

# Understanding Parameterized Quantum Circuit Learning for Quantum Chemical Applications

Grier M. Jones,<sup>†,‡</sup> Nick Taylor,<sup>†</sup> Viki Kumar Prasad,<sup>†,‡,¶,§</sup> Ulrich Fekl,<sup>\*,‡</sup> and Hans-Arno Jacobsen<sup>\*,†</sup>

<sup>†</sup> *The Edward S. Rogers Sr. Department of Electrical and Computer Engineering, University of Toronto, 10 Kings College Road, Toronto, Ontario, Canada M5S 3G4*

<sup>‡</sup> *Department of Chemical and Physical Sciences, University of Toronto Mississauga, 3359 Mississauga Road, Mississauga, Ontario, Canada L5L 1C6*

<sup>¶</sup> *Data Science Institute, University of Toronto, 700 University Ave 10th floor, Toronto, ON M7A 2S4*

<sup>§</sup> *Current Affiliation: Department of Chemistry, University of Calgary, 2500 University Drive NW, Calgary, AB T2N 1N4*

E-mail: [ulrich.fekl@utoronto.ca](mailto:ulrich.fekl@utoronto.ca); [jacobsen@eetcg.toronto.edu](mailto:jacobsen@eetcg.toronto.edu)

## Abstract

Within the quantum machine learning (QML) field, parameterized quantum circuits (PQCs), built using fixed and parameterized gates, offer a hybrid approach for complex machine learning tasks. While many potential use cases have been proposed, the exploration of relevant datasets for chemists is lacking. Our study seeks to understand the possible advantages and disadvantages of PQCs for two chemically relevant datasets: one based on the bond separation energies of 49 different classes

of bonds, called the BSE49 dataset, and another consisting of water confirmations, where coupled-cluster singles and doubles (CCSD) wave functions are predicted using electronic structure theory data from lower-level methods using the data-driven coupled-cluster (DDCC) method. In our study, we examine a combinatorial space of 14 data encoding layers and 12 variational (ansatz) layers, for a combined total of 168 PQCs. To calibrate our PQCs, we utilize a dataset of noisy linear, quadratic, and sine functions to explore the effects of the circuit width and depth, the effects of the feature set size, and various error mitigation techniques. Following this step, we similarly examine our chemically relevant datasets. Our work highlights the difficulties in encoding classical molecular representations in a PQC for predicting bond separation energies and the aptitude for PQCs for predicting molecular wave functions.

## 1 Introduction

In recent years, machine learning (ML) has emerged as a popular tool in chemistry for revealing new patterns in data, providing new insights beyond simple models and human experience, accelerating computations, and analyzing chemical space. For computational chemists, the primary goal of applying ML is often to bypass the explicit calculation of molecular properties, which can be computationally expensive for large datasets.<sup>1</sup> ML can be applied to a diverse set of applications including, but not limited to, accelerating molecular simulations,<sup>2–4</sup> determining molecular properties,<sup>5–9</sup> and for discovering new catalysts,<sup>10–12</sup> drugs,<sup>13,14</sup> and materials.<sup>15–17</sup> Since these applications can become resource intensive, regarding the generation of training data using traditional computational chemistry approaches and the training of large-scale ML models, computational chemistry and ML practitioners have explored new acceleration platforms, such as graphical processing units (GPUs) and tensor processing units (TPUs).<sup>3,13,18–22</sup>

Alternatively, computational approaches incorporating the quantum mechanical principles of superposition and entanglement, called quantum computing (QC), have become increasingly popular for chemical applications due to possible quantum speedups for quantum chemical calcula-

tions.<sup>23</sup> For computational chemistry, methods such as the quantum phase estimation (QPE)<sup>24–29</sup> algorithm have been shown to offer exponential speedups over classical methods. Despite the promising speedups, QPE requires long coherence times, while the current generation of quantum processing units (QPUs) are often too noisy for practical applications. Alternatively, methods based on the variational principle, such as the variational quantum eigensolver (VQE),<sup>30–33</sup> have been proposed as a quantum-classical hybrid approach, capable of running on noisy, near-term quantum devices.

While most QC studies that are relevant to computational chemists focus on creating more efficient electronic structure methods on quantum computers,<sup>23,34–36</sup> an approach that combines both ML and QC, is quantum machine learning (QML). Using either formal mathematical proofs or numerical results based on empirical observations, QML has shown potential quantum speedups for various applications using a diverse set of implementations.<sup>37</sup> While several classes of QML algorithms have shown promise for providing flexible ML models, parameterized quantum circuits (PQCs) are capable of achieving non-trivial results on near-term quantum hardware. PQCs formulate the ML algorithm as a variational problem optimized using a hybrid approach using both classical and quantum hardware.<sup>38</sup> Like classical ML approaches, PQCs have been applied for several chemistry use cases such as drug<sup>39–45</sup> and materials discovery,<sup>46–48</sup> the prediction of proton affinities,<sup>49</sup> and experimental molecular properties, including the log solubility in water, melting point, octanol/water distribution coefficient, and hydration free energy of small molecules in water.<sup>50</sup> Despite the broad range of topics covered in these studies and the interest among computational chemists in exploring PQCs for chemical applications, studies analyzing the potential benefits or drawbacks of using QML for quantum chemistry are lacking.

In this study, we address this by analyzing a diverse set of PQCs using two datasets related to quantum chemistry. The first dataset, BSE49 consists of bond separation energies (BSEs) of 49 unique bond types, calculated using the highly accurate (RO)CBS-QB3 composite method.<sup>51</sup> The second dataset consists of water conformers calculated with coupled-cluster singles and doubles (CCSD) using the data-driven coupled-cluster (DDCC) scheme of Townsend and Vogiatzis.<sup>52,53</sup>

Both datasets offer a unique perspective on the aptitude of applying PQCs on classical and quantum data<sup>54</sup> since the models based on BSE49 rely on classical molecular representations,<sup>55</sup> such as Molecular ACCess Systems (MACCS)<sup>56</sup> or Morgan fingerprints,<sup>57,58</sup> as input, while the input features in the DDCC method encode explicit quantum information related to the molecular electronic structure. In this study, we introduce *qgress* a modular Python framework, based on PennyLane<sup>59</sup> and Qiskit,<sup>60</sup> for exploring PQCs for regression-based QML tasks. To this end, we explore the effects of classical and quantum data using a broad set of 168 unique PQCs, based on a combination of 14 data encoding and 12 variational layers. We then perform an analysis of circuit depth on model performance, using two different expansion strategies, one based on data re-uploading<sup>61</sup> and the other based on increasing the number of model parameters using additional variational layers. Using these insights, we then analyze how the best model performs using fake and real Qiskit backends. Lastly, we provide a detailed discussion on the efficiency and performance of PQCs, with insights into what quantum advantage could mean concerning classical ML models.

## 2 Methods

PQCs typically consist of three components: encoding layers that map features onto a quantum circuit, variational layers with classically optimized parameters, and measurement layers that provide numerical estimates of the regression target values.<sup>39</sup> Choosing the optimal encoding layer can be a challenging task due to the costs associated with mapping the input data to qubits.<sup>37</sup> Due to this fact, we choose existing encoding layers that have shown promising results for regression tasks such as Mitarai (M),<sup>62</sup> single- (A1) and double-angle (A2) encoding layers found in Ref.<sup>39</sup>, along with the instantaneous quantum polynomial (IQP) circuit found in Refs.<sup>63</sup> and<sup>64</sup>.

To maintain a general notation, in the following section, we follow the notations derived from Ref.<sup>39</sup>. Encoding layers work by mapping a given  $d$ -dimensional feature vector,  $\mathbf{x} = (x_1, x_2, \dots, x_d)^T \in \mathbb{R}^d$ , normalized on the range  $[-1, 1]$ , onto a quantum circuit using a unitary matrix, denoted as

$U_{\Phi(\mathbf{x})}$ , to produce the quantum state  $U_{\Phi(\mathbf{x})} |0\rangle^{\otimes n}$ , where  $n$  is the number of qubits. More generally, the encoding layer can also incorporate both unitary and entanglement gates, such that

$$U_{\Phi(x)} = \prod_l E_{\text{ent}}^l U_{\phi_l(\mathbf{x})}, \quad (1)$$

where,  $E_{\text{ent}}^l$  denotes the entangling gates, e.g., CNOT or CZ, and  $U_{\phi_l(\mathbf{x})}$  denotes the choice of encoding unitaries.

In this study, like in Ref.<sup>39</sup>, we explore encoders with  $l = 1, 2$ . When  $l = 1$ , this forms the simplest encoding layers with  $E_{\text{ent}}^1$  being absent, e.g. corresponding to the identity (**I**). Using the previously defined general notation, the simplest encoding layer, single-angle encoding (Fig. 1 **(a)**), is then defined as,

$$U_{A1} = \prod_{i=0}^n R_i^Y(x_i), \quad (2)$$

where  $R_i^Y$  denotes a parameterized Y rotation gate on qubit  $i$ . Single-angle encoding, or qubit encoding, is structured similarly to a product of unentangled quantum states and have a similar mathematical structure as a product of local kernels where each  $x_i$  is encoded in a local feature map.<sup>38,65</sup>

The next encoder, double-angle encoding (Fig. 1 **(b)**) utilizes a parameterized Y rotation gate on qubit  $i$ , similar to  $A1$ , with the addition of a parameterized Z rotation gate on qubit  $i$ , denoted as

$$U_{A2} = \prod_{i=0}^n R_i^Z(x_i) R_i^Y(x_i), \quad (3)$$

which introduces additional redundancy by encoding two angles on the Bloch sphere. An extension of the double-angle encoding is the Mitarai encoding layer (Fig. 1 **(c)**), which includes an arccosine function on the parameterized Z gate and arcsine on the parameterized Y gate,

$$U_M = \prod_{i=0}^n R_i^Z(\arccos(x_i^2)) R_i^Y(\arcsin(x_i^2)). \quad (4)$$

This unitary is physically motivated by expanding a density operator in terms of a set of Pauli

operators, as highlighted in Ref.<sup>62</sup>

The last and most complex encoding layer is the instantaneous quantum polynomial (IQP) (Fig. 1 (d)) proposed by Havlicek *et al.*,<sup>64</sup>

$$U_{\text{IQP}} = \prod_{i=0}^n H_i R_i^Z(x_i) \prod_{i < j} ZZ_{ij}, \quad (5)$$

where  $H_i$  denotes a Hadamard gate on qubit  $i$  and  $ZZ_{ij}$  denotes a two-qubit entangling gate defined as  $ZZ_{ij} = e^{-ix_i x_j \sigma_z \otimes \sigma_z}$ . It should be noted that under specific complexity and theoretic assumptions, IQP circuits cannot be efficiently simulated using classical resources and therefore offer a circuit that can only be simulated efficiently using quantum resources.<sup>66,67</sup>

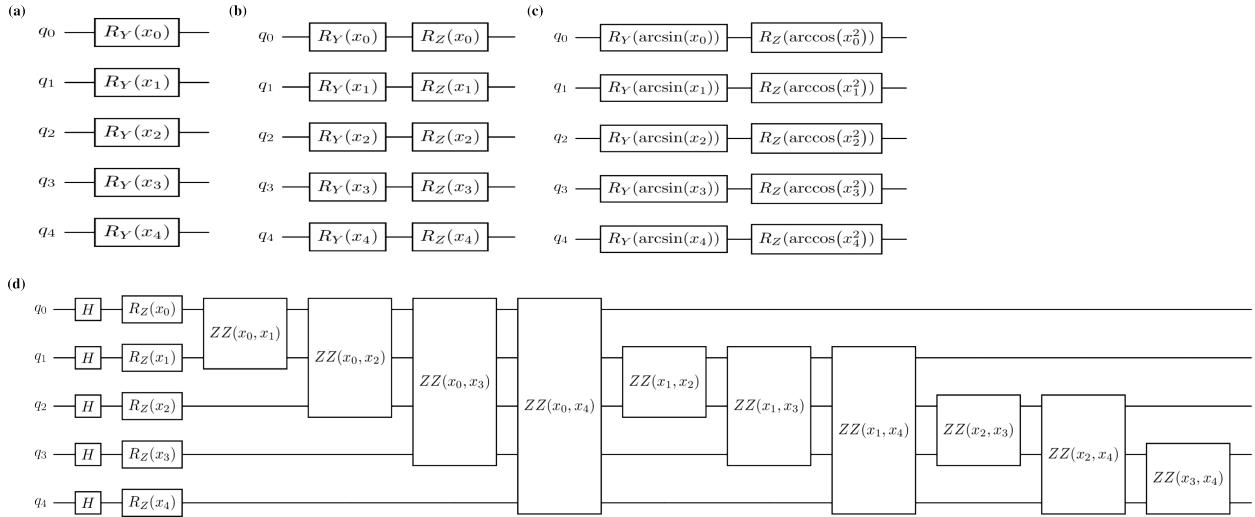


Figure 1: Five qubit examples of the (a) single angle (A1), (b) double angle (A2), (c) Mitarai (M), and (d) Instantaneous Quantum Polynomial (IQP) encoders used in this study.

When  $l = 2$ , more complex circuits can be generated, like in Ref.<sup>39</sup>, where we choose the entanglement gates,  $E_{\text{ent}}^1$  and  $E_{\text{ent}}^2$ , to be equivalent and the encoding layer takes the following form,  $U_{\Phi(\mathbf{x})} = E_{\text{ent}} U_{\phi_2(\mathbf{x})} E_{\text{ent}} U_{\phi_1(\mathbf{x})}$ . Additionally, we exclude IQP encoding when  $l = 2$  due to the increased computational costs associated with the circuit depth when compared to the expanded models based on A1, A2, and M encoding. Therefore, when we exclude IQP, there are five unique combinations of  $U_{\phi_1(\mathbf{x})}$  and  $U_{\phi_2(\mathbf{x})}$  (M-M, A1-A1, A2-A2, M-A1, and M-A2) and two different en-

tanglement layer options (CNOT and CZ) for a total of 10 encoding circuits with  $l = 2$ . The common notation for these circuits is  $U_{\phi_1(\mathbf{x})} - U_{\phi_2(\mathbf{x})} - E_{\text{ent}}$  where two example encoding circuits are M–M–CNOT and M–A1–CNOT. The 14 encoding circuits used throughout this study are shown in Table 1, where the first column shows the label, second column the first unitary ( $U_{\phi_1(\mathbf{x})}$ ), third column the second unitary ( $U_{\phi_2(\mathbf{x})}$ ), and last column the entanglement gates ( $E_{\text{ent}}$ ).

Table 1: The 14 encoding layers used in this study where the first column shows the encoding circuit name, second column the unitary for  $l = 1$ , third column the unitary for  $l = 2$ , and last column the corresponding entanglement gates.

Name	$U_{\phi_1(\mathbf{x})}$	$U_{\phi_2(\mathbf{x})}$	$E_{\text{ent}}$
A1	$U_{A1}$	—	—
A2	$U_{A2}$	—	—
M	$U_M$	—	—
IQP	$U_{\text{IQP}}$	—	—
A1–A1–CNOT	$U_{A1}$	$U_{A1}$	$E_{\text{CNOT}}$
A2–A2–CNOT	$U_{A2}$	$U_{A2}$	$E_{\text{CNOT}}$
M–M–CNOT	$U_M$	$U_M$	$E_{\text{CNOT}}$
M–A1–CNOT	$U_M$	$U_{A1}$	$E_{\text{CNOT}}$
M–A2–CNOT	$U_M$	$U_{A2}$	$E_{\text{CNOT}}$
A1–A1–CZ	$U_{A1}$	$U_{A1}$	$E_{\text{CZ}}$
A2–A2–CZ	$U_{A2}$	$U_{A2}$	$E_{\text{CZ}}$
M–M–CZ	$U_M$	$U_M$	$E_{\text{CZ}}$
M–A1–CZ	$U_M$	$U_{A1}$	$E_{\text{CZ}}$
M–A2–CZ	$U_M$	$U_{A2}$	$E_{\text{CZ}}$

The next layer in a PQC which follows the encoding layer, is the variational layer. The variational layers introduce trainable parameters, that can be optimized using classical compute resources, into the quantum circuit and provide flexibility to the QML models. These layers have a general form, defined as,

$$U(\boldsymbol{\theta}) = \prod_v U_v(\boldsymbol{\theta}_v), \quad (6)$$

where  $\boldsymbol{\theta}$  denote the variational parameters,  $v$  denote the number of times that the layer is repeated within the circuit, and all entanglement gates are implicitly included in  $U_v(\boldsymbol{\theta}_v)$ . In our study, we examine 12 variational circuits, found in,<sup>68</sup> as shown in Fig. 2. The unique names of the 12 variational circuits are as follows: Modified-Pauli-CRZ (Fig. 2(a)), Modified-Pauli-CRX (Fig.

2(**b**)), Efficient-CRZ (Fig. 2(**c**)), Efficient-CRX (Fig. 2(**d**)),, HWE-CNOT (Fig. 2(**e**)), HWE-CZ (Fig. 2(**f**)), ESU2 (Fig. 2(**g**)), Full-Pauli-CRZ (Fig. 2(**h**)), Full-Pauli-CRX (Fig. 2(**i**)), Hadamard (Fig. 2(**j**)), Full-CRZ (Fig. 2(**k**)), and Full-CRX (Fig. 2(**l**)).

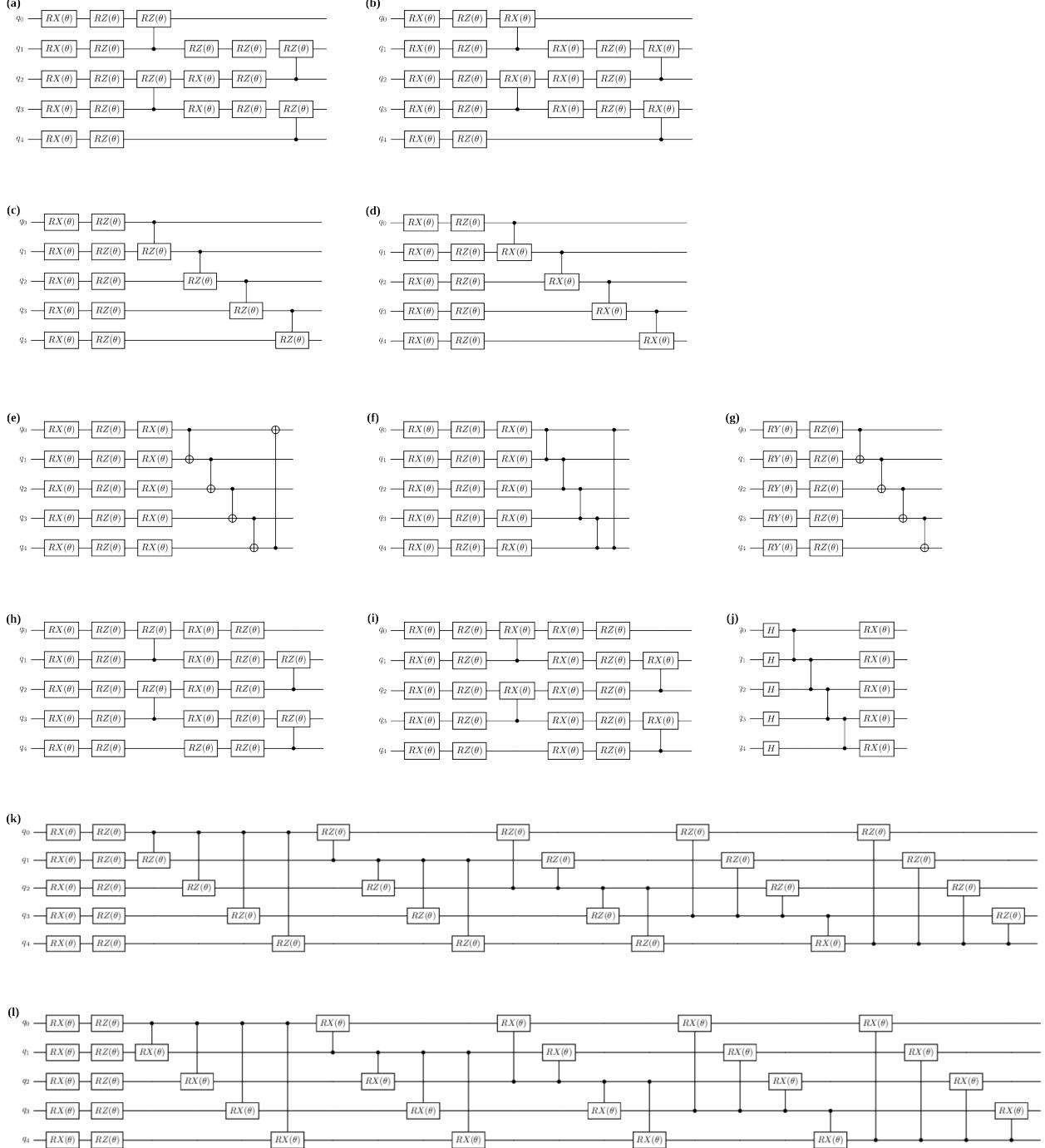


Figure 2: The 12 variational circuits include (a) Modified-Pauli-CRZ, (b) Modified-Pauli-CRX, (c) Efficient-CRZ, (d) Efficient-CRX, (e) HWE-CNOT, (f) HWE-CZ, (g) ESU2, (h) Full-Pauli-CRZ, (i) Full-Pauli-CRX, (j) Hadamard, (k) Full-CRZ, and (l)Full-CRX

To introduce more model parameters and increase the expressibility and nonlinearity of the PQCs, two expansion strategies can be explored. The first we denoted as the number of ansatz

layers (AL), where  $v$  can be increased to introduce more unique trainable parameters into the PQC. The second expansion strategy builds on the work of Pérez-Salinas *et al.* that showed data re-uploading in PQCs is equivalent to the Universal Approximation Theorem for artificial neural networks.<sup>61</sup> This is achieved by combining Eqs. 1 and 6 into a general circuit,

$$|\Psi\rangle = U(\boldsymbol{\theta})U_{\Phi(\mathbf{x})}|0\rangle^{\otimes n} = \prod_k \left( \prod_v U_v(\boldsymbol{\theta}_v) \prod_l E_{\text{ent}}^l U_{\phi_l(\mathbf{x})} \right) |0\rangle^{\otimes n}, \quad (7)$$

where  $k$  denotes the re-upload depth (RUD) of the circuit.

The last component of a PQC is measurement, which is required to recover the predicted target values,  $\hat{y}_i$ , of the machine learning model that is used to update the model parameters. This is performed by measuring the quantum state,  $|\Psi\rangle$ , using the Pauli Z operator on the first qubit denoted as,

$$\hat{y}_i = \langle \Psi | Z_0 | \Psi \rangle_i, \quad (8)$$

and passing the set of predicted target values,  $\hat{\mathbf{y}} = (\hat{y}_1, \dots, \hat{y}_N) \in \mathbb{R}^N$ , where  $N$  is the number of samples, to the loss function,  $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$ , where  $y_i$  belongs to the set of true target values  $\mathbf{y} = (y_1, \dots, y_N) \in \mathbb{R}^N$ . While  $\mathcal{L}$  can be any loss function relevant for regression-based ML tasks, we choose to mean square error to be the loss function,

$$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2. \quad (9)$$

### 3 Implementation

To explore the large combination of PQCs previously mentioned, we introduce *qregress* (Fig. 3), a modular Python package based on PennyLane<sup>59</sup> and Qiskit.<sup>60</sup> The portion of the code based on PennyLane is capable of performing state-vector simulation using Qulacs,<sup>69</sup> noisy simulation using *qiskit-aer* with the *FakeQuebec* backend, along with access to real devices accessible on the IBM Quantum Platform using the PennyLane-Qiskit plugin. Additionally, to train and test regression models that require many circuit executions, we ported our PennyLane code to Qiskit so that the

Qiskit Batch Execution mode could be utilized. This was a vital step for running on *ibm\_quebec* since Qiskit Sessions, as implemented in the PennyLane-Qiskit plugin, are too cumbersome to train PQCs as  $N$ , the number of training samples, increases. In practice, all state-vector simulations were performed using PennyLane, while all calculations using *FakeQuebec* and *ibm\_quebec* are run using Qiskit. All state-vector simulations with 5 qubits were optimized using 1000 iterations, while 16 qubit are initially ran using 250 iterations and longer runs, using 1000 iterations, use early stopping due to computational cost considerations. State-vector simulations were also optimized using the Simultaneous Perturbation Stochastic Approximation method (SPSA), as implemented in PennyLane since we found that SPSA requires less steps to optimize when compared to other optimizers for state-vector simulation. For the experiments ran using *FakeQuebec* and *ibm\_quebec*, we utilized the Constrained Optimization By Linear Approximation (COBYLA) optimizer, as implemented in SciPy.<sup>70</sup> For all models, features ( $\mathbf{x}$ ) and target values ( $\mathbf{y}$ ) were scaled using the MinMaxScaler, as implemented in Scikit-learn,<sup>71</sup> such that all features and target values are  $\mathbb{R} \in [-1, 1]$ .

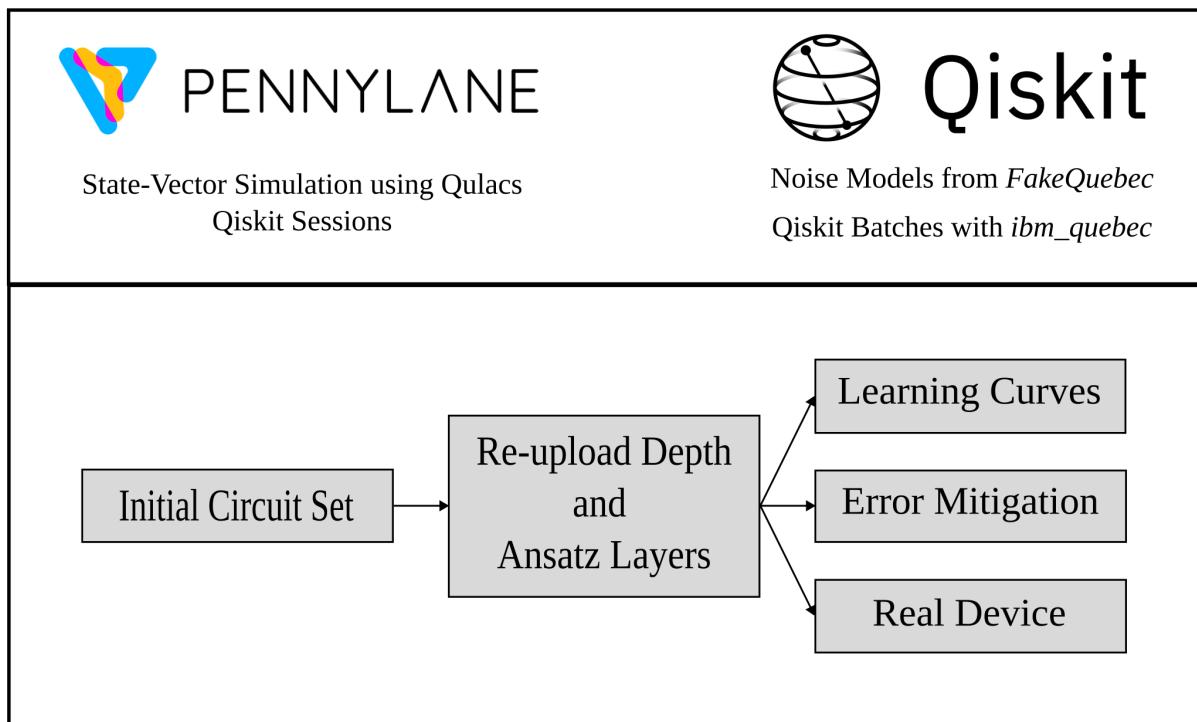


Figure 3: REMAKE to include cool code pic. Make this figure highlight the modularity of *qregress*

## 4 Datasets

The first quantum chemistry dataset we explore is BSE49, which contains bond separation energies (BSEs), for homolytic bond cleavage of covalently bonded molecules, such as  $A-B \longrightarrow A^+ + B^-$ .<sup>51</sup> This dataset consists of 49 unique A-B type bonds with 4394 datapoints, 1951 of which are existing structures, while the remaining 2443 are hypothetical structures. BSE49 provides highly accurate BSEs, calculated using the (RO)CBS-QB3<sup>76–78</sup> composite quantum chemistry method, that is useful for benchmarking ML-based applications.

An important aspect of ML in chemistry is the choice of molecular representation, or how the molecule is represented in the machine learning models.<sup>55</sup> In general, molecular representations can be partitioned into three groups, graph-, topology-, and physics-based representations, all of which were explored to provide a comprehensive overview of how they perform using classical models as a baseline. During the preprocessing stage, before converting to the XYZ coordinates into our molecular representation of choice, the set of hypothetical structures was reduced to 2436 molecules due to issues with valency exceptions when converting XYZ coordinates into RDKit mol objects. Using RDKit<sup>79</sup> we examined three commonly applied graph-based molecular representations, Molecular ACCess Systems (MACCS),<sup>56</sup> Morgan or extended-connectivity fingerprints,<sup>57,58</sup> and RDKit fingerprints. All three of these methods use traversals of the molecular graphs to encode various structural details into bit vectors. Lastly, we explore both topology- and physics-based molecular representations, both of which encode the three-dimensional structure of molecules in various, unique ways. Persistent images (PIs) are a topology-based fingerprint that uses persistence homology to encode topological information of three-dimensional molecular structures into fixed-dimension images.<sup>80–82</sup> We use the implementation from Townsend *et al.*,<sup>81</sup> which uses the Ripser Python package to generate PIs.<sup>83</sup> Lastly, we explore two physics-based representations, Coulomb matrices (CMs)<sup>84</sup> and smooth overlap of atomic positions (SOAPs), that were generated using DScribe.<sup>85</sup> Both methods encode physical information regarding the atomic environments of each molecule, where CMs encode the Coulomb repulsion between atoms using nuclear charges, while SOAPs encode the similarities between atomic neighborhoods us-

ing kernels. Due to the computational cost associated with computing the regularized entropy match (REMatch) kernel over the set of 2436 molecules with the SOAPs representation, we excluded SOAPs from the overall discussion. For each of the molecular representations, we analyzed two different methods for representing the components of the BSE reaction. The first represents the feature vector as the products subtracted from the reactants, denoted as *sub* (e.g.,  $\text{CM}_{\text{sub}} = (\text{CM}_{\text{A}\cdot} + \text{CM}_{\text{B}\cdot}) - \text{CM}_{\text{A-B}}$ ), similar to the method used in Ref.,<sup>86</sup> and the second represents the feature vector using only the reactant (A-B), denoted as *AB* (e.g.,  $\text{CM}_{\text{AB}} = \text{CM}_{\text{A-B}}$ ).

Using this set of molecular representations, we also examine a diverse set of classical regression models to help choose the optimal representation for our diverse set of PQCs. The classical models we analyze include ridge, lasso, elastic net, *k*-nearest-neighbors, random forest, gradient boosting, support vector machines, kernel ridge, and Gaussian process regression as implemented in scikit-learn.<sup>71</sup> As highlighted in Fig. S1, we found that the best molecular representation across all models test was Morgan fingerprints using the *sub* formulation, which we use for all experiments using the PQCs. Additionally, one challenge of applying classical molecular representations for ML tasks using PQCs is reducing the set of features, often containing hundreds or thousands of features per sample, to the number of qubits used on the quantum device. For example, at the onset of this project the standard number of qubits on the IBM quantum devices was 5 or 16 qubits, which requires reducing the initial set of 2048 features per sample for Morgan fingerprints down to 5 or 16 features. To perform this reduction, we explored two different methods, SHapley Additive ExPlanation analysis (SHAP),<sup>87</sup> which uses cooperative game theory to determine feature importance, and principal component analysis (PCA), which is a dimensionality reduction technique that captures the direction a datasets variance, as implemented in scikit-learn.<sup>71</sup> In Figs. S2b and S2a, we compare the two reduction techniques, for both 5 and 16 features, with the initial model containing 2048 features. The initial model has mean absolute errors (MAE) of 1.91 and 4.98 kcal/mol and  $R^2$ s of 0.99 and 0.91 for the training and test set, respectively. Using SHAP, we found that the model containing 5 features has an MAE of 16.08 kcal/mol and an  $R^2$  of 0.39 for the training set, while the test set has an MAE of 15.86 kcal/mol and an  $R^2$  of 0.42. For the

model with 16 features, we observe improvements in both the training and test sets, with MAEs of 10.48 and 11.08 kcal/mol and  $R^2$  values of 0.69 and 0.68, respectively. For the models using 5 and 16 features reduced through PCA, the training sets yield MAEs of 4.09 and 3.23 kcal/mol, while the test sets have MAEs of 10.17 and 8.40 kcal/mol, respectively. The  $R^2$ s also show slight improvement over the feature reductions using SHAP, where the  $R^2$ s using 5 features are 0.95 and 0.69 for the training and test set, respectively, and the model using 16 features has  $R^2$ s of 0.97 and 0.78 for the training and test set, respectively. Overall, the models using SHAP to reduce the feature set offer models with less overfitting, while sacrificing accuracy, when compared to the models reduced using PCA, which offer more accurate models that are more overfit. Based on these insights, for the experiments using PQCs, all BSE49 data is reduced to 5 or 16 qubits using PCA applied on Morgan fingerprint using the *sub* formulation.

The second quantum chemistry dataset we analyze in this paper is generated using the data-driven coupled-cluster (DDCC) scheme of Townsend and Vogiatzis.<sup>52,53</sup> DDCC is an ML-based approach for accelerating the convergence of coupled-cluster singles and doubles (CCSD) calculations by predicting the  $t_2$ -amplitudes of the CCSD wave function (Eq. 10) with features generated using lower-level methods, such as Hartree-Fock (HF) and Møller-Plesset second-order perturbation theory (MP2), that are used to initialize CCSD calculations. Herein, we provide a brief overview of the equations required to understand the DDCC method, starting with the coupled-cluster wave function, which takes the general form,

$$|\Psi_{\text{CC}}\rangle = \exp(\hat{T}) |\Psi_0\rangle \quad (10)$$

where  $\hat{T}$  is the cluster operator and  $|\Psi_0\rangle$  is the reference (Hartree-Fock) wave function. In CCSD, the  $\hat{T}$  operator is truncated to only include single ( $\hat{T}_1$ ) and double ( $\hat{T}_2$ ) excitations. After solving the coupled-cluster equations, the CCSD correlation energy can be formulated as,

$$E_{\text{corr}}^{\text{CCSD}} = \sum_{\substack{a < b \\ i < j}} \langle ij || ab \rangle t_{ij}^{ab} + \sum_{\substack{a < b \\ i < j}} \langle ij || ab \rangle t_i^a t_j^b \quad (11)$$

where  $i$  and  $j$  denote occupied orbitals,  $a$  and  $b$  denote virtual orbitals,  $t_{ij}^{ab}$  correspond to two-electron excitation amplitudes ( $t_2$ -amplitudes),  $t_i^a$  and  $t_j^b$  correspond to one-electron excitation amplitudes ( $t_1$ -amplitudes), and  $\langle ij||ab\rangle$  are two-electron integrals. The objective of the DDCC method is to learn the CCSD  $t_2$ -amplitudes using features generated using HF and MP2, since the CCSD  $t_2$ -amplitudes are initialized using MP2  $t_2$ -amplitudes, defined as,

$$t_{ij(\text{MP2})}^{ab} = \frac{\langle ij||ab\rangle}{\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b} \quad (12)$$

where  $\varepsilon_i$  and  $\varepsilon_j$  denote the orbital energies of the occupied orbitals  $i$  and  $j$ , while the virtual orbitals  $a$  and  $b$  are denoted by  $\varepsilon_a$  and  $\varepsilon_b$ . Due to this important relationship, MP2  $t_2$ -amplitudes are included in the DDCC feature set, along with the numerator ( $\langle ij||ab\rangle$ ), denominator ( $\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b$ ), a binary feature to denote whether the excitation goes to the same virtual orbital, and the orbital energies ( $\varepsilon_i, \varepsilon_j, \varepsilon_a, \varepsilon_b$ ). Additionally, the feature set includes terms related to the individual contributions to the orbital energies, such as the one-electron Hamiltonian ( $h$ ), Coulomb matrix ( $J$ ), exchange matrix  $K$ , and Coulomb and exchange integrals ( $J_a^i, J_b^j, K_i^a, K_j^b$ ). In total, there are 30 features for each CCSD  $t_2$ -amplitude due to the addition of features that denote the sign and magnitude of the MP2  $t_2$ -amplitudes.

Using the DDCC method, we generate a dataset consisting of 199 water molecules obtained from the study of Townsend and Vogiatzis using the STO-3G basis set<sup>88</sup> with the frozen core approximation. All HF, MP2, and CCSD data was generated using Psi4<sup>89</sup> and Psi4Numpy.<sup>90</sup> We will also note that despite using a minimal basis set, this is a data intensive quantum chemistry application for PQCs since for each molecule there are  $(N_{occ})^2(N_{virt})^2$   $t_2$ -amplitudes, where  $N_{occ}$  denotes the number of occupied orbitals and  $N_{virt}$  denotes the number of virtual orbitals. For the water confirmers with 4 occupied and 2 virtual orbitals, there are a total of 64  $t_2$ -amplitudes per molecule. Further details regarding the feature set and implementation can be found in Ref.<sup>52</sup>

Lastly, like the BSE49 dataset, the 30 DDCC features must be reduced to 5 or 16 features using either SHAP or PCA. Unlike the BSE49 dataset, SHAP outperforms PCA due to the direct

relationship between the feature set and target values. Additionally, as highlighted in Fig. S3, when the feature set is reduced to 5 or 16 features, both models have train and test  $R^2$ 's of 1.00. Due to the computational costs of simulating state-vector models on a broad set of PQCs, we will only perform our DDCC models with 5 features, which includes the two-electron integrals ( $\langle ij||ab \rangle$ ), MP2  $t_2$ -amplitudes ( $t_{ij}^{ab}$ ), the magnitude of the MP2  $t_2$ -amplitudes, the difference in orbital energies ( $\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b$ ), and the binary feature denoting whether two-electrons are promoted to the same virtual orbital ( $a = b$ ).

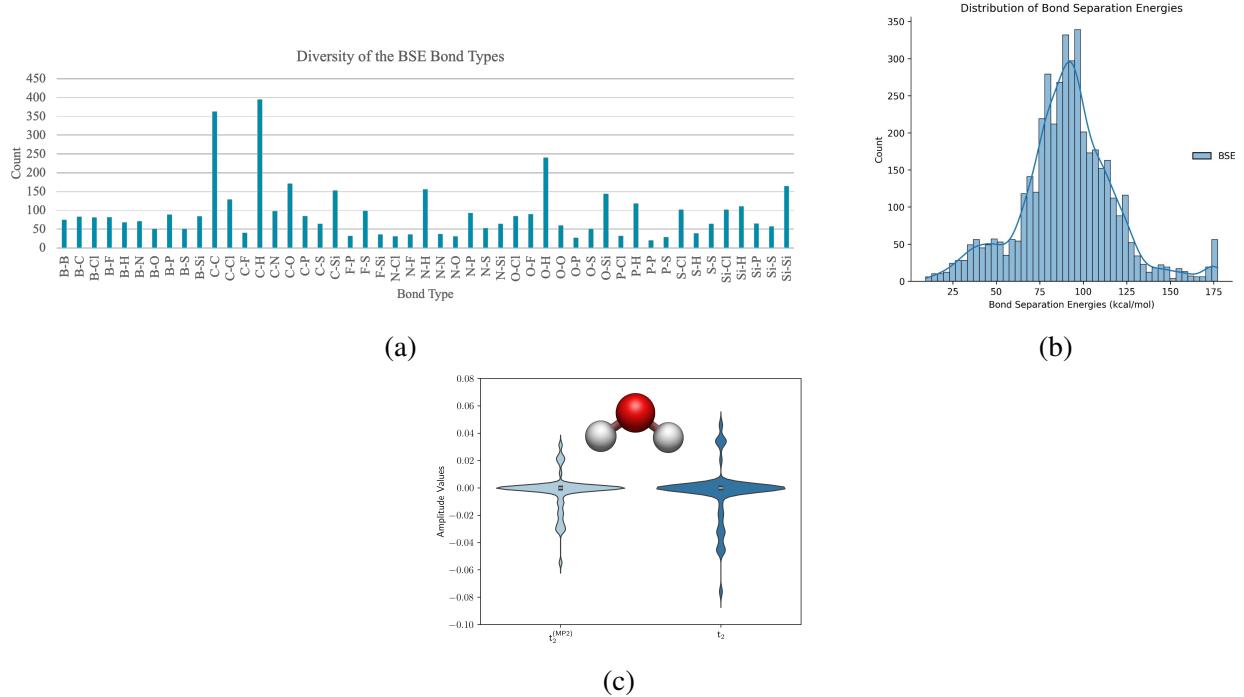


Figure 4: REMAKE: Examples of the datasets explored in this study. For the BSE49 dataset the distributions of the (d) bond types and the (e) bond separation energies in kcal/mol. For the DDCC dataset distributions of the initial MP2  $t_2$ -amplitudes and the optimized CCSD  $t_2$ -amplitudes are shown in (f).

## 5 Results and Discussion

[introduction paragraph] In this section, for both the DDCC and BSE49 datasets, we start our analysis of by examining the performance of the initial set of 168 PQCs with re-upload depths of 1 ( $k = 1$ ) and the number ansatz layers set to 1 ( $v = 1$ ). Using this data, we select the best encoder-

ansatz pair and increase the re-upload depth and number of ansatz layers to 3 and 5 ( $k, v \in 3, 5$ ) to determine how model performance is effected by circuit depth. We then analyze how the best circuit from this set performs using a learning curve, where we hold the number of test points set to 20% and vary the number of training points, to determine how PQC behaviors with varying numbers of training data. Lastly, we provide insights into how the best PQC model behaves on simulated and real IBM backends using *FakeQuebec* and *ibm\_quebec*.

Starting with the 5 qubit BSE49 data, Fig. 5a highlights the performance of the 168 PQC using a heat plot of the coefficients of determination ( $R^2$ ) of the training (left) and test sets (right). The training set  $R^2$  values range from a minimum of -1.4673 to a maximum of 0.1775, with a mean of -0.1865 and a standard deviation of 0.3433. For the test set, the  $R^2$  values range from a minimum of -1.4774 to a maximum of 0.1974, with a mean of -0.1765 and a standard deviation of 0.3446. General trends regarding the performance of the encoders (left) and ansatz (right) layers are highlighted in Fig. 5b using box plots. On average, the best encoding layer is M-M-CNOT with a training and test  $R^2$ 's of -0.0173 and -0.0157, respectively. The best ansatz layer, is Full-CRX with an average training  $R^2$  of 0.1054 and test  $R^2$  of 0.1165. Overall, for the BSE49 dataset using 5 qubits, the best encoder-ansatz pair is A1-A1-CNOT\_Full-CRX training set  $R^2$  of 0.1775 and test set  $R^2$  of 0.1974. Despite A1-A1-CNOT\_Full-CRX being the best encoder-ansatz pair, for explorations of the circuit depth, we chose M-M-CZ\_HWE-CNOT, which offers similar performance training set  $R^2$  of 0.1566 and test set  $R^2$  of 0.1740 but has offers a less deep circuit that is less costly to perform state-vector simulations on.

[This is bad; the 70 removed were originally ran with 250 iter... not 1000 like the rest...] Following this analysis, 70 PQC's were removed based on performance regarding the computational costs of state-vector simulation and model accuracy.

Using the remaining 98 PQC's, we perform a similar analysis using 16 qubits.

sixteen qubit 5c 5d removed the really bad ones from five qubit

pair A2\_Hadamard train  $R^2$  0.0028 test  $R^2$  0.0140 train MAE 19.8071 kcal/mol test MAE 20.0745 kcal/mol

Encoder A1 train  $R^2$  -0.0501 test  $R^2$  -0.0458 train MAE 20.7795 kcal/mol test MAE 21.3186 kcal/mol

variational HWE-CNOT train  $R^2$  -0.0147 test  $R^2$  -0.0025 train MAE 20.1601 kcal/mol test MAE 20.5422 kcal/mol

Talk about the cost of going wider, and inherently deeper

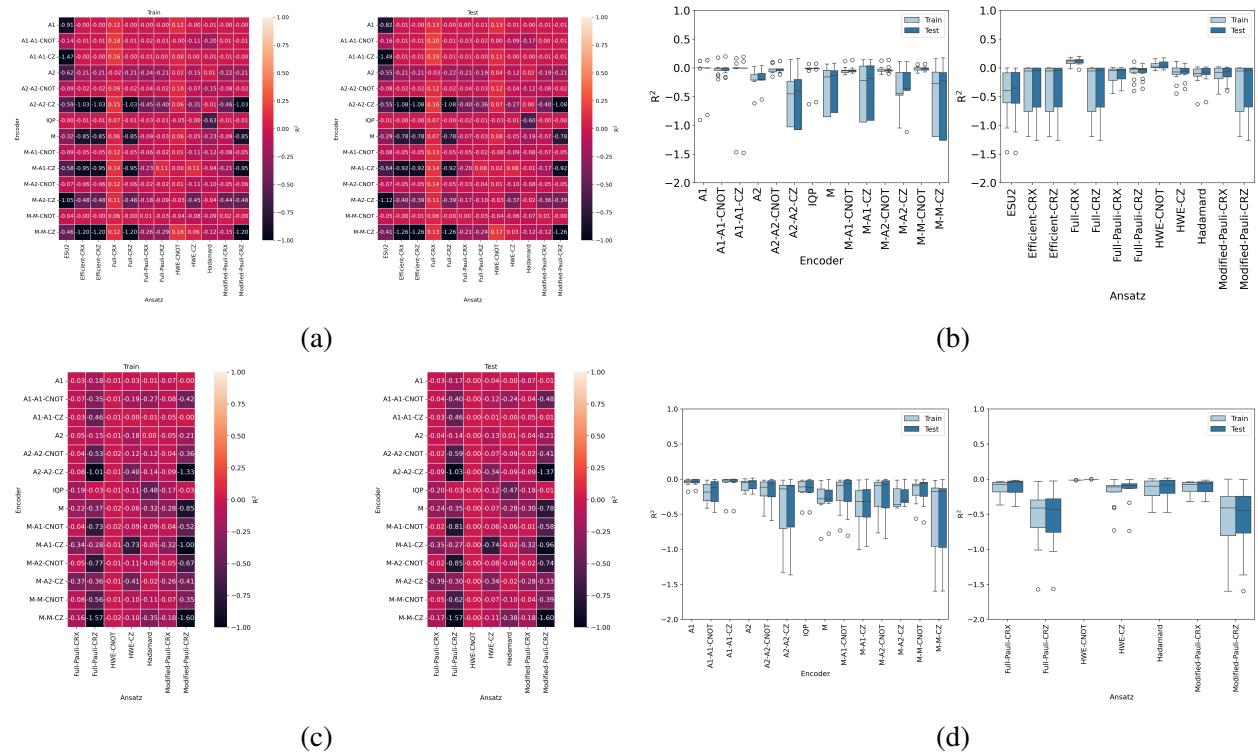


Figure 5

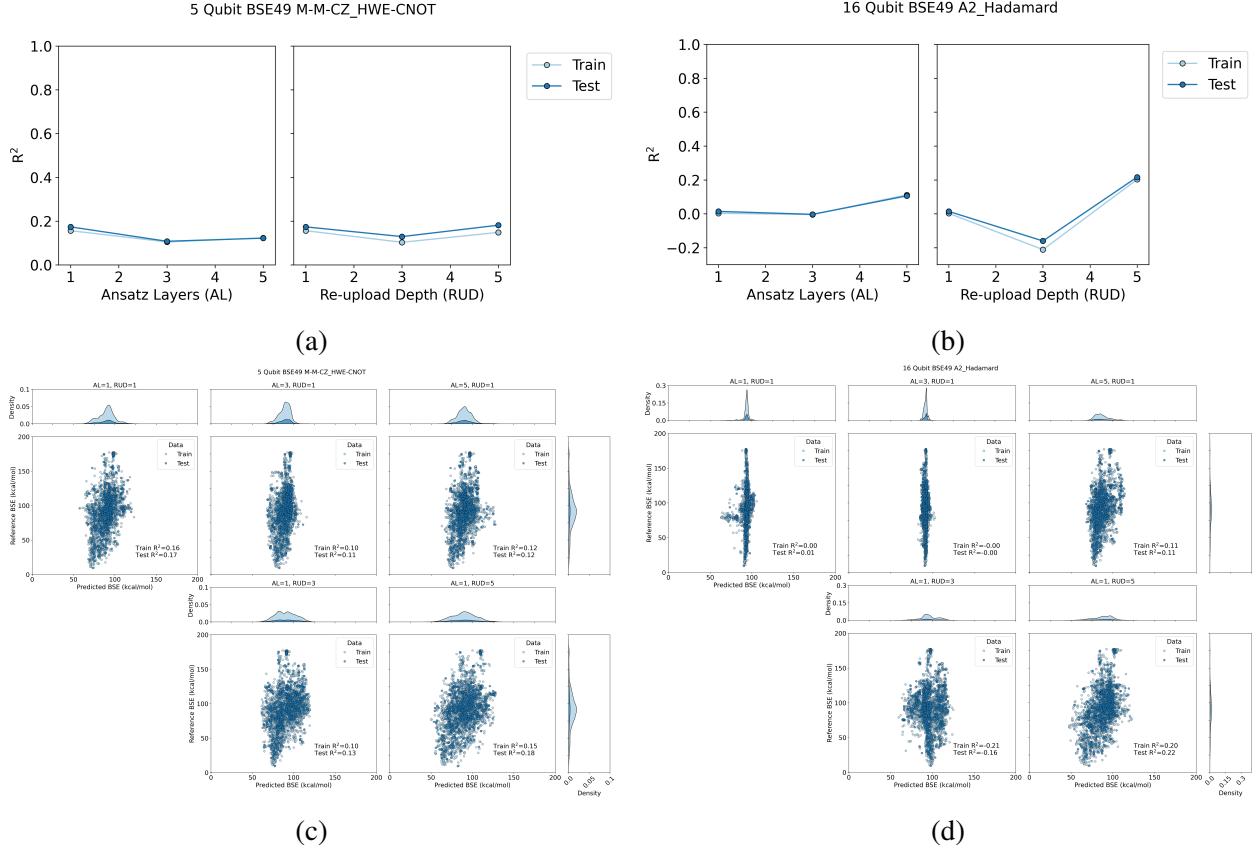


Figure 6

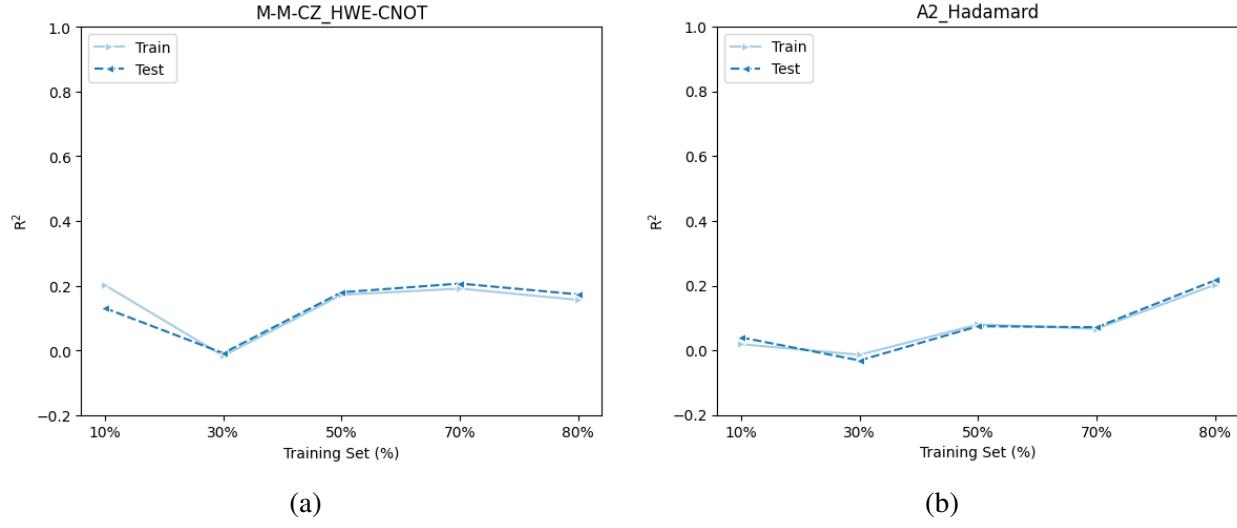


Figure 7

## Structure

- Broad set
- RUD/AL tests
- learning curve (maybe)
- real device

Train R<sup>2</sup> 0.6191 Test R<sup>2</sup> 0.6184 train MAE 5.9739  $mE_h$  test MAE 5.9660  $mE_h$  pair: A2\_HWE-CNOT

Train R<sup>2</sup> 0.0163 Test R<sup>2</sup> 0.0183 train MAE 12.7282  $mE_h$  test MAE 12.7116  $mE_h$  encoder: A1

Train R<sup>2</sup> 0.2817 Test R<sup>2</sup> 0.2853 train MAE 11.6967  $mE_h$  test MAE 11.6871  $mE_h$  variational:

HWE-CNOT

Best encoder average Train R<sup>2</sup> X/test R<sup>2</sup> Y Best ansatz HWE-CNOT average Train R<sup>2</sup> X/test R<sup>2</sup> Y

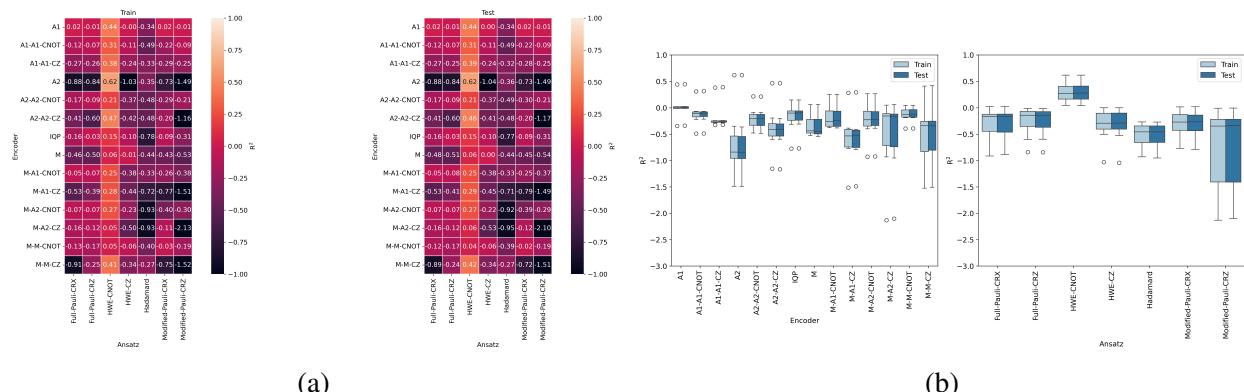


Figure 8

(AL,RUD)=(1,1) Train R<sup>2</sup> 0.62/test R<sup>2</sup> 0.62 (AL,RUD)=(1,3) Train R<sup>2</sup> 0.85/test R<sup>2</sup> 0.85 (AL,RUD)=(1,5) Train R<sup>2</sup> 0.82/test R<sup>2</sup> 0.83 (AL,RUD)=(3,1) Train R<sup>2</sup> 0.71/test R<sup>2</sup> 0.71 (AL,RUD)=(5,1) Train R<sup>2</sup> 0.77/test R<sup>2</sup> 0.77

Talk about the cost of going wider, and inherently deeper

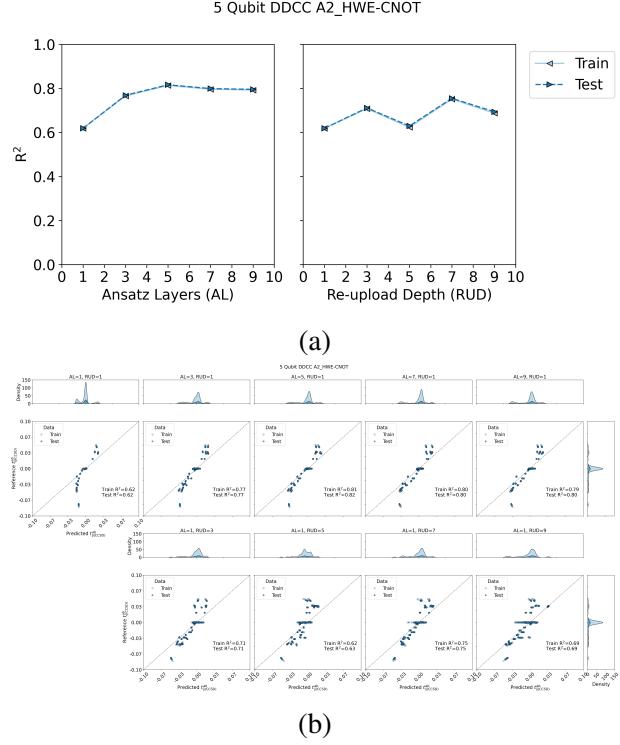


Figure 9: Model evaluation, using  $R^2$  (y-axis), of re-upload depths (RUD) and ansatz layers (AL) of 1, 3, and 5 for the A2\_HWE-CNOT using the DDCC dataset. The left side of the plot denotes the training set and the right side the test set.

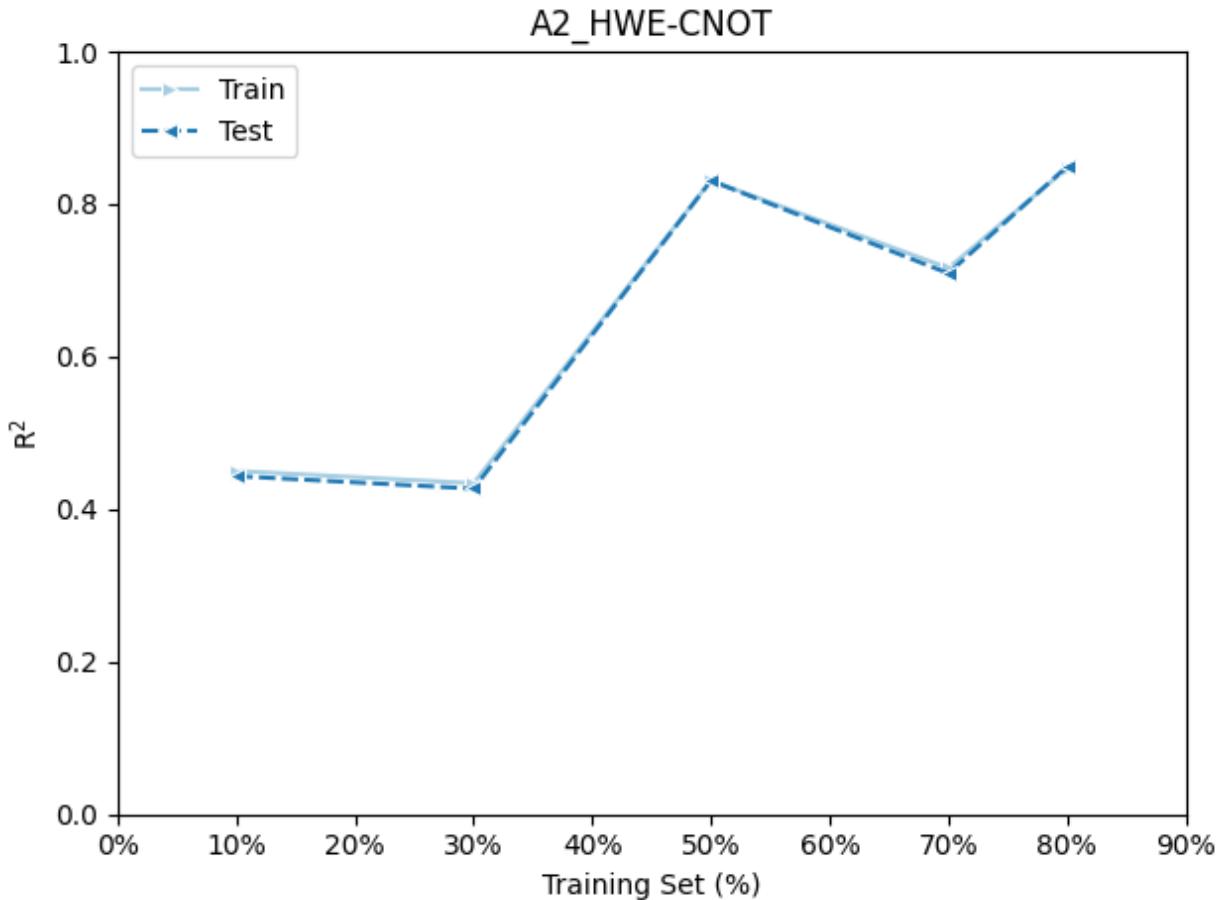


Figure 10

Ansaetze analysis<sup>68</sup> “In particular, a substantial improvement in performance of two-qubit gates in a ring or all-to-all connected arrangement, compared to that of those on a line, is observed.”

“Furthermore, improvement in both descriptors is achieved by sequences of controlled X-rotation gates compared to sequences of controlled Z-rotation gates.”

“investigated how expressibility ‘saturates’ with increased circuit depth, finding that the rate and saturated value appear to be distinguishing features of a PQC”

5 qubit/5 ansatz layers:  $2*5 + 3 * 5 * 5 = 85$  parameters (75 trainable,  $5 * 2$  features) A2  $2n$  parameters HWE-CNOT  $3nL$  parameters 2 qubit gates  $nL$  Number of parameters  $(2n+3nL)=(2+3L)*n$   
 $n$ , number of qubits and  $L$ , number of circuit layers

To efficiently run DDCC on IBM Quebec, splitting the data into batches of  $\approx 4$  samples with 64  $t_2$ -amplitudes each.

One iteration would require approximately  $N_{\text{samples}} * N_{\text{occ}}^2 * N_{\text{virt}}^2 * N_{\text{shots}} * N_{\text{observables}}$  circuit executions (+ whatever SPSA costs to run per iteration)

Ran using the state vector model parameters for one iteration to test the optimization and resilience levels using Fake Quebec before running on the real device

$$\{1024 \times x | x \in [1, 10]\}$$

Fake Optimization level 2 resilience level 0

Regarding the number of circuit executions vs performance 3072 (1024 times 3) is the best number of shots

Fig. S4 Fig. S5

Real Optimization level 2 resilience level 1 3072

resilience level 2 is too expensive on the real device Regarding the number of circuit executions vs performance 3072 (1024 times 3) is the best number of shots

Error handling for real run with iter>1

For the experiments using the *FakeQuebec* and *ibm\_quebec* backends, we performed small tests, initialized using the state-vector parameters with one iteration, to determine the best circuit optimization and error mitigation (resilience) levels.

optimization levels none (0), light (1), medium (2), high (3)

resilience levels (error mitigation) none (0) level 1 readout error mitigation and measurement twirling using Twirled Readout Error eXtinction (TREX)<sup>72</sup> level 2 level 1 + gate twirling and zero noise extrapolation (ZNE)<sup>73-75</sup>

## 6 Conclusion

Quantum advantage in terms of computational complexity but not in model performance?

Depth is not always better! Molecular representations specifically for QML Distributed QC to incorporate more features Noiseless simulation is costly and does not offer the desired accuracy for BSE49 or DDCC

DDCC could be a useful dataset to benchmark PQC models since it is trivial to perform classically, yet hard for PQCs...

## 7 Acknowledgements

We would like to acknowledge the Government of Canada’s New Frontiers in Research Fund (NFRF), for grant NFRFE-2022-00226, and the Quantum Software Consortium (QSC), financed under grant #ALLRP587590-23 from the National Sciences and Engineering Research Council of Canada (NSERC) Alliance Consortia Quantum Grants. This research was enabled in part by support provided by IBM Quantum via the Quantum Software Consortium and the Digital Research Alliance of Canada. The authors would also like to acknowledge Aviraj Newatia, who provided valuable insights with preliminary results excluded from this work. [somethings were ran on the MSRG cloud... do we need to cite the funders of our cloud???] [Viki: Did Nick have any special undergrad funding?]

## References

- (1) Janet, J. P.; Kulik, H. J. *Machine Learning in Chemistry*; ACS In Focus; American Chemical Society, 2020.
- (2) Behler, J. Perspective: Machine learning potentials for atomistic simulations. *JOURNAL OF CHEMICAL PHYSICS* **2016**, 145.
- (3) S. Smith, J.; Isayev, O.; E. Roitberg, A. ANI-1: an extensible neural network potential with DFT accuracy at force field computational cost. *Chemical Science* **2017**, 8, 3192–3203, Publisher: Royal Society of Chemistry.
- (4) Gao, X.; Ramezanghorbani, F.; Isayev, O.; Smith, J. S.; Roitberg, A. E. TorchANI: A Free and Open Source PyTorch-Based Deep Learning Implementation of the ANI Neural Net-

work Potentials. *J. Chem. Inf. Model.* **2020**, *60*, 3408–3415, Publisher: American Chemical Society.

- (5) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; Palmer, A.; Settels, V.; Jaakkola, T.; Jensen, K.; Barzilay, R. Analyzing Learned Molecular Representations for Property Prediction. *JOURNAL OF CHEMICAL INFORMATION AND MODELING* **2019**, *59*, 3370–3388.
- (6) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *SCIENTIFIC DATA* **2014**, *1*.
- (7) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Big Data Meets Quantum Chemistry Approximations: The \$\Delta\$-Machine Learning Approach. *JOURNAL OF CHEMICAL THEORY AND COMPUTATION* **2015**, *11*, 2087–2096.
- (8) Hansen, K.; Biegler, F.; Ramakrishnan, R.; Pronobis, W.; von Lilienfeld, O. A.; Mueller, K.-R.; Tkatchenko, A. Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *JOURNAL OF PHYSICAL CHEMISTRY LETTERS* **2015**, *6*, 2326–2331.
- (9) Unke, O. T.; Meuwly, M. PhysNet: A Neural Network for Predicting Energies, Forces, Dipole Moments, and Partial Charges. *JOURNAL OF CHEMICAL THEORY AND COMPUTATION* **2019**, *15*, 3678–3693.
- (10) Zhong, M. et al. Accelerated discovery of CO<sub>2</sub> electrocatalysts using active machine learning. *NATURE* **2020**, *581*, 178+.
- (11) Nandy, A.; Duan, C.; Taylor, M. G.; Liu, F.; Steeves, A. H.; Kulik, H. J. Computational Discovery of Transition-metal Complexes: From High-throughput Screening to Machine Learning. *Chem. Rev.* **2021**, *121*, 9927–10000, Publisher: American Chemical Society.

- (12) M. Jones, G.; A. Smith, B.; K. Kirkland, J.; D. Vogiatzis, K. Data-driven ligand field exploration of Fe( iv )–oxo sites for C–H activation. *Inorganic Chemistry Frontiers* **2023**, *10*, 1062–1075, Publisher: Royal Society of Chemistry.
- (13) Goh, G. B.; Hodas, N. O.; Vishnu, A. Deep learning for computational chemistry. *JOURNAL OF COMPUTATIONAL CHEMISTRY* **2017**, *38*, 1291–1307.
- (14) Yang, X.; Wang, Y.; Byrne, R.; Schneider, G.; Yang, S. Concepts of Artificial Intelligence for Computer-Assisted Drug Discovery. *Chem. Rev.* **2019**, *119*, 10520–10594, Publisher: American Chemical Society.
- (15) Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine learning for molecular and materials science. *NATURE* **2018**, *559*, 547–555.
- (16) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *SCIENCE* **2018**, *361*, 360–365.
- (17) Raccuglia, P.; Elbert, K. C.; Adler, P. D. F.; Falk, C.; Wenny, M. B.; Mollo, A.; Zeller, M.; Friedler, S. A.; Schrier, J.; Norquist, A. J. Machine-learning-assisted materials discovery using failed experiments. *NATURE* **2016**, *533*, 73+.
- (18) Ufimtsev, I. S.; Martínez, T. J. Graphical Processing Units for Quantum Chemistry. *Computing in Science & Engineering* **2008**, *10*, 26–34, Conference Name: Computing in Science & Engineering.
- (19) Götz, A. W.; Wölfle, T.; Walker, R. C. In *Annual Reports in Computational Chemistry*; Wheeler, R. A., Ed.; Elsevier, 2010; Vol. 6; pp 21–35.
- (20) Pederson, R.; Kozlowski, J.; Song, R.; Beall, J.; Ganahl, M.; Hauru, M.; Lewis, A. G. M.; Yao, Y.; Mallick, S. B.; Blum, V.; Vidal, G. Large Scale Quantum Chemistry with Tensor Processing Units. *J. Chem. Theory Comput.* **2023**, *19*, 25–32, Publisher: American Chemical Society.

- (21) Gawehn, E.; Hiss, J. A.; Brown, J. B.; Schneider, G. Advancing drug discovery via GPU-based deep learning. *Expert Opinion on Drug Discovery* **2018**, *13*, 579–582, Publisher: Taylor & Francis \_eprint: <https://doi.org/10.1080/17460441.2018.1465407>.
- (22) Pandey, M.; Fernandez, M.; Gentile, F.; Isayev, O.; Tropsha, A.; Stern, A. C.; Cherkasov, A. The transformational role of GPU computing and deep learning in drug discovery. *Nat Mach Intell* **2022**, *4*, 211–221, Publisher: Nature Publishing Group.
- (23) Cao, Y.; Romero, J.; Olson, J. P.; Degroote, M.; Johnson, P. D.; Kieferová, M.; Kivlichan, I. D.; Menke, T.; Peropadre, B.; Sawaya, N. P. D.; Sim, S.; Veis, L.; Aspuru-Guzik, A. Quantum Chemistry in the Age of Quantum Computing. *Chem. Rev.* **2019**, *119*, 10856–10915, Publisher: American Chemical Society.
- (24) Abrams, D. S.; Lloyd, S. Simulation of many-body Fermi systems on a universal quantum computer. *Physical Review Letters* **1997**, *79*, 2586, Publisher: APS.
- (25) Abrams, D. S.; Lloyd, S. Quantum algorithm providing exponential speed increase for finding eigenvalues and eigenvectors. *Physical Review Letters* **1999**, *83*, 5162, Publisher: APS.
- (26) Aspuru-Guzik, A.; Dutoi, A. D.; Love, P. J.; Head-Gordon, M. Simulated quantum computation of molecular energies. *Science* **2005**, *309*, 1704–1707, Publisher: American Association for the Advancement of Science.
- (27) Lanyon, B. P.; Whitfield, J. D.; Gillett, G. G.; Goggin, M. E.; Almeida, M. P.; Kassal, I.; Biamonte, J. D.; Mohseni, M.; Powell, B. J.; Barbieri, M.; others Towards quantum chemistry on a quantum computer. *Nature chemistry* **2010**, *2*, 106–111, Publisher: Nature Publishing Group.
- (28) Whitfield, J. D.; Biamonte, J.; Aspuru-Guzik, A. Simulation of electronic structure Hamiltonians using quantum computers. *Molecular Physics* **2011**, *109*, 735–750, Publisher: Taylor & Francis.

- (29) Aspuru-Guzik, A.; Walther, P. Photonic quantum simulators. *Nature physics* **2012**, *8*, 285–291, Publisher: Nature Publishing Group.
- (30) 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, Publisher: Nature Publishing Group.
- (31) Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S. C.; Endo, S.; Fujii, K.; McClean, J. R.; Mitarai, K.; Yuan, X.; Cincio, L.; Coles, P. J. Variational quantum algorithms. *Nat Rev Phys* **2021**, *3*, 625–644, Publisher: Nature Publishing Group.
- (32) McClean, J. R.; Romero, J.; Babbush, R.; Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.* **2016**, *18*, 023023, Publisher: IOP Publishing.
- (33) Bharti, K.; Cervera-Lierta, A.; Kyaw, T. H.; Haug, T.; Alperin-Lea, S.; Anand, A.; Degroote, M.; Heimonen, H.; Kottmann, J. S.; Menke, T.; Mok, W.-K.; Sim, S.; Kwek, L.-C.; Aspuru-Guzik, A. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* **2022**, *94*, 015004, Publisher: American Physical Society.
- (34) Romero, J.; Babbush, R.; McClean, J. R.; Hempel, C.; Love, P. J.; Aspuru-Guzik, A. Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz. *QUANTUM SCIENCE AND TECHNOLOGY* **2019**, *4*.
- (35) McArdle, S.; Endo, S.; Aspuru-Guzik, A.; Benjamin, S.; Yuan, X. Quantum computational chemistry. *REVIEWS OF MODERN PHYSICS* **2020**, *92*.
- (36) Bauer, B.; Bravyi, S.; Motta, M.; Chan, G. K.-L. Quantum Algorithms for Quantum Chemistry and Quantum Materials Science. *CHEMICAL REVIEWS* **2020**, *120*, 12685–12717.
- (37) Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195–202.

- (38) Benedetti, M.; Lloyd, E.; Sack, S.; Fiorentini, M. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.* **2019**, *4*, 043001.
- (39) Suzuki, T.; Katouda, M. Predicting toxicity by quantum machine learning. *J. Phys. Commun.* **2020**, *4*, 125012.
- (40) 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, Publisher: American Chemical Society.
- (41) Bhatia, A. S.; Saggi, M. K.; Kais, S. Quantum Machine Learning Predicting ADME-Tox Properties in Drug Discovery. *J. Chem. Inf. Model.* **2023**, *63*, 6476–6486, Publisher: American Chemical Society.
- (42) 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, Publisher: American Chemical Society.
- (43) Li, J.; Topaloglu, R. O.; Ghosh, S. Quantum Generative Models for Small Molecule Drug Discovery. *IEEE Transactions on Quantum Engineering* **2021**, *2*, 1–8, Conference Name: IEEE Transactions on Quantum Engineering.
- (44) Avramouli, M.; Savvas, I.; Vasilaki, A.; Garani, G.; Xenakis, A. Quantum Machine Learning in Drug Discovery: Current State and Challenges. Proceedings of the 26th Pan-Hellenic Conference on Informatics. New York, NY, USA, 2023; pp 394–401.
- (45) Avramouli, M.; Savvas, I. K.; Vasilaki, A.; Garani, G. Unlocking the Potential of Quantum Machine Learning to Advance Drug Discovery. *Electronics* **2023**, *12*, 2402, Number: 11 Publisher: Multidisciplinary Digital Publishing Institute.

- (46) Ishiyama, Y.; Nagai, R.; Mieda, S.; Takei, Y.; Minato, Y.; Natsume, Y. Noise-robust optimization of quantum machine learning models for polymer properties using a simulator and validated on the IonQ quantum computer. *Sci Rep* **2022**, *12*, 19003, Publisher: Nature Publishing Group.
- (47) Ryu, J.-Y.; Elala, E.; Rhee, J.-K. K. Quantum Graph Neural Network Models for Materials Search. *Materials* **2023**, *16*, 4300, Number: 12 Publisher: Multidisciplinary Digital Publishing Institute.
- (48) Vitz, M.; Mohammadbagherpoor, H.; Sandeep, S.; Vlasic, A.; Padbury, R.; Pham, A. Hybrid Quantum Graph Neural Network for Molecular Property Prediction. 2024; <http://arxiv.org/abs/2405.05205>, arXiv:2405.05205 [quant-ph].
- (49) Jin, H.; Merz, K. M. J. Integrating Machine Learning and Quantum Circuits for Proton Affinity Predictions. *J. Chem. Theory Comput.* **2025**, Publisher: American Chemical Society.
- (50) Hatakeyama-Sato, K.; Igarashi, Y.; Kashikawa, T.; Kimura, K.; Oyaizu, K. Quantum circuit learning as a potential algorithm to predict experimental chemical properties. *Digital Discovery* **2023**, *2*, 165–176.
- (51) Prasad, V. K.; Khalilian, M. H.; Otero-de-la Roza, A.; DiLabio, G. A. BSE49, a diverse, high-quality benchmark dataset of separation energies of chemical bonds. *Sci Data* **2021**, *8*, 300, Publisher: Nature Publishing Group.
- (52) Townsend, J.; Vogiatzis, K. D. Data-Driven Acceleration of the Coupled-Cluster Singles and Doubles Iterative Solver. *J. Phys. Chem. Lett.* **2019**, *10*, 4129–4135.
- (53) Jones, G. M.; S. Pathirage, P. D. V.; Vogiatzis, K. D. In *Quantum Chemistry in the Age of Machine Learning*; Dral, P. O., Ed.; Elsevier, 2023; pp 509–529.
- (54) Cerezo, M.; Verdon, G.; Huang, H.-Y.; Cincio, L.; Coles, P. J. Challenges and opportunities in

quantum machine learning. *Nat Comput Sci* **2022**, *2*, 567–576, Publisher: Nature Publishing Group.

- (55) Jones, G. M.; Story, B.; Maroulas, V.; Vogiatzis, K. D. *Molecular Representations for Machine Learning*; ACS In Focus; American Chemical Society, 2023.
- (56) Durant, J. L.; Leland, B. A.; Henry, D. R.; Nourse, J. G. Reoptimization of MDL Keys for Use in Drug Discovery. *Journal of Chemical Information and Computer Sciences* **2002**, *42*, 1273–1280.
- (57) Morgan, H. L. The Generation of a Unique Machine Description for Chemical Structures-A Technique Developed at Chemical Abstracts Service. *Journal of Chemical Documentation* **1965**, *5*, 107–113, Type: Journal Article.
- (58) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling* **2010**, *50*, 742–754, Publisher: American Chemical Society.
- (59) Bergholm, V. et al. PennyLane: Automatic differentiation of hybrid quantum-classical computations. 2022; <http://arxiv.org/abs/1811.04968>, arXiv:1811.04968 [quant-ph].
- (60) 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. 2024; \_eprint: 2405.08810.
- (61) Pérez-Salinas, A.; Cervera-Lierta, A.; Gil-Fuster, E.; Latorre, J. I. Data re-uploading for a universal quantum classifier. *Quantum* **2020**, *4*, 226, Publisher: Verein zur Förderung des Open Access Publizierens in den Quantenwissenschaften.
- (62) Mitarai, K.; Negoro, M.; Kitagawa, M.; Fujii, K. Quantum circuit learning. *Phys. Rev. A* **2018**, *98*, 032309.

- (63) Bremner, M. J.; Montanaro, A.; Shepherd, D. J. Average-case complexity versus approximate simulation of commuting quantum computations. *Phys. Rev. Lett.* **2016**, *117*, 080501, arXiv:1504.07999 [quant-ph].
- (64) Havlicek, V.; Córcoles, A. D.; Temme, K.; Harrow, A. W.; Kandala, A.; Chow, J. M.; Gambetta, J. M. Supervised learning with quantum enhanced feature spaces. *Nature* **2019**, *567*, 209–212, arXiv:1804.11326 [quant-ph].
- (65) Stoudenmire, E.; Schwab, D. J. Supervised Learning with Tensor Networks. Advances in Neural Information Processing Systems. 2016.
- (66) Lund, A. P.; Bremner, M. J.; Ralph, T. C. Quantum sampling problems, BosonSampling and quantum supremacy. *npj Quantum Inf* **2017**, *3*, 1–8, Publisher: Nature Publishing Group.
- (67) Harrow, A. W.; Montanaro, A. Quantum computational supremacy. *Nature* **2017**, *549*, 203–209, Publisher: Nature Publishing Group.
- (68) Sim, S.; Johnson, P. D.; Aspuru-Guzik, A. Expressibility and Entangling Capability of Parameterized Quantum Circuits for Hybrid Quantum-Classical Algorithms. *Advanced Quantum Technologies* **2019**, *2*, 1900070.
- (69) Suzuki, Y. et al. Qulacs: a fast and versatile quantum circuit simulator for research purpose. *Quantum* **2021**, *5*, 559, arXiv:2011.13524 [quant-ph].
- (70) Virtanen, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **2020**, *17*, 261–272.
- (71) Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
- (72) van den Berg, E.; Minev, Z. K.; Temme, K. Model-free readout-error mitigation for quantum expectation values. *Phys. Rev. A* **2022**, *105*, 032620, Publisher: American Physical Society.

- (73) Kandala, A.; Temme, K.; Córcoles, A. D.; Mezzacapo, A.; Chow, J. M.; Gambetta, J. M. Error mitigation extends the computational reach of a noisy quantum processor. *Nature* **2019**, *567*, 491–495, Publisher: Nature Publishing Group.
- (74) Li, Y.; Benjamin, S. C. Efficient Variational Quantum Simulator Incorporating Active Error Minimization. *Phys. Rev. X* **2017**, *7*, 021050, Publisher: American Physical Society.
- (75) Temme, K.; Bravyi, S.; Gambetta, J. M. Error Mitigation for Short-Depth Quantum Circuits. *Phys. Rev. Lett.* **2017**, *119*, 180509, Publisher: American Physical Society.
- (76) Wood, G. P. F.; Radom, L.; Petersson, G. A.; Barnes, E. C.; Frisch, M. J.; Montgomery, J. A., Jr. A restricted-open-shell complete-basis-set model chemistry. *The Journal of Chemical Physics* **2006**, *125*, 094106.
- (77) Montgomery, J. A., Jr.; Frisch, M. J.; Ochterski, J. W.; Petersson, G. A. A complete basis set model chemistry. VI. Use of density functional geometries and frequencies. *The Journal of Chemical Physics* **1999**, *110*, 2822–2827.
- (78) Montgomery, J. A., Jr.; Frisch, M. J.; Ochterski, J. W.; Petersson, G. A. A complete basis set model chemistry. VII. Use of the minimum population localization method. *The Journal of Chemical Physics* **2000**, *112*, 6532–6542.
- (79) RDKit. <https://www.rdkit.org/>.
- (80) Adams, H.; Emerson, T.; Kirby, M.; Neville, R.; Peterson, C.; Shipman, P.; Chepushtanova, S.; Hanson, E.; Motta, F.; Ziegelmeier, L. Persistence Images: A Stable Vector Representation of Persistent Homology. *Journal of Machine Learning Research* **2017**, *18*, 1–35.
- (81) Townsend, J.; Micucci, C. P.; Hymel, J. H.; Maroulas, V.; Vogiatzis, K. D. Representation of molecular structures with persistent homology for machine learning applications in chemistry. *Nature Communications* **2020**, *11*, 3230, Publisher: Nature Research.

- (82) Schiff, Y.; Chenthamarakshan, V.; Hoffman, S. C.; Natesan Ramamurthy, K.; Das, P. Augmenting Molecular Deep Generative Models with Topological Data Analysis Representations. ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). 2022; pp 3783–3787.
- (83) Tralie, C.; Saul, N.; Bar-On, R. Ripser.py: A Lean Persistent Homology Library for Python. *Journal of Open Source Software* **2018**, *3*, 925.
- (84) Rupp, M.; Tkatchenko, A.; Müller, K. R.; Lilienfeld, O. A. v. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical Review Letters* **2012**, *108*.
- (85) De, S.; Bartók, A. P.; Csányi, G.; Ceriotti, M. Comparing molecules and solids across structural and alchemical space. *Phys. Chem. Chem. Phys.* **2016**, *18*, 13754–13769, Publisher: The Royal Society of Chemistry.
- (86) García-Andrade, X.; García Tahoce, P.; Pérez-Ríos, J.; Martínez Núñez, E. Barrier Height Prediction by Machine Learning Correction of Semiempirical Calculations. *J. Phys. Chem. A* **2023**, *127*, 2274–2283, Publisher: American Chemical Society.
- (87) Lundberg, S. M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems. 2017.
- (88) Hehre, W. J.; Ditchfield, R.; Stewart, R. F.; Pople, J. A. Self-Consistent Molecular Orbital Methods. IV. Use of Gaussian Expansions of Slater-Type Orbitals. Extension to Second-Row Molecules. *The Journal of Chemical Physics* **1970**, *52*, 2769–2773.
- (89) Parrish, R. M. et al. Psi4 1.1: An Open-Source Electronic Structure Program Emphasizing Automation, Advanced Libraries, and Interoperability. *J. Chem. Theory Comput.* **2017**, *13*, 3185–3197, Publisher: American Chemical Society.
- (90) Smith, D. G. A. et al. Psi4NumPy: An Interactive Quantum Chemistry Programming Envi-

ronment for Reference Implementations and Rapid Development. *J. Chem. Theory Comput.* **2018**, *14*, 3504–3511, Publisher: American Chemical Society.