JCTC Journal of Chemical Theory and Computation

pubs.acs.org/JCTC Article

A Hybrid Transformer Architecture with a Quantized Self-Attention Mechanism Applied to Molecular Generation

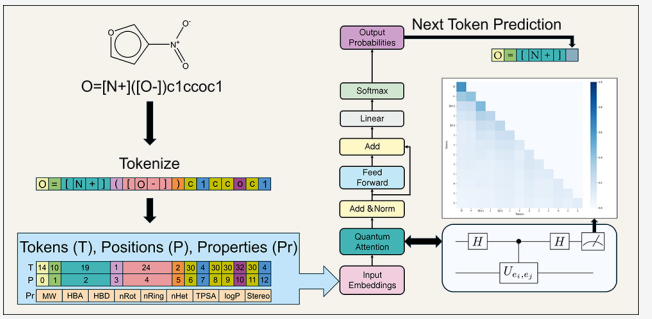
Anthony M. Smaldone,* Yu Shee, Gregory W. Kyro, Marwa H. Farag, Zohim Chandani, Elica Kyoseva, and Victor S. Batista*

Cite This: *J. Chem. Theory Comput.* 2025, 21, 5143−5154

Read Online

ACCESS | Metrics & More | Article Recommendations

ABSTRACT: The success of the self-attention mechanism in classical machine learning models has inspired the development of quantum analogs aimed at reducing the computational overhead. Self-attention integrates learnable *query* and *key* matrices to calculate attention scores between all pairs of tokens in a sequence. These scores are then multiplied by a learnable *value* matrix to obtain the output self-attention matrix, enabling the model to effectively capture long-range dependencies within the input sequence. Here, we propose a hybrid quantum-classical self-attention mechanism as part of a transformer decoder, the architecture underlying large language models (LLMs). To demonstrate its utility in chemistry, we train this model on the QM9 dataset for conditional generation, using SMILES strings as input, each labeled with a set of physicochemical properties that serve as conditions during inference. Our theoretical analysis shows that the time complexity of the query-key dot product is reduced from $O(n^2d)$ in a classical model to $O(n^2\log d)$ in our quantum model, where n and d represent the sequence length and the embedding dimension, respectively. We perform simulations using NVIDIA's CUDA-Q platform, which is designed for efficient GPU scalability. This work provides a promising avenue for quantum-enhanced natural language processing (NLP).



INTRODUCTION

Motivation. The self-attention mechanism, a cornerstone of the Transformer architecture[1] has revolutionized numerous machine learning (ML) domains, including natural language processing (NLP)[2] computer vision[3,4] and computational biology.[5,6] By capturing long-range dependencies in sequential data, it enables efficient and scalable learning, driving the Transformer's widespread adoption. Its versatility has spurred extensive research into refining and extending its applications within and beyond this framework.

Quantum machine learning (QML) has emerged as a rapidly growing field[7−12] leveraging quantum computation to potentially enhance learning and optimization tasks. Notably, a recent hybrid quantum−classical model successfully proposed and experimentally validated KRAS inhibitors, demonstrating that quantum machine learning techniques can already contribute meaningfully to real-world drug discovery workflows.[13] This field explores whether manipulating quantum states in Hilbert space outperforms classical vector operations in deep learning. Inspired by the success of the self-attention mechanism and the Transformer architecture, researchers are increasingly exploring quantum analogs to investigate potential performance gains achievable through the learning of information encoded into quantum states. Recently, Loshchilov et al.[14] introduced a normalized transformer with representation learning on a hypersphere. This approach bears similarity to quantum state evolution, where unitary operators move states across a hypersphere, suggesting that high-dimensional normalized representations may offer advantages for quantum self-attention mechanisms.

Background. The earliest application of self-attention in QML came from Li et al.[15] who used classical Gaussian projections of query and key quantum states for text classification. Some works depart from the classical formulation of the scaled dot-product attention mechanism and "mix" tokens together in Hilbert space to capture correlations instead of computing query-key dot products. For instance, Khatri et al.[16] develop a quantum algorithm of the skip-k-gram NLP technique using linear combinations of unitaries (LCU) and the quantum singular value transform (QSVT). Zheng et al.[17] encode both query and key vectors into a parametric quantum

Received: February 26, 2025
Revised: April 14, 2025
Accepted: April 16, 2025
Published: May 7, 2025

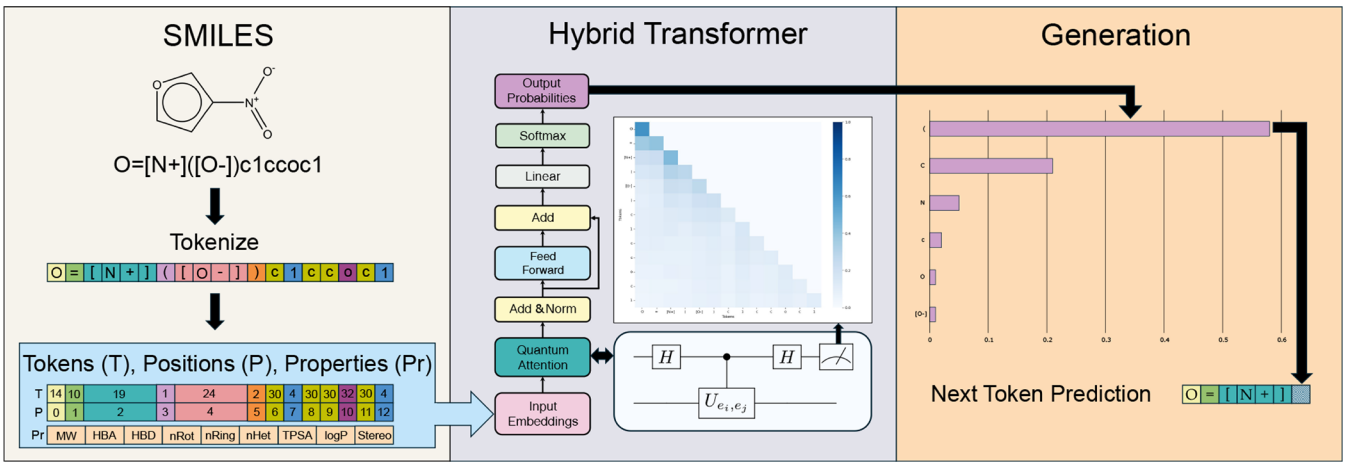


Figure 1. Proposed hybrid quantum-classical transformer model generates molecules by processing SMILES strings (e.g., O = [N+]([O-])c1ccccc1). Each string is split into a sequence of tokens, which are assigned token, positional, and physicochemical property embeddings. These embeddings pass through a hybrid self-attention mechanism: quantum circuits compute attention scores, which are combined with classical value matrices. The output then flows through the remaining classical transformer decoder to predict the next token in the sequence. This enables conditional molecular generation targeting specific physicochemical properties.

circuit (PQC) and measure the qubits to learn their correlations. Evans et al.[18] replace the explicit dot product with a PQC that blends tokens in the Fourier domain via quantum Fourier transformers (QFTs).

Other efforts focus on quantum analogs of self-attention and transformers that closely adhere to the classical framework, preserving the core principles of their operation. Xue et al.[19] propose an end-to-end quantum vision transformer; however, reliance of analog encoding for quantum random access memory (qRAM), leads to exponential scaling unless binary-tree structured data is assumed[20] limiting its feasibility for noisy intermediate-scale quantum (NISQ) and general-purpose applications. Cherrat et al.[21] introduce a hybrid approach, learning query and key states with $O(d)$ qubits—where d represents the embedding dimension—to compute the squared dot product as an attention score. Meanwhile, Liao and Ferrie[22] and Guo et al.[23] propose theoretical quantum transformer models based on algorithms like LCU and block encoding to utilize quantum linear algebra techniques. While these methods offer improved scaling under sparsity assumptions, their quantum resource requirement continues to make them impractical for the NISQ era, underscoring the demand for NISQ-friendly quantum self-attention approaches.

In this work, we introduce a novel quantum-classical hybrid self-attention mechanism, integrated into a transformer decoder for molecular generation. This approach uses $O(\log d)$ qubits and CNOT gates to learn all embeddings, as well as query and key representations, quantum mechanically. Unlike prior methods, it directly yields attention scores without squaring the dot product. We further incorporate positional embeddings and establish a general framework for additional embeddings, such as physicochemical molecular features, enabling control over generated molecular properties. Our results demonstrate that this hybrid model performs on par with classical baselines in SMILES validity, uniqueness, novelty, and property-targeted molecular generation (Figure 1).

FRAMEWORK

Classical Attention Score Calculation. For a given input sequence of tokens $\{x_1, x_2, ..., x_n\}$, each token x_i is mapped to an embedding $\mathbf{e}_i$ via a learned embedding matrix. Positional embeddings $\mathbf{p}_i$ are added to preserve token order, yielding the final input embeddings:

$$\mathbf{z}_i = \mathbf{e}_i + \mathbf{p}_i \tag{1}$$

where the embedding dimension is d (i.e., dim $\mathbf{e}_i$ = dim $\mathbf{p}_i = d$). Additional embeddings can enhance next-token prediction and condition the model to generate data with specific properties during inference. These are incorporated by summing κ additional vectors $\mathbf{c}_{i,v}$ with the token and positional embeddings, as in

$$\mathbf{z}_i = \mathbf{e}_i + \mathbf{p}_i + \sum_{v=1}^{\kappa} \mathbf{c}_{i_v} \tag{2}$$

following established practice.[24]

The input embeddings are then linearly projected into *query* ($\mathbf{Q}$), *key* ($\mathbf{K}$), and *value* ($\mathbf{V}$) matrices:

$$\mathbf{Q} = \mathbf{Z}\mathbf{W}^Q, \quad \mathbf{K} = \mathbf{Z}\mathbf{W}^K, \quad \mathbf{V} = \mathbf{Z}\mathbf{W}^V \tag{3}$$

where $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_n]^T$ stacks the input embeddings and $\mathbf{W}^Q$, $\mathbf{W}^K$, and $\mathbf{W}^V$ are learned weight matrices. The scaled dot-product attention mechanism[1] computes attention scores across all token pairs as

$$\mathbf{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathbf{A}(\mathbf{Q}, \mathbf{K})\mathbf{V} \tag{4}$$

with

$$\mathbf{A}(\mathbf{Q}, \mathbf{K}) = \mathrm{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right) \tag{5}$$

where d_k is the dimension of the key vectors.

Learning Attention Scores with Quantum States. In the attention matrix $\mathbf{A}$, each element $a_{i,j}$ is the scaled dot product of the i-th query vector $\mathbf{q}_i$ and the j-th key vector $\mathbf{k}_j$, followed by a softmax operation (see eq 5). In this work, we use quantum circuits to compute individual attention scores. We learn representations of the embedding vectors $\mathbf{z}_i$, query

https://doi.org/10.1021/acs.jctc.5c00331
J. Chem. Theory Comput. 2025, 21, 5143−5154

vectors $\mathbf{q}_i$, and key vectors $\mathbf{k}_i$ as quantum states, denoted $|q_i\rangle$ and $|k_j\rangle$, and determine their inner product $\langle q_i|k_j\rangle$. The value matrix $\mathbf{V}$ and subsequent operations, however, remain classical. Figure 2 illustrates our hybrid quantum-classical self-attention framework. The next subsection details the quantum circuit used to compute $\mathbf{A}$'s matrix elements.

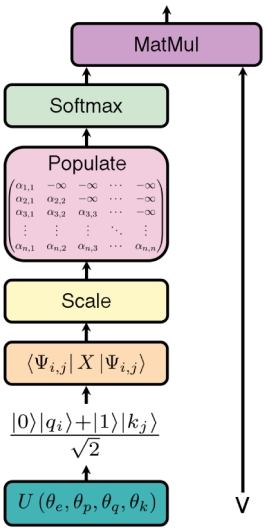


Figure 2. Quantum self-attention layer combining $\mathbf{QK}^{\mathrm{T}}$ calculated with quantum circuits and a classically computed value matrix $\mathbf{V}$ (eq 3). Quantum token embeddings $\boldsymbol{\theta}_{\mathbf{e}}$, positional angles $\boldsymbol{\theta}_{\mathbf{qp}}$, and learnable parameters $\boldsymbol{\theta}_{\mathbf{q}}$, $\boldsymbol{\theta}_{\mathbf{k}}$ are used in the unitary U as the circuit that evolves the quantum states depicted in eqs 7−12. Each quantum circuit produces an attention score, and thus, there are $\frac{n^2+n}{2}$ instances of U with their respective angles to ensure a fully populated masked attention matrix. The expectation value of the Pauli-X observable on the ancilla qubit (equivalent to a Hadamard transform and measurement in the computational basis as shown in eqs 13 and 14) is obtained and represents the dot product between query and key vectors. The original transformer implementation[1] scales attention scores by $\frac{1}{\sqrt{d_k}}$ to maintain a variance of 1. Since the dot products herein are obtained from the expectations of the quantum subsystems, they are bound on the closed interval $[-1, 1]$. To maintain a variance of 1, they must be scaled by $\sqrt{d_k}$. The scaled values are stored in the masked attention matrix, softmax is applied to the rows, and the resulting matrix is multiplied with $\mathbf{V}$.

Quantum Encoding of Token and Positional Information. Similar to the classical framework, we construct an embedding matrix to assign token embeddings $\theta_{\mathbf{e}_i}$, where each vector's entries are scaled to $[0, \pi]$ and its dimension equals the number of learnable parameters m in the ansatz U_e, which prepares the quantum state $|e_i\rangle$ for each token. In this work, we define learnable positional encoding angles $\theta_{\mathbf{p}_i}$, initialized to zero. The states $|e_i\rangle$ and $|p_i\rangle$ are prepared by applying unitaries $U_{e_i} = U_e(\theta_{\mathbf{e}_i})$ and $U_{p_i} = U_p(\theta_{\mathbf{p}_i})$ to initial states:

$$U_e(\theta_{\mathbf{e}_i})|0\rangle^{\otimes\frac{\log d}{2}} = |e_i\rangle,\ U_p(\theta_{\mathbf{p}_i})|0\rangle^{\otimes\frac{\log d}{2}} = |p_i\rangle \tag{6}$$

Just as the elements of the token and positional embeddings are learned classically, the parameters of the unitary operators U_{e_i} and U_{p_i} are learned as well. These states are prepared independently, yielding the composite state Ψ_1 as shown in Figure 4, which encodes token and positional information for a sequence:

$$\Psi_1 = |e_i\rangle \otimes |p_i\rangle = |z_i\rangle \tag{7}$$

All PQC ansatzes in this work use a single layer of R_y gates followed by an entangling layer of CNOT gates as shown in Figure 3.

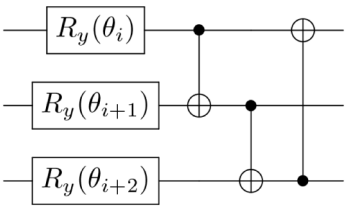


Figure 3. Structure of the parametric quantum circuits used in this work.

Learning Query and Key States. The separable quantum states of token and positional encodings are entangled via a PQC U_q, resulting in a quantum state analogous to a query vector.

$$\Psi_2 = U_q|z_i\rangle = |q_i\rangle \tag{8}$$

An ancilla qubit in the $|+\rangle$ state is introduced, and the entire circuit is conditionally reversed under its control. If the ancilla qubit is in the $|0\rangle$ state, the working register remains $|q_i\rangle$. If the ancilla qubit is in the $|1\rangle$ state, the working register is reset to $|0\rangle^{\otimes \log d}$ as shown in eqs 9 and 10

$$\Psi_3 = CU_q^{\dagger}\left(\frac{|0\rangle + |1\rangle}{\sqrt{2}} \otimes |q_i\rangle\right) = \frac{|0\rangle|q_i\rangle + |1\rangle|z_i\rangle}{\sqrt{2}} \tag{9}$$

$$\Psi_4 = \mathrm{CU}_{e_i}^{\dagger}\left(\mathrm{CU}_{p_i}^{\dagger}\left(\frac{|0\rangle|q_i\rangle + |1\rangle|z_i\rangle}{\sqrt{2}}\right)\right) = \frac{|0\rangle|q_i\rangle + |1\rangle|0\rangle^{\otimes\log d}}{\sqrt{2}} \tag{10}$$

where $CU = |0\rangle\langle 0| \otimes I + |1\rangle\langle 1| \otimes U$ and I is the identity operator.

Analogous to the preparation of $|z_i\rangle$ in eq 7, $|z_j\rangle$ is prepared with the difference being that all PQCs are controlled by the ancilla qubit. This controlled preparation ensures the modified Hadamard test yields a real-valued dot product $\langle q_i|k_j\rangle$ between query and key states. After preparing $|z_j\rangle$, a controlled PQC CU_k transforms it into $|k_j\rangle$, applied only when the ancilla is in $|1\rangle$, resulting in the state:

$$\Psi_5 = CU_{p_j}\left(CU_{e_j}\left(\frac{|0\rangle|q_i\rangle + |1\rangle|0\rangle^{\otimes\log d}}{\sqrt{2}}\right)\right) = \frac{|0\rangle|q_i\rangle + |1\rangle|z_j\rangle}{\sqrt{2}} \tag{11}$$

https://doi.org/10.1021/acs.jctc.5c00331
J. Chem. Theory Comput. 2025, 21, 5143−5154

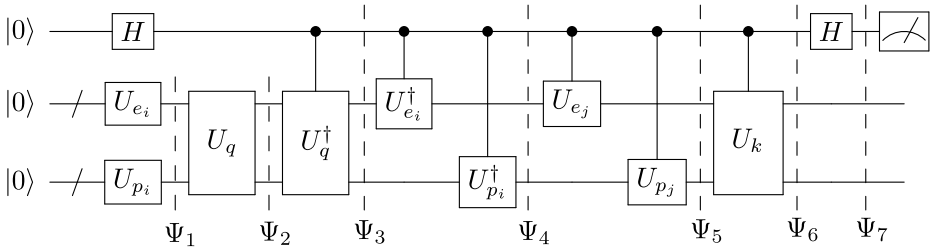


Figure 4. Quantum circuit used to create query and key states from a given quantum token and positional encoding to produce an attention score when measured. U_{e_i}/U_{e_j} and U_{p_i}/U_{p_j} are the unitaries that create the token and positional encoding of the i/j-th token into the quantum state. U_q and U_k are the unitaries to learn query and key representations of the quantum states containing token and positional information. The mathematical description of states Ψ_1 to Ψ_7 are found in eqs 7−13. The expectation value on the ancilla qubit yields the desired query-key dot product $\mathrm{Re}\langle q_i|k_j\rangle$.

$$\Psi_6 = CU_k\left(\frac{|0\rangle|q_i\rangle + |1\rangle|z_j\rangle}{\sqrt{2}}\right) = \frac{|0\rangle|q_i\rangle + |1\rangle|k_j\rangle}{\sqrt{2}}. \tag{12}$$

A final Hadamard gate is applied to the ancilla qubit, which transforms the state to

$$\Psi_7 = \frac{|0\rangle \otimes (|q_i\rangle + |k_j\rangle) + |1\rangle \otimes (|q_i\rangle - |k_j\rangle)}{2} \tag{13}$$

The ancilla qubit is measured yielding an expectation value of

$$\langle\Psi_7|(Z \otimes I^{\otimes \log d})|\Psi_7\rangle = \mathrm{Re}\langle q_i|k_j\rangle \tag{14}$$

which is equivalent to the classical analog of the i,j-th entry of the $\mathbf{QK}^{\mathrm{T}}$ matrix, where Z is the Pauli-Z gate Figure 4.

Because each attention score is computed by an independent quantum circuit, this process can be parallelized across multiple quantum processing units (QPUs). For next-token prediction, where the upper triangle of the $\mathbf{QK}^{\mathrm{T}}$ matrix is masked, a maximum of $\frac{n^2+n}{2} - 1$ unique quantum circuits are required in the worst-case scenario. This "worst case" occurs because circuits with identical parameters, which produce the same results, need not be recomputed, thus eliminating redundancy. Moreover, the first circuit, corresponding to $a_{1,1}$, need not be executed since its softmax output is always 1 due to masking. The overall architecture is shown in Figure 5.

Extension to Additional Embeddings. Beyond token and positional encodings, our method can incorporate additional embeddings. To mirror the classical approach of equal embedding dimensions, we choose all Hilbert subspaces for each embedding to be of equal dimension. Thus, to form the quantum state $|\tilde{z}_i\rangle$, which includes token, positional, and κ additional embeddings, we prepare

$$|\tilde{z}_i\rangle = U_e(\theta_{\mathbf{e}_i})|0\rangle^{\otimes(\frac{\log d}{\kappa+2})} \otimes U_p(\theta_{\mathbf{p}_i})|0\rangle^{\otimes(\frac{\log d}{\kappa+2})} \otimes \bigotimes_{v=1}^{\kappa} U_{c_v}(\theta_{\mathbf{c}_v})|0\rangle^{\otimes(\frac{\log d}{\kappa+2})}. \tag{15}$$

We note that the $\kappa + 2$ term arises because the total number of qubits ($\log d$) is divided into registers of equal size for each embedding—two registers for tokens and positions, along with κ additional registers. Here, we incorporate $\kappa = 1$ additional embeddings for physicochemical properties. Frequently, additional embeddings like molecular properties ($\mathbf{c}$) use the same

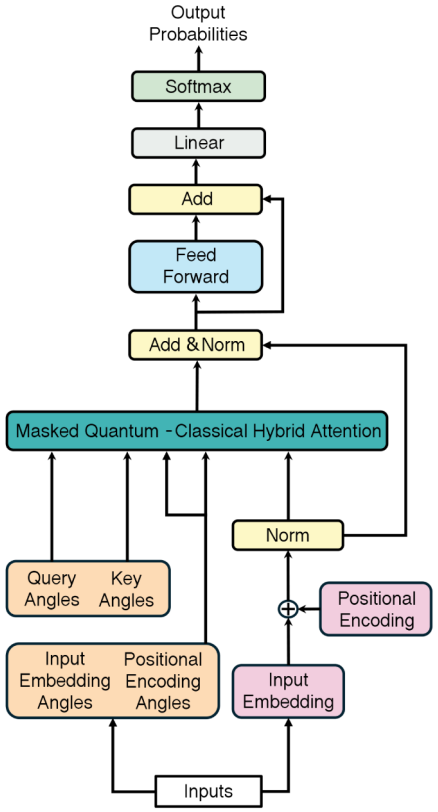


Figure 5. Architecture of the hybrid quantum-classical transformer decoder where embeddings are learned both quantum and classically. $\mathbf{QK}^{\mathrm{T}}$ is computed with quantum circuits and $\mathbf{V}$ is computed classically. Embedding and parameter (learnable position) matrices of dimension $n \times \dim\boldsymbol{\theta}_e = \mathrm{n} \times \dim\boldsymbol{\theta}_p$ and $n \times d$ are defined for a given input sequence to obtain quantum parameters (orange) and classical (pink) parameters of the model, respectively. The query and key angles are learned and used to transform the quantum embedding states into quantum query and key states. Modified-Hadamard tests are performed, and the output of the masked quantum-classical hybrid attention mechanism (teal) is a matrix of dimensions $n \times d$.

embedding vector for each sequence element. For this case, eq 2 reduces to

$$\mathbf{z}_i = \mathbf{e}_i + \mathbf{p}_i + \sum_{v=1}^{\kappa} \mathbf{c}_v = \mathbf{e}_i + \mathbf{p}_i + \mathbf{c} \tag{16}$$

Likewise, the quantum circuit can be simplified when embeddings are uniform across the sequence. Instead of reversing the entire register to $|0\rangle^{\otimes \log d}$ under control, as in eqs 9 and 10, subspaces with uniform embeddings remain

https://doi.org/10.1021/acs.jctc.5c00331
J. Chem. Theory Comput. 2025, 21, 5143−5154

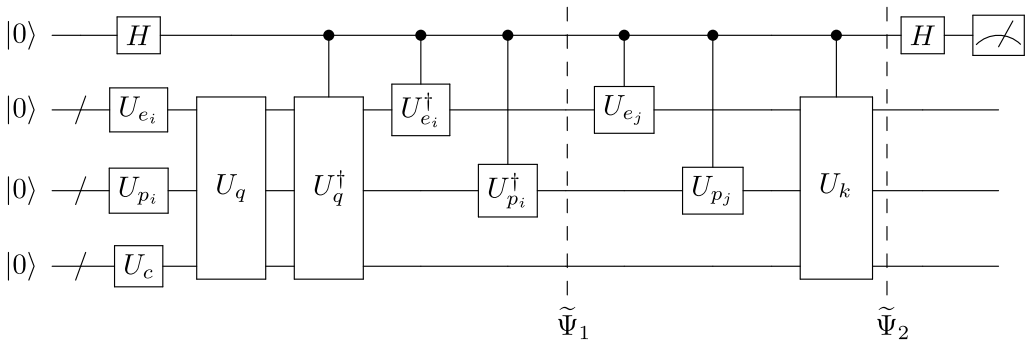


Figure 6. Quantum circuit in the hybrid quantum-classical self-attention mechanism. The circuit learns all embeddings, query and key representations, and produces a query-key dot product upon repeated measurement. U_c is a set of angles representing the physicochemical property embeddings. Mathematical descriptions of $\tilde{\Psi}_1$ and $\tilde{\Psi}_2$ are found in eqs 17 and 18, respectively.

unchanged, as they are identical across query-key pairs. This simplification yields the circuit in Figure 6, where

$$\tilde{\Psi}_1 = \frac{|0\rangle|q_i\rangle + |1\rangle|0\rangle^{\otimes \frac{2\log d}{3}}|c\rangle}{\sqrt{2}} \tag{17}$$

$$\tilde{\Psi}_2 = CU_{p_j}(CU_{e_j}(\tilde{\Psi}_1)) = \frac{|0\rangle|q_i\rangle + |1\rangle|k_j\rangle}{\sqrt{2}}. \tag{18}$$

The term $\frac{2\log d}{3}$ arises because, as explained in eq 15, the quantum registers for each embedding are designed to have equal sizes. Consequently, with token, positional, and an additional $\kappa = 1$ embedding representing molecular properties, there are three working quantum registers. Two of these three registers (the token and position quantum states) have been reversed under control.

Quantum Gradient Calculation. *Parameter Shift.* The inability to access intermediate quantum states significantly complicates traditional backpropagation via reverse-mode automatic differentiation for PQCs compared to classical methods.[25] The *parameter shift rule* offers an approach for computing exact gradients of PQCs. The expectation value of an observable $\hat{O}$ (e.g., a Pauli operator) is given by

$$f(\theta) = \langle 0|U^\dagger(\theta)\hat{O}U(\theta)|0\rangle \tag{19}$$

and the parameter shift rule computes its derivative with respect to θ as follows:

$$\frac{\partial f(\theta)}{\partial \theta} = \frac{1}{2}\left[f\left(\theta + \frac{\pi}{2}\right) - f\left(\theta - \frac{\pi}{2}\right)\right] \tag{20}$$

This approach computes the gradient by evaluating $f(\theta)$ at shifted values $\theta \pm \frac{\pi}{2}$, avoiding the need for finite differences but introducing an overhead of two circuit evaluations per parameter. As a result, its computational cost scales linearly with the number of parameters, rendering it impractical for large-scale QML models and motivating alternative methods.

Simultaneous Perturbation Stochastic Approximation. To address the parameter shift method's linear scalability with parameter count, we employ approximate quantum gradient calculations via the simultaneous perturbation stochastic approximation (SPSA) algorithm.[26] SPSA seeks a parameter vector $\mathbf{x} \in \mathbb{R}^m$ that minimizes

$$\min_{\mathbf{x}} f(\mathbf{x}) = \min_{\mathbf{x}} \mathbb{E}_{\xi}[F(\mathbf{x}, \xi)] \tag{21}$$

where $F: \mathbb{R}^m\mathbb{R}$ depends on the parameter vector $\mathbf{x} \in \mathbb{R}^m$ and a noise term ξ. SPSA approximates the gradient $\nabla f(\mathbf{x})$ as follows:

$$\nabla f(\mathbf{x}) \approx \frac{f(\mathbf{x} + \epsilon\cdot\mathbf{\Delta}) - f(\mathbf{x} - \epsilon\cdot\mathbf{\Delta})}{2\epsilon} \odot \frac{1}{\mathbf{\Delta}} \tag{22}$$

where $\mathbf{\Delta}$ is an m-dimensional vector with elements randomly chosen as ± 1, $\odot\frac{1}{\mathbf{\Delta}}$ represents the element-wise reciprocal and multiplication of the perturbation vector $\mathbf{\Delta}$, and the perturbation step ϵ is set to 0.01 in this work. We note that the noise term ξ in the objective function (eq 21) arises from the stochasticity of mini-batch sampling. This is separate from the perturbation vector Δ, which is generated by the SPSA algorithm to approximate the gradient.

SPSA requires just two PQC evaluations regardless of the parameter count, making it ideal for the NISQ era. Convergence analysis[27] indicates that larger batch sizes yield more stable updates by reducing gradient variance, aligning well with our training approach using a batch size of 256.

Complexity Analysis. *Computational Complexity.* The time complexity of the classical calculation of the attention matrix $\mathbf{A}$ in eq 5 is $O(n^2d)$ stemming from the multiplication of $\mathbf{Q}$ and $\mathbf{K}^T$, which are matrices of dimensions $n \times d$ and $d \times n$, respectively. Under the assumption of efficient preparation of quantum states, this modified Hadamard test approach produces each inner product $\langle q_i|k_j\rangle$ in $O(1)$ time compared to the classical $O(d)$, leading to a complexity of $O(n^2)$ for the population of $\mathbf{A}$. In practice, we prepare states with $O(\log d)$ depth, bringing the overall practical complexity to $O(n^2\log d)$. It is important to note that preparing quantum states in $O(\log d)$ time may result in highly structured states. Specifically, low-depth circuits are likely to produce quantum states with sparse amplitudes or pattern-like structures, which could make direct comparisons to classical algorithms that are designed for dense and unstructured data less equitable.

Query Complexity. The measurement of the ancilla qubit produces a Bernoulli random variable where the probability of measuring 1 is given by $p_0 = \frac{1 + \mathrm{Re}\langle q_i|k_j\rangle}{2}$. Thus, the Chernoff-Hoeffding theorem demonstrates $\mathrm{Re}\langle q_i|k_j\rangle$ can be found in $O\left(\frac{1}{\epsilon^2}\right)$ query complexity with additive error ϵ. However, since $\mathrm{Re}\langle q_i|k_j\rangle = 2p_0 - 1$ is obtained by measuring the ancilla qubit

https://doi.org/10.1021/acs.jctc.5c00331
J. Chem. Theory Comput. 2025, 21, 5143−5154

in the $|0\rangle$ state, quantum amplitude amplification can be performed to improve the query complexity to $O\left(\frac{1}{\epsilon}\right)$. To illustrate this, the final state before measurement as described in eq 13 can be written equivalently:

$$\Psi_7 = \frac{|0\rangle}{2} \otimes (|q_i\rangle + |k_j\rangle) + \frac{|1\rangle}{2} \otimes (|q_i\rangle - |k_j\rangle) = \sqrt{p_0}|\psi_{\text{good}}\rangle + \sqrt{1-p_0}|\psi_{\text{bad}}\rangle. \quad (23)$$

In this form, it is clear to see that a quadratic speed up in query complexity can be achieved using amplitude amplification with Grover operator

$$G = R_{\Psi_7} R_{\text{good}} \quad (24)$$

where the reflector R_{Ψ_7} is the unitary that prepares the entire Hadamard-test circuit $\Psi_7 = R_{\Psi_7}|0\rangle^{\otimes(\log d+1)}$ and the reflector R_{good} is the Pauli-Z gate since the ancilla qubit is always in state $|0\rangle$ for $|\psi_{\text{good}}\rangle$. While we leave the implementation of amplitude amplification to future researchers, we note that the overall practical complexity of the attention score calculation may be reduced to $O\left(\frac{n^2 \log d}{\epsilon}\right)$.

EXPERIMENTS

Dataset and Features. We employ the QM9 dataset,[28] a benchmark of 133,885 small organic molecules represented as SMILES strings,[29] as the basis for our study. The SMILES are canonicalized with RDKit[30] and duplicates were removed leaving 133,798 remaining data points. After preprocessing, the dataset was split into training and validation sets at a 20:1 ratio. From the dataset, we extracted nine key physicochemical features using RDKit: molecular weight (MW), number of hydrogen bond acceptors (HBA), number of hydrogen bond donors (HBD), number of rotatable bonds (nRot), number of rings (nRing), number of heteroatoms (nHet), topological polar surface area (TPSA), logP (partition coefficient), and the number of stereocenters (Stereo). To incorporate these descriptors as additional embeddings into our quantum-classical hybrid model, we transform them via a classical linear layer from 9 dimensions to $\dim\{\theta_e\} = \dim\{\theta_p\}$. We then scale the batch linearly between between 0 and π to produce θ_c for quantum circuit encoding. The SMILES strings are tokenized by breaking them into meaningful substructures, such as atoms, rings, branches, and bond types, and converting them into a sequence of discrete tokens that can be mapped to a high-dimensional embedding vector. The QM9 SMILES strings consist of 30 unique tokens, along with padding, start-of-sequence, and end-of-sequence tokens, resulting in a total vocabulary size of 33.

Benchmarks and Trainings. We trained two models in this study: one learning SMILES strings using only text sequences and another incorporating physicochemical embeddings for conditional molecular generation. The architecture of the proposed hybrid quantum-classical transformer used for this conditional generation task is illustrated in Figure 1, where molecular properties are embedded alongside token and positional information to guide the quantum attention mechanism. For the sequence-only setup, we assigned 3 qubits to each of the token and positional registers. For the condition-based setup, we allocated 2 qubits to each of the token, positional, and physicochemical registers, maintaining 6 working qubits across both configurations.

We compared the quantum-classical model's performance to that of fully classical models with equivalent architectures. All training setups employed one decoder layer and one attention head. The quantum model computed attention scores using 6 active qubits, producing a Hilbert space of dimension $2^6 = 64$, and thus, we set the classical token and positional embedding vectors to 64 dimensions. We also trained a classical model with equal parameter counts, denoted Classical−eq, for further comparison. Since each PQC ansatz uses a single layer of R_y gates, the number of learnable parameters per register matches the qubit count of 3 for token and positional registers in sequence-only training ($\frac{\log 64}{2} = 3$, eq 6), and 2 for token, positional, and physicochemical registers in condition-based training ($\frac{\log 64}{3} = 2$, eq 15). To match the parameter count between the Quantum and Classical−eq models, the weight matrices $\mathbf{W}^Q$ and $\mathbf{W}^K$ (eq 3) have shapes 3×2 and 2×3 for sequence-only and condition-based setups, respectively, yielding 6 parameters each for query and key transformations. All models in this work use 64-dimensional embeddings for the value matrix $\mathbf{W}^V$. To summarize, the total number of parameters for the hybrid quantum-classical model (Quantum), fully classical model with an equal number of parameters (Classical−eq) model were 47,704 and 48,307 for the sequence and condition-based models, respectively. The fully classical model with an equivalent architecture to the Quantum model but with traditionally sized query-key weight matrices $\mathbf{W}^Q$ and $\mathbf{W}^K$ of shape 64×64 (denoted as Classical) has 55,713 and 56,535 parameters for the sequence and condition-based model, respectively. To fairly evaluate the performance of these models, we initialized shared parameters across all models with identical random values.

We implemented the machine learning components using PyTorch.[31] All models underwent training for 20 epochs with the AdamW optimizer[32] set to a learning rate of 0.005 and a weight decay of 0.1. We applied gradient clipping with a maximum norm of 1.0 per layer to stabilize gradients and used cross-entropy loss as the objective function. To support a batch size of 256, we conducted quantum circuit simulations with CUDA-Q,[33] an open-source QPU-agnostic platform designed for accelerated quantum-classical supercomputing. All quantum simulations were performed using the state-vector simulator available in CUDA-Q. Training times for a single epoch on a CPU versus a single GPU are shown in Table 1, with a single GPU achieving a speedup of 1.34 over the CPU. Distributing simulations across four GPUs further yielded a speedup of 3.84 relative to one GPU. Consequently, we

Table 1. Training Time for a Single Epoch on the QM9 Dataset with the Condition-Based Quantum Model Using a Batch Size of 256[a]

Hardware	Epoch Time (hrs)
CPU	41.28
1 GPU	30.85
4 GPUs	8.03

[a]Runtimes for the full training are extrapolated from the average runtime over 4 batch updates. The GPUs and CPU used here are NVIDIA A100s and an AMD EPYC 7763, respectively.

https://doi.org/10.1021/acs.jctc.5c00331
J. Chem. Theory Comput. 2025, 21, 5143−5154

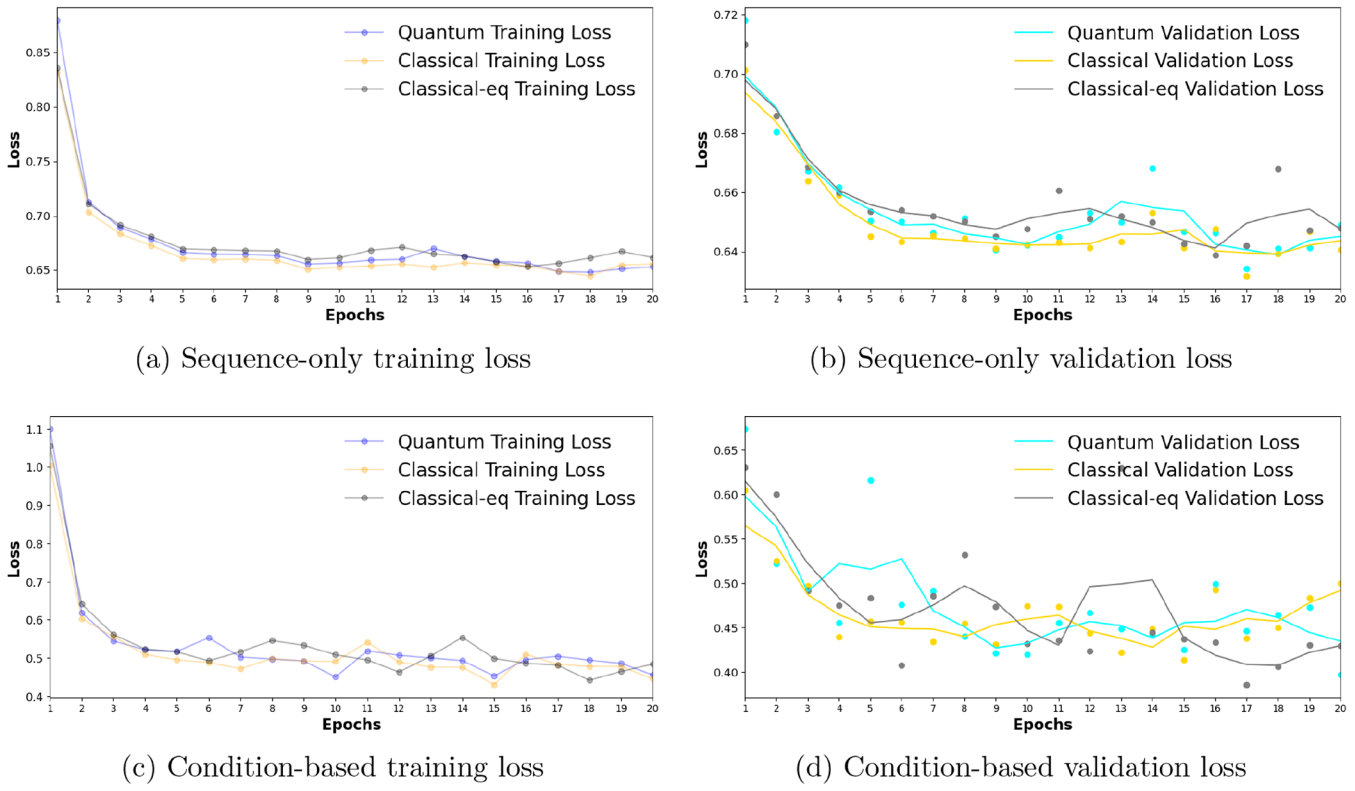


Figure 7. Learning curves for the training and validation losses of the quantum-classical model (Quantum), the fully classical model with an equal number of parameters to the quantum-classical model (Classical−eq), and the fully classical model with an equivalent architecture but with traditionally sized weight matrices (Classical). To better illustrate the model's learning progress, the validation curves (7b, 7d) display the 3-epoch moving average of the loss.

Table 2. Performance of Each Model at the Epoch with the Lowest Validation Loss[a]

	Model	Loss	Accuracy (%)	Validity (%)	Uniqueness (%)	V × U (%)	Novelty (%)
Sequence Only	Quantum	0.634	62.0	68.6	81.9	56.2	52.6
	Classical-eq	0.639	61.6	69.4	79.4	55.1	53.9
	Classical	0.632	62.4	72.5	81.2	55.9	52.0
Property Embeddings	Quantum	0.397	69.9	50.5	38.8	19.6	69.6
	Classical-eq	0.386	68.3	50.7	40.0	20.3	70.4
	Classical	0.414	68.0	38.5	48.0	18.5	71.2

[a]Quantum denotes the quantum-classical hybrid model. Classical−eq denotes a fully classical model with an qqual number of learnable parameters as the quantum model. Classical denotes a fully classical model with an equivalent architecture as the quantum model but with traditionally sized weight matrices. Accuracy % is the percentage of tokens correctly predicted. Validity % is the percentage of generated sequences that create a valid mol structure in RDKit Out of 100,000 queries to the trained model. The product of validity (V) and uniqueness (U) shows the percentage of model queries that result in unique compounds. Novelty % is the percentage of valid and unique SMILES strings that do not appear in the training set.

utilized four NVIDIA A100 GPUs on NERSC's Perlmutter supercomputer to accelerate training with this large batch size.

RESULTS

Evaluation of SMILES String Learning and Generation. Training and validation loss curves for the sequence-based and condition-based models are presented in Figure 7. For performance evaluation, we used the epoch with the lowest validation loss per model for inference, reporting next-token accuracy on the validation set as the ratio of correctly predicted tokens (highest output probability matching the true token) to total tokens. The validity, uniqueness, and novelty percentages from 100,000 inference queries to the trained models are reported. The results in Table 2 for the condition-based model used the mean values of each physicochemical property in the training set to construct the property embedding vector during inference, as it produced the greatest rate of valid and unique SMILES (shown as $V \times U$ in Table 2) across all models. The results using the mean, median, and mode of each property to guide the conditional generation were tested and are shown in Table A1 in the appendix.

The decrease in $V \times U$-from 55.1 to 56.2% for the sequence-only model to 18.5−20.3% for the condition-based model arises from the trade-off between structural validity and diversity when optimizing for property constraints. The condition-based model samples molecules from a narrower chemical space to satisfy both structural and physicochemical constraints. The benefit of incorporating property embeddings

https://doi.org/10.1021/acs.jctc.5c00331
J. Chem. Theory Comput. 2025, 21, 5143−5154

Table 3. Conditional Generation Results[a]

	MW	HBA	HBD	nRot	nRing	nHet	TPSA	logP	Stereo
Mean	122.77	2.23	0.83	0.92	1.74	2.47	37.16	0.30	1.71
Quantum	125.13	**2.31**	0.55	0.63	2.05	**2.37**	32.58	**0.30**	2.11
Classical−eq	**123.42**	2.15	0.55	0.55	2.03	2.21	31.74	0.46	1.98
Classical	128.04	2.15	**1.00**	**0.79**	**1.67**	2.19	**34.53**	0.52	**1.81**
Mean + (2 × σ)	137.88	4.34	2.50	3.10	4.16	4.84	79.67	2.30	4.77
Quantum	135.73	3.95	2.79	2.71	**3.74**	**4.82**	81.61	**2.24**	**4.46**
Classical−eq	**137.12**	3.98	**2.75**	2.52	3.62	4.76	**78.03**	2.36	4.34
Classical	136.09	**4.05**	2.90	**2.91**	3.53	4.78	83.44	2.37	4.27
Mean − (2 × σ)	107.66	0.12	0.00	0.00	0.00	0.10	0.00	−1.71	0.00
Quantum	112.83	0.63	**0.15**	**0.11**	0.00	0.05	**0.55**	**−1.74**	0.41
Classical−eq	**112.12**	0.58	0.19	0.11	0.00	**0.06**	0.67	−1.49	0.26
Classical	113.33	**0.30**	0.33	0.11	0.00	0.06	0.66	−1.46	**0.15**

[a]The top section demonstrates how well each model is able to generate molecules targeting the mean values of each property from the training data. The middle and lower sections indicate a target that is above and below the mean value for each property by 2 × the standard deviation (σ), respectively. For each model, the average value for that property of all valid generated molecules is shown. In the middle and lower sections, each numerical entry represents the result from an inference experiment where only that property was specified, and the remaining 8 properties were imputed from the training data with *k*-nearest neighbors. Bold values indicate which model generated molecules closer to the target value. Quantum indicates the quantum-classical hybrid model, Classical−eq denotes the fully classical model with an equal number of parameters as the quantum model, and classical denotes the fully classical model with an equivalent architecture to the quantum model but with traditionally sized weight matrices. All inferences were performed with the epoch that possessed the lowest validation loss for each model.

is reflected in the improved next-token prediction accuracy across all models, increasing from 61.6−62.4% to 68.0−69.9%. Across all metrics in both the sequence-only and condition-based trainings, the quantum and classical models exhibited comparable performances.

In-Distribution Modeling Performance. Following SMILES generation, we evaluated models trained with physicochemical embeddings for their ability to reproduce the training set's property distribution. The first row of Table 3 presents the mean values of the nine molecular properties in the training set. To assess how well each model aligns with this distribution, we performed inference using the epoch with the lowest validation loss per model employing a physicochemical embedding vector based on these mean property values. After 100,000 queries, we computed the properties of all valid SMILES strings and reported their averages in the upper section of Table 3 for each model. Bold values in each column indicate the model generating valid molecules closest to the target mean. Results revealed comparable performance across models, with the quantum model producing molecules nearest to the target means for 3 of 9 properties: hydrogen bond acceptors (HBA), heteroatoms (nHet), and logP (octanol−water partition coefficient). The Classical−eq model excelled at generating molecules with on-target molecular weights (MW), while the Classical model outperformed others for the remaining five properties: hydrogen bond donors (HBD), rotatable bonds (nRot), rings (nRing), topological polar surface area (TPSA), and number of stereocenters (Stereo).

Out-Of-Distribution Modeling Performance. The middle and bottom sections of Table 3 examined the models' ability to generate molecules beyond the training distribution. For each experiment, we set the target for one property two standard deviations above and below the mean ($\mu \pm 2\sigma$), imputing the other eight properties from the training data using the *k*-nearest neighbors (k-NN) method in scikit-learn.[34]

For targets 2σ above the mean, the quantum model generated molecules closest to the targets for four properties: nRing, nHet, LogP, and Stereo. The Classical−eq model matched three properties (MW, HBD, TPSA), while the Classical model outperformed others on two (HBA, nRot). Below the mean by 2σ, the quantum model excelled at HBD, nRot, TPSA, and LogP; the Classical−eq model at MW and nHet; and the Classical model at HBA and Stereo. Notably, all models generated molecules with zero rings equally well. To assess whether skewed distributions disproportionately affect any model, we repeated the experiment using median ± 1.5 IQR targets, where IQR is the interquartile range. Results in Table A2 confirm further that all models exhibit comparable performance in generating molecules beyond the training distribution.

Comparison of Attention Maps. To visualize and qualitatively compare the features learned by the attention mechanisms, attention maps for an example molecule, shown in Figure 8a, are presented in Figure 8b−d. While the aggregate quantitative performance of the models is similar, it is evident that they do not learn the same features to the same extent. This divergence in feature learning highlights the potential utility of hybrid quantum-classical self-attention mechanisms. Combining quantum and classical self-attention heads could enhance the extraction of a broader range of sequence features compared to relying solely on either. Such an approach could improve downstream task performance, an avenue for future research.

LIMITATIONS, FUTURE WORK, AND CONCLUSIONS

Our primary contribution is demonstrating that quantum states and learnable unitary evolutions can replace classical self-attention components in a generative model, achieving a NISQ-friendly solution that maintains performance parity with fully classical models. Additionally, we introduce a novel method for incorporating positional encodings to enhance the model's ability to learn sequence-based information, alongside the integration of supplementary embeddings, such as molecular properties. This approach enabled targeted molec-

https://doi.org/10.1021/acs.jctc.5c00331
J. Chem. Theory Comput. 2025, 21, 5143−5154

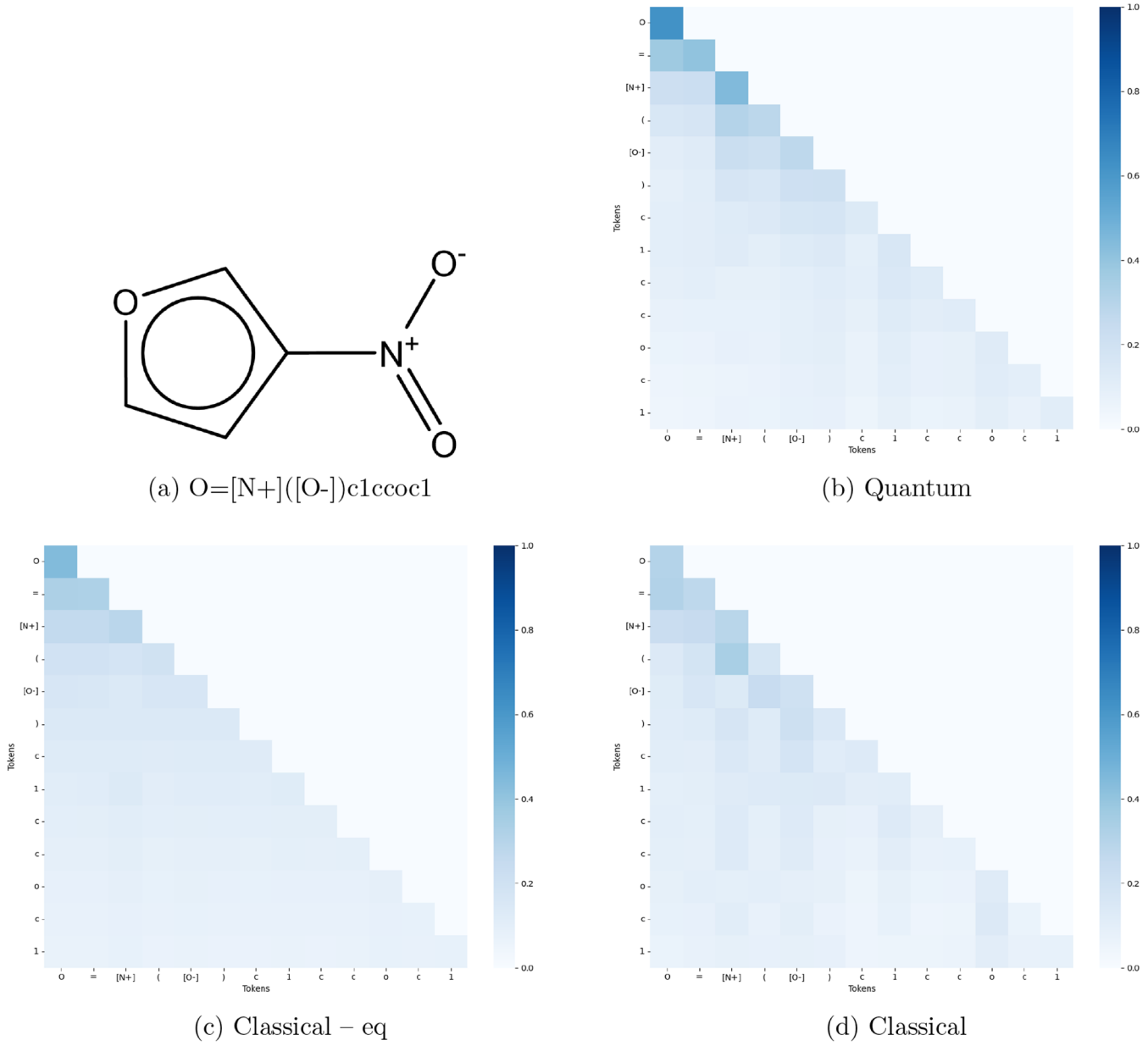


Figure 8. Attention maps of O = [N+]([O-])c1ccoc1 (a) for the quantum-classical hybrid model (Quantum) (b), the fully classical model with an equal number of parameters (Classical−eq) (c), and the fully classical model with an equivalent architecture as the quantum-classical model but with traditionally sized weight matrices (Classical) (d). All attention maps were computed using the model parameters from the epoch with the lowest validation loss per condition-based model.

ular generation, producing molecules with desired properties, thus demonstrating the potential of hybrid quantum-classical architectures for generative tasks. Notably, we achieved comparable performance between our quantum and classical baselines while training with the Simultaneous Perturbation Stochastic Approximation algorithm. Since parameter-shift gradients are computationally expensive, and backpropagation remains a common criticism of quantum neural networks, demonstrating competitive performance using SPSA is a promising result.

Despite these advances, our method has limitations to consider. The primary bottleneck in a self-attention mechanism's complexity is its quadratic scaling with sequence length, $O(n^2d)$. Our proposed method reduces the attention matrix computation time complexity to $O(n^2 \log d)$ but fails to address the dominant n^2 quadratic scaling term. Additionally, the complexity to multiply the attention matrix and value matrix still scales $O(n^2d)$. While prior quantum transformer and self-attention formulations suggest further reductions are theoretically possible (such as a complexity reduction to a polylogarithmic $O((\log n)^2 d)$ dependence on sequence length[23] they demand quantum resources impractical for the NISQ era such as efficient state preparation and block encodings of matrices, reinforcing our focus on NISQ compatibility over exhaustive complexity optimization. We defer exploring additional attention heads, decoder layers where *query* and *key* encodings could be prepared from the embedding vectors of the previous layer via a small neural network to map them to angles in a unitary, and more expressive ansatzes to future research. Additionally, while our

https://doi.org/10.1021/acs.jctc.5c00331
J. Chem. Theory Comput. 2025, 21, 5143−5154

Table A1. Inference Performance of Each Model at the Epoch with the Lowest Validation Loss, Where the Conditions Used for Generation are Chosen from the Mean, Median, and Mode of Those Properties from within the Training Data[a]

	Model	Validity %	Uniqueness %	V × U %	Novelty %
Mean	Quantum	50.5	38.8	19.6	69.6
	Classical−eq	50.7	40.0	20.3	70.4
	Classical	38.5	48.0	18.5	71.2
Median	Quantum	70.2	14.5	10.2	66.3
	Classical−eq	80.3	14.5	11.6	66.3
	Classical	70.9	23.1	16.4	75.2
Mode	Quantum	73.8	21.7	16.0	95.1
	Classical−eq	65.8	22.6	14.9	94.6
	Classical	64.9	28.0	18.2	95.4

[a]Quantum indicates the quantum-classical hybrid model, Classical−eq denotes the fully classical model with an equal number parameters as the quantum model, Classical denotes the fully classical model with an equivalent architecture to the quantum model, but with traditionally sized weight matrices. Validity % is the percentage of generated sequences that create a valid Mol structure in RDKit out of 100,000 queries to the trained model. The product of validity (V) and uniqueness (U) shows the percentage of model queries, which result in unique compounds. Novelty % is the percentage of valid and unique SMILES strings that do not appear in the training set.

Table A2. Conditional Generation Results[a]

	MW	HBA	HBD	nRot	nRing	nHet	TPSA	log P	Stereo
Median	125.13	2.00	1.00	1.00	1.00	2.00	35.82	0.28	2.00
Quantum	122.51	1.93	**1.24**	0.70	1.00	1.94	**35.58**	**0.24**	2.17
Classical−eq	**124.51**	1.95	1.26	0.81	1.00	**1.97**	34.89	0.33	**2.05**
Classical	123.55	**1.96**	0.70	**0.84**	1.00	2.04	31.51	0.53	1.58
Median + (1.5 × IQR)	134.07	3.50	2.50	2.50	2.50	3.50	81.29	2.23	6.50
Quantum	131.98	3.08	2.61	2.05	**2.39**	3.23	71.37	2.14	5.29
Classical−eq	**134.31**	3.32	2.96	1.99	2.25	**3.37**	86.42	**2.27**	**5.32**
Classical	134.66	**3.50**	**2.50**	**2.29**	2.12	3.70	**81.63**	2.47	5.12
Median − (1.5 × IQR)	116.19	0.50	0.00	0.00	0.00	0.50	0.00	−1.66	0.00
Quantum	116.09	**1.07**	0.23	0.15	0.00	**0.45**	0.92	**−1.60**	0.21
Classical−eq	119.34	1.12	0.48	0.13	0.00	0.77	**0.87**	−1.58	0.24
Classical	**116.22**	1.14	**0.16**	**0.11**	0.00	0.81	1.02	−1.41	**0.10**

[a]The top section demonstrates how well each model is able to generate molecules targeting the median values of each property from the training data. The middle and lower sections indicate a target that is above and below the median value for each property by 1.5 × interquartile range (IQR), respectively. For each model, the average value for that property of all valid generated molecules are shown. In the middle and lower sections, each numerical entry represents the result from an inference experiment where only that property was specified and the remaining 8 properties were inputed from the training data with k-nearest neighbors. Bold values indicate which model generated molecules closer to the target value. Quantum indicates the quantum-classical hybrid model, Classical−eq denotes the fully classical model with an equal number of parameters as the quantum model, Classical denotes the fully classical model with an equivalent architecture to the quantum model, but with traditionally sized weight matrices. All inferences were performed with the epoch that possessed the lowest validation loss for each model.

simulations do not account for quantum noise due to computational constraints, we provide a query complexity analysis to theoretically assess robustness. Future work will aim to incorporate noise models to study the model's behavior in realistic quantum scenarios. Remarkably, both the Quantum and Classical−eq models, with as few as two learnable parameters for tokens and positions, effectively learn SMILES strings. We hope these findings spur further development of practical, NISQ-ready designs that balance efficiency and performance for generative modeling.

APPENDIX A

Property Choices for Inference

For all models (Quantum, Classical−eq, Classical), the ranking of property value selection methods that maximize valid and unique molecules ($V \times U$) is mean > mode > median, as shown in Table A1. Interestingly, this statistical choice affects novelty, with mode-based inference increasing the fraction of novel compounds to over 95%. We tested the models' ability to generate molecules beyond the training distribution using median ± 1.5 IQR targets, as presented in Table A2, to assess whether distributional skew disproportionately impacts any model. For the upper range, the Quantum model performed best for nRing; the Classical−eq model for MW, nHet, LogP, and Stereo; and the Classical model for HBA, HBD, nRot, and TPSA. For the lower range, the Quantum model performed best for HBA, nHet, and LogP; the Classical−eq model for TPSA; and the Classical model for MW, HBD, nRot, and Stereo. These results—given the margins observed—suggest no model outperforms or underperforms in generating target molecules when using $\mu \pm 2\sigma$ versus median ± 1.5 IQR approaches.

https://doi.org/10.1021/acs.jctc.5c00331
J. Chem. Theory Comput. 2025, 21, 5143−5154

■ ASSOCIATED CONTENT

Data Availability Statement

The datasets and code to reproduce the figures and results from this work are available at https://github.com/anthonysmaldone/Quantum-Transformer, and as an application tutorial on the CUDA-Q documentation page https://nvidia.github.io/cuda-quantum/latest/applications/python/quantum_transformer.html.

■ AUTHOR INFORMATION

Corresponding Authors

Anthony M. Smaldone − *Department of Chemistry, Yale University, New Haven, Connecticut 06520, United States;* orcid.org/0009-0008-7265-0017; Email: anthony.smaldone@yale.edu

Victor S. Batista − *Department of Chemistry, Yale University, New Haven, Connecticut 06520, United States; Yale Quantum Institute, New Haven, Connecticut 06511, United States;* orcid.org/0000-0002-3262-1237; Email: victor.batista@yale.edu

Authors

Yu Shee − *Department of Chemistry, Yale University, New Haven, Connecticut 06520, United States;* orcid.org/0000-0002-3728-0021

Gregory W. Kyro − *Department of Chemistry, Yale University, New Haven, Connecticut 06520, United States;* orcid.org/0000-0002-0095-8548

Marwa H. Farag − *NVIDIA Corporation, Santa Clara, California 95051, United States*

Zohim Chandani − *NVIDIA Corporation, Santa Clara, California 95051, United States*

Elica Kyoseva − *NVIDIA Corporation, Santa Clara, California 95051, United States*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jctc.5c00331

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors acknowledge support from the National Science Foundation Engines Development Award: Advancing Quantum Technologies (CT) under award number 2302908. A.M.S. and G.W.K. acknowledge financial support from the National Science Foundation Graduate Research Fellowship under award number DGE-2139841. V.S.B. also acknowledges partial support from the National Science Foundation Center for Quantum Dynamics on Modular Quantum Devices (CQD-MQD) under award number 2124511, as well as computational resources provided by the National Energy Research Scientific Computing Center (NERSC) under the DOE Mission Science award ERCAP0031864.

■ REFERENCES

(1) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. *Adv. Neural Inf. Process. Syst.* arXiv:1706.03762, **2017**.

(2) OpenAI; Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S. *GPT-4 Technical Report*, 2024. http://arxiv.org/abs/2303.08774. arXiv:2303.08774 [cs].

(3) Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S. *An Image is Worth 16 × 16 Words: Transformers for Image Recognition at Scale*, 2021. http://arxiv.org/abs/2010.11929. arXiv:2010.11929 [cs].

(4) Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; Zagoruyko, S. *End-to-End Object Detection with Transformers*, 2020. http://arxiv.org/abs/2005.12872. arXiv:2005.12872 [cs].

(5) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583−589.

(6) Abramson, J.; Adler, J.; Dunger, J.; Evans, R.; Green, T.; Pritzel, A.; Ronneberger, O.; Willmore, L.; Ballard, A. J.; Bambrick, J.; Bodenstein, S. W.; Evans, D. A.; Hung, C.-C.; O'Neill, M. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **2024**, *630*, 493−500.

(7) Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum Machine Learning. *Nature* **2017**, *549*, 195−202.

(8) Wiebe, N.; Braun, D.; Lloyd, S. Quantum Algorithm for Data Fitting. *Phys. Rev. Lett.* **2012**, *109*, 050505.

(9) Lloyd, S.; Mohseni, M.; Rebentrost, P. Quantum principal component analysis. *Nat. Phys.* **2014**, *10*, 631−633.

(10) Rebentrost, P.; Mohseni, M.; Lloyd, S. Quantum Support Vector Machine for Big Data Classification. *Phys. Rev. Lett.* **2014**, *113*, 130503.

(11) Cong, I.; Choi, S.; Lukin, M. D. Quantum convolutional neural networks. *Nat. Phys.* **2019**, *15*, 1273−1278.

(12) Zoufal, C.; Lucchi, A.; Woerner, S. Quantum Generative Adversarial Networks for learning and loading random distributions. *npj Quantum Inf.* **2019**, *5* (1), 103.

(13) Ghazi Vakili, M.; Gorgulla, C.; Snider, J.; Nigam, A.; Bezrukov, D.; Varoli, D.; Aliper, A.; Polykovsky, D.; Padmanabha Das, K. M.; Cox, H., III Quantum-computing-enhanced algorithm unveils potential KRAS inhibitors. *Nature Biotechnology* **2025**, DOI: 10.1038/s41587-024-02526-3.

(14) Loshchilov, I.; Hsieh, C.-P.; Sun, S.; Ginsburg, B. *nGPT: Normalized Transformer with Representation Learning on the Hypersphere*, 2024. http://arxiv.org/abs/2410.01131. arXiv:2410.01131 [cs].

(15) Li, G.; Zhao, X.; Wang, X. *Quantum Self-Attention Neural Networks for Text Classification*, 2023. http://arxiv.org/abs/2205.05625. arXiv:2205.05625 [quant-ph].

(16) Khatri, N.; Matos, G.; Coopmans, L.; Clark, S. *Quixer: A Quantum Transformer Model*, 2024. http://arxiv.org/abs/2406.04305. arXiv:2406.04305 [quant-ph].

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

(18) Evans, E. N.; Cook, M.; Bradshaw, Z. P.; LaBorde, M. L. *Learning with SASQuaTCh: a Novel Variational Quantum Transformer Architecture with Kernel-Based Self-Attention*, 2024; http://arxiv.org/abs/2403.14753. arXiv:2403.14753 [quant-ph].

(19) Xue, C.; Chen, Z.-Y.; Zhuang, X.-N.; Wang, Y.-J.; Sun, T.-P.; Wang, J.-C.; Liu, H.-Y.; Wu, Y.-C.; Wang, Z.-L.; Guo, G.-P. *End-to-End Quantum Vision Transformer: Towards Practical Quantum Speedup in Large-Scale Models*, 2024. http://arxiv.org/abs/2402.18940. arXiv:2402.18940 [quant-ph].

(20) Mitarai, K.; Kitagawa, M.; Fujii, K. Quantum Analog-Digital Conversion. *Phys. Rev. A* **2019**, *99*, 012301.

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

(22) Liao, Y.; Ferrie, C. *GPT on a Quantum Computer*, 2024. http://arxiv.org/abs/2403.09418. arXiv:2403.09418 [quant-ph].

(23) Guo, N.; Yu, Z.; Choi, M.; Agrawal, A.; Nakaji, K.; Aspuru-Guzik, A.; Rebentrost, P. *Quantum linear algebra is all you need for*

Transformer architectures, 2024. http://arxiv.org/abs/2402.16714. arXiv:2402.16714 [quant-ph].
(24) Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*, 2019. http://arxiv.org/abs/1810.04805, arXiv:1810.04805.
(25) Abbas, A.; King, R.; Huang, H.-Y.; Huggins, W. J.; Movassagh, R.; Gilboa, D.; McClean, J. R. *On quantum backpropagation, information reuse, and cheating measurement collapse*, 2023. http://arxiv.org/abs/2305.13362. arXiv:2305.13362 [quant-ph].
(26) Spall, J. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Trans. Autom. Control* **1992**, *37*, 332−341.
(27) Hoffmann, T.; Brown, D. *Gradient Estimation with Constant Scaling for Hybrid Quantum Machine Learning*, 2022; http://arxiv.org/abs/2211.13981. arXiv:2211.13981 [quant-ph].
(28) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Sci. Data* **2014**, *1* (1), 140022.
(29) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31−36.
(30) Landrum, G.; Tosco, P.; Kelley, B.; Rodriguez, R.; Cosgrove, D.; Vianello, R.; Sriniker; Gedeck, P.; Jones, G.; NadineSchneider, *rdkit/rdkit: 2024_09_4 (Q3 2024) Release*, 2024. https://zenodo.org/doi/10.5281/zenodo.591637.
(31) Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L., *PyTorch: An Imperative Style, High-Performance Deep Learning Library*, 2019. http://arxiv.org/abs/1912.01703. arXiv:1912.01703.
(32) Loshchilov, I.; Hutter, F. *Decoupled Weight Decay Regularization*, 2019. http://arxiv.org/abs/1711.05101. arXiv:1711.05101.
(33) *The CUDA-Q development team, CUDA-Q*. https://github.com/NVIDIA/cuda-quantum.
(34) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.

https://doi.org/10.1021/acs.jctc.5c00331
J. Chem. Theory Comput. 2025, 21, 5143−5154