Quantum Machine Learning in Drug Discovery: Applications in Academia and Pharmaceutical Industries

Published as part of Chemical Reviews special issue "Quantum Computing".

Anthony M. Smaldone,[⊥] Yu Shee,[⊥] Gregory W. Kyro, Chuzhi Xu, Nam P. Vu, Rishab Dutta, Marwa H. Farag, Alexey Galda, Sandeep Kumar, Elica Kyoseva, and Victor S. Batista*
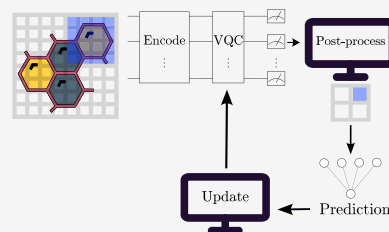
Cite This: https://doi.org/10.1021/acs.chemrev.4c00678

Read Online

ACCESS | 📊 Metrics & More | 📰 Article Recommendations

ABSTRACT: The nexus of quantum computing and machine learning—quantum machine learning—offers the potential for significant advancements in chemistry. This Review specifically explores the potential of quantum neural networks on gate-based quantum computers within the context of drug discovery. We discuss the theoretical foundations of quantum machine learning, including data encoding, variational quantum circuits, and hybrid quantum-classical approaches. Applications to drug discovery are highlighted, including molecular property prediction and molecular generation. We provide a balanced perspective, emphasizing both the potential benefits and the challenges that must be addressed.

CONTENTS

1. INTRODUCTION

Quantum Machine Learning (QML) refers to the use of quantum algorithms to train models by encoding classical or quantum data into quantum states and adaptively refining the state to fit the data through methods such as gradient descent or variational approaches. Post-training, QML models leverage the learned quantum representations for prediction or data generation. A useful conceptual framework in the field categorizes machine learning (ML) approaches based on whether the data and computational methods involved are quantum or classical. Within this context, our focus lies on leveraging quantum algorithms for both quantum and classical data, which aligns with the core strengths of QML in capturing quantum-enhanced representations.

While the field has seen many review articles on QML for chemistry,[1−6] significant gaps remain, particularly in the context of quantum neural networks (QNN) on gate-based quantum computing frameworks. Existing literature often provides a broad overview without delving into the specifics of the algorithms underlying QNNs. Additionally, many reviews limit their scope to quantum support vector machines (QSVMs) or generative models such as quantum generative adversarial networks (QGANs). In contrast, this review concentrates on QNNs, exploring both predictive and generative architectures in detail. While quantum reinforcement learning[7,8] has been extended to QNNs in recent studies,[9] we focus our review on supervised and unsupervised learning paradigms. We bridge gapsby presenting in-depth discussions of variational quantum circuits, QNN-specific algorithms, and their implementations. Furthermore, our inclusion of practical toolkits and benchmarks provides actionable insights for researchers and practitioners, while offering forward-looking suggestions to address challenges in the field. This comprehensive approach positions our review as a resource for advancing QML applications in chemistry and pharmaceutical sciences.

1.1. Quantum Computing

In this introduction, we discuss the general methodology of quantum computing based on unitary transformations (gates) of quantum registers, which underpin the potential advancements in computational power over classical systems. We introduce the unique properties of quantum bits, or qubits, quantum calculations implemented by algorithms that evolve qubit states through unitary transformations, followed by measurements that collapse the superposition states to produce specific outcomes, and last, the challenges faced in practical quantum computing limited by noise, with hybrid approaches that integrate quantum and classical computing to address current limitations. This introductory discussion sets the stage for a deeper exploration into quantum computing for machine learning applications in subsequent sections.

Calculations with quantum computers generally require evolving the state of a quantum register by applying a sequence of pulses that implement unitary transformations according to a designed algorithm. A measurement of the resulting quantum state then collapses the coherent state, yielding a specific outcome of the calculation. To obtain reliable results, the process is typically repeated thousands of times, with averages taken over all of the measurements to account for quantum randomness and ensure statistical accuracy. This repetition is essential to achieve convergence, as each individual measurement only provides probabilistic information about the quantum state.

Quantum registers are commonly based on qubits. Like classical bits, qubits can be observed in either of two possible states (0 or 1). However, unlike classical bits, they can be prepared in superposition states, representing both 0 and 1 simultaneously with certain probability. In fact, the state of a single qubit can be described using the ket notation, as follows:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{1}$$

where α and β are complex amplitudes satisfying the normalization condition $|\alpha|^2 + |\beta|^2 = 1$. Such a state represents the states $|0\rangle$ and $|1\rangle$ simultaneously with probability $|\alpha|^2$ and $|\beta|^2$, respectively.

Quantum registers with n qubits represent states that are linear combinations of tensor products of qubit states. Therefore, a register with n qubits represents 2^n states simultaneously, offering a representation with exponential advantage over classical registers. For instance, the state of a register with two qubits represents four states simultaneously, as follows:

$$|\psi\rangle = \alpha_{00}|0\rangle\otimes|0\rangle + \alpha_{01}|0\rangle\otimes|1\rangle + \alpha_{10}|1\rangle\otimes|0\rangle + \alpha_{11}|1\rangle\otimes|1\rangle \tag{2}$$

with complex coefficients α_{jk} satisfying the normalization condition $|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1$, and defining the probabilities $P_{jk} = |\alpha_{jk}|^2$ of observing the state collapsed onto state $|j\rangle \otimes|k\rangle$ when measuring the two qubits.

Quantum gates, analogous to classical logic gates, are used to represent the effect of the pulses that manipulate the states according to unitary transformations. Commonly used gates for the transformation of a single qubit are the gates represented by the Pauli matrices:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \; Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \; Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{3}$$

For example, the X (or, NOT) gate, flips the state of a qubit from $|0\rangle$ to $|1\rangle$, and vice versa. Another important class of transformations of a single qubit are the rotation gates $R_x(\theta)$, $R_y(\theta)$, and $R_z(\theta)$. The rotation around the Y-axis, for instance, is expressed as

$$R_y(\theta) = \exp\left(-i\frac{\theta}{2}Y\right) = \cos\left(\frac{\theta}{2}\right)I - i\sin\left(\frac{\theta}{2}\right)Y \tag{4}$$

where I is the identity matrix.

For multi-qubit systems, universal computing can be achieved with single qubit gates (such as Pauli, rotation, or T gates) and the two-qubit CNOT (Controlled-NOT) gate, defined as follows:

$$\mathrm{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{5}$$

The CNOT gate operates with respect to a specific ordering of qubits, where the first qubit in the pair serves as the control, and the second qubit acts as the target. This ordering dictates the action of the gate, ensuring that the target qubit is flipped only when the control qubit is in the $|1\rangle$ state.

Measurements of individual qubits project the superposition states onto one of the basis states of the operator used for the measurement. Averages over many measurements (i.e., many shots) are required to achieve statistical converge of the

calculation. For example, for a single qubit prepared in state $|\psi\rangle$, given by eq 1, measurements with the Z operator yield either 1 when the state is collapsed by the measurement onto state $|0\rangle$ (with probability $|\alpha|^2$), or -1 when the state is is collapsed into state $|1\rangle$ (with probability $|\beta|^2$).

Quantum mechanics introduces concepts such as superposition and entanglement, enabling computational parallelism. Quantum superposition allows for the representation and manipulation of an exponential number of states simultaneously, offering potential quantum advantage. Likewise, quantum entanglement, a form of correlation not present in classical systems, could further enhance this advantage. The simplest example is a register prepared in the Bell state $|\psi\rangle = \alpha_{00}|0\rangle \otimes|0\rangle + \alpha_{11}|1\rangle \otimes|1\rangle$ where measurement of one of the two qubits collapses the state of *both* qubits in highly correlated way so that both qubits end-up in the same collapsed state (both $|0\rangle$, or both $|1\rangle$, but never one $|0\rangle$ and the other $|1\rangle$). These quantum properties offer great potential for computational advantage over classical computers and thus could lead to significant advancements in many areas of chemistry, and beyond.

Quantum algorithms can achieve significant speed up compared to their classical counterparts. For example, the Quantum Fourier Transform (QFT)[10] can enable exponential speedup when compared to the best-known classical Fourier transform algorithms. Algorithms like Quantum Phase Estimation[11] and the Shor's algorithm[12,13] can also enable factorization of large numbers with exponential quantum advantage. Amplitude amplification techniques, such as those used in Grover's algorithm, provide a quadratic speed up for unstructured search problems, while the Harrow-Hassidim-Lloyd (HHL) algorithm[14] offers logarithmic speed up for solving linear systems within bounded error, highlighting the potential of quantum computing to outperform classical methods in a wide range of applications. The actual implementation of these quantum algorithms, however, would require fault-tolerant quantum computers that are not currently available to achieve quantum advantage over classical algorithms.

Due to the current limitations of quantum hardware, including noise and limited qubit counts, significant efforts have been focused on near-term calculations based on hybrid quantum-classical approaches where only part of the calculation is performed on the quantum computer while the rest of the computation is delegated to conventional high-performance computers.[15] For example, variational algorithms,[16] such as the Variational Quantum Eigensolver (VQE)[17] and Quantum Imaginary Time Evolution (QITE),[18−28] implement hybrid quantum-classical approaches. These algorithms generate quantum states and employ classical computations to combine the results of the measurements performed on the quantum states. This synergy leverages the strengths of both quantum and classical resources, making it feasible to solve problems with current noisy intermediate-scale quantum (NISQ) devices.

Despite significant advances in the field, an outstanding challenge is to achieve advantage over classical high performance computing. One promising direction is the use of quantum computers to implement machine learning algorithms. Harnessing the speed-up of quantum algorithms could address complex problems in data analysis and pattern recognition. Given that quantum computing has many potential applications in chemistry and biological science, there's a lot of hope that quantum machine learning can be extended to these areas of research.

1.2. Machine Learning

Machine learning (ML) algorithms are able to *learn* from data, where *learning* in this context can be defined according to "a program is said to learn from experience E with respect to some other class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E".[29] In practice, machine learning can be used to approximate a function of the input data to predict some variable (e.g., predict chemical toxicity from molecular features),[30,31] or can be used to learn the distribution of the input data to generate synthetic data akin to the training distribution (e.g., generating virtual compounds with specific drug-like properties).[32]

There are now many machine learning methods that have demonstrated exceptional, unprecedented abilities in many areas of research pertaining to drug development, with AlphaFold 2 and its later iterations being particularly recognizable.[33,34] AlphaFold is able to predict protein structures from their input sequences with high accuracy, although it is less capable in cases where the input sequence corresponds to a structure that is not well represented in the training distribution. Nonetheless, there is a lot of excitement and anticipation that AlphaFold will enable a lot of innovation within the domains of studying protein dynamics and hit identification in drug discovery.[35]

Machine learning has become pervasive in cheminformatics, and there have been many tools developed to predict molecular properties, generate compounds with prespecified properties, and ultimately reduce an incredible vast chemical search space to something tractable given the specific task at hand.[36] Specifically, there are a lot of efforts leveraging machine learning to reveal molecular mechanisms,[37] analyze complex biochemical data,[38] process and optimize chemical data,[39] predict protein structure,[33,34] virtual screening and drug design,[40,41] protein−ligand docking,[42] as well as many other tasks.[43]
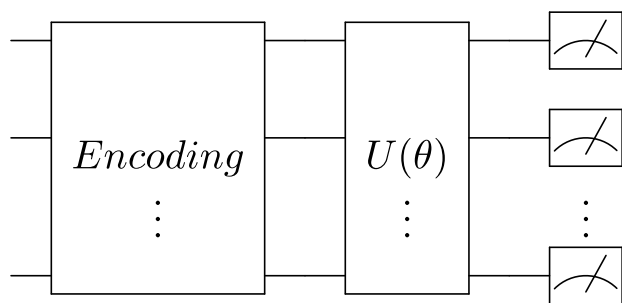
1.3. Quantum Neural Networks

Machine learning has had a transformative effect on all facets of modern life and has led to increasing computational demands, thus motivating the development of QML methods. The promising capabilities of quantum computing have already motivated the development of quantum analogs for a wide range of classical machine learning methods. Bayesian inference,[44] least-squares fitting,[45] principal component analysis,[46] and support vector machines[47] are some of the algorithms for which quantum counterparts have already been developed. While quantum analogs for these traditional ML methods have a demonstrable quantum speed-up,[48] some of the most awe-inspiring advances are due to artificial neural networks (ANNs).

Perhaps the earliest discussions of quantum neural networks (QNNs) were motivated by studies of neural function through the lens of quantum mechanics.[49] Since then, the field has evolved to exploiting the computational parallelization enabled by superposition states and entanglement.[50] In the early stages of research for QNNs, much effort was dedicated toward developing quantum systems that preserved the mechanisms of classical ANNs.[51−53] However, those efforts have largely failed to reconcile the linear dynamics of a quantum state evolving through a circuit and the nonlinear behavior of classical neural networks.[54] One notable attempt to overcome this issue is the development of Sigmoid Quantum Perceptrons (SQPs), which are parametrized quantum gates designed to emulate the nonlinear activation functions of classical perceptrons.[55] While SQPs theoretically exhibit a universal approximation property

similar to classical perceptrons, numerical experiments have shown that, at low qubit and gate counts, they do not perform well. In fact, simpler quantum circuits consisting of standard gates (such as Y-rotations and CNOTs) often outperform networks of SQPs in these cases. This highlights the challenge of using SQPs in the NISQ era, where the available quantum resources are limited. Additionally, researchers are developing methods to approximate nonlinear activation functions[56,57] to alleviate this concern. Increasingly, the field has consolidated around the use of variational quantum circuits to learn data representations[58] rather than directly creating a quantum analog of a neural network. Accordingly, quantum versions of the most popular classical neural network architectures, such as convolutional neural networks (QCNNs), graph neural networks (QGNNs), variational autoencoders (QVAEs), and generative adversarial networks (QGANs) have been realized and centered around variational quantum circuits.

QNNs require data encoding, variational quantum gates with learnable parameters θ, and measurements, as depicted in Figure 1. Data encoding converts classical data into a quantum state.

Figure 1. Structure of a typical quantum neural network. The input is encoded into a quantum state, followed by a variational quantum circuit and measurements.

The choice of strategy for data encoding can be of paramount importance in QNNs, as it can significantly affect performance and impact the underlying computational complexity. While other data encoding strategies exist,[59,60] three of the most popular methods are discussed in the following subsections.

1.3.1. Basis Encoding. Basis encoding is a straightforward and inexpensive method to encode binary data into a quantum system. Explicitly, let $\mathcal{D}$ be a classical binary dataset such that each element $x^m \in \mathcal{D}$ is an N-bit binary string of the form $x^m = (b_1^m, b_2^m, \cdots, b_N^m)$, with $b_j^m = 0$ or 1. Then the classical dataset can be represented by the quantum state $|\mathcal{D}\rangle$ of N qubits, where M is the total number of basis states used for the encoding:

$$|\mathcal{D}\rangle = \frac{1}{\sqrt{M}} \sum_{m=1}^{M} |x^m\rangle \tag{6}$$

where x^m corresponds to the m-th element of the dataset. For encoding a specific element (e.g., the binary string $[1,0,1]$) we simply place a Pauli X gate on the qubits that should be one (e.g., on the first and third qubits, as shown in Figure 2a).

1.3.2. Angle Encoding. Unlike basis encoding where the data is restricted to binary values, angle encoding allows data to take the form of real, floating point numbers. This encoding method entails rotating the state of a qubit around an axis of the Bloch sphere by an angle corresponding to the classical data. Explicitly, for an element x^m of a classical dataset $\mathcal{D}$ where $x^m \in$
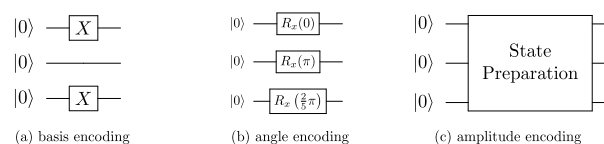
Figure 2. Data encoding methods. (a) Quantum circuit to prepare the $[1,0,1]$ vector with basis encoding. (b) Quantum circuit to prepare the prescaled $\left[0, \pi, \frac{2\pi}{5}\right]$ vector with angle encoding, choosing the x-rotation axis. (c) Amplitude encoding is equivalent to quantum state preparation.

$[0, 2\pi]$, then the value of x^m may be encoded into a single qubit by a rotation operator:

$$x \to R_k(x)|0\rangle = e^{-ix\sigma_k/2}|0\rangle \tag{7}$$

where k indicates the rotation axis (e.g., $k = y$).

Classical datasets seldom satisfy the 2π-periodicity requirement of rotation gates. Nevertheless, the data can always be normalized such that $x^m \in [0, 2\pi]$, or commonly $x^m \in [0, \pi]$. For example, suppose the task is to encode the vector $x = [0, 5, 2]$ into a quantum state via angle encoding with a maximum rotation angle of π. The vector is first normalized to the range $[0, \pi]$:

$$x_{\text{angles}} = \pi \cdot \frac{x - \min(x)}{\max(x) - \min(x)} = \left[0, \pi, \frac{2\pi}{5}\right] \tag{8}$$

After scaling, the angles can be encoded with the R_x or R_y gates, as shown in Figure 2b.

1.3.3. Amplitude Encoding. Amplitude encoding allows one to encode complex valued floats into the amplitudes of a quantum state. Thus, for a given classical dataset $\mathcal{D}$, an L2-normalized complex vector $x \in \mathcal{D}$ of length N can be encoded into $\log(N)$ qubits. Namely,

$$x \to U_x|0\rangle^{\otimes \log(N)} = |x\rangle = \sum_{k=0}^{2^N-1} \alpha_k|k\rangle \tag{9}$$

Many quantum neural networks rely on this encoding strategy, as it enables an exponential reduction in the number of required bits to represent data, and thus has the potential to allow for a speed-up that is not possible on classical computers. Despite this, the unitary operator U_x shown in eq 9 may demand a significant number of gates - a challenge discussed further in section 6.2.

1.3.4. Variational Quantum Circuits and Readout. Variational quantum circuits (VQCs), also commonly known as parametrized quantum circuits (PQCs), are typically used to introduce learnable parameters θ of unitary gates (Figure 3). After the VQC, measurements are performed. Measurements typically undergo classical post-processing to obtain averages. The set of parameters θ are iteratively adjusted by a classical
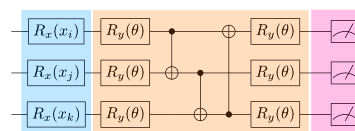

Figure 3. Generic three qubit quantum neural network using x-axis angle encoding (blue), a variational quantum circuit (orange), and measurements (pink).

computer to minimize a cost function $C(\theta)$ defined by the average expectation values $\langle\phi|U^\dagger(\theta)\hat{O}U(\theta)|\phi\rangle$, as follows:

$$C(\theta) = f(\langle\phi|U^\dagger(\theta)\hat{O}U(\theta)|\phi\rangle) \qquad (10)$$

where $|\phi\rangle$ are the encoded states and $U(\theta)$ is the ansatz of choice with learnable parameters and the function f is any classical post-processing function. The overall hybrid quantum-classical machine learning scheme is depicted in Figure 4.
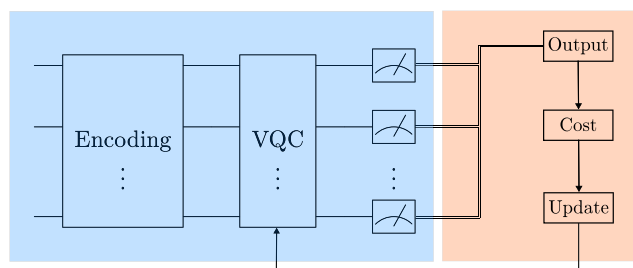
Figure 4. Training a general variational quantum circuit (blue) via classical post-processing and optimization (orange).

2. PREDICTIVE QUANTUM MACHINE LEARNING

2.1. Quantum Graph Neural Networks

Graph neural networks (GNNs) are popular models in applications of machine learning methods to chemistry because molecules can be intuitively represented as graphs where nodes are atoms and edges are bonds (Figure 5). In a typical GNN, messages (i.e., features used to describe each node) are passed between neighboring nodes, ultimately resulting in an aggregated graph-level encoding which can subsequently be processed to predict some value (e.g., protein–ligand binding affinity, hERG activity, etc.)[61]

QGNNs were first introduced with the Networked Quantum System.[62] In this system, a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ with the set of nodes $\mathcal{V}$ and edges $\mathcal{E}$ is defined as tensor products of Hilbert subspaces representing nodes and edges. The Hilbert space representing nodes, $\mathcal{H}_\mathcal{V} = \otimes_{v \in \mathcal{V}} \mathcal{H}_v$ and the space representing edges $\mathcal{H}_\mathcal{E} = \otimes_{e \in \mathcal{E}} \mathcal{H}_e$ are joined to create the full networked Hilbert space $H_\mathcal{G} = \mathcal{H}_\mathcal{V} \otimes \mathcal{H}_\mathcal{E}$ that comprise the space for the complete graph. Since then, various quantum theoretical formulations of QGNNs have been introduced.[63−66] Alongside quantum graph convolutional networks, quantum learning on equivariant graphs has also been demonstrated,[67,68] which has been of increasing interest in classical ML for drug discovery.[69−71]

Equivariant QGNNs and hybrid quantum-classical QGNNs have been used to predict the HOMO−LUMO gap in the QM9 dataset, which can provide insights on molecular stability.[72] An interesting observation from this work is that comparisons of their QGNN models to their corresponding classical models with the same number of parameters shows that the quantum models typically outperform the classical counterparts. Additionally, training of the quantum model is generally more efficient. These are exciting results that suggest favorable scalability and generalization of QGNNs, as previously suggested.[73] Another study,[74] has implemented a hybrid QGNN to predict the formation energy of perovskite materials. While their method underperforms compared to the fully classical GNN, it has been pointed out that advantages will emerge once state preparation techniques improve due to their usage of amplitude encoding.

Quantum isomorphic graph networks and quantum graph convolutional networks have been used to predict protein ligand binding affinities, showing that hybrid models already perform on par with state-of-the-art models.[75] In this work, features are amplitude encoded into a quantum state and a PQC replaces the classical multilayer perceptron (MLP) to perform convolutions. The models provide a good balance between number of parameters and generalization.

QGNNs are truly promising methods. For example, Liao et al.[66] has analyzed quantum implementations of the Simple Graph Convolutional network[76] and the linear graph convolutional network[77] that exhibit quantum advantage in terms of both space and time complexity. As the utility of graph networks is extended to both small molecules and large protein structures alike, solutions with complexity advantages are expected to be the dominant driver of the success of QGNNs.

2.2. Quantum Convolutional Neural Networks

Convolutional neural networks (CNNs) gained initial popularity for their success in image detection and classification.[78] They have been applied in chemistry to predict molecular properties, interaction strengths, and other chemically significant tasks.[79] The most fundamental architectural components of CNNs are the kernels of convolutional layers.[80] Each kernel creates a linear combination of the values in the spatial neighborhood of a given voxel (i.e., a pixel in the 2D case or a point in a 3D grid) of the input data and then propagates the resulting scalar to a corresponding spatial index in the output array. The coefficients for this linear combination are learned throughout training and constitute the weights of the kernel, which are applied uniformly across the input voxels.

QCNNs were first introduced for quantum phase recognition,[81] outperforming existing approaches with a significantly reduced number of variational parameters, scaling as $O(\log(N))$ with N the number of qubits. This initial success sparked significant interest, leading to the development of many QCNN variants,[82−86] tutorials,[87−90] and applications to a large range of complex tasks in many fields of science and technology. For example, in high energy physics, QCNNs have been used to
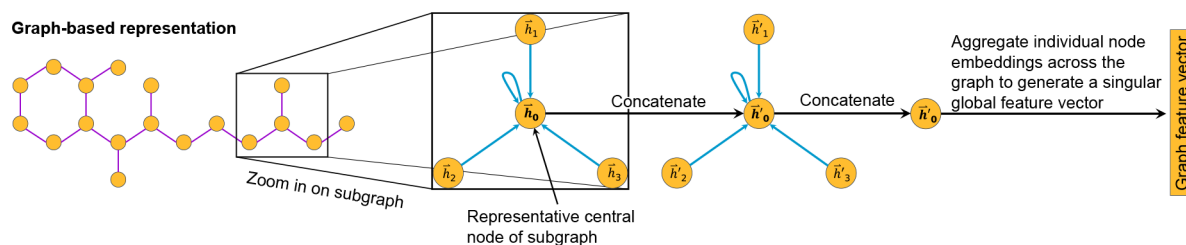
Figure 5. A classical graph neural network for extracting features from a molecule.

classify particles with a level of accuracy and speed of convergence that surpasses classical Convolutional Neural Networks (CNNs) with the same number of learnable parameters.[91] In the field of biochemistry, they have shown the ability to generate protein distance matrices[92] and predict protein−ligand binding affinities,[75,93] demonstrating their potential to contribute to our understanding of complex biological systems.

The appeal of QCNNs over many quantum counterparts of classical neural networks is multifaceted. In a QCNN, the classical convolutional filters are replaced by quantum circuits (Figure 6). In CNNs, the computation involves the discrete
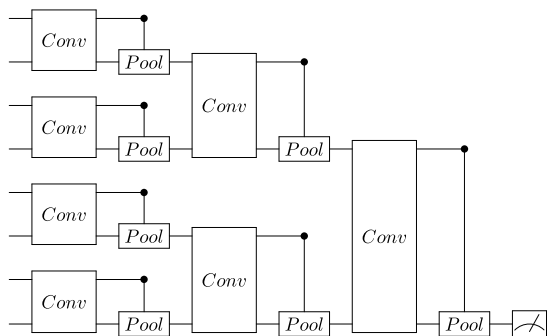
Figure 6. QCNN architecture introduced by Cong et al.[81] with $\log(N)$ parameters, where N is the number of qubits.

convolution between a relatively small kernel and the input data. This is attractive, as it allows the quantum approach to load only a small amount of information at a time onto quantum devices, as determined by the kernel size, which is of paramount importance during the NISQ era. This feature of QCNNs can be particularly useful in a biological context, as full-size feature maps would be too demanding.

Broadly speaking, there are two classes of QCNNs that could offer quantum advantage. This first class is akin to the general structure shown in Figure 6.[81] QCNNs with that structure incorporate pooling layers that halve the number of active qubits with each successive layer. This architectural choice involves only $O(\log(N))$ parameters and effectively circumvents the issue of barren plateaus − a significant challenge discussed further in Section 6.2. The second class can be termed Hybrid-QCNNs (HQCNNs). HQCNN models replace the forward pass of a convolutional filter with a quantum circuit, but perform

pooling layers classically after a measurement. HQCNNs are popular choices since they allow for more classical control over the network, with the mixing of quantum and classical components potentially offering performance gains at the expense of trainability and complexity brought by the original QCNN architecture.

QCNNs and HQCNNs offer distinct advantages that are attractive for chemical and pharmaceutical applications. While QCNNs require only $O(\log(N))$ parameters and avoid barren plateaus, this by itself does not deem them to be advantageous over classical CNNs. In a rigorous analysis of QCNNs (to which they later extend to all QML models), the generalization bounds of these models were investigated.[73] The reported analysis offers a guide to determine whether a QML model can exhibit better performance on unseen (test) data when compared to their classical counterpart. It is shown that when a QML model achieves a small training error on a given task, while the classical model with the same training error is significantly more complex, then the QML model will most likely outperform the classical model on unseen data.

This simple guide is particularly useful for drug discovery applications where datasets can often be limited but good generalization is paramount for discovery of life-saving compounds. Given that in the NISQ era QML models can only include a limited number of parameters, it is commonplace and intuitive when designing QML models to compare their performance to a classical network of equal parameters. Therefore, it is important to temper claims of advantage in the event of comparing a quantum and classical models, wherein the classical model might be heavily restricted for the sole purpose of fair comparisons with equal number of parameters. Instead, it is more significant to identify tasks which satisfy the criteria which guarantee good generalization bounds.[73] Shifting focus to this task identification, we anticipate that applications that are more likely to benefit from demonstrable quantum advantage are those for which the training data is scarce.

HQCNNs operate differently and more flexibly than QCNNs. So, the ways in which quantum advantage might be demonstrated is likely different from QCNNs. While the above criteria to identify potential generalization quantum advantage would still apply to HQCNNs[94], this becomes less straightforward as HQCNNs do not necessarily operate with $O(\log(N))$ parameters like their fully quantum counterparts. HQCNNs have been proposed to enable quantum speed-up in the CNN architecture (and neural network architectures in
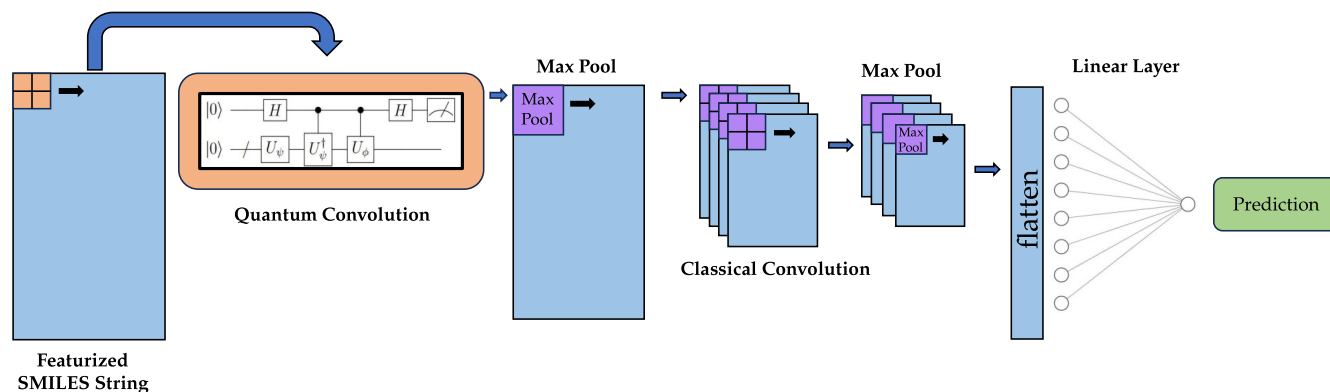
Figure 7. Hybrid Quantum-Convolutional Neural Network (HQCNN). The full architecture contains a quantum VQC layer, followed by classical pooling and classical convolutional layers. Adapted with permission from Smaldone and Batista.[30] Copyright 2024 American Chemical Society.

general) by directly calculating the inner product of the filter and input data (Figure 7).[30,95,96] These approaches are attractive when searching for quantum advantage, as they are task-agnostic and the potential for realization on quantum hardware is dictated almost exclusively by data representation - a much more straightforward litmus test of advantage compared to that required for a generalizability advantage.

The success of classical CNNs in drug discovery has prompted the exploration of QCNNs, as in the domain of biophysics where the relatively large input data can be broken up into tractable quantum circuits using the HQCNN methodology. An early biophysical application of HQCNNs has involved a model capable of predicting protein structure,[92] where the sequence lengths of the protein chains range from 50 to 500 residues and 50 to 266 residues in the training and testing sets, respectively. The reported results indicate commensurate performance to predictions by the popular classical model DeepCov[97] for protein contact maps while offering faster training convergence. Both Domingo et al.[93] and Dong et al.[75] trained HQCNNs to predict protein−ligand binding affinities. Domingo et al. demonstrated that their HQCNN architecture is able to reduce the number of parameters by 20% compared to the equivalent classical counterpart while maintaining performance. They noted that depending on the hardware, including a quantum layer into the CNN architecture resulted in a 20% to 42% reduction in training times to achieve the same performance as a fully classical network. Similarly, Dong et al. demonstrated competitive results with force field-based MM/GBSA and MM/PBSA calculations while reducing the overall number of parameters to their classical counterparts. In the work by Smaldone and Batista,[30] a HQCNN has been trained to predict drug toxicity (Figure 8). This work has demonstrated a

method where the weights of a convolutional layer are learned via quantum circuits while performing the underlying matrix multiplication of discrete dot products with quadratic quantum speed-up. This strategy performs at the level of classical models with equal number of parameters and can be transferred to a classical CNN mid-training to allow for noiseless training convergence. While this model and those presented by Domingo et al.[93] and Dong et al.[75] rely on an assumption of noiseless and efficiently prepared quantum states, they provide insights as to where quantum advantages may lie as the quantum hardware is concurrently improved.

2.3. Looking Ahead: Quantum Machine Learning for Large Molecules

mRNA and antibody-based biotherapeutics are critical for the development of next-generation therapies, yet both pose complex challenges, such as determining mRNA structures and understanding antibody−antigen interactions. Quantum computing has already shown promise by predicting mRNA secondary structures (see Figure 9),[98] and quantum neural
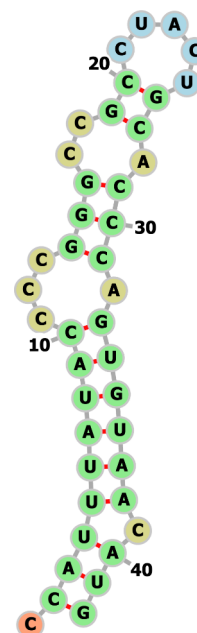
Figure 9. Optimal folded mRNA structure of the 42-nucleotide sequence computed using the VQE algorithm with 80 physical qubits on the IBM quantum processor Heron. Reproduced with permission from ref 99. Copyright 2024 IEEE.

networks are now being applied to tackle antibody−antigen interactions. Notably, Paquet et al.[100] introduced Quantum-Bound, a hybrid quantum neural network designed to predict the physicochemical properties of ligands within receptor−ligand complexes. Furthermore, Jin et al.[101] developed a QNN model to predict potential COVID-19 variant strains using available SARS-CoV-2 RNA sequences. These early successes highlight the potential of quantum neural networks to address key challenges in biotherapeutics.

3. GENERATIVE QUANTUM MACHINE LEARNING

3.1. Quantum Autoencoders

The primary motivation behind the development of autoencoders is to compress data into a latent space, reducing dimensionality while preserving essential information of the


Figure 8. Quantum CNN summary. (Top) Quantum circuit by Smaldone and Batista[30] employed to train a QCNN that predicts drug toxicities with a quadratic quantum speed-up for matrix multiplication, assuming efficiently prepared quantum states. (Bottom) Learning curve for the prediction of drug activity to the androgen receptor. The yellow region indicates epochs where the model was trained with reduced complexity using quantum circuits, and the green region shows where the weights were derived and training was continued using a classical CNN. Adapted with permission from Smaldone and Batista.[30] Copyright 2024 American Chemical Society.

(a) QAE.

(b) HQA.

(c) QAE for classical data.

Figure 10. Different types of Quantum Autoencoders (QAEs). (a) QAE utilizing a fully quantum circuit as the model architecture. (b) Hybrid Quantum Autoencoder (HQA) with classical latent representation. (c) QAE with classical data as the input and output.

training data. Similarly, the original motivation for development of Quantum Autoencoders (QAEs) is to compress quantum data (Figure 10). Variational Autoencoders (VAEs), a specific type of autoencoders, have gained popularity for molecular generation due to their ability to learn compact representations of molecular structures and generate new molecules with similar properties. QVAEs can compress quantum states and could therefore enable new avenues for molecular generation, though the exact benefits of QVAEs in this domain require further investigation.

There are two primary types of QAE, both utilizing hybrid quantum-classical schemes where classical computers are used for parameter optimization. The first type employs a quantum circuit as the model architecture,[102−108] aiming to leverage quantum gates and operations to encode and decode quantum states (see Figure 10a). The second type, known as the Hybrid Quantum Autoencoder (HQA),[109] employs measurement outcomes as the latent representations. This approach combines classical networks with QNNs in a hybrid model architecture, where classical vectors derived from quantum measurements are accessible for further analysis and processing (see Figure 10b). Note that the compression is effective (with no loss of information) only if the set of states to be compressed has support on a subspace (lower dimension) of its Hilbert space.[110] For example, the success of the Hubbard model example from Romero et al.[102] is due to the fact that these physical states exhibit certain symmetries.

A proposal for QVAE[111] involves the model architecture of the first type of QAE shown in Figure 10a and a latent representation regularized as in classical VAE. The regularized latent space can enhance classification performance compared to QAE. However, the regularization process requires mid-circuit quantum state tomography, which may represent a practical challenge for fully characterizing the state and scaling up.

Despite the promising aspects of QAE, several challenges remain. First, training relies on classical optimization algorithms, which can obscure statements about the overall computational complexity. Second, these models assume that input states can be efficiently prepared, a relatively straightforward task for quantum data but challenging for classical data (see Figure 10c). The encoding of classical data into quantum states might negate the computational benefits offered by quantum computers. Consequently, no immediate advantage can be claimed for QAE on classical data over classical methods at present. However, advancements in quantum computing hardware and more efficient optimization schemes could lead to significant improvements, making QAE a more viable and efficient tool in the future, particularly as the training optimization and data encoding complexity becomes comparable to the quantum components of the models.

3.2. Quantum Generative Adversarial Networks

Generative Adversarial Networks (GANs) are machine learning models designed to generate new data samples that mimic samples from a given distribution. GANs consist of three primary components: the prior distribution/noise sampling, the generator, and the discriminator. The generator creates data samples from random noise sampling, while the discriminator evaluates the authenticity of the generated samples by comparison against real data. This adversarial training process helps the generator improve over time, creating increasingly realistic samples. GANs have found applications in molecular generation, much like VAEs, and have been shown to generate novel molecular structures that adhere to desired properties.[112−115]

A QGAN was proposed by Dallaire-Demers and Killoran.[116] This work introduced the concept of using quantum circuits within the GAN framework, specifically leveraging quantum

circuits to measure gradients. Romero and Aspuru-Guzik[117] extended the concept of QGANs by modeling continuous classical probability distributions using a hybrid quantum–classical approach. While their results were promising, they noted that further theoretical investigations were necessary to determine whether their methodology offers practical advantages over classical approaches.

QGANs have been applied to generation of small molecules,[118] in a study that applied QGANs to the QM9 dataset.[119] That study reported better learning behavior due to the claimed superior expressive power and fewer parameters required by the quantum models. However, these QGANs struggled to generate valid molecules, and subsequent tests by other researchers indicated that these QGANs struggled to generate train-like molecules.[120]

Kao et al.[120] explored the advantages of QGANs in generative chemistry by testing different components of the GAN framework with quantum counterparts. They demonstrated that using a quantum noise generator (prior distribution sampling) could yield compounds with better drug properties. However, they found that quantum generators struggled to generate molecules that resembled those in the training set and encountered computational restrictions during further training. Additionally, they showed that a quantum discriminator with just 50 parameters could achieve a better KL score than a classical discriminator with 22000 parameters, indicating that quantum components can enhance expressive power even with a much fewer number of parameters. Nevertheless, these advancements often compromised the validity and uniqueness of the generated molecules, potentially undermining the efficiency of the sampling and generation processes.

Anoshin et al.[121] introduced a hybrid quantum cycle generative adversarial network for small molecule generation, utilizing the cycle-consistent framework from prior research.[122,123] Their approach featured a hybrid generator where a quantum circuit processed the noise vector (prior distribution) and connected to a MLP to generate molecular graphs. This method demonstrated comparable, or even improved, performance across various metrics, including uniqueness, validity, diversity, drug-likeness, as well as synthesizability and solubility, highlighting the potential of hybrid quantum-classical architectures in enhancing generative models. However, the study did not provide a detailed comparison of the total number of parameters used, limiting claims about its expressive power.

While QGANs show some promising results in molecular generation, particularly in areas like enhanced drug properties and the potential for better expressive power in discriminators, significant challenges persist. The expressive power derived from full quantum discriminators may come at the cost of compromising other crucial metrics in molecular generation. Additionally, when hybrid networks achieve improvements in drug properties and other metrics, the exact contribution of expressive power offered by the quantum component becomes less clear. Thus, an outstanding challenge is to achieve enhanced expressive power without sacrificing performance across other critical metrics.

3.3. Looking Ahead: Quantum Transformers

Much of the AI revolution is due to the transformer architecture introduced in the "Attention is All You Need" paper out of Google DeepMind.[124] This architecture was originally developed for language translation, and consisted of encoder and decoder components which are connected via a cross-attention mechanism. The encoder alone is useful for learning a context-rich representation for a given input sequence by masking some of the sequence and learning to predict the masked parts. The decoder is useful for generating new sequences by learning to predict the next parts of some sequence given a context. Within the realm of biochemistry and drug discovery, transformer encoders have been developed to extract feature vectors from SMILES strings to be used for downstream predictive tasks, and transformer decoders have been used to generate SMILES strings with prespecified characteristics.[125−129] The fundamental capabilities of the transformer architecture are due to the self-attention mechanism where query, key, and value vectors are computed for each input token (e.g., a subword in text or a character in a SMILES strings), attention scores are derived via a scaled dot product of query and key vectors, and softmax normalizes these scores to obtain weights that modulate the aggregation of the value vector, effectively capturing the magnitude with which each token will attend to every other token in the sequence. The self-attention mechanism is often executed multiple times in parallel through what is referred to as multi-head attention.

The overwhelming success of the classical transformer in ML has naturally piqued the interest of QML researchers. Most implementations of quantum transformers have been adapted as Vision Transformers (ViTs) rather than for Natural Language Processing (NLP).[130−133] While classical ViT models have been utilized in predictive tasks in chemistry and biophysics,[134−136] the primary role of transformers in the context of drug discovery has remained with transformer-based generative models.

Quantum-based attention for generative pretrained transformers (GPTs) are still in their infancy, and while many of the results presented thus far have been largely theoretical, the field is rapidly advancing. In 2022, DiSipio et al.[138] discuss the beginnings of quantum NLP, and highlighted that the underlying mathematical operations of the transformer's self-attention mechanism all have implementable quantum formulations. In 2023, both Gao et al.[139] and Li et al.[140] show implementations for a quantum self-attention mechanism. Guo et al.[141] and Liao et al.[142] independently present full end-to-end GPT quantum algorithms. Notably, the work from Guo et al. presents a rigorous complexity analysis and demonstrates a theoretical quantum advantage for numerous normalization operations throughout the architecture.

The success of the transformer architecture can be attributed to its self-attention mechanism, which has motivated researchers to explore quantum analogs for potential performance gains. The traditional classical self-attention mechanism scales $O(n^2 d)$ for sequence length n and embedding dimension d. This arises from multiplying the query and key matrices QK^T as well as applying the resulting pairwise attention matrix to the value matrix V. Unfortunately, the current quantum implementations that potentially achieve a complexity advantage rely on algorithms that are unsuitable for quantum hardware in the NISQ-era. Smaldone et al.[137] presented a quantum-classical self-attention mechanism in a transformer for molecular generation that reduces the complexity of the attention score calculation to $O(n^2 \log d)$ without relying on unrealistic assumptions of state preparation or matrix sparsity. Their model performs similarly to the equivalent classical analog and is able to generate molecules conditioned on physicochemical properties. The workflow of
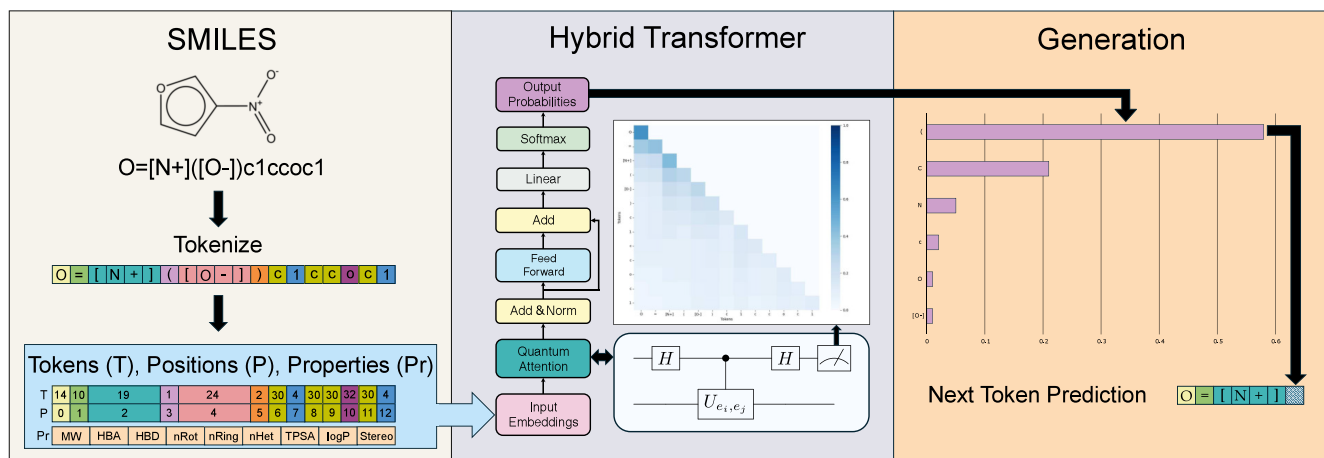
Figure 11. Overall workflow of the quantum−classical hybrid transformer model for molecular generation, proposed by Smaldone et al.[137] Attention scores are generated with quantum circuits and used inside a classical transformer decoder model to generate SMILES strings.

their model is outlined in Figure 11 and is available as an application tutorial on the CUDA-Q documentation page.[143]

Some classical techniques try to avoid the $O(n^2 d)$ complexity of scaled dot-product attention through alternative methods.[144−146] Similarly - instead of scaled dot-product attention - Quantinuum released an open-source model, Quixer,[147] that proposes a quantum analog of the k-skip-n-gram NLP technique for learning relationships between tokens. Quixer mixes embedded tokens by using linear combination of unitaries (LCU),[148] and further computes skip-bigrams between words using quantum singular value transformations (QSVT).[149] Quixer's model scales $O(\log(nd))$ in the number of qubits and $O(n \log d)$ in the number of gates. In contemporary transformer applications, sequence length n is often much larger than the embedding d which makes the logarithmic scaling in the number of qubits with respect to n a promising look into the future of transformers.

While the present models largely do not claim an explicit complexity quantum advantage, this should not dissuade future researchers from utilizing the available methods for their pharmacological applications. The nascency of the field presents an opportunity for researchers in academia and pharmaceutical industry alike to hunt for advantages elsewhere. Presently with few current works in the literature applying quantum transformers to chemical, biological, or pharmaceutical tasks, this should inspire researchers to investigate if these quantum transformers can learn hidden features inaccessible to classical learning styles as indicated by Li et al.[140] In this event, combining features extracted from both a quantum transformer component and a classical transformer component could present a model with a richer understanding of chemical and biological function, leading to exciting downstream effects in drug design. Beyond quantum transformer architectures, hybrid generative models that incorporate quantum components in other neural architectures have demonstrated early experimental success. Vakili et al.[150] demonstrated that a hybrid QCBM− LSTM model can generate valid SMILES strings for KRAS inhibitors, two of which showed micromolar activity in cell-based assays. This highlights the practical promise of quantum generative models even outside transformer architectures.

4. POTENTIAL OF BOSONIC QUANTUM PROCESSORS FOR QUANTUM MACHINE LEARNING

Nearly all gate-based QNNs are performed on qubit-based quantum processors, however the potential to extend these models beyond this platform remains relatively unexplored. There has been some recent efforts building quantum machine learning models with the multilevel discrete[151,152] and continuous[153,154] representation of a quantum harmonic oscillator, also known as a qumode. In this section, we discuss the potential of a new paradigm of quantum computing consisting of qubits coupled with qumodes for quantum machine learning.

### 4.1. Basics of Bosonic Quantum Computing

Hybrid qubit-qumode devices[155−157] have the potential to augment the power of qubit architectures by allowing for data encoding in a much larger Hilbert space with hardware efficiency. For example, qumodes could amplify the impact of VQCs in applications to QML beyond the implementations discussed in Section 1.3.4.

An arbitrary qumode state $|\psi\rangle$, corresponding to the state of a quantum harmonic oscillator, can be expanded in its Fock basis state representation as a superposition of a countably infinite set of orthonormal photon-number states $\{|n\rangle\}$. In practice, however, the expansion is truncated with a Fock cutoff d, as follows:

$$|\psi\rangle = \sum_{n=0}^{d-1} c_n |n\rangle \tag{11}$$

According to eq 11, a qumode generalizes the two-level qubit into a d-level state (also known as *qudit*[158]), thus offering an expanded basis set. Beyond the expanded basis, the hardware of bosonic modes are relatively weakly affected by amplitude damping errors,[157] which leads to extended lifetimes, and the possibility of implementing efficient error correction codes.[159−161]

Recent advancements in bosonic quantum hardware have significantly progressed, enhancing the implementation of qumodes across various architectures[155] However, achieving universal quantum computing remains challenging when relying solely on native qumode operations. This is where hybrid qubit-qumode hardware have made notable strides. For example, in the circuit quantum electrodynamics (cQED) framework, a
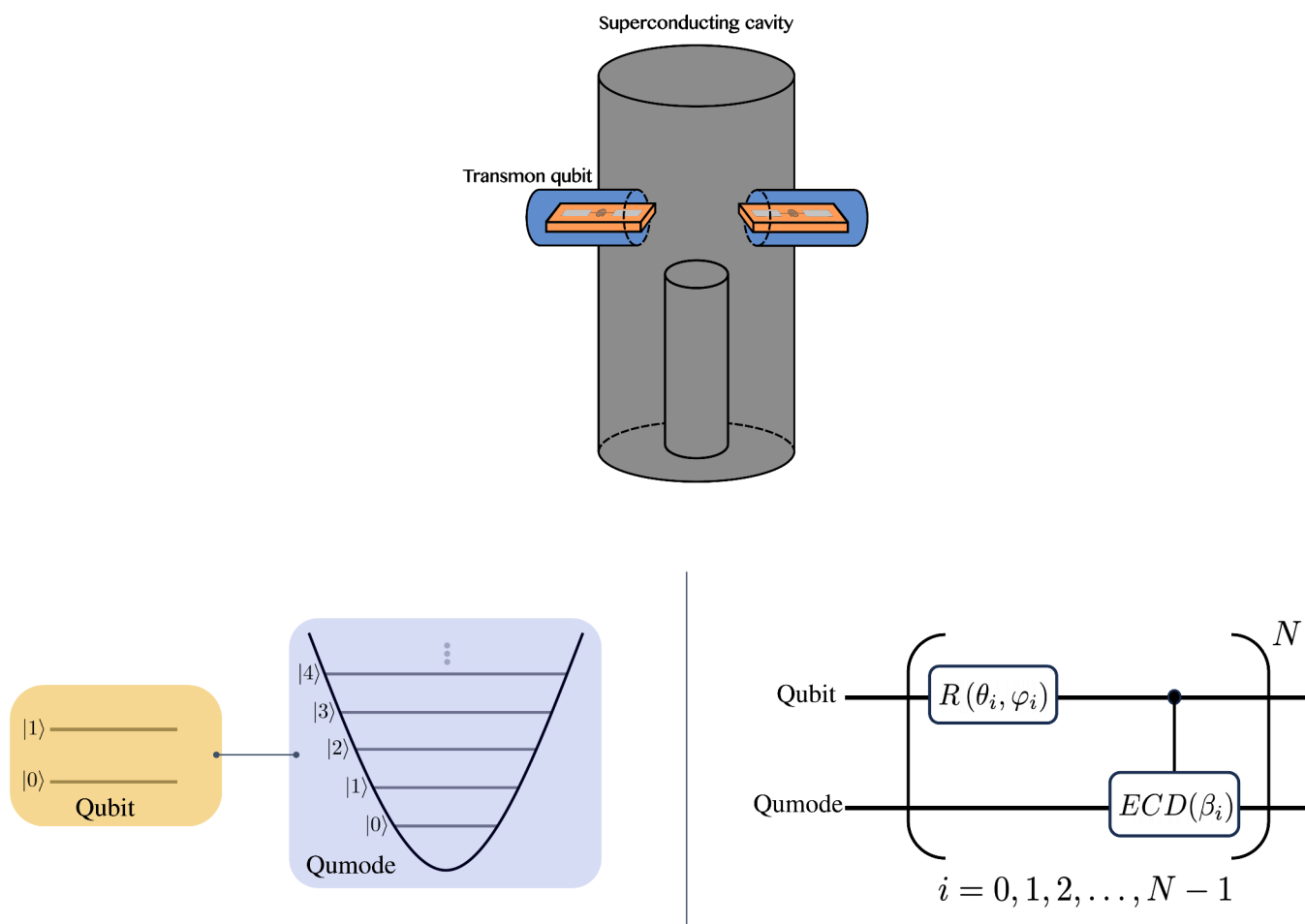
Figure 12. (Top) Schematic representation of a superconducting cavity resonator coupled to a qubit transmon. (Bottom Left) Visual schematic of a two-level qubit (transmon) coupled to a multilevel oscillator (superconducting cavity). (Bottom Right) Example of a qubit-qumode circuit that allows for universal control, where the qubit rotation gate $R(\theta, \varphi)$ is defined in eq 13 and the echoed-conditional displacement (ECD) gate is defined in eq 14.

microwave cavity coupled to a transmon qubit has demonstrated considerable potential (Figure 12).[162] The interplay between qubit and qumode dynamics enables the development of hybrid qubit-oscillator gate sets, which are efficient in achieving universality.[157,163,164]

Additionally, photonic processors offer programmability that facilitates the simulation of bosonic systems.[165,166] In contrast, qubit based hardware is inherently suited for simulating Fermions through the Jordan-Wigner transformation.[167−169] Therefore, a hybrid qubit-qumode architecture is particularly attractive, since qubit-only or bosonic-only native gates might require deeper circuits for specific applications, although methods have been developed to represent bosons using qubits and vice versa.[170−172]

Incorporating efficient bosonic representation could enable practical simulations beyond the capabilities of conventional qubit-based quantum computers, as already shown for example in calculations of vibrational spectra of small polyatomic molecules.[173] This can be achieved with photonic quantum processors,[174−176] cQED devices,[173] and even hybrid qudit-boson simulators.[177]

Another unique feature of qumodes is that they can also be represented by continuous variable (CV) bases corresponding to position and momentum operators of a quantum harmonic oscillator,[178] with no counterpart for qubits. For example, in the position representation, an arbitrary qumode state $|\psi\rangle$ can be expressed, as follows:

$$|\psi\rangle = \int_{-\infty}^{+\infty} dx \psi(x)|x\rangle \tag{12}$$

where $\psi(x) = \langle x|\psi\rangle$ is the oscillator complex valued amplitude at x. As state and process tomography are necessary to calibrate and model hardware noise, hybrid processors offer simple protocols to determine the Wigner function of qumode states,[179−183] allowing further development of abstract machine models.[157]

4.2. Potential Advantages of Qubit-Qumode Circuits in QML

Hybrid qubit-qumode circuits, such as the one shown in Figure 12, can be used to approximate any unitary transformation of the qubit-qumode system. An attractive choice of a circuit structure[164] applies repeating modules of a qubit rotation gate,

$$R(\theta, \varphi) = e^{-i\theta/2(\sigma_x \cos \varphi + \sigma_y \sin \varphi)} \tag{13}$$

where σ_x and σ_y are Pauli X and Y matrices, followed by an echoed conditional displacement (ECD) gate,

$$ECD(\beta) = |1\rangle\langle 0|\otimes D(\beta/2) + |0\rangle\langle 1| \otimes D(-\beta/2) \tag{14a}$$

$$D(\alpha) = e^{\alpha \hat{a}^\dagger - \alpha^* \hat{a}} \tag{14b}$$
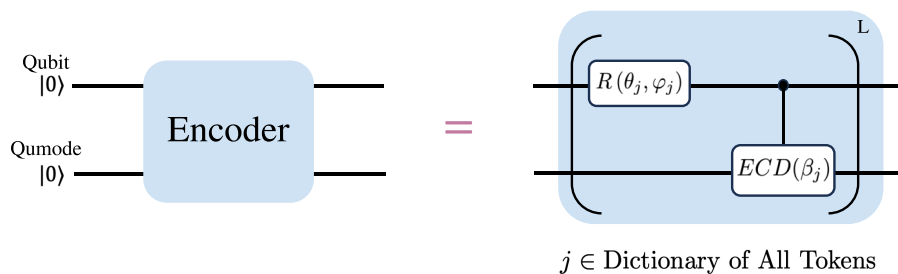
(a) General encoder architecture with the R-ECD circuit.
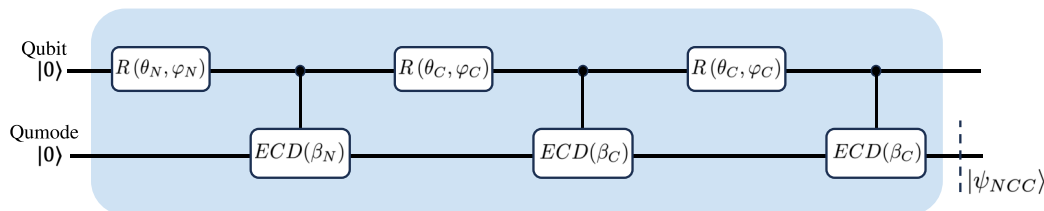
(b) An example showing the encoding circuit of NCC, the SMILES representation of ethylamine.

Figure 13. Circuit diagrams for the R-ECD encoding method. (a) General encoder architecture with a set of θ, φ, and β assigned to each unique token in the dictionary, encoding a string of length L. (b) Specific circuit for the encoding of ethylamine, where the R-ECD block corresponding to N is applied, followed by two blocks of R-ECD corresponding to C.

where $\hat{a}^{\dagger}$ and $\hat{a}$ are bosonic creation and annihilation operators, respectively.

The qumode Hilbert space may offer advantages over qubit-based registers since it allows for more efficient representations for predictive and generative tasks.[151,152,158] For example, a system with 8 qubits involves a Hilbert space with $2^8 = 256$ basis states, which could be represented by two qumodes with control over $d = 16$ Fock states (each mode offering a Hilbert space equivalent to the space expanded by 4 qubits).[173] Therefore, encoding of complex molecular information that typically requires many qubits would potentially benefit from hybrid qubit-qumode circuits, as these systems offer significant hardware efficiency compared to qubit circuits with a similarly sized Hilbert space. Additionally, the circuits of qumode states can be based on efficient or shallow circuits that bypass the need of deep circuits based on elementary logic gates.[158,173]

4.3. Encoding Classical Information in Qubit-Qumode Circuits

We introduce two possible methods for encoding classical (or quantum) data in the form of quantum states of a qumode coupled to a qubit. Similar to amplitude encoding for qubit systems, we can adapt the method discussed in Section 1.3.3 for qumodes. We simply modify eq 9 to encode a vector of length d into the amplitudes α_k of a d-level qudit, as follows:

$$x \rightarrow U_x|0\rangle_d = |x\rangle = \sum_{k=0}^{d-1} \alpha_k|k\rangle \tag{15}$$

where U_x is the unitary transformation that encodes the data provided by the amplitudes in the form of the qumode state $|x\rangle$. Here, $|0\rangle_d$ is the initial vacuum state of the qumode corresponding to an empty cavity without photons. Preparing U_x requires parameterization of a circuit with universal qumode control such as the one with blocks of a qubit rotation gate

followed by an ECD gate (R-ECD circuit) outlined in Figure 13 and Section 4.2.[164] Other circuit structures are also available which can be parametrized to encode any arbitrary dataset by amplitude encoding in the form of a qumode state.[157,163]

Another practical method for encoding molecular information (e.g., a list of tokens defining a specific molecule) in a qubit-qumode state involves a generalization of phase encoding. A dictionary is used to correlate the input tokens to the values of parameters used in the circuit. When using the R-ECD circuit structure defined by eq 13 and eq 14, specific parameters θ, φ, and β of each module are assigned to each specific token of the input. So, the sequence of tokens defining the input molecule is encoded as a specific parametrization of the R-ECD ansatz. This generalization of phase encoding is not limited to molecular encodings, or the specific choice of circuit structure, and can be applied for a wide range of studies, including the encoding of tokens in qubit-qumode devices.

One technical challenge of these generalized phase encoding methods is that the encoded states for different states could partially overlap with each other, unless an orthogonalization procedure is enforced. The partial overlap could lead to some level of confusion due to ambiguity of the encoding. To address this challenge, the parameters assigned to each token can be made learnable parameters such that the encodings are optimized to be as different as possible.

5. EFFICIENT CIRCUIT SIMULATION FOR NEAR-TERM RESEARCH AND COMPUTING UNIT INTEGRATION

Despite recent progress, current Quantum Processing Units (QPUs) remain limited in size and computational capabilities due to noise and scaling challenges, which impedes progress in algorithmic research. To address this challenge, circuit simulation techniques are meeting the critical need to advance research boundaries. An open-source platform for seamlessly
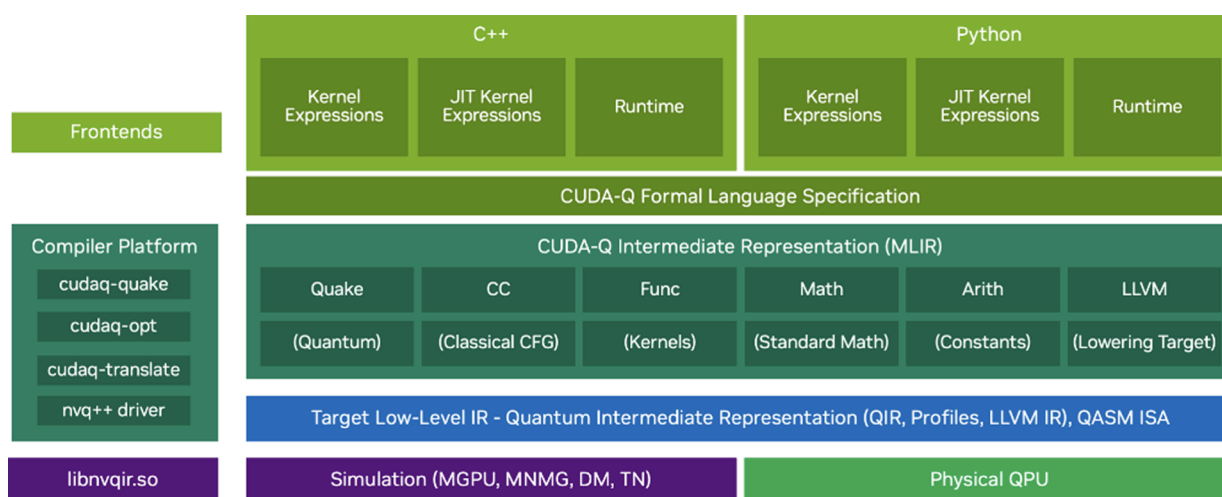
Figure 14. CUDA-Q software stack. CUDA-Q builds off of a core MLIR-based intermediate representation for representing hybrid quantum−classical code with control flow. The compiler workflow lowers to target specific code for backend QPU execution. This state-of-the-art compiler stack is exposed to programmers via a library-based C++ language extension and a JIT compiled language representation in Python.

integrating and programming QPUs, GPUs, and CPUs within a single system is provided by NVIDIA's CUDA-Q[184] (see Figure 14). Various quantum computing frameworks, including Cirq, Qiskit, TorchQuantum, and Pennylane,[185−188] utilize GPU-accelerated simulation through the cuQuantum libraries[189] featured in the CUDA-Q simulation backend. By employing the CUDA-Q compiler alongside cuQuantum APIs as simulation backends, users can achieve near-optimal GPU acceleration and exceptional performance at scale.

In this section, we demonstrate how CUDA-Q can be utilized to accelerate and scale up quantum circuit simulations. This is applicable to various fields including quantum machine learning for chemistry. We use CUDA-Q v0.8 for simulations and show how the compute resources scale with the size of the simulation. Examples used to reproduce the results presented in this section are available in GitHub.[190]

5.1. Circuit Simulator with State Vector and GPU Acceleration

Desktop CPUs can handle the simulation of small numbers of qubits; for instance, on a laptop with at least 8 GB of memory, noiseless simulations can reach up to 24 qubits, while noisy simulations are feasible with up to 18 qubits.[191] However, as the memory required to store the full state vector grows exponentially with the number of qubits, GPUs are needed for larger simulations. For example, an NVIDIA DGX A100 can simulate 20 qubits with exceptional speed, while a CPU would be very slow at performing the state vector simulation of similar size, as shown in Figure 15.

Figure 15 compares the logarithmic ($\log_{10}$) runtime for computing the expectation value of a quantum circuit similar to the one shown in Figure 16 using a state vector simulator on one CPU (AMD EPYC 7742 64-Core Processor) as compared to one NVIDIA A100 GPU. The quantum circuit in Figure 16 is a standard parametrized quantum circuit employed in QNNs for different applications such as QGANs applied for drug discovery and molecular generation.[120,192] Specifically, Figure 15 shows the comparison of the runtime on a single CPU as compared to a single GPU for data points of 10,000 (i.e., 10,000 expectation values) as a function of the number of qubits. It is shown that the runtime on the CPU significantly increases as we increase the number of qubits while increases only modestly on the NVIDIA
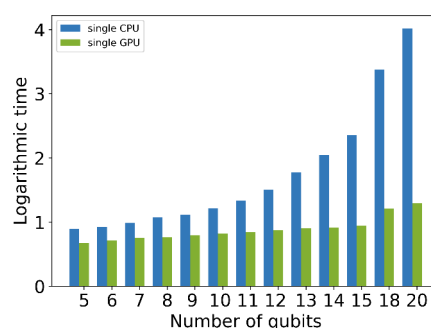
Figure 15. Logarithmic ($\log_{10}$) execution time for one "observe" call (i.e., measure the observable operator applied to the state vector/ wavefunction, also known as expectation value) for each data point (10,000 data points; i.e., in total there are 10,000 expectation values) on a single CPU versus a single GPU for a one layer of the parametrized quantum circuit (PQC) similar to the PQC shown in Figure 16. The code used to generate the data in this figure is available on GitHub.[193]
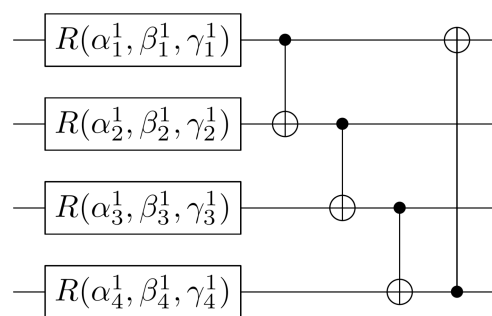
Figure 16. An example of a parametrized quantum circuit employed in QNNs.

A100 GPU. For example, for the 18 qubit circuit, there is a ≈150× speed up on the single GPU. When increasing the number of qubits to 20, the speed up is ≈530×. These results emphasize the need for GPU supercomputing to accelerate simulations of quantum algorithms and applications to research and development. Such simulations would enable studies beyond small-scale proof-of-concept calculations in application studies to real-world scenarios.

Another example demonstrating the capabilities of CUDA-Q are the implementations of the VQE-Quantum approximate optimization algorithm (VQE-QAOA) algorithm for simulations of molecular docking[194] and protein folding.[195] For example, the VQE-QAOA algorithm has been applied to find the optimal configuration of a ligand bound to a protein, implementing the molecular docking simulation as a weighted maximum clique problem.[194] Simulations were performed with up to 12 qubits. The CUDA-Q tutorial[196] reproduces the results using DC-QAOA and compares the CPU and GPU runtimes (Table 1). For 12 qubits, a 16.6× speed up is observed on a single GPU when compared to a single CPU.

Table 1. Execution Time of One "Observe" Call (i.e., Expectation Value) Using DC-QAOA Ansatz[a]

qubits	CPU time (s)	GPU time (s)
6	0.322	0.160
8	1.398	0.390
12	6.863	0.412

[a]Simulations were run with 3, 8, and 13 layers for 6, 8, and 12 qubits, respectively.

CUDA-Q also allows for gate fusion to enhance state vector simulations with deep circuits, thereby improving performance.[197,198] Gate fusion is an optimization technique that combines consecutive quantum gates into a single gate (see Figure 17), which reduces the overall computational cost and
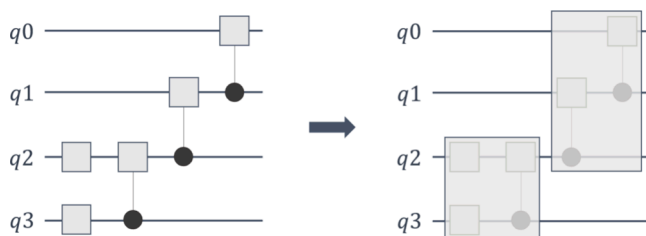
Figure 17. Gate fusion fuses multiple gates into one larger gate.

increases the circuit efficiency.[199,200] By grouping small gate matrices into a single multi-qubit gate, the fused gate can be applied in one operation, eliminating the need for multiple applications of small gate matrices. This optimization reduces memory bandwidth usage, as applying a gate matrix G to a state $|\Psi\rangle = G|\phi\rangle$ involves reading and writing the state vector. The memory bandwidth (in bytes, including reads and writes) can be calculated, as follows:

$$\text{memory bandwidth} = 2 \times \frac{\text{svSizeBytes}}{2^{\text{ncontrols}}} \quad (16)$$

where 'svSizeBytes' represents the state vector size in bytes and 'ncontrols' is the number of control qubits (e.g., a CNOT gate has one control). Applying two gates, $G_2 G_1 |\phi\rangle$, requires two reads and two writes, whereas applying the combined gate $(G_1 G_2)|\phi\rangle$ only needs one read and one write.

Gate fusion can significantly enhance simulation performance for deep circuits which are crucial for quantum applications in chemistry. A notable example is the unitary coupled cluster singles and doubles (UCCSD) ansatz, widely used in quantum computational chemistry calculations. For instance, when running a single observation call (i.e., computing one expectation value) for the C_2H_4 molecule using the UCCSD ansatz

with 24 qubits on an NVIDIA A100, the total elapsed time is 30.02 s without gate fusion. In contrast, with gate fusion, the elapsed time is reduced to 12.44 s, demonstrating a 2.4× speedup. The code for this comparison is available on GitHub.[201]

5.2. Parallelization and Scaling

NVIDIA's CUDA-Q platform provides a clear overview of the various devices in a quantum-classical compute node, including GPUs, CPUs, and QPUs. Researchers and application developers can work with a diverse array of these devices. Although the integration of multiple QPUs into a single supercomputer is still in progress, the current availability of GPU-based circuit simulators on NVIDIA multi-GPU architectures enables the programming of multi-QPU systems today.

5.2.1. Enabling Multi-QPU Workflows. CUDA-Q enables application developers to design workflows for multi-QPU architectures that utilize multiple GPUs. This can be achieved using either the 'NVIDIA-mQPU' backend[202] or the 'remote-mQPU' backend, which we discuss further in Section 5.2.3. The 'NVIDIA-mQPU' backend simulates a QPU for each available NVIDIA GPU on the system, allowing researchers to run quantum circuits in parallel and thus accelerating simulations. This capability is crucial for applications such as quantum machine learning algorithms. For example, in training QNNs, computing expectation values for numerous data points is often required to train the model. By batching these data points, they can be processed simultaneously across multiple GPUs.

Figure 18 compares results obtained by running a QNN workflow running on a single GPU versus those obtain by
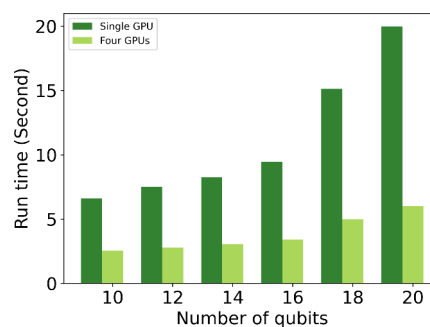
Figure 18. Execution time for an "observe" call (expectation value) made for 10,000 data-points for a one layer of the parametrized quantum circuit similar to the one shown in Figure 16. All simulations were run on an NVIDIA DGX A100 GPU device. For a single GPU, the total 10,000 data-points are dealt within a single GPU (i.e, 10,000 "observe" calls in sequence on a single GPU). For the four GPU case, the data points are split into four batches, each containing 2500 data-points (i.e., 2500 "observe" calls on each GPU).

distributing the workflow across four GPUs (in a single CPU node with 4 GPUs). The code for this comparison is available in GitHub.[203] For an application using 20 qubits, we find that the runtime with four-GPUs is approximately 3.3 times faster than using a single GPU. Although parallelization requires some synchronization and communication across the GPUs, which slightly limits the speedup to being less than 4x, this still demonstrates strong scaling performance. It highlights the efficient utilization of GPU resources when available.

Another example of a commonly used application primitive that benefits from parallelization using the 'NVIDIA-mQPU' backend is the Hadamard test. The Hadamard test is crucial for

computing the overlap between different states, as necessary to evaluate correlation functions, and expectation values which involve calculating $O(n^2)$ independent circuits in a wide range of applications, including prediction of drug toxicity[30] and determining the electronic ground state energy of molecules.[204,205] By leveraging parallelism, these $O(n^2)$ circuits can be efficiently executed across as many QPUs — whether physical or simulated— as are available.

5.2.2. Scaling Circuit Simulations with Multi-GPUs. The conventional state-vector simulation method requires storing 2^n complex amplitudes in memory when simulating n qubits. This results in exponentially increasing memory requirements for circuits with a large number of qubits. If each complex amplitude requires 8 bytes of memory, the total memory required for an n qubit quantum state is 8 bytes $\times 2^n$. For instance, with $n = 30$ qubits, the memory requirement is approximately 8 GB, while for $n = 40$ qubits, it jumps to about 8700 GB. CUDA-Q addresses this challenge by enabling the distribution of state-vector simulation across multiple GPUs via the 'NVIDIA-mGPU' backend.[202] For detailed information of the algorithm, see Sec. II−C in ref 189. Additionally, examples of using the 'NVIDIA-mGPU' backend are available on GitHub.[206]

The 'NVIDIA-mGPU' backend combines the memory of multiple GPUs within a single DGX compute node and across multiple DGX compute nodes in a cluster. DGX compute nodes, part of NVIDIA's DGX platform, are high-performance computing (HPC) servers, specifically designed for HPC and artificial intelligence workloads, leveraging NVIDIA GPUs to accelerate intensive computations. By pooling GPU memory, this backend allows for greater scalability and eliminate the memory limitations of individual GPUs. Consequently, the capacity to simulate larger numbers of qubits is constrained only by the available GPU resources in the system.

Intra-node NVlink[207] is a powerful tool for large-scale simulations. An NVLink-based system enables greater performance optimization by providing direct access to the full NVLink feature set, bypassing the CUDA-Aware MPI layer. CUDA-Q v0.8 introduces an improved algorithm for intra-node NVLink, leveraging CUDA Peer-to-peer (P2P) communication.[208] Table 2 compares the performance of CUDA-Q 0.7 (using CUDA-

Table 2. Quantum Algorithm Performance Improvements Enabled by NVLink Optimizations[a]

algorithm	qubits	speed up (in simulation time)
H-Gates	35	2.47
QAOA	32	1.28
QFT	35	1.13
UCCSD	32	1.30

[a]The speed up in time is reported for CUDA-Q v0.8 with CUDA P2P compared to CUDA-Q v0.7 with CUDA-aware MPI. Simulation times accounts for a single VQE execution. H-Gates refers to applying one Hadamard gate per qubit. All simulations were run on a DGX H100 device.

aware MPI) and CUDA-Q 0.8 (using P2P) on an NVlink-enabled DGX H100 system. In these simulations, the state vector was distributed across a single node with 8 GPUs. Four large-scale quantum algorithms were benchmarked using the MPI and P2P API in CUDA Runtime. As shown in Table 2, CUDA-Q v0.8 with P2P achieves up to 2.5x speedup for H-gates compared to CUDA-Q v0.7 with CUDA-aware MPI.

Additionally, developers can now use CUDA-Q to fully exploit the performance of the NVIDIA GH200 AI superchip,[209] further enhancing the capabilities of quantum simulation in CUDA-Q. With a combined CPU and GPU memory of 1.2TB, the GH200 AI superchip significantly accelerates quantum simulations, reducing the number of required nodes by 75%. This reduction is particularly crucial for quantum applications research, which is often constrained by memory limitations.

Table 3 compares the performance of the GH200 superchip and the DGX H100 for running a quantum algorithm using a

Table 3. Comparison of Performance of the GH200 Superchip and the DGX H100 for Running a Quantum Algorithm Using a State Vector Simulation[a]

algorithm	qubits	speed up (in simulation time)
H-Gate	37	4.10
QFT	37	2.58

[a]Simulations were run with 37 qubits, and the state vector was distributed across 8 GPUs in a single node in the DGX H100 and four nodes in the GH200 superchip. H-Gates refers to applying one Hadamard gate per qubit.

state vector simulator. In this comparison, we employed 37 qubits and distributed the state vector across 8 GPUs on four nodes in the GH200 superchip and a single node in the DGX-H100. Our findings show that the GH200 superchip achieves up to 2.58x speed up for the quantum Fourier transform (QFT) and a 4x speed up for H-Gates.

5.2.3. Combining Backends For Large Scale Simulations. Quantum circuit simulations can be scaled up using the 'NVIDIA-mGPU' backend and parallelized with the 'NVIDIA-mQPU' backend, as described in the previous section. CUDA-Q provides the capability to combine both backends through the 'remote-mQPU' backend, enabling large-scale simulations (Figure 19). In this configuration, multiple GPUs comprise a virtual QPU. A practical example of using 'remote-mQPU' for QNNs is available on GitHub.[210]
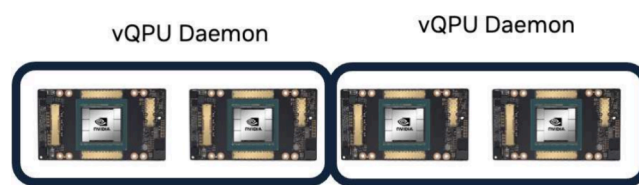
Figure 19. An example of multi-QPU backend with multi-GPU. Here, there are two virtual QPUs (vQPU), and each virtual QPU is made of two GPUs.

5.3. Quantum Circuit Simulator With Tensor Networks

The state vector method is effective for simulating deep quantum circuits, however, it becomes impractical for simulations of circuits with large numbers of qubits due to the exponential growth in computational resources required — making them unmanageable even on the most powerful supercomputers available today. As an alternative, the tensor network method represents the quantum state of N qubits through a series of tensor contractions (see Figure 20). This approach allows quantum circuit simulators to efficiently handle circuits with many qubits.

Tensors (see Figure 21) generalize scalars (0D), vectors (1D), and matrices (2D) to an arbitrary number of dimensions.
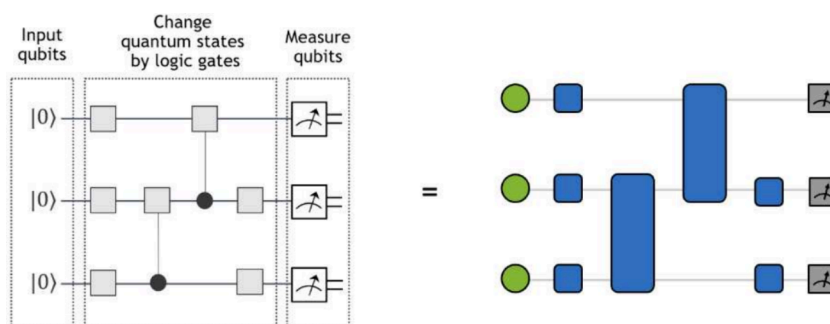
Figure 20. Single-qubit and two-qubit gates translate to rank-2 and rank-4 tensors, respectively. The initial single-qubit states $|0\rangle$ and single-qubit measurement operations can be viewed as vectors (projectors) of size 2. The contraction of the tensor network on the right yields the wavefunction amplitude of the quantum circuit on the left for a particular basis state.
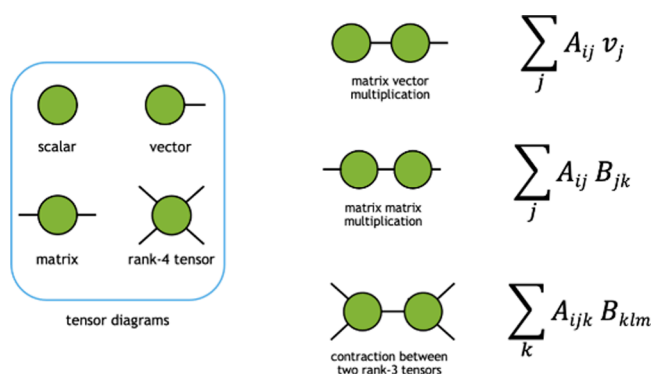
Figure 21. Tensor diagram (left) and example of matrix-like contractions (right).

A tensor network consists of a set of tensors connected together through tensor contractions to form an output tensor. In Einstein summation notation, a tensor contraction involves summing over pairs of repeated indices (see Figure 21). For example, a rank-four tensor M can be formed by contracting two rank-three tensors C and B, as follows: $M_{ijlm} = \sum_k C_{ijk} B_{klm}$. Here, the contraction is performed by summing over the shared index k. Identifying an efficient contraction sequence is essential for minimizing the computational cost of the tensor networks.[189,211] The contractions between the constituent tensors define the topology of the network.[212,213]

CUDA-Q offers two GPU-accelerated tensor network backends: 'tensornet' and 'tensornet-mps'.[214] For a detailed explanation of tensor network algorithms and their performance, see refs[189, 215].

The 'tensornet' backend represents quantum states and circuits as tensor networks without any approximations. It computes measurement samples and expectation values through tensor network contractions.[216] This backend supports the distribution of tensor operations across multiple nodes and GPUs, enabling efficient evaluation and simulation of quantum circuits.

The 'tensornet-mps' backend utilizes the matrix product state (MPS) representation of the state vector, exploiting low-rank approximations of the tensor network through decomposition techniques such as QR and singular value decomposition. As an approximate simulator, it allows truncation of the number of singular values to keep the MPS size manageable. The 'tensornet-mps' backend supports only single-GPU simulations. Its approximate nature enables it to handle a large number of

qubits for certain classes of quantum circuits while maintaining a relatively low memory footprint.

As hybrid quantum-classical workflows and scalable simulation frameworks continue to evolve, tools like CUDA-Q provide a vital foundation for real-world deployment of quantum machine learning models. Yet despite these technological advancements, key challenges remain across quantum hardware, algorithms, and scalability. The next section outlines these broader limitations and critical areas of research required to unlock the full potential of quantum machine learning.

6. CHALLENGES AND OUTLOOK

6.1. Hardware

When evaluating the physical implementation of quantum computers, it is essential to consider the widely recognized five criteria proposed by DiVincenzo:[217]

(1) **Scalable physical systems with well-characterized qubits**: The system should contain qubits that are not only distinguishable from each other but also manipulable either individually or collectively. This requirement ensures that qubits can be controlled with precision for complex quantum computations.

(2) **Ability to initialize qubits to a simple, known state**: Typically referred to as a "fiducial" state, this criterion emphasizes the importance of preparing qubits in a well-defined, simple initial state, such as the zero state. This initialization process is crucial for the reliability and predictability of subsequent quantum operations.

(3) **Decoherence times much longer than gate operation times**: Quantum systems must exhibit long coherence times relative to the time it takes to perform quantum gate operations. This ensures that quantum information is preserved long enough to complete computations before being lost to decoherence.

(4) **A universal set of quantum gates**: The hardware must support a set of quantum gates capable of performing any quantum computation. This typically includes a variety of single-qubit gates along with a two-qubit entangling gate, such as the CNOT gate, enabling the construction of complex quantum circuits.

(5) **Qubit-specific measurement capability**: The system should allow for accurate measurement of individual qubits' states after computation. This criterion is essential for retrieving the final output of quantum computations.

Gate-based quantum computer designs generally adhere to these criteria, yet achieving the most optimal performance

remains a significant challenge. For QML, these hardware requirements introduce additional complexities.

QNNs often claim superior expressive power compared to classical neural networks. This advantage typically necessitates high connectivity among qubits, aligning with the need for well-characterized and scalable qubit systems described in Criterion (1). Ensuring such connectivity while maintaining system scalability and qubit fidelity is a nontrivial challenge in current hardware implementations.

Moreover, QML algorithms frequently utilize amplitude encoding, a technique that effectively encodes classical data into quantum states. This approach, however, is equivalent to preparing arbitrary quantum states, which goes beyond the simpler requirement of initializing qubits to a fiducial state as outlined in Criterion (2). Consequently, specific QML applications may require either modifications to the existing hardware criteria or the development of more advanced state preparation algorithms to achieve the desired outcomes.

Finally, when the final output of a QNN necessitates precise amplitude measurements of quantum states, the hardware must extend the measurement capabilities described in Criterion (5). Specifically, accurate and scalable quantum state tomography becomes essential to extract the necessary information from the quantum system. This represents another area where current quantum hardware may need further refinement to fully support the demands of QML.

6.2. Algorithms

Loading classical data into a quantum state is the often first step in a QNN, and is a step that will largely dictate the performance of the model, the potential advantages the quantum model possess over the classical, and the model's quantum resource complexity. For example, angle encoding (eq 7) is inexpensive to implement on quantum hardware, but it is difficult to extract a complexity advantage. Alternatively, amplitude encoding easily enables a complexity advantage due to the exponentially larger Hilbert space in which information can be stored, but at the expense of quantum resources to prepare such quantum state. In particular, state preparation techniques to prepare arbitrary state vectors scale exponentially with respect to the number of two-qubit gates required to prepare the quantum state.[218,219] While this problem of state preparation may be daunting, promising data encoding workarounds are being developed. Data reuploading is a strategy that allows circuits to handle more complex data by breaking the information into smaller quantum circuits.[220] Shin et al. presents a method for QML that utilizes a quantum Fourier-featured linear model to exponentially encode data in a hardware efficient manner.[221] The authors demonstrate the method achieves high expressivity and exhibits better learning performance compared to data reuploading, notably when learning the potential energy surface of ethanol. These promising directions should motivate QML researchers to identify tasks where their input data exists in or can be transformed into a form that is known to be efficiently prepared[222−224] or where the exact input vector does not need to be known a priori and is learned through training. Furthermore, as QPUs evolve to include more qubits and improved interconnected topologies, state preparation algorithms that utilize ancillary qubits will help address the challenges associated with poor decoherence times and prolonged gate execution times, as they are capable of preparing arbitrary states with shallower depths.[225]

Similar to how classical ML architectures have the potential to suffer from vanishing gradients, VQCs have the potential to suffer from barren plateaus. Barren plateaus occur when the loss differences used to compute quantum weight gradients exponentially vanish with the size of the system. Larocca et al. presents a comprehensive review where the authors outline strategies to avoid and mitigate the problem of barren plateaus.[226] Some of these methods that an aspiring QML researcher should be aware of are shallow circuits and clever weight initialization strategies. Notably, Ragone et al.[227] present a theorem to determine exactly if any noiseless quantum circuit will exhibit barren plateaus regardless of the circuit's structure. The authors note that among the implications of their work, it is possible to design variational quantum circuits that exhibit high entanglement and use nonlocal measurements while still avoiding barren plateaus, going against conventional wisdom. This lifts restrictions and gives researchers a much deeper insight into the trainability of their circuits. On the other hand, researchers should seek possible ways to obtain structured optimization landscapes and linear/superlinear loss convergence with inspiration from quantum chemistry simulations to avoid barren plateaus.[228,229] Recent evidence indicates that models capable of avoiding barren plateaus can often be simulated classically.[230,231] This does not mean that parametrized quantum circuits (PQCs) lack value; rather, it invites us to reconsider the unique advantages they might provide. For instance, there may still be a trade-off between accessing a broader Hilbert space and maintaining trainability.

In addition to the difficulties of determining quantum gradients, updating the quantum weights can prove difficult as well. Classical neural networks have had tremendous success using backpropagation to update the model's weights, however methods for updating quantum weights is still being intensely researched. QNNs most commonly employ the parameter-shift method[232,233] to estimate quantum gradients for each weight, however this can prove expensive as it requires running at least $2M$ quantum circuits for M trainable parameters during the backward pass computation, giving a total time complexity of $O(M^2)$. New methods for quantum backpropagation are emerging that is making the evaluation of quantum gradients more efficient, most recently the work by Abbas et al.[234] that reduces the complexity from quadratic parameter-shift method to $O(M\text{polylog}(M))$ time. The expensive nature of required quantum resources to update weights encourage many to explore different optimization methods. Many quantum neural networks in the literature often employ the Constrained Optimization by Linear Approximations algorithm[235] for weight optimization, however this method is only applicable for models with few trainable parameters. Work is being done to improve gradient-free based optimization of VQC parameters that are more efficient than the parameter-shirt method. Kulshrestha et al. devise an optimization scheme with good scalability potential that trains at the level of classical optimizers while outperforming them in computation time.[236] Weidmann et al. present an optimization method that significantly improves convergence of QNNs compared to the parameter-shift method.[237]

6.3. Outlook

In this review, we have examined the use of QNNs implemented on gate-based quantum computers for applications in chemistry and pharmaceuticals. While the integration of quantum computing into these fields holds the potential for significant

advancements, it also presents unique challenges that must be addressed.

As discussed in the previous subsections, the hardware and algorithmic challenges for QML are substantial. The requirements for coherence, qubit connectivity, and state preparation introduce significant hurdles that have yet to be fully overcome. QNNs often require precise qubit control and extended coherence times, which current quantum hardware struggles to provide consistently. On the algorithmic front, issues such as state preparation, barren plateaus, and efficient quantum gradient computation remain critical bottlenecks that demand innovative solutions.

Recent progress in quantum error correction, highlighted by Google Quantum AI's breakthrough,[238] and Brock et al.'s demonstration of qudit-based error correction beyond the break-even point[239] marks a significant milestone. This achievement suggests that we are nearing the development of more reliable quantum systems, which is crucial for the practical implementation of QML in real-world scenarios. However, there remains a pressing need for improved scalability of quantum hardware and the development of more robust error correction protocols.

Looking ahead, as quantum technology continues to mature, we anticipate the emergence of more sophisticated applications, such as the discovery of new drugs and materials, the optimization of chemical reactions, and the exploration of molecular structures with unprecedented accuracy. The intersection of quantum computing and machine learning offers a unique opportunity to transform how we tackle some of the most complex challenges in science and industry.

AUTHOR INFORMATION

Corresponding Author

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

Authors

Anthony M. Smaldone — *Department of Chemistry, Yale University, New Haven, Connecticut 06520, United States;* ◎ orcid.org/0009-0008-7265-0017

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

Chuzhi Xu — *Department of Chemistry, Yale University, New Haven, Connecticut 06520, United States;* ◎ orcid.org/0009-0003-6622-4752

Nam P. Vu — *Department of Chemistry, Lafayette College, Easton, Pennsylvania 18042, United States; Department of Chemistry, Yale University, New Haven, Connecticut 06520, United States;* ◎ orcid.org/0000-0002-3849-5139

Rishab Dutta — *Department of Chemistry, Yale University, New Haven, Connecticut 06520, United States;* ◎ orcid.org/0000-0001-7675-1431

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

Alexey Galda — *Moderna Inc., Cambridge, Massachusetts 02142, United States*

Sandeep Kumar — *Moderna Inc., Cambridge, Massachusetts 02142, 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.chemrev.4c00678

Author Contributions

[⊥]A.M.S. and Y.S. contributed equally to this article.

Notes

A.G. and S.K. are employees of Moderna, Inc. and may own stock/stock options in the company. A.M.S., Y.S., G.W.K., V.S.B., M.H.F., and E.K. have ongoing collaborative projects using CUDA-Q which do not alter the scientific integrity of the work presented herein. Other authors declare no conflict of interest.
The authors declare no competing financial interest.

Biographies

Anthony M. Smaldone is a National Science Foundation Graduate Research Fellow and a Ph.D. candidate in Theoretical Chemistry at Yale University. He earned an Associate of Science in Mathematics and Science from Corning Community College and a Bachelor of Science in Chemistry and Mathematics from Binghamton University. His research focuses on developing quantum machine learning methods and applying them to drug discovery problems.

Yu Shee received his B.S. in Chemical Biology and Computational Chemistry from the University of California, Berkeley. He is currently a Ph.D. candidate in the Batista Group at Yale University, where his research focuses on applying machine learning methods to organic reactions.

Gregory W. Kyro is a National Science Foundation Graduate Research Fellow and Ph.D. candidate in Chemistry at Yale University. He is best known for developing multiple state-of-the-art deep learning methods for modeling protein-small molecule interactions.

Chuzhi Xu received his B.A. in Chemistry and Computer Science from Washington University in St. Louis. He is currently a first-year Ph.D. student in Theoretical Chemistry in the Batista Group at Yale University, where he focuses on machine learning, quantum computing, and their applications to chemistry.

Nam P. Vu is currently an undergraduate student at Lafayette College majoring in Electrical & Computer Engineering and Mathematics. He is also a visiting undergraduate researcher at Yale University, where he is developing quantum algorithms for chemical graph theory problems and simulations of open quantum dynamics to further understand charge and energy transfer processes in molecular systems. In addition, Nam is engaged in molecular dynamics simulation and analysis of biomolecular systems, computational analysis of organic reaction mechanisms, as well as machine-learning-assisted electronic and photonic sensing devices.

Dr. Rishab Dutta received his Ph.D. in Chemistry from Rice University, where he developed computational methods for strongly correlated quantum many-body systems. He is currently a postdoctoral associate in the Department of Chemistry at Yale University, specializing in quantum algorithms for molecular sciences and the algebraic structures emerging from many-body quantum mechanics.

Dr. Marwa Farag is a quantum algorithm engineer at NVIDIA. Her work at NVIDIA focuses on developing algorithms using the NVIDIA CUDA-Q platform and AI techniques in collaboration with researchers across academia and industry. Before joining NVIDIA, she led quantum

computing efforts at Ford Motor Company, focusing on the development of advanced materials simulation. She holds a Ph.D. in theoretical chemistry and computational modeling from the University of Murcia. After her Ph.D., she worked in academia as a postdoc at the Max-Planck Institute, the University of Groningen, and the University of Southern California. During her doctoral and postdoctoral research, she developed quantum chemical methods for applications including electronic structure theory and dynamics of chemical systems.

Dr. Alexey Galda received his Ph.D. in Theoretical Physics from the University of Birmingham (UK), specializing in the physics of strongly correlated quantum systems. Following his Ph.D., he held a postdoctoral fellowship at Argonne National Laboratory and served as a Research Assistant Professor at the University of Chicago for five years. Transitioning from a decade in academia to the tech industry, Dr. Galda held Chief Scientist roles at various startups, focusing on quantum computing and AI. His expertise encompasses quantum optimization algorithms, experimental implementations of ternary quantum computation on NISQ devices, and the development of tensor-network simulators for quantum circuits. In 2023, Dr. Galda joined Moderna as the Associate Scientific Director of Quantum Algorithms and Applications, where he leads the quantum computing program aimed at pioneering quantum algorithms for mRNA drug design.

Sandeep Kumar is a computational biophysicist with more than 25 years of research experience in protein structure−function relationships and development of antibody-based biopharmaceuticals. He holds a bachelor's degree in Physics, a Master's degree in Molecular Biology Biotechnology, and a doctorate in Theoretical Biophysics. Sandeep has contributed to 113 peer reviewed research articles, book chapters, and topical reviews, along with two edited books. In recent years, Sandeep has been advocating for Biopharmaceutical Informatics, a strategic vision for synergistic use of computation and experiments in drug discovery. As part of this, Sandeep has proposed a conceptual roadmap for AI-enabled discovery of antibodies in silico (DAbI). Sandeep Kumar is excited about the potential applications of quantum computing to the discovery and development of therapeutic antibodies.

Dr. Elica Kyoseva is the Director of Quantum Algorithm Engineering at NVIDIA, where she leads the quantum algorithm engineering and applied research teams. Her focus is on using AI supercomputing to overcome the bottlenecks of quantum computing, which include enabling large-scale device modeling, quantum circuit generation, and algorithm development, among others. Prior to her role at NVIDIA, Elica envisioned, launched, and led the Quantum for Bio program at Wellcome Leap, a $50 million program aimed at accelerating the applications of quantum computing in human health. Elica earned her PhD in Quantum Optics from the University of Sofia, Bulgaria. She spent time in academia as a postdoc at the Center for Quantum Technologies in Singapore and, subsequently, as a tenure-track assistant professor at the Singapore University of Technology and Design. Elica also spent a year at the Massachusetts Institute of Technology as a Research Fellow. She was a Marie Curie fellow and was awarded the 2016 John Atanasoff Award of President of Bulgaria for internationally recognized scientists.

Victor S. Batista is the John Gamble Kirkwood Professor of Chemistry at Yale University, where he has been a faculty member since 2001. He earned his B.Sc. in Chemistry from the University of Buenos Aires in 1989 and his Ph.D. in Theoretical Chemistry from Boston University in 1996. After completing postdoctoral research at the University of California, Berkeley, and the University of Toronto, he joined Yale's Department of Chemistry. Professor Batista's research focuses on theoretical and computational chemistry, particularly on developing semiclassical and quantum dynamics methods to study photoinduced reactions and catalytic processes. He has published over 425 articles in peer-reviewed journals, contributing significantly to the understanding of photosynthetic systems and quantum control of chemical dynamics. As the Director of the NSF Center for Quantum Dynamics on Modular Quantum Devices, Professor Batista leads efforts to develop new paradigms for quantum simulations of complex chemical systems using programmable Kerr-cat platforms. The Center aims to demonstrate the unique capabilities of bosonic modular devices in simulating chemical dynamics and correlated many-body systems. Throughout his career, Professor Batista has received numerous accolades. He is a Fellow of the Royal Society of Chemistry and an elected member of the Connecticut Academy of Science and Engineering. His recent publications include work on quantum machine learning applications in drug discovery as well as the development of quantum algorithms for simulating non-Markovian dynamics in chemical systems.

ACKNOWLEDGMENTS

REFERENCES

(1) Batra, K.; Zorn, K. M.; Foil, D. H.; Minerali, E.; Gawriljuk, V. O.; Lane, T. R.; Ekins, S. Quantum Machine Learning Algorithms for Drug Discovery Applications. *J. Chem. Inf. Model.* **2021**, *61*, 2641−2647.

(2) Avramouli, M.; Savvas, I. K.; Vasilaki, A.; Garani, G. Unlocking the Potential of Quantum Machine Learning to Advance Drug Discovery. *Electronics* **2023**, *12*, 2402.

(3) Mensa, S.; Sahin, E.; Tacchino, F.; Barkoutsos, P. K.; Tavernelli, I. Quantum Machine Learning Framework for Virtual Screening in Drug Discovery: a Prospective Quantum Advantage. *arXiv* **2022**, 2204.04017.

(4) Pyrkov, A.; Aliper, A.; Bezrukov, D.; Lin, Y.-C.; Polykovskiy, D.; Kamya, P.; Ren, F.; Zhavoronkov, A. Quantum computing for near-term applications in generative chemistry and drug discovery. *Drug Discovery Today* **2023**, *28*, 103675.

(5) Sajjan, M.; Li, J.; Selvarajan, R.; Sureshbabu, S. H.; Kale, S. S.; Gupta, R.; Singh, V.; Kais, S. Quantum machine learning for chemistry and physics. *Chem. Soc. Rev.* **2022**, *51*, 6475−6573.

(6) Weidman, J. D.; Sajjan, M.; Mikolas, C.; Stewart, Z. J.; Pollanen, J.; Kais, S.; Wilson, A. K. Quantum computing and chemistry. *Cell Reports Physical Science* **2024**, *5*, 102105.

(7) Meyer, N.; Ufrecht, C.; Periyasamy, M.; Scherer, D. D.; Plinge, A.; Mutschler, C. A Survey on Quantum Reinforcement Learning. *arXiv* 2024, 2211.03464. DOI: 10.48550/arXiv.2211.03464.

(8) Park; Kim, J. Trends in Quantum Reinforcement Learning: State-of-the-Arts and the Road Ahead. *ETRI J.* **2024**, *46* (5), 748−758.

(9) Chen, S. Y.-C.; Yang, C.-H. H.; Qi, J.; Chen, P.-Y.; Ma, X.; Goan, H.-S. Variational Quantum Circuits for Deep Reinforcement Learning. *IEEE Access* **2020**, *8*, 141007−141024.

(10) Coppersmith, D. An approximate Fourier transform useful in quantum factoring. *arXiv* 2002, 0201067 DOI: 10.48550/arXiv.quant-ph/0201067.

(11) Kitaev, A. Y. Quantum measurements and the Abelian Stabilizer Problem. *arXiv* 1995, 9511026 DOI: 10.48550/arXiv.quant-ph/9511026.

(12) Shor, P. Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science* **1994**, 124−134, DOI: 10.1109/SFCS.1994.365700.

(13) Shor, P. W. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Journal on Computing* **1997**, *26*, 1484−1509.

(14) Harrow, A. W.; Hassidim, A.; Lloyd, S. Quantum algorithm for solving linear systems of equations. *Phys. Rev. Lett.* **2009**, *103*, 150502.

(15) Bharti, K.; Cervera-Lierta, A.; Kyaw, T. H.; Haug, T.; Alperin-Lea, S.; Anand, A.; Degroote, M.; Heimonen, H.; Kottmann, J. S.; Menke, T.; et al. others Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* **2022**, *94*, 015004.

(16) Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S. C.; Endo, S.; Fujii, K.; McClean, J. R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. others Variational quantum algorithms. *Nat. Rev. Phys.* **2021**, *3*, 625−644.

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

(18) Moll, N.; Barkoutsos, P.; Bishop, L. S.; Chow, J. M.; Cross, A.; Egger, D. J.; Filipp, S.; Fuhrer, A.; Gambetta, J. M.; Ganzhorn, M.; et al. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Sci. Technol.* **2018**, *3*, 030503.

(19) McArdle, S.; Jones, T.; Endo, S.; Li, Y.; Benjamin, S. C.; Yuan, X. Variational ansatz-based quantum simulation of imaginary time evolution. *npj Quantum Inf.* **2019**, *5*, 75.

(20) Yuan, X.; Endo, S.; Zhao, Q.; Li, Y.; Benjamin, S. C. Theory of variational quantum simulation. *Quantum* **2019**, *3*, 191.

(21) Gomes, N.; Zhang, F.; Berthusen, N. F.; Wang, C.-Z.; Ho, K.-M.; Orth, P. P.; Yao, Y. Efficient Step-Merged Quantum Imaginary Time Evolution Algorithm for Quantum Chemistry. *J. Chem. Theory Comput.* **2020**, *16*, 6256−6266.

(22) Motta, M.; Sun, C.; Tan, A. T. K.; O'Rourke, M. J.; Ye, E.; Minnich, A. J.; Brandão, F. G. S. L.; Chan, G. K.-L. Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution. *Nat. Phys.* **2020**, *16*, 205−210.

(23) Nishi, H.; Kosugi, T.; Matsushita, Y.-i. Implementation of quantum imaginary-time evolution method on NISQ devices by introducing nonlocal approximation. *npj Quantum Inf.* **2021**, *7*, 85.

(24) Selvarajan, R.; Dixit, V.; Cui, X.; Humble, T. S.; Kais, S. Prime factorization using quantum variational imaginary time evolution. *Sci. Rep.* **2021**, *11*, 20835.

(25) Huang, Y.; Shao, Y.; Ren, W.; Sun, J.; Lv, D. Efficient quantum imaginary time evolution by drifting real-time evolution: An approach with low gate and measurement complexity. *J. Chem. Theory Comput.* **2023**, *19*, 3868−3876.

(26) Yeter-Aydeniz, K.; Moschandreou, E.; Siopsis, G. Quantum imaginary-time evolution algorithm for quantum field theories with continuous variables. *Phys. Rev. A* **2022**, *105*, 012412.

(27) Kyaw, T. H.; Soley, M. B.; Allen, B.; Bergold, P.; Sun, C.; Batista, V. S.; Aspuru-Guzik, A. Boosting quantum amplitude exponentially in variational quantum algorithms. *Quantum Science and Technology* **2024**, *9*, 01LT01.

(28) Benedetti, M.; Fiorentini, M.; Lubasch, M. Hardware-efficient variational quantum algorithms for time evolution. *Physical Review Research* **2021**, *3*, 033083.

(29) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016. http://www.deeplearningbook.org, accessed 2024-08-27.

(30) Smaldone, A. M.; Batista, V. S. Quantum-to-Classical Neural Network Transfer Learning Applied to Drug Toxicity Prediction. *J. Chem. Theory Comput.* **2024**, *20*, 4901−4908.

(31) Kyro, G. W.; Martin, M. T.; Watt, E. D.; Batista, V. S. CardioGenAI: A Machine Learning-Based Framework for Re-Engineering Drugs for Reduced hERG Liability. *arXiv* **2024**, 2403.07632 DOI: 10.48550/arXiv.2403.07632.

(32) Kyro, G. W.; Morgunov, A.; Brent, R. I.; Batista, V. S. ChemSpaceAL: An Efficient Active Learning Methodology Applied to Protein-Specific Molecular Generation. *J. Chem. Inf. Model.* **2024**, *64*, 653−665.

(33) Jumper, J.; et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583−589.

(34) Abramson, J.; et al. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **2024**, *630*, 493−500.

(35) Varadi, M.; Velankar, S. The impact of AlphaFold Protein Structure Database on the fields of life sciences. *PROTEOMICS* **2023**, *23*, 2200128.

(36) Niazi, S. K.; Mariam, Z. Recent Advances in Machine-Learning-Based Chemoinformatics: A Comprehensive Review. *International Journal of Molecular Sciences* **2023**, *24*, 11488.

(37) Paul, S.; Olymon, K.; Martinez, G. S.; Sarkar, S.; Yella, V. R.; Kumar, A. MLDSPP: Bacterial Promoter Prediction Tool Using DNA Structural Properties with Machine Learning and Explainable AI. *J. Chem. Inf. Model.* **2024**, *64*, 2705−2719.

(38) Zhou, Y.; Qi, T.; Pan, M.; Tu, J.; Zhao, X.; Ge, Q.; Lu, Z. Deep-Cloud: A Deep Neural Network-Based Approach for Analyzing Differentially Expressed Genes of RNA-seq Data. *J. Chem. Inf. Model.* **2024**, *64*, 2302−2310.

(39) Khashei, M.; Nazgouei, E.; Bakhtiarvand, N. Intelligent Discrete Deep Learning Based Classification Methodology in Chemometrics. *J. Chem. Inf. Model.* **2023**, *63*, 1935−1946.

(40) Kyro, G. W.; Brent, R. I.; Batista, V. S. HAC-Net: A Hybrid Attention-Based Convolutional Neural Network for Highly Accurate Protein−Ligand Binding Affinity Prediction. *J. Chem. Inf. Model.* **2023**, *63*, 1947−1960.

(41) Oliveira, T. A. d.; Silva, M. P. d.; Maia, E. H. B.; Silva, A. M. d.; Taranto, A. G. Virtual Screening Algorithms in Drug Discovery: A Review Focused on Machine and Deep Learning Methods. *Drugs and Drug Candidates* **2023**, *2*, 311−334.

(42) Corso, G.; Stärk, H.; Jing, B.; Barzilay, R.; Jaakkola, T. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking. *arXiv* **2023**, 2210.01776 DOI: 10.48550/arXiv.2210.01776.

(43) Schrier, J.; Norquist, A. J.; Buonassisi, T.; Brgoch, J. Pursuit of the Exceptional: Research Directions for Machine Learning in Chemical and Materials Science. *J. Am. Chem. Soc.* **2023**, *145*, 21699−21716.

(44) Low, G. H.; Yoder, T. J.; Chuang, I. L. Quantum inference on Bayesian networks. *Phys. Rev. A* **2014**, *89*, 062315.

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

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

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

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

(49) Kak, S. C. Advances in Imaging and Electron Physics. *Elsevier* **1995**, *94*, 259−313.

(50) Panella, M.; Martinelli, G. Neural networks with quantum architecture and quantum learning. *International Journal of Circuit Theory and Applications* **2011**, *39*, 61−77.

(51) Gupta, S.; Zia, R. K. P. Quantum Neural Networks. *Journal of Computer and System Sciences* **2001**, *63*, 355−383.

(52) Zhou, R.; Wang, H.; Wu, Q.; Shi, Y. Quantum Associative Neural Network with Nonlinear Search Algorithm. *International Journal of Theoretical Physics* **2012**, *51*, 705−723.

(53) Gonçalves, C. P. Quantum Cybernetics and Complex Quantum Systems Science - A Quantum Connectionist Exploration. *arXiv* **2014**, 1402.1141 DOI: 10.48550/arXiv.1402.1141.

(54) Schuld, M.; Sinayskiy, I.; Petruccione, F. The quest for a Quantum Neural Network. *Quantum Information Processing* **2014**, *13*, 2567−2586.

(55) Wilkinson, S. A.; Hartmann, M. J. Evaluating the performance of sigmoid quantum perceptrons in quantum neural networks. *arXiv* **2022**, 2208.06198 DOI: 10.48550/arXiv.2208.06198.

(56) Maronese, M.; Destri, C.; Prati, E. Quantum activation functions for quantum neural networks. *Quantum Information Processing* **2022**, *21*, 128.

(57) Zi, W.; Wang, S.; Kim, H.; Sun, X.; Chattopadhyay, A.; Rebentrost, P. Efficient quantum circuits for machine learning activation functions including constant T-depth ReLU. *Physical Review Research* **2024**, *6*, 043048.

(58) Farhi, E.; Neven, H. Classification with Quantum Neural Networks on Near Term Processors. *arXiv* **2018**, 1802.06002 DOI: 10.48550/arXiv.1802.06002.

(59) LaRose, R.; Coyle, B. Robust data encodings for quantum classifiers. *Phys. Rev. A* **2020**, *102*, 032420.

(60) Pande, M. B. A Comprehensive Review of Data Encoding Techniques for Quantum Machine Learning Problems. In *2024 Second International Conference on Emerging Trends in Information Technology and Engineering (ICETITE)*; IEEE, 2024; pp 1−7. DOI: 10.1109/ic-ETITE58242.2024.10493306

(61) Reiser, P.; Neubert, M.; Eberhard, A.; Torresi, L.; Zhou, C.; Shao, C.; Metni, H.; van Hoesel, C.; Schopmans, H.; Sommer, T.; Friederich, P. Graph neural networks for materials science and chemistry. *Communications Materials* **2022**, *3*, 93.

(62) Verdon, G.; McCourt, T.; Luzhnica, E.; Singh, V.; Leichenauer, S.; Hidary, J. Quantum Graph Neural Networks. *arXiv* **2019**, 1909.12264 DOI: 10.48550/arXiv.1909.12264.

(63) Zheng, J.; Gao, Q.; Lü, Y. Quantum Graph Convolutional Neural Networks. *40th Chinese Control Conference (CCC)* **2021**, 6335−6340.

(64) Tüysüz, C.; Rieger, C.; Novotny, K.; Demirköz, B.; Dobos, D.; Potamianos, K.; Vallecorsa, S.; Vlimant, J.-R.; Forster, R. Hybrid quantum classical graph neural networks for particle track reconstruction. *Quantum Machine Intelligence* **2021**, *3*, 29.

(65) Beer, K.; Khosla, M.; Köhler, J.; Osborne, T. J. Quantum machine learning of graph-structured data. *arXiv* **2021**, 2103.10837 DOI: 10.48550/arXiv.2103.10837.

(66) Liao, Y.; Zhang, X.-M.; Ferrie, C. Graph Neural Networks on Quantum Computers. *arXiv* **2024**, 2405.17060 DOI: 10.48550/arXiv.2405.17060.

(67) Mernyei, P.; Meichanetzidis, K.; Ceylan, s. l. Equivariant Quantum Graph Circuits. *arXiv* **2022**, 2112.05261 DOI: 10.48550/arXiv.2112.05261.

(68) Skolik, A.; Cattelan, M.; Yarkoni, S.; Bäck, T.; Dunjko, V. Equivariant quantum circuits for learning on weighted graphs. *arXiv* **2023**, 2205.06109 DOI: 10.48550/arXiv.2205.06109.

(69) Chen, S.; Tang, Z.; You, L.; Chen, C. Y.-C. A knowledge distillation-guided equivariant graph neural network for improving protein interaction site prediction performance. *Knowledge-Based Systems* **2024**, *300*, 112209.

(70) Cremer, J.; Medrano Sandonas, L.; Tkatchenko, A.; Clevert, D.-A.; De Fabritiis, G. Equivariant Graph Neural Networks for Toxicity Prediction. *Chem. Res. Toxicol.* **2023**, *36* (10), 1561−1573.

(71) Dhakal, A.; Gyawali, R.; Cheng, J. Predicting Protein-Ligand Binding Structure Using E(n) Equivariant Graph Neural Networks. *bioRxiv* **2023**, DOI: 10.1101/2023.08.06.552202.

(72) Ryu, J.-Y.; Elala, E.; Rhee, J.-K. K. Quantum Graph Neural Network Models for Materials Search. *Materials* **2023**, *16*, 4300.

(73) Caro, M. C.; Huang, H.-Y.; Cerezo, M.; Sharma, K.; Sornborger, A.; Cincio, L.; Coles, P. J. Generalization in quantum machine learning from few training data. *Nat. Commun.* **2022**, *13*, 4919.

(74) Vitz, M.; Mohammadbagherpoor, H.; Sandeep, S.; Vlasic, A.; Padbury, R.; Pham, A. Hybrid Quantum Graph Neural Network for Molecular Property Prediction. *arXiv* **2024**, 2405.05205 DOI: 10.48550/arXiv.2405.05205.

(75) Dong, L.; Li, Y.; Liu, D.; Ji, Y.; Hu, B.; Shi, S.; Zhang, F.; Hu, J.; Qian, K.; Jin, X.; Wang, B. Prediction of Protein−Ligand Binding Affinity by a Hybrid Quantum−Classical Deep Learning Algorithm. *Advanced Quantum Technologies* **2023**, *6*, 2300107.

(76) Wu, F.; Zhang, T.; Souza, Jr. A. H. d.; Fifty, C.; Yu, T.; Weinberger, K. Q. Simplifying Graph Convolutional Networks. *arXiv* **2019**, 1902.07153 DOI: 10.48550/arXiv.1902.07153.

(77) Pasa, L.; Navarin, N.; Erb, W.; Sperduti, A. Empowering Simple Graph Convolutional Networks. *IEEE Transactions on Neural Networks and Learning Systems* **2024**, *35*, 4385−4399.

(78) Dhillon, A.; Verma, G. K. Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence* **2020**, *9*, 85−112.

(79) Jiang, S.; Zavala, V. M. Convolutional neural nets in chemical engineering: Foundations, computations, and applications. *AIChE J.* **2021**, *67*, No. e17282.

(80) O'Shea, K.; Nash, R. An Introduction to Convolutional Neural Networks. *arXiv* **2015**, 1511.08458 DOI: 10.48550/arXiv.1511.08458.

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

(82) Kerenidis, I.; Landman, J.; Prakash, A. Quantum Algorithms for Deep Convolutional Neural Networks. *arXiv* **2019**, 1911.01117 DOI: 10.48550/arXiv.1911.01117.

(83) Henderson, M.; Shakya, S.; Pradhan, S.; Cook, T. Quanvolutional neural networks: powering image recognition with quantum circuits. *Quantum Machine Intelligence* **2020**, *2*, 2.

(84) Liu, J.; Lim, K. H.; Wood, K. L.; Huang, W.; Guo, C.; Huang, H.-L. Hybrid quantum-classical convolutional neural networks. *Sci. China Phys. Mech. Astron.* **2021**, *64*, 290311.

(85) MacCormack, I.; Delaney, C.; Galda, A.; Aggarwal, N.; Narang, P. Branching quantum convolutional neural networks. *Physical Review Research* **2022**, *4*, 013117.

(86) Smaldone, A. M.; Kyro, G. W.; Batista, V. S. Quantum convolutional neural networks for multi-channel supervised learning. *Quantum Machine Intelligence* **2023**, *5*, 41.

(87) Oh, S.; Choi, J.; Kim, J. A Tutorial on Quantum Convolutional Neural Networks (QCNN). *2020 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, South Korea; IEEE, 2020; pp 236−239. DOI: 10.1109/ICTC49870.2020.9289439

(88) Mari, A. Quanvolutional Neural Networks. *Pennylane*, 2020. https://pennylane.ai/qml/demos/tutorial_quanvolution/, accessed 2024-08-27.

(89) The Quantum Convolution Neural Network. In *Qiskit Machine Learning 0.8.2*; GitHub, 2024. https://qiskit-community.github.io/qiskit-machine-learning/tutorials/11_quantum_convolutional_neural_networks.html, accessed 2024-08-27.

(90) Quantum Convolutional Neural Network. *TensorFlow*. https://www.tensorflow.org/quantum/tutorials/qcnn, accessed 2024-08-27.

(91) Chen, S. Y.-C.; Wei, T.-C.; Zhang, C.; Yu, H.; Yoo, S. Quantum convolutional neural networks for high energy physics data analysis. *Physical Review Research* **2022**, *4*, 013231.

(92) Hong, Z.; Wang, J.; Qu, X.; Zhu, X.; Liu, J.; Xiao, J. Quantum Convolutional Neural Network on Protein Distance Prediction. In *2021 International Joint Conference on Neural Networks (IJCNN)*; IEEE, 2021; pp 1−8. DOI: 10.1109/IJCNN52387.2021.9533405

(93) Domingo, L.; Djukic, M.; Johnson, C.; Borondo, F. Binding affinity predictions with hybrid quantum-classical convolutional neural networks. *Sci. Rep.* **2023**, *13*, 17951.

(94) Wu, T.; Bentellis, A.; Sakhnenko, A.; Lorenz, J. M. Generalization Bounds in Hybrid Quantum-Classical Machine Learning Models. *arXiv* 2025, 2504.08456 DOI: 10.48550/arXiv.2504.0845.

(95) Stein, S. A.; Mao, Y.; Ang, J.; Li, A. QuCNN: A Quantum Convolutional Neural Network with Entanglement Based Backpropagation. *arXiv* **2022**, 2210.05443 DOI: 10.48550/arXiv.2210.05443.

(96) Zhao, J.; Zhang, Y.-H.; Shao, C.-P.; Wu, Y.-C.; Guo, G.-C.; Guo, G.-P. Building quantum neural networks based on swap test. *Phys. Rev. A* **2019**, *100*, 012334.

(97) Jones, D. T.; Kandathil, S. M. High precision in protein contact prediction using fully convolutional neural networks and minimal sequence features. *Bioinformatics* **2018**, *34*, 3308−3315.

(98) Kumar, V.; Alevras, D.; Metkar, M.; Welling, E.; Cade, C.; Niesen, I.; Friedhoff, T.; Park, J.; Shivpuje, Sa.; LaDue, M.; Davis, W.; Galda, A. Towards secondary structure prediction of longer mRNA sequences using a quantum-centric optimization scheme. *arXiv* 2025, 2505.05782, DOI: 10.48550/arXiv.2505.05782.

(99) Alevras, D.; Metkar, M.; Yamamoto, T.; Kumar, V.; Friedhoff, T.; Park, J.-E.; Takeori, M.; LaDue, M.; Davis, W.; Galda, A. mRNA Secondary Structure Prediction Using Utility-Scale Quantum Computers. In *IEEE International Conference on Quantum Computing and*

Engineering (QCE); IEEE, 2024; pp 488−499. DOI: 10.1109/QCE60285.2024.00064

(100) Paquet, E.; Soleymani, F.; St-Pierre-Lemieux, G.; Viktor, H. L.; Michalowski, W. QuantumBound − Interactive protein generation with one-shot learning and hybrid quantum neural networks. *Artificial Intelligence Chemistry* **2024**, *2*, 100030.

(101) Jin, Y.-X.; Hu, J.-J.; Li, Q.; Luo, Z.-C.; Zhang, F.-Y.; Tang, H.; Qian, K.; Jin, X.-M. Quantum Deep Learning for Mutant COVID-19 Strain Prediction. *arXiv* **2022**, 2203.03556 DOI: 10.48550/arXiv.2203.03556.

(102) Romero, J.; Olson, J. P.; Aspuru-Guzik, A. Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology* **2017**, *2*, 045001.

(103) Lamata, L.; Alvarez-Rodriguez, U.; Martín-Guerrero, J. D.; Sanz, M.; Solano, E. Quantum autoencoders via quantum adders with genetic algorithms. *Quantum Science and Technology* **2019**, *4*, 014007.

(104) Bondarenko, D.; Feldmann, P. Quantum Autoencoders to Denoise Quantum Data. *Phys. Rev. Lett.* **2020**, *124*, 130502.

(105) Cao, C.; Wang, X. Noise-Assisted Quantum Autoencoder. *Physical Review Applied* **2021**, *15*, 054012.

(106) Pepper, A.; Tischler, N.; Pryde, G. J. Experimental Realization of a Quantum Autoencoder: The Compression of Qutrits via Machine Learning. *Phys. Rev. Lett.* **2019**, *122*, 060501.

(107) Ding, Y.; Lamata, L.; Sanz, M.; Chen, X.; Solano, E. Experimental Implementation of a Quantum Autoencoder via Quantum Adders. *Advanced Quantum Technologies* **2019**, *2*, 1800065.

(108) Huang, C.-J.; Ma, H.; Yin, Q.; Tang, J.-F.; Dong, D.; Chen, C.; Xiang, G.-Y.; Li, C.-F.; Guo, G.-C. Realization of a quantum autoencoder for lossless compression of quantum data. *Phys. Rev. A* **2020**, *102*, 032412.

(109) Srikumar, M.; Hill, C. D.; Hollenberg, L. C. L. Clustering and enhanced classification using a hybrid quantum autoencoder. *Quantum Science and Technology* **2022**, *7*, 015020.

(110) Ma, H.; Huang, C.-J.; Chen, C.; Dong, D.; Wang, Y.; Wu, R.-B.; Xiang, G.-Y. On compression rate of quantum autoencoders: Control design, numerical and experimental realization. *arXiv* **2022**, 2005.11149 DOI: 10.48550/arXiv.2005.11149.

(111) Wang, G.; Warrell, J.; Emani, P. S.; Gerstein, M. ζ-QVAE: A Quantum Variational Autoencoder utilizing Regularized Mixed-state Latent Representations. *arXiv* **2024**, 2402.17749 DOI: 10.48550/arXiv.2402.17749.

(112) Guimaraes, G. L.; Sanchez-Lengeling, B.; Outeiral, C.; Farias, P. L. C.; Aspuru-Guzik, A. Objective-reinforced generative adversarial networks (organ) for sequence generation models. *arXiv* **2017**, 1705.10843 DOI: 10.48550/arXiv.1705.10843.

(113) Sanchez-Lengeling, B.; Outeiral, C.; Guimaraes, G. L.; Aspuru-Guzik, A. Optimizing distributions over molecular space. An objective-reinforced generative adversarial network for inverse-design chemistry (ORGANIC). *ChemRxiv* **2017**, DOI: 10.26434/chemrxiv.5309668.v2.

(114) Putin, E.; Asadulaev, A.; Ivanenkov, Y.; Aladinskiy, V.; Sanchez-Lengeling, B.; Aspuru-Guzik, A.; Zhavoronkov, A. Reinforced adversarial neural computer for de novo molecular design. *J. Chem. Inf. Model.* **2018**, *58*, 1194−1204.

(115) Putin, E.; Asadulaev, A.; Vanhaelen, Q.; Ivanenkov, Y.; Aladinskaya, A. V.; Aliper, A.; Zhavoronkov, A. Adversarial threshold neural computer for molecular de novo design. *Mol. Pharmaceutics* **2018**, *15*, 4386−4397.

(116) Dallaire-Demers, P.-L.; Killoran, N. Quantum generative adversarial networks. *Phys. Rev. A* **2018**, *98*, 012324.

(117) Romero, J.; Aspuru-Guzik, A. Variational Quantum Generators: Generative Adversarial Quantum Machine Learning for Continuous Distributions. *Advanced Quantum Technologies* **2021**, *4*, 2000003.

(118) Li, J.; Alam, M.; Sha, C. M.; Wang, J.; Dokholyan, N. V.; Ghosh, S. Drug Discovery Approaches using Quantum Machine Learning. *arXiv* **2021**, 2104.00746.

(119) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific Data* **2014**, *1*, 140022.

(120) Kao, P.-Y.; Yang, Y.-C.; Chiang, W.-Y.; Hsiao, J.-Y.; Cao, Y.; Aliper, A.; Ren, F.; Aspuru-Guzik, A.; Zhavoronkov, A.; Hsieh, M.-H.; Lin, Y.-C. Exploring the Advantages of Quantum Generative Adversarial Networks in Generative Chemistry. *J. Chem. Inf. Model.* **2023**, *63*, 3307−3318.

(121) Anoshin, M.; Sagingalieva, A.; Mansell, C.; Zhiganov, D.; Shete, V.; Pflitsch, M.; Melnikov, A. Hybrid Quantum Cycle Generative Adversarial Network for Small Molecule Generation. *IEEE Transactions on Quantum Engineering* **2024**, *5*, 1−15.

(122) Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A. A. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *arXiv* **2020**, 1703.10593 DOI: 10.48550/arXiv.1703.10593.

(123) Maziarka, u.; Pocha, A.; Kaczmarczyk, J.; Rataj, K.; Danel, T.; Warchoł, M. Mol-CycleGAN: a generative model for molecular optimization. *Journal of Cheminformatics* **2020**, *12*, 2.

(124) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2023**, 1706.03762 DOI: 10.48550/arXiv.1706.03762.

(125) Wang, J.; Hsieh, C.-Y.; Wang, M.; Wang, X.; Wu, Z.; Jiang, D.; Liao, B.; Zhang, X.; Yang, B.; He, Q.; et al. others Multi-constraint molecular generation based on conditional transformer, knowledge distillation and reinforcement learning. *Nature Machine Intelligence* **2021**, *3*, 914−922.

(126) Luong, K.-D.; Singh, A. Application of Transformers in Cheminformatics. *J. Chem. Inf. Model.* **2024**, *64*, 4392−4409.

(127) Shee, Y.; Li, H.; Morgunov, A.; Batista, V. DirectMultiStep: Direct Route Generation for Multi-Step Retrosynthesis. *J. Chem. Inf. Model.* **2025**, *65* (8), 3903−3914.

(128) Shee, Y.; Li, H.; Zhang, P.; Nikolic, A. M.; Lu, W.; Kelly, H. R.; Manee, V.; Sreekumar, S.; Buono, F. G.; Song, J. J.; Newhouse, T. R.; Batista, V. S. Site-specific template generative approach for retrosynthetic planning. *Nat. Commun.* **2024**, *15*, 7818.

(129) Li, H.; Shee, Y.; Allen, B.; Maschietto, F.; Morgunov, A.; Batista, V. Kernel-elastic autoencoder for molecular design. *PNAS Nexus* **2024**, *3*, pgae168.

(130) 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. *arXiv* **2024**, 2402.18940 DOI: 10.48550/arXiv.2402.18940.

(131) 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. *arXiv* **2024**, 2403.14753 DOI: 10.48550/arXiv.2403.14753.

(132) Tariq, S.; Arfeto, B. E.; Khalid, U.; Kim, S.; Duong, T. Q.; Shin, H. Deep Quantum-Transformer Networks for Multi-Modal Beam Prediction in ISAC Systems. *IEEE Internet of Things Journal* **2024**, *11*, 29387.

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

(134) Khokhlov, I.; Krasnov, L.; Fedorov, M. V.; Sosnin, S. Image2SMILES: Transformer−Based Molecular Optical Recognition Engine. *Chemistry Methods* **2022**, *2*, No. e202100069.

(135) Ding, W.; Chen, H.; Ji, H. Efficiently Predicting Reaction Rates and Revealing Reactive Sites with a Molecular Image-Vision Transformer and Fukui Function Validation. *Ind. Eng. Chem. Res.* **2024**, *63* (16), 7064−7072.

(136) Jha, K.; Saha, S.; Karmakar, S. Prediction of Protein-Protein Interactions Using Vision Transformer and Language Model. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* **2023**, *20*, 3215−3225.

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

(138) Di Sipio, R.; Huang, J.-H.; Chen, S. Y.-C.; Mangini, S.; Worring, M. The Dawn of Quantum Natural Language Processing. *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal*

Processing (ICASSP), Singapore; Brrokhaven Science Associates, LLC, 2022; pp 8612−8616.

(139) Gao, Y.; Song, Z.; Yang, X.; Zhang, R. Fast Quantum Algorithm for Attention Computation. *arXiv* **2023**, 2307.08045.

(140) Li, G.; Zhao, X.; Wang, X. Quantum Self-Attention Neural Networks for Text Classification. *arXiv* **2023**, 2205.05625.

(141) 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. *arXiv* **2024**, 2402.16714.

(142) Liao, Y.; Ferrie, C. GPT on a Quantum Computer. *arXiv* **2024**, 2403.09418.

(143) CUDA-Q Applications. NVIDIA CUDA-Q; *NVIDIA*. https://nvidia.github.io/cuda-quantum/latest/applications/python/quantum_transformer.html, accessed 2025-05-05

(144) Katharopoulos, A.; Vyas, A.; Pappas, N.; Fleuret, F. Transformers are RNNs: Fast Autoregressive Transformers with Linear Attention. *arXiv* **2020**, 2006.16236.

(145) Wang, S.; Li, B. Z.; Khabsa, M.; Fang, H.; Ma, H. Linformer: Self-Attention with Linear Complexity. *arXiv* **2020**, 2006.04768.

(146) Lee-Thorp, J.; Ainslie, J.; Eckstein, I.; Ontanon, S. FNet: Mixing Tokens with Fourier Transforms. *arXiv* **2022**, 2105.03824.

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

(148) Childs, A. M.; Wiebe, N. Hamiltonian Simulation Using Linear Combinations of Unitary Operations. *Quantum Information and Computation* **2012**, *12*, 901.

(149) Gilyén, A.; Su, Y.; Low, G. H.; Wiebe, N. Quantum singular value transformation and beyond: exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, 2019; pp 193−204.

(150) Ghazi Vakili, M.; Gorgulla, C.; Snider, J. et al. Quantum-computing-enhanced algorithm unveils potential KRAS inhibitors. *Nat. Biotechnol.*, **2025** DOI: 10.1038/s41587-024-02526-3.

(151) Roca-Jerat, S.; Román-Roche, J.; Zueco, D. Qudit machine learning. *Machine Learning: Science and Technology* **2024**, *5*, 015057.

(152) Mandilara, A.; Dellen, B.; Jaekel, U.; Valtinos, T.; Syvridis, D. Classification of data with a qudit, a geometric approach. *Quantum Machine Intelligence* **2024**, *6*, 17.

(153) Ghasemian, E.; Razminia, A.; Rostami, H. Quantum machine learning based on continuous variable single-photon states: an elementary foundation for quantum neural networks. *Quantum Information Processing* **2023**, *22*, 378.

(154) Ranjan, A.; Patel, T.; Silver, D.; Gandhi, H.; Tiwari, D. ProxiML: Building Machine Learning Classifiers for Photonic Quantum Computing. *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* **2024**, *3*, 834−849.

(155) Copetudo, A.; Fontaine, C. Y.; Valadares, F.; Gao, Y. Y. Shaping photons: Quantum information processing with bosonic cQED. *Appl. Phys. Lett.* **2024**, *124*, 080502.

(156) Dutta, R.; Cabral, D. G.; Lyu, N.; Vu, N. P.; Wang, Y.; Allen, B.; Dan, X.; Cortiñas, R. G.; Khazaei, P.; Schafer, M.; et al. Simulating chemistry on bosonic quantum devices. *J. Chem. Theory Comput.* **2024**, *20*, 6426−6441.

(157) Liu, Y.; Singh, S.; Smith, K. C.; Crane, E.; Martyn, J. M.; Eickbusch, A.; Schuckert, A.; Li, R. D.; Sinanan-Singh, J.; Soley, M. B.; et al. Hybrid Oscillator-Qubit Quantum Processors: Instruction Set Architectures, Abstract Machine Models, and Applications. *arXiv* **2024**, 2407.10381.

(158) Wang, Y.; Hu, Z.; Sanders, B. C.; Kais, S. Qudits and high-dimensional quantum computing. *Front. Phys.* **2020**, *8*, 589504.

(159) Ofek, N.; Petrenko, A.; Heeres, R.; Reinhold, P.; Leghtas, Z.; Vlastakis, B.; Liu, Y.; Frunzio, L.; Girvin, S. M.; Jiang, L.; Mirrahimi, M.; Devoret, M. H.; Schoelkopf, R. J. Extending the lifetime of a quantum bit with error correction in superconducting circuits. *Nature* **2016**, *536*, 441−445.

(160) Sivak, V. V.; Eickbusch, A.; Royer, B.; Singh, S.; Tsioutsios, I.; Ganjam, S.; Miano, A.; Brock, B. L.; Ding, A. Z.; Frunzio, L.; Girvin, S.

M.; Schoelkopf, R. J.; Devoret, M. H. Real-time quantum error correction beyond break-even. *Nature* **2023**, *616*, 50−55.

(161) Ni, Z.; Li, S.; Deng, X.; Cai, Y.; Zhang, L.; Wang, W.; Yang, Z.-B.; Yu, H.; Yan, F.; Liu, S.; Zou, C.-L.; Sun, L.; Zheng, S.-B.; Xu, Y.; Yu, D. Beating the break-even point with a discrete-variable-encoded logical qubit. *Nature* **2023**, *616*, 56−60.

(162) Blais, A.; Grimsmo, A. L.; Girvin, S. M.; Wallraff, A. Circuit quantum electrodynamics. *Rev. Mod. Phys.* **2021**, *93*, 025005.

(163) Krastanov, S.; Albert, V. V.; Shen, C.; Zou, C.-L.; Heeres, R. W.; Vlastakis, B.; Schoelkopf, R. J.; Jiang, L. Universal control of an oscillator with dispersive coupling to a qubit. *Phys. Rev. A* **2015**, *92*, 040303.

(164) Eickbusch, A.; Sivak, V.; Ding, A. Z.; Elder, S. S.; Jha, S. R.; Venkatraman, J.; Royer, B.; Girvin, S. M.; Schoelkopf, R. J.; Devoret, M. H. Fast universal control of an oscillator with weak dispersive coupling to a qubit. *Nat. Phys.* **2022**, *18*, 1464−1469.

(165) Kalajdzievski, T.; Weedbrook, C.; Rebentrost, P. Continuous-variable gate decomposition for the Bose-Hubbard model. *Phys. Rev. A* **2018**, *97*, 062311.

(166) Zhang, Y.; Gaddie, A.; Do, H.-V.; Biedermann, G. W.; Lewis-Swan, R. J. Simulating a two-component Bose-Hubbard model with imbalanced hopping in a Rydberg tweezer array. *Phys. Rev. A* **2024**, *109*, 053317.

(167) Bravyi, S. B.; Kitaev, A. Y. Fermionic Quantum Computation. *Annals of Physics* **2002**, *298*, 210−226.

(168) Ortiz, G.; Gubernatis, J. E.; Knill, E.; Laflamme, R. Quantum algorithms for fermionic simulations. *Phys. Rev. A* **2001**, *64*, 022319.

(169) Ortiz, G.; Gubernatis, J. E.; Knill, E.; Laflamme, R. Erratum: Quantum algorithms for fermionic simulations [Phys. Rev. A 64, 022319 2001]. *Phys. Rev. A* **2002**, *65*, 029902.

(170) Sawaya, N. P. D.; Huh, J. Quantum Algorithm for Calculating Molecular Vibronic Spectra. *J. Phys. Chem. Lett.* **2019**, *10*, 3586−3591.

(171) Shaw, A. F.; Lougovski, P.; Stryker, J. R.; Wiebe, N. Quantum Algorithms for Simulating the Lattice Schwinger Model. *Quantum* **2020**, *4*, 306.

(172) Dutta, R.; Vu, N. P.; Lyu, N.; Wang, C.; Batista, V. S. Simulating electronic structure on bosonic quantum computers. *arXiv* **2024**, 2404.10222.

(173) Wang, C. S.; Curtis, J. C.; Lester, B. J.; Zhang, Y.; Gao, Y. Y.; Freeze, J.; Batista, V. S.; Vaccaro, P. H.; Chuang, I. L.; Frunzio, L.; Jiang, L.; Girvin, S. M.; Schoelkopf, R. J. Efficient Multiphoton Sampling of Molecular Vibronic Spectra on a Superconducting Bosonic Processor. *Phys. Rev. X* **2020**, *10*, 021060.

(174) Huh, J.; Guerreschi, G. G.; Peropadre, B.; McClean, J. R.; Aspuru-Guzik, A. Boson sampling for molecular vibronic spectra. *Nat. Photonics* **2015**, *9*, 615−620.

(175) Shen, Y.; Lu, Y.; Zhang, K.; Zhang, J.; Zhang, S.; Huh, J.; Kim, K. Quantum optical emulation of molecular vibronic spectroscopy using a trapped-ion device. *Chemical Science* **2018**, *9*, 836−840.

(176) Sparrow, C.; Martín-López, E.; Maraviglia, N.; Neville, A.; Harrold, C.; Carolan, J.; Joglekar, Y. N.; Hashimoto, T.; Matsuda, N.; O'Brien, J. L.; Tew, D. P.; Laing, A. Simulating the vibrational quantum dynamics of molecules using photonics. *Nature* **2018**, *557*, 660−667.

(177) MacDonell, R. J.; Navickas, T.; Wohlers-Reichel, T. F.; Valahu, C. H.; Rao, A. D.; Millican, M. J.; Currington, M. A.; Biercuk, M. J.; Tan, T. R.; Hempel, C.; Kassal, I. Predicting molecular vibronic spectra using time-domain analog quantum simulation. *Chemical Science* **2023**, *14*, 9439−9451.

(178) Weedbrook, C.; Pirandola, S.; García-Patrón, R.; Cerf, N. J.; Ralph, T. C.; Shapiro, J. H.; Lloyd, S. Gaussian quantum information. *Rev. Mod. Phys.* **2012**, *84*, 621−669.

(179) Flühmann, C.; Home, J. P. Direct Characteristic-Function Tomography of Quantum States of the Trapped-Ion Motional Oscillator. *Phys. Rev. Lett.* **2020**, *125*, 043602.

(180) Bertet, P.; Auffeves, A.; Maioli, P.; Osnaghi, S.; Meunier, T.; Brune, M.; Raimond, J. M.; Haroche, S. Direct Measurement of the Wigner Function of a One-Photon Fock State in a Cavity. *Phys. Rev. Lett.* **2002**, *89*, 200402.

(181) Hofheinz, M.; Wang, H.; Ansmann, M.; Bialczak, R. C.; Lucero, E.; Neeley, M.; O'Connell, A. D.; Sank, D.; Wenner, J.; Martinis, J. M.; Cleland, A. N. Synthesizing arbitrary quantum states in a superconducting resonator. *Nature* **2009**, *459*, 546−549.

(182) Vlastakis, B.; Petrenko, A.; Ofek, N.; Sun, L.; Leghtas, Z.; Sliwa, K.; Liu, Y.; Hatridge, M.; Blumoff, J.; Frunzio, L.; Mirrahimi, M.; Jiang, L.; Devoret, M. H.; Schoelkopf, R. J. Characterizing entanglement of an artificial atom and a cavity cat state with Bell's inequality. *Nat. Commun.* **2015**, *6*, 8970.

(183) Lutterbach, L. G.; Davidovich, L. Method for Direct Measurement of the Wigner Function in Cavity QED and Ion Traps. *Phys. Rev. Lett.* **1997**, *78*, 2547−2550.

(184) The CUDA-Q development team CUDA-Q. *GitHub*. https://github.com/NVIDIA/cuda-quantum, accessed 2024−08−27.

(185) Bergholm, V.; et al. PennyLane: Automatic differentiation of hybrid quantum-classical computations. *arXiv* **2022**, 1811.04968.

(186) Javadi-Abhari, A.; Treinish, M.; Krsulich, K.; Wood, C. J.; Lishman, J.; Gacon, J.; Martiel, S.; Nation, P. D.; Bishop, L. S.; Cross, A. W.; Johnson, B. R.; Gambetta, J. M. Quantum computing with Qiskit. *arXiv* **2024**, 2405.08810.

(187) https://github.com/quantumlib/Cirq.git, accessed 2024−08−27.

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

(189) Bayraktar, H.; et al. cuQuantum SDK: A High-Performance Library for Accelerating Quantum Science. *arXiv* **2023**, 2308.01999.

(190) CUDA-Q Hands-on Workshop Materials. *cudaq-libraries*; GitHub, 2024. https://github.com/cudaq-libraries/workshops/tree/2024-chem-review, accessed 2024−08−27.

(191) Choosing hardware for your qsim simulation. *Google*. https://quantumai.google/qsim/choose_hw, accessed 2024−08−27.

(192) Anoshin, M.; Sagingalieva, A.; Mansell, C.; Zhiganov, D.; Shete, V.; Pflitsch, M.; Melnikov, A. Hybrid Quantum Cycle Generative Adversarial Network for Small Molecule Generation. *IEEE Transactions on Quantum Engineering* **2024**, *5*, 1−14.

(193) QNN-1gpu.py. *cudaq-libraries*; GitHub, 2024. https://github.com/cudaq-libraries/workshops/blob/2024-chem-review/cpu-gpu/QNN-1gpu.py, accessed 2024−08−27.

(194) Ding, Q.-M.; Huang, Y.-M.; Yuan, X. Molecular docking via quantum approximate optimization algorithm. *Phys. Rev. Appl.* **2024**, *21*, 034036.

(195) Robert, A.; Barkoutsos, P. K.; Woerner, S.; Tavernelli, I. Resource-efficient quantum algorithm for protein folding. *npj Quantum Inf* **2021**, *7*, 38.

(196) QAOA-molecular-docking.py. *cudaq-libraries*; GitHub, 2024. https://github.com/cudaq-libraries/workshops/blob/2024-chem-review/cpu-gpu/QAOA-molecular-docking.py, accessed 2024−08−27.

(197) Wolf, M.; Shabtai, E.; Howe, B. New NVIDIA CUDA-Q Features Boost Quantum Application Performance. *NVIDIA Developer*, 2024. https://developer.nvidia.com/blog/new-nvidia-cuda-q-features-boost-quantum-application-performance/, accessed 2024−08−27.

(198) Smelyanskiy, M.; P. D. Sawaya, N.; Aspuru-Guzik, A. qHiPSTER: The Quantum High Performance Software Testing Environment. *arXiv* **2016**, 1601.07195.

(199) Gate fusion. *NVIDIA cuQuantum*, ver. 25.03.0; NVIDIA. https://docs.nvidia.com/cuda/cuquantum/latest/custatevec/overview.html#gate-fusion, accessed 2024−08−27.

(200) CUDA-Q Circuit Simulation Backends. *NVIDIA CUDA-Q*; NVIDIA. https://nvidia.github.io/cuda-quantum/latest/using/backends/simulators.html#single-gpu, accessed 2024−08−27.

(201) c2h4-24q.py. *cudaq-libraries*; GitHub, 2024. https://github.com/cudaq-libraries/workshops/blob/2024-chem-review/gate-fusion/c2h4-24q.py, accessed 2024−08−27.

(202) CUDA-Q Backends. *NVIDIA CUDA-Q*; NVIDIA. https://nvidia.github.io/cuda-quantum/latest/using/backends/backends.html, accessed 2024−08−27.

(203) QNN-mgpu.py. *cudaq-libraries*; GitHub, 2024. https://github.com/cudaq-libraries/workshops/blob/2024-chem-review/multi-QPU/QNN-mgpu.py, accessed 2024−08−27.

(204) Stair, N. H.; Huang, R.; Evangelista, F. A. a Multireference Quantum Krylov Algorithm for Strongly Correlated Electrons. *J. Chem. Theory Comput.* **2020**, *16*, 2236.

(205) Cohn, J.; Motta, M.; Parrish, R. M. Quantum Filter Diagonalization with Compressed Double-Factorized Hamiltonians. *PRX Quantum* **2021**, *2*, 040352.

(206) mgpu_ghz.py. *cudaq-libraries*; GitHub, 2024. https://github.com/cudaq-libraries/workshops/blob/2024-chem-review/multi-GPU/mgpu_ghz.py, accessed 2024−08−27.

(207) NVLink and NVLink Switch. *NVIDIA*. https://www.nvidia.com/en-us/data-center/nvlink/, accessed 2024−08−27.

(208) CUDA Toolkit Documentation. *CUDA Toolkit*, ver. 12.8.1; NVIDIA. https://docs.nvidia.com/cuda/cuda-runtime-api/index.html, accessed 2024−08−27.

(209) McPhee, S. NVIDIA Announces DGX GH200 AI Supercomputer. *NVIDIA*, 2023. https://nvidianews.nvidia.com/news/nvidia-announces-dgx-gh200-ai-supercomputer, accessed 2024−08−27.

(210) QNN-remote-mQPU.ipynb. *cudaq-libraries*; GitHub, 2024. https://github.com/cudaq-libraries/workshops/blob/2024-chem-review/remote-mQPU/QNN-remote-mQPU.ipynb, accessed 2024−08−27.

(211) Gray, J.; Kourtis, S. Hyper-optimized tensor network contraction. *Quantum* **2021**, *5*, 410.

(212) Orús, R. A practical introduction to tensor networks: Matrix product states and projected entangled pair states. *Annals of Physics* **2014**, *349*, 117−158.

(213) Orús, R. Tensor networks for complex quantum systems. *Nat. Rev. Phys.* **2019**, *1*, 538.

(214) https://nvidia.github.io/cuda-quantum/latest/using/backends/simulators.html#tensor-network-simulators, accessed 2024−08−27.

(215) Haidar, A.; Fang, L. Scaling Quantum Circuit Simulation with NVIDIA cuTensorNet. *NVIDIA Developer*. https://developer.nvidia.com/blog/scaling-quantum-circuit-simulation-with-cutensornet/, accessed 2024−08−27.

(216) Overview. *NVIDIA cuQuantum*, ver. 25.03.0; NVIDIA. https://docs.nvidia.com/cuda/cuquantum/latest/cutensornet/overview.html, accessed 2024−08−27.

(217) DiVincenzo, D. P. The physical implementation of quantum computation. *Fortschritte der Physik: Progress of Physics* **2000**, *48*, 771−783.

(218) Shende, V. V.; Bullock, S. S.; Markov, I. L. Synthesis of Quantum Logic Circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* **2006**, *25*, 1000−1010.

(219) Mottonen, M.; Vartiainen, J. J.; Bergholm, V.; Salomaa, M. M. Transformation of quantum states using uniformly controlled rotations. *arXiv* **2004**, 0407010.

(220) Pérez-Salinas, A.; Cervera-Lierta, A.; Gil-Fuster, E.; Latorre, J. I. Data re-uploading for a universal quantum classifier. *Quantum* **2020**, *4*, 226.

(221) Shin, S.; Teo, Y. S.; Jeong, H. Exponential data encoding for quantum supervised learning. *Phys. Rev. A* **2023**, *107*, 012422.

(222) Zylberman, J.; Debbasch, F. Efficient quantum state preparation with Walsh series. *Phys. Rev. A* **2024**, *109*, 042401.

(223) Shukla, A.; Vedula, P. An efficient quantum algorithm for preparation of uniform quantum superposition states. *Quantum Information Processing* **2024**, *23*, 38.

(224) Gleinig, N.; Hoefler, T. An Efficient Algorithm for Sparse Quantum State Preparation. In *Proceedings of the 2021 58th ACM/IEEE Design Automation Conference (DAC)*; IEEE, 2021; pp 433−438. DOI: 10.1109/DAC18074.2021.9586240

(225) Zhang, X.-M.; Yuan, X. Circuit complexity of quantum access models for encoding classical data. *npj Quantum Information* **2024**, *10*, 42.

(226) Larocca, M.; Thanasilp, S.; Wang, S.; Sharma, K.; Biamonte, J.; Coles, P. J.; Cincio, L.; McClean, J. R.; Holmes, Z.; Cerezo, M. A Review of Barren Plateaus in Variational Quantum Computing. *arXiv* **2024**, 2405.00781.

(227) Ragone, M.; Bakalov, B. N.; Sauvage, F.; Kemper, A. F.; Ortiz Marrero, C.; Larocca, M.; Cerezo, M. A Lie algebraic theory of barren plateaus for deep parameterized quantum circuits. *Nat. Commun.* **2024**, *15*, 7172.

(228) Warren, S.; Wang, Y.; Benavides-Riveros, C. L.; Mazziotti, D. A. Exact ansatz of fermion-boson systems for a quantum device. *Phys. Rev. Lett.* **2024**, *133*, 080202.

(229) Smart, S. E.; Mazziotti, D. A. Verifiably exact solution of the electronic Schrödinger equation on quantum devices. *Phys. Rev. A* **2024**, *109*, 022802.

(230) Bermejo, P.; Braccia, P.; Rudolph, M. S.; Holmes, Z.; Cincio, L.; Cerezo, M. Quantum Convolutional Neural Networks are (Effectively) Classically Simulable. *arXiv* **2024**, 2408.12739.

(231) Cerezo, M.; Larocca, M.; García-Martín, D.; Diaz, N. L.; Braccia, P.; Fontana, E.; Rudolph, M. S.; Bermejo, P.; Ijaz, A.; Thanasilp, S.; Anschuetz, E. R.; Holmes, Z. Does provable absence of barren plateaus imply classical simulability? Or, why we need to rethink variational quantum computing. *arXiv* **2024**, 2312.09121.

(232) Mitarai, K.; Negoro, M.; Kitagawa, M.; Fujii, K. Quantum Circuit Learning. *Phys. Rev. A* **2018**, *98*, 032309.

(233) Schuld, M.; Bergholm, V.; Gogolin, C.; Izaac, J.; Killoran, N. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A* **2019**, *99*, 032331.

(234) 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. *arXiv* **2023**, 2305.13362.

(235) Powell, M. J. D. In *Advances in Optimization and Numerical Analysis*; Gomez, S., Hennart, J.-P., Eds.; Springer Netherlands: Dordrecht, 1994; pp 51−67.

(236) Kulshrestha, A.; Liu, X.; Ushijima-Mwesigwa, H.; Safro, I. Learning To Optimize Quantum Neural Network Without Gradients. *arXiv* **2023**, 2304.07442.

(237) Wiedmann, M.; Hölle, M.; Periyasamy, M.; Meyer, N.; Ufrecht, C.; Scherer, D. D.; Plinge, A.; Mutschler, C. An Empirical Comparison of Optimizers for Quantum Machine Learning with SPSA-based Gradients. *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* **2023**, 450−456.

(238) Google Quantum AI and Collaborators; et al. Quantum error correction below the surface code threshold. *Nature* **2025**, *638*, 920−926.

(239) Brock, B. L.; Singh, S.; Eickbusch, A.; Sivak, V. V.; Ding, A. Z.; Frunzio, L.; Girvin, S. M.; Devoret, M. H. Quantum Error Correction of Qudits Beyond Break-Even. *Nature* **2025**, *641* (8063), 612−618.