text{en}}^{\text{out}} - \text{mean}(r_{\text{en}}^{\text{out}})}{\text{std}(r_{\text{en}}^{\text{out}})} + \text{mean}(r_{\text{en}}^{\text{in}}), \tag{7}$$

where $\text{mean}(A)$ and $\text{std}(A)$ denote for the scalar mean and standard deviation of all elements in matrix A, and $r_{\text{en}}^{\text{in}}$ and $r_{\text{en}}^{\text{out}}$ share the $n \times m \times 1$ dimension. Here, $\times, +, -$ are broadcast operators applied element-wise to $r_{\text{en}}^{\text{out}}$.

The output of $\text{Amp}_{r_{\text{en}}}$ with dimension $n \times m \times d_{\text{emb}}$ is processed by an FC module with trainable weights to obtain features of dimension $n \times m \times (2d_{\text{emb}})$. These features are then separately processed along the nuclear dimension direction (second axis) by an averaging (Avg.) module and a repeated one-dimensional convolutional (1D-Conv.) module (kernel size 2, stride 2) until m becomes 1. The separated processed features are summed to produce an $n \times (2d_{\text{emb}})$ feature matrix. This matrix is normalized and adjusted by the amplification module $\text{Amp}_{r_{\text{e}}}$, where r_{e} (dimension n) is the

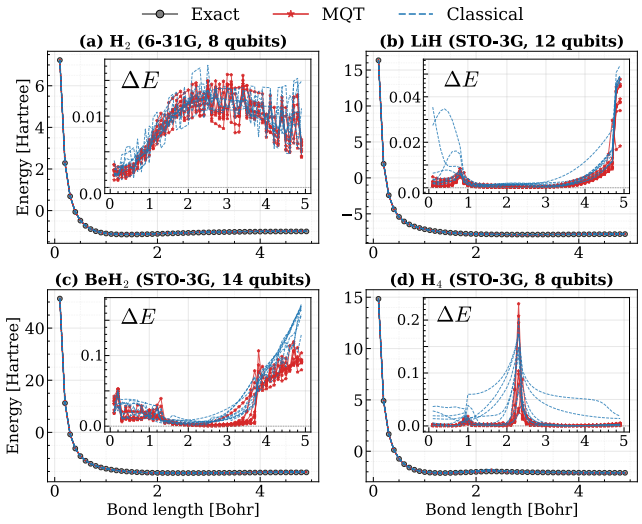

Figure 4: Potential energy curves and estimation errors (ΔE) in the inset plots for varying bond lengths in (a) H_2 (b) LiH, (c) BeH_2, and (d) H_4 molecules using the quantum (red lines) and classical (dotted blue lines) Transformers.

average r_{en} across its second dimension. The amplification module Amp_{r_e} multiplies each element at the index (i, k) $(1 \leq i \leq n, 1 \leq k \leq 2d_{\text{emb}})$ of the feature matrix to $e^{-r_e(i)}$ where $r_e(i)$ is the i-element in r_e. The output is then separately processed by Avg. and 1D-Conv. modules along the electron dimension direction (the first axis), summed to form a $2d_{\text{emb}}$-dimensional feature vector, and further normalized and amplified by Amp_r. Here r is the scalar average of r_e, and each element of the vector is multiplied by e^{-r}.

Numerical Experiments

In the following numerical experiments, we apply MQT to estimate the potential energy curves (PEC) of the molecular Hamiltonian for H_2, LiH, BeH_2, and H_4. For the second quantization of the Hamiltonians, we employ the Bravyi-Kitaev mapping for H_2 (8 qubits), LiH (12 qubits), and BeH_2 (14 qubits), and the Jordan-Wigner transformation for H_4 (8 qubits). The 6-31G basis set is used for H_2, while STO-3G basis set is applied to the other molecules.

We first evaluate MQT in a plain training scenario, where it is trained and tested on data from the same molecule. Subsequently, we investigate the pretraining on H_2, BeH_2, and H_4, followed by fine-tuning on a different molecule (LiH). We set $d_{\text{emb}} = 8$ as the default setting in these experiments.

Estimating the PEC with Plain Training

In plain training, the token matrix is generated in each iteration from a configuration with a random bond length sampled from (0.0, 5.0) [Bohr]. After training, the PEC is estimated for bond lengths ranging from 0.1 to 4.9 [Bohr] in 0.1 [Bohr] increments. We use $L = 6$ layers of QT modules in each block B_i. We compare the performance of MQT against a classical Transformer replacing the QT module in Fig. 2. To align with the settings of MQT, we

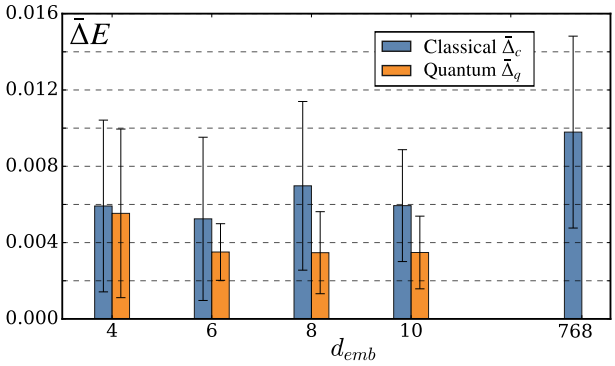

Figure 5: Bar plot comparing the average ground-state energy estimation error of LiH over potential energy curves between the classical Transformer ($\overline{\Delta}_c$) and MQT ($\overline{\Delta}_q$) as a function of the token embedding dimension d_{emb}. Error bars represent standard deviations. The quantum result at $d_{\text{emb}} = 768$ is not displayed due to resource limitations.

consider the classical Transformer with d_{emb} embedding dimensions, $4 \times d_{\text{emb}}$ intermediate dimensions of the feed-forward networks, and a single-head attention mechanism. We use AdamW (Loshchilov and Hutter 2019) optimizer with a weight decay rate of 0.001. The learning rate is set to 0.008 for the classical Transformer with H_2; and 0.004 for other cases. In the optimization, we run four processes in parallel on a single GPU (Recht et al. 2011), with each process performing 2500 iterations, for a total of 10^4 iterations. We use the TorchQuantum (Wang et al. 2022) library for the simulation of quantum circuits in MQT.

Metric	H_2	LiH	BeH_2	H_4
$\overline{\Delta}_c$	9.3e-3	6.0e-3	3.3e-2	1.9e-2
$\overline{\Delta}_q$	8.9e-3	3.5e-3	2.7e-2	0.5e-2

Table 1: Average ground-state energy estimation error by classical Transformer ($\overline{\Delta}_c$) and MQT ($\overline{\Delta}_q$)

Figure 4 compares the energy estimations of the MQT and classical Transformer against the theoretically calculated ground-state energies for each molecule. The inset figures show the estimation errors, with each line representing one of nine trials. MQT exhibits lower estimation errors than the classical Transformer for (b) LiH, (c) BeH_2, and (d) H_4. For (a) H_2, the estimation errors of MQT are nearly identical to those of the classical Transformer (note the differing y-axis scales across plots). The average estimation errors across all tested bond lengths for the classical Transformer ($\bar{\Delta}_c$) and MQT ($\bar{\Delta}_q$) are summarized in Tab. 1. These results suggest that MQT performs comparably to the classical Transformer for H_2 but outperforms it for other molecules, reducing the average estimation error by 42% in LiH, 18% in BeH_2, and 74% in H_4 compared to the classical Transformer. In Fig. 5, we further compare $\overline{\Delta}_c$ and $\overline{\Delta}_q$ of LiH for different $d_{\text{emb}} \in \{4, 6, 8, 10\}$. MQT shows lower estimation error, consistent with all d_{emb}, and achieves the saturated

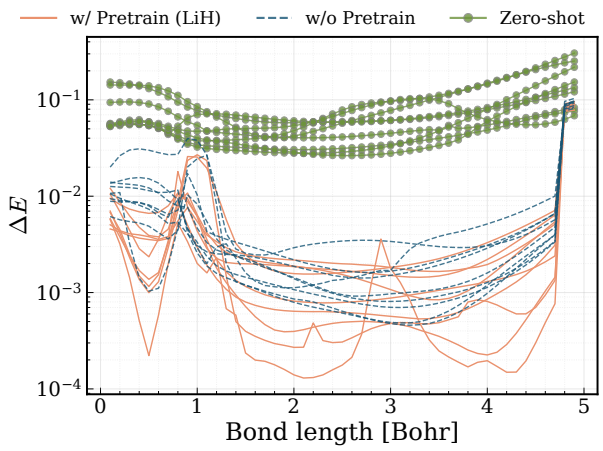

Figure 6: Energy estimation errors (ΔE) in nine trials show MQT trained on LiH with few-shot learning, comparing pretraining on H_2, BeH_2, and H_4 (orange lines) versus no pretraining (dotted teal-blue lines) and zero-shot learning (green line with circle markers) across varying bond lengths.

value when $d_{\text{emb}} \geq 6$. Interestingly, even with $d_{\text{emb}} = 768$, the classical Transformer exhibits significantly higher estimation errors compared to MQT at much lower d_{emb}. This discrepancy may be attributed to the optimization difficulty associated with the large number of parameters in the classical model, or it may indicate that MQT is inherently better suited to modeling molecular data. These findings underscore the potential advantages of integrating quantum structures into the Transformer framework.

Estimating the PEC with Pretraining

The MQT model with $L = 12$ layers in each QT block is pretrained on multiple molecules (H_2, BeH_2, H_4). Here we employ Bravyi-Kitaev mapping and train with 2×10^4 iterations. Then the model is fine-tuned with a small number of data points (few-shot learning) to estimate the PEC of LiH.

During fine-tuning, the training mirrors the setting in the pretraining stage, employing the AdamW optimizer with a weight decay of 10^{-3} and a learning rate of 4×10^{-3}. The training data for LiH consisted of five randomly selected bond lengths: $\{0.5, 1.5, 2.5, 3.5, 4.5\}$ [Bohr]. We run four processes in parallel on a single GPU, with each process performing 500 iterations, for a total of 2000 iterations for fine-tuning. After the fine-tuning, the PEC is estimated for bond lengths from 0.1 to 4.9 Bohr in 0.1 Bohr increments. Figure 6 depicts the average estimated error curves for pretraining and fine-tuning (orange lines, labeled "w/ Pretrain (LiH)") and fine-tuning only (green lines, labeled "w/o Pretrain") across nine trials. The pretrained model yields more accurate estimations in few-shot learning compared to only using fine-tuning, as indicated by reducing nearly 19% of error to the theoretical values from 7.6×10^{-3} ("w/o Pretrain") to 6.2×10^{-3} ("w/ Pretrain (LiH)"). Notably, with the same number of training data points, our model achieves an 11% improvement in few-shot learning performance compared to the neural network-based meta-VQE (Cervera-Lierta, Kottmann, and Aspuru-Guzik 2021), which yields an average estimation error of 7.0×10^{-3}. The neural network-based meta-VQE (Cervera-Lierta, Kottmann, and Aspuru-Guzik 2021) is reimplemented in our experiments on LiH.

We further evaluate the zero-shot learning, where the pretrained MQT is tested on LiH without fine-tuning. As shown in Fig. 6, the estimation errors of zero-shot MQT significantly exceed those of few-shot MQT, indicating that zero-shot learning is inadequate. At least a few-shot learning is necessary to enhance estimation accuracy.

Discussion

We proposed the MQT model, which leverages the quantum attention mechanism to revolutionize the calculation of ground-state energies. The key contribution is that MQT can be adapted to train on multiple molecules and multiple configurations without altering the model structure. MQT consistently outperformed the classical model in estimating potential energy curves for H_2, LiH, BeH_2, and H_4. In scenarios with small molecules and readily available training data, pretraining MQT can significantly reduce the number of quantum circuit runs required for larger and more complex molecular systems. To maximize this pretraining advantage, designing suitable pretraining datasets for specific molecules is crucial.

Scaling MQT to larger molecules requires addressing several limitations. A primary bottleneck is the output quantum state representation via amplitude embedding, where direct state preparation is challenging due to the complexity scaling exponentially with the number of qubits n_q. The qRAM scheme could reduce this complexity to near-linear in n_q, but fault-tolerant qRAM remains an engineering challenge. The FC-HF block tied to amplitude embedding in Fig. 2 could be replaced with a VQC to directly produce states without amplitude embedding. Inspired by the divide-and-conquer approach in VQE (Fujii et al. 2022), we can concatenate MQTs to enable the application to large systems with strong intrasubsystem and weak inter-subsystem interactions on small-scale quantum computers.

In the near term, while improvements in the architectural completeness of QT models and their compatibility with classical components are still necessary, establishing a unified benchmarking framework is critical for accurately evaluating their real-world performance across different approaches. Generative tasks involving molecular data represent particularly promising applications in this context. Looking ahead to the early era of fault-tolerant quantum computing, QT models based on quantum linear algebra offer strong theoretical potential (Guo et al. 2024; Liao and Ferrie 2024; Khatri et al. 2024). As quantum hardware continues to evolve, QTs are expected to play an increasingly impactful role in solving complex tasks, unlocking new possibilities in quantum-enhanced machine learning. In this perspective, MQT serves as a compelling example of how quantum-enhanced architectures can go beyond traditional text and image applications, addressing fundamental challenges in quantum chemistry and materials science, domains where classical neural networks are inherently limited.

Acknowledgments

The authors acknowledge Shintaro Sato, Koki Chinzei, and Nasa Matsumoto for their fruitful discussions. Special thanks are extended to Koki Chinzei for his valuable comments on the design of the MQT model.

References

Azad, U. 2023. PennyLane Quantum Chemistry Datasets. https://pennylane.ai/datasets/collection/qchem.

Bergholm, V.; and et al. 2022. PennyLane: Automatic differentiation of hybrid quantum-classical computations. *arXiv*.

Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; and Lloyd, S. 2017. Quantum machine learning. *Nature*, 549(7671): 195–202.

Ceroni, J.; Stetina, T. F.; Kieferova, M.; Marrero, C. O.; Arrazola, J. M.; and Wiebe, N. 2023. Generating Approximate Ground States of Molecules Using Quantum Machine Learning. arXiv:2210.05489.

Cervera-Lierta, A.; Kottmann, J. S.; and Aspuru-Guzik, A. 2021. Meta-Variational Quantum Eigensolver: Learning Energy Profiles of Parameterized Hamiltonians for Quantum Simulation. *PRX Quantum*, 2: 020329.

Cherrat, E. A.; Kerenidis, I.; Mathur, N.; Landman, J.; Strahm, M.; and Li, Y. Y. 2024. Quantum Vision Transformers. *Quantum*, 8: 1265.

Dunjko, V.; Taylor, J. M.; and Briegel, H. J. 2016. Quantum-Enhanced Machine Learning. *Phys. Rev. Lett.*, 117: 130501.

Evans, E. N.; Cook, M.; Bradshaw, Z. P.; and LaBorde, M. L. 2025. Learning with SASQuaTCh: a Novel Variational Quantum Transformer Architecture with Kernel-Based Self-Attention. arXiv:2403.14753.

Fujii, K.; Mizuta, K.; Ueda, H.; Mitarai, K.; Mizukami, W.; and Nakagawa, Y. O. 2022. Deep Variational Quantum Eigensolver: A Divide-And-Conquer Method for Solving a Larger Problem with Smaller Size Quantum Computers. *PRX Quantum*, 3: 010346.

Giovannetti, V.; Lloyd, S.; and Maccone, L. 2008. Quantum Random Access Memory. *Phys. Rev. Lett.*, 100: 160501.

Gonthier, J. F.; Radin, M. D.; Buda, C.; Doskocil, E. J.; Abuan, C. M.; and Romero, J. 2022. Measurements as a roadblock to near-term practical quantum advantage in chemistry: Resource analysis. *Phys. Rev. Res.*, 4: 033154.

Guo, N.; Yu, Z.; Choi, M.; Agrawal, A.; Nakaji, K.; Aspuru-Guzik, A.; and Rebentrost, P. 2024. Quantum linear algebra is all you need for Transformer architectures. arXiv:2402.16714.

Hammond, B. L.; Lester, W. A., Jr.; and Reynolds, P. J. 1994. *Monte Carlo Methods in Ab Initio Quantum Chemistry*. World Scientific. ISBN 978-981-02-0321-4.

Havlíček, V.; Córcoles, A. D.; Temme, K.; Harrow, A. W.; Kandala, A.; Chow, J. M.; and Gambetta, J. M. 2019. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747): 209–212.

Kandala, A.; Mezzacapo, A.; Temme, K.; Takita, M.; Brink, M.; Chow, J. M.; and Gambetta, J. M. 2017. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671): 242–246.

Khatri, N.; Matos, G.; Coopmans, L.; and Clark, S. 2024. Quixer: A Quantum Transformer Model. arXiv:2406.04305.

Kitaev, A. Y. 1995. Quantum measurements and the Abelian Stabilizer Problem. *Electron. Colloquium Comput. Complex.*, TR96.

Kohn, W.; Becke, A. D.; and Parr, R. G. 1996. Density Functional Theory of Electronic Structure. *J. Phys. Chem.*, 100(31): 12974–12980.

Lee, J.; Berry, D. W.; Gidney, C.; Huggins, W. J.; McClean, J. R.; Wiebe, N.; and Babbush, R. 2021. Even More Efficient Quantum Computations of Chemistry Through Tensor Hypercontraction. *PRX Quantum*, 2: 030305.

Li, G.; Zhao, X.; and Wang, X. 2024. Quantum Self-Attention Neural Networks for Text Classification. *Sci. China Inf. Sci.*, 67: 142501.

Liao, Y.; and Ferrie, C. 2024. GPT on a Quantum Computer. arXiv:2403.09418.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Mitarai, K.; Negoro, M.; Kitagawa, M.; and Fujii, K. 2018. Quantum circuit learning. *Phys. Rev. A*, 98: 032309.

Nielsen, M. A.; and Chuang, I. L. 2010. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition edition. ISBN 978-1-107-00217-3.

Peruzzo, A.; McClean, J.; Shadbolt, P.; Yung, M.-H.; Zhou, X.-Q.; Love, P. J.; Aspuru-Guzik, A.; and O'Brien, J. L. 2014. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.*, 5(4213): 4213.

Recht, B.; Re, C.; Wright, S.; and Niu, F. 2011. Hogwild!: A Lock-Free Approach to Parallelizing Stochastic Gradient Descent. In *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc.

Seeley, J. T.; Richard, M. J.; and Love, P. J. 2012. The Bravyi-Kitaev transformation for quantum computation of electronic structure. *J. Chem. Phys.*, 137(22): 224109.

Sipio, R. D.; Huang, J.-H.; Chen, S. Y.-C.; Mangini, S.; and Worring, M. 2021. The Dawn of Quantum Natural Language Processing. arXiv:2110.06510.

Smaldone, A. M.; Shee, Y.; Kyro, G. W.; Farag, M. H.; Chandani, Z.; Kyoseva, E.; and Batista, V. S. 2025. A Hybrid Transformer Architecture with a Quantized Self-Attention Mechanism Applied to Molecular Generation. arXiv:2502.19214.

Tilly, J.; Chen, H.; Cao, S.; Picozzi, D.; Setia, K.; Li, Y.; Grant, E.; Wossnig, L.; Rungger, I.; Booth, G. H.; and Tennyson, J. 2022. The Variational Quantum Eigensolver: A review of methods and best practices. *Physics Reports*, 986: 1–128.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

von Glehn, I.; Spencer, J. S.; and Pfau, D. 2023. A Self-Attention Ansatz for Ab-initio Quantum Chemistry. In *The Eleventh International Conference on Learning Representations*.

Wang, H.; Ding, Y.; Gu, J.; Li, Z.; Lin, Y.; Pan, D. Z.; Chong, F. T.; and Han, S. 2022. Quantumnas: Noise-adaptive search for robust quantum circuits. In *The 28th IEEE International Symposium on High-Performance Computer Architecture (HPCA-28)*.

White, S. R. 1992. Density matrix formulation for quantum renormalization groups. *Phys. Rev. Lett.*, 69: 2863–2866.

Xu, X.-F.; Xue, C.; Zhuang, X.-N.; Wang, Y.-J.; Sun, T.-P.; Fang, Y.; Wang, J.-C.; Liu, H.-Y.; Wu, Y.-C.; Chen, Z.-Y.; and Guo, G.-P. 2025. Towards Fault-Tolerant Quantum Deep Learning: Designing and Analyzing Quantum ResNet and Transformer with Quantum Arithmetic and Linear Algebra Primitives. arXiv:2402.18940.

Zheng, J.; Gao, Q.; and Miao, Z. 2023. Design of a Quantum Self-Attention Neural Network on Quantum Circuits. In *2023 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 1058–1063. IEEE.