

# A Combinatorial Search of 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*

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

[Problem: QML methods related to quantum chemistry are rarely studied so we chose two different sets of data, one that's classical in-quantum out and the other that is quantum in-quantum out]

### [FOCUS ON QUANTUM CHEMISTRY]

In recent years, machine learning (ML) has emerged as a popular tool in chemistry to reveal new patterns in data, provide new insights beyond simple models and human experience, and to accelerate computations and the analysis of chemical space. Applications of ML in chemistry include accelerating molecular simulations,<sup>1–3</sup> determining molecular properties,<sup>4–8</sup> and for the discovery of new catalysts,<sup>9–11</sup> drugs,<sup>12,13</sup> and materials.<sup>14–16</sup> Often, the goal of ML in chemistry is to bypass the explicit calculation of molecular properties that can be expensive to calculate for large sets of data.<sup>17</sup>

Similar to the aforementioned machine learning methods, quantum computing (QC) for chemical applications has become increasingly popular due to possible quantum speedups for quantum chemical calculations.<sup>18</sup> While most quantum computing studies focus on creating new and more efficient electronic structure methods on quantum computers,<sup>18–21</sup> an approach that combines both ML and QC, is quantum machine learning (QML). In recent years, QML has shown potential for quantum speedups, either through formal mathematical proofs or numerical experiments based on empirical observations.<sup>22</sup> Among the classes of QML algorithms, parameterized quantum circuits (PQCs), which are based on a hybrid quantum–classical optimization loop, have shown promise as

flexible QML models, capable of achieving non-trivial results on near-term quantum hardware.<sup>23,24</sup> In the context of chemistry, quantum machine learning (QML) has gained traction for tasks such as drug discovery,<sup>24–30</sup> materials discovery,<sup>31–33</sup> proton affinities,<sup>34</sup> and experimental molecular properties, including the log solubility in water, melting point, octanol/water distribution coefficient, hydration free energy of small molecules in water.<sup>35</sup>

Like the previously mentioned studies, we wanted to determine whether QML, specifically using PQCs, can offer a quantum advantage (or speedup) on chemical applications. Herein, we explore two datasets that require the explicit calculation of (RO)CBS-QB3 and CCSD (RO)CBS-QB3 can be used for X Y and Z as found in the Wood et al pub CCSD can be used for energies and properties

Despite the broad range of applications covered in the previously mentioned studies,

A common feature among most of the previously mentioned studies is that these studies do not focus on aspects related to quantum chemistry, such as the calculation of correlation energies or bond separation energies, which require the explicit calculation using quantum chemical methods. Motivated by this, we analyze two datasets related to quantum chemistry, one of which requires the use of classical molecular representations as output and predicts the bond separation energies

the BSE49 dataset,<sup>36</sup> which consists of bond separation energies of 49 unique bond types calculated using a high-level composite method called (RO)CBS-QB3

, and a dataset of water confirmers using data-driven coupled cluster (DDCC),<sup>37</sup> a machine learning approach for predicting the coupled-cluster singles and doubles wave function parameters using lower-level quantum chemistry methods.

Motivated by these recent studies, we wanted to perform an exhaustive benchmark of a large set of PQCs on chemically relevant datasets. To this end, we introduce *qregress*, a modular Python-based code built on PennyLane for exploring PQCs for regression based learning tasks. Our code base includes a set of 14 data-encoding and 12 variational (ansatz) subcircuits for a total of 168 PQCs. We include support for state-vector simulation using Qulacs, noisy fake backends, and real devices—along with error mitigation techniques implemented in Qiskit. We explore two datasets

related to classical machine learning in chemistry tasks, which includes

Chose two datasets for which the authors are familiar with their classical performance [cite the classification schemes quantum in/quantum out and classical in/quantum out] The BSE49 dataset is a classical input-quantum output problem The DDCC is a quantum input-quantum output problem

Using these datasets, we benchmark the initial set of PQCs, followed by taking the best performing circuits and exploring the effects of circuit depth on model performance. Lastly, we use the best circuit to run on a real device, *ibm\_quebec*, to...?

[Explore in what contexts quantum advantage can be achieved]

## 2 Methods

PQCs are often constructed of three parts: encoding layers that are used to encode the features onto a quantum circuit, variational layers which include parameters that are optimized classically, and measurements which provide numerical estimations of the regression target values.<sup>24</sup> In this study, we utilize the Mitarai (M),<sup>38</sup> single- (A1) and double-angle (A2) encoding layers found in Ref.<sup>24</sup>, along with the instantaneous quantum polynomial (IQP) circuit found in Refs.<sup>39</sup> and<sup>40</sup>. In the following section, we follow the notations derived from Ref.<sup>24</sup>. Encoding layers work mapping a  $d$ -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  $n$  are the number of qubits. The encoding layer takes the following general form,

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

where,  $E_{\text{ent}}^l$  denotes the entangling gates, which can be a CNOT, CZ, or identity (**I**) gates,  $U_{\phi_l(\mathbf{x})}$  denotes the choice of encoding unitaries. Like in Ref.<sup>24</sup>, we choose  $l \in \{1, 2\}$ , such that when  $l = 1$ ,  $E_{\text{ent}}$  corresponds to an identity matrix and  $U_{\Phi(x)} = U_{\phi_1(\mathbf{x})}$ .

When  $l = 1$ ,  $U_{\Phi(x)}$  can be one of the following four encoding layers:  $U_{\text{A1}}$ ,  $U_{\text{A2}}$ ,  $U_{\text{M}}$ , or  $U_{\text{IQP}}$ .

The single-angle encoding (Fig. 1 **(a)**) is the simplest and takes the following form,

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

where  $R_i^Y$  denotes a parameterized Y rotation gate on qubit  $i$ . Like the single-angle encoding, the double-angle encoding (Fig. 1 **(b)**) utilizes a parameterized Y rotation gate on qubit  $i$ , with the addition of a parameterized Z rotation gate on qubit  $i$ , denoted as

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

The Mitarai encoding layer (Fig. 1 **(c)**) is a double-angle encoding layer with the addition of an arccosine function on the parameterized Z gate and arcsine on the parameterized Y gate,

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

Following the formulation provided in the PennyLane<sup>41</sup> software package, the IQP encoding layer (Fig. 1 **(d)**) is defined as,

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

where  $H_i$  denotes a Hadamard gate on qubit  $i$  and  $ZZ_{ij}$  denotes a two-qubit entanglement gate defined as  $ZZ_{ij} = e^{-ix_i x_j \sigma_z \otimes \sigma_z}$ .

**Quantum advantage of IQP:** “For example, under well-believed complexity-theoretic assumptions, the class of PQCs called instantaneous quantum polynomial-time cannot be efficiently simulated by classical resources (see Lund et al [3] and Harrow and Montanaro [4] for accessible Reviews of quantum supremacy proposals).”<sup>23</sup>

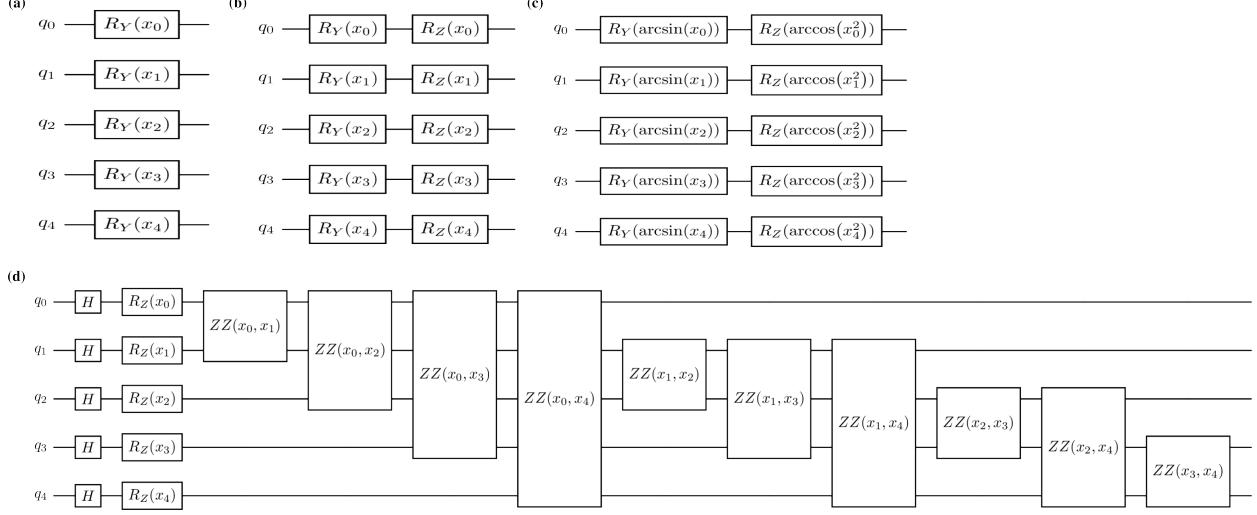


Figure 1: (a) Single angle (A1) encoding, (b) double angle (A2) encoding, (c) Mitarai (M) encoding, and (d) Instantaneous Quantum Polynomial (IQP) encoding

When  $l = 2$ , like in Ref.<sup>24</sup> we choose entanglement gates,  $E_{\text{ent}}^1$  and  $E_{\text{ent}}^2$  to be equivalent, and the encoding layer takes the following form,  $U_{\Phi(\mathbf{x})} = E_{\text{ent}} U_{\phi_2(\mathbf{x})} E_{\text{ent}} U_{\phi_1(\mathbf{x})}$ . We also exclude IQP encoding when  $l = 2$  due to the increased circuit depth, when compared to A1, A2, and M encoding. Therefore, 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 entanglement layer options (CNOT and CZ) for a total of 10 encoding circuits. These circuits are denoted as  $U_{\phi_1(\mathbf{x})} - U_{\phi_2(\mathbf{x})} - E_{\text{ent}}$ , for example, two example encoding circuits are M–M–CNOT and M–A1–CNOT. Table 1 shows all fourteen encoding circuits examined in this study.

Following the encoding layers, variational (or ansatz) layers are used to introduce trainable parameters into the quantum circuit. We use a mixed notation from Refs.<sup>24</sup> and<sup>42</sup>, since Ref.<sup>42</sup> contains all of the variational layers used within this work. We relegate the discussion of the expressibility and entanglement examined in that work to Section 3. A general variational layer can be denoted as,

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

where  $\boldsymbol{\theta}$  denotes the variational parameters and  $v$  denotes the number of times that the layer is repeated within the circuit. As  $v$  increases and the number of trainable parameters ( $\boldsymbol{\theta}$ ) increase,

Table 1: Add something smart

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 theoretical assumption is that the model expressibility should also increase. In our study, we choose  $v \in \{1, 3, 5\}$  and refer to this as the number of ansatz layers (ALs). We examine 12 different variational circuits, as shown in Fig. 2, which are denoted using the following labels: 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)),

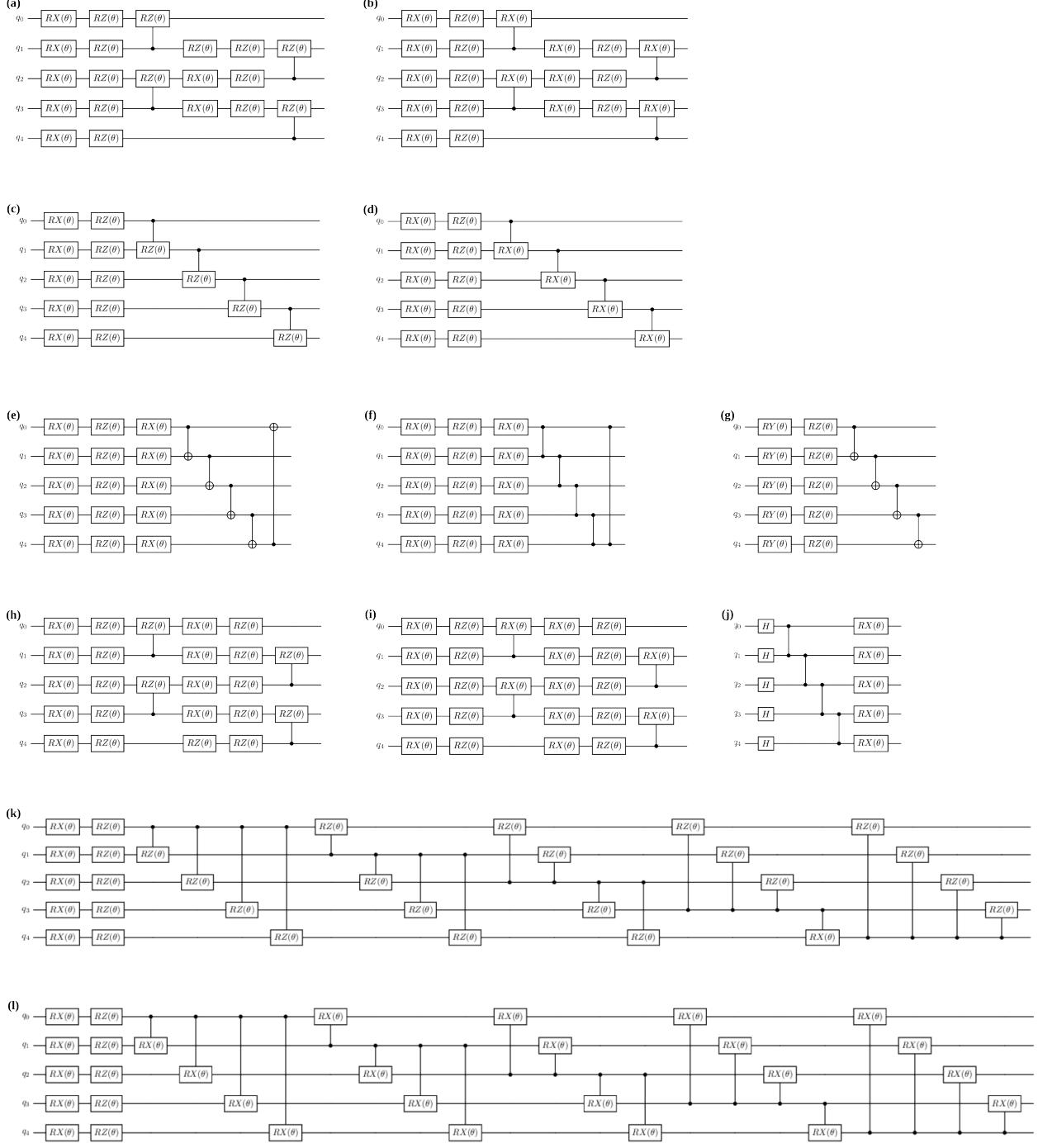


Figure 2: (a) Modified-Pauli-CRZ, (b) Modified-Pauli-CRX, (c) Efficient-CRZ, (d) Efficient-CRX, (e) HWE-CNOT, (f) HWE-CZ, (g) ESU2, (h) Full-Pauli-CRZ, (i) Full-Pauli-CRX, (j) Hadamard, (k) Full-CRZ, and (l)Full-CRX

Now that we have define the encoding (Eq. 1) and variational (Eq. 6) circuits, we can then

combine them to denote a general, complete circuit as,

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

where we choose  $k \in \{1, 3, 5\}$ , which denotes the re-upload depth (RUD) of the circuit. When a sufficient number of data re-uploading occur, it has been shown by Pérez-Salinas *et al.* that data re-uploading is equivalent to the Universal Approximation Theorem for artificial neural networks.<sup>43</sup>

Lastly, to recover the predicted target values,  $\hat{y}_i$ , from our quantum circuits, measurement of the quantum state,  $|\Psi\rangle$ , must be performed. To perform this operation, we apply the Pauli Z operator on the first qubit denoted as,

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

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, is then passed to the loss function,  $\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}})$ , where  $y_i$  belongs to the set of true target values  $\mathbf{y} = (y_1, \dots, y_N) \in \mathbb{R}^N$ . In practice,  $\mathcal{L}$  can be any loss function but we choose to use the mean square error loss function denoted as,

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

## Implementation

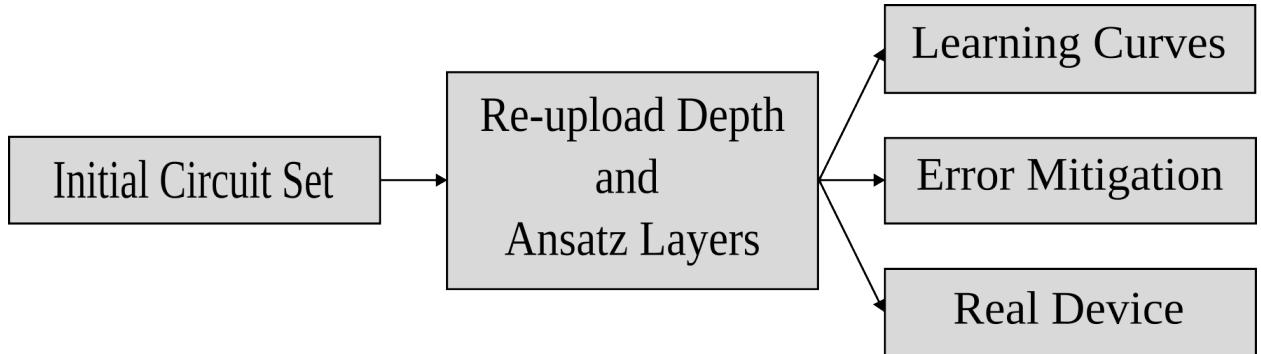


Figure 3: Make this figure highlight the modularity of *qregress*

We introduce *qregress*, a modular Python package for regression-based PQCs.

We perform all simulation calculations using PennyLane,<sup>41</sup> either using Qulacs<sup>44</sup> for state vector calcualtions, while noisy calculations were performed using *qiskit-aer* with the *FakeQuebec* backend as implemented in the PennyLane-Qiskit plugin.<sup>45</sup> We perform calculations in the *ibm\_quebec* device using circuits implemented using Qiskit,<sup>45</sup> due to issues we initially faced with running experiments on the real device using the PennyLane-Qiskit plugin. For the experiments using PennyLane, we utilize the Simultaneous Perturbation Stochastic Approximation method (SPSA) as implemented in PennyLane, while for the experiments run on *ibm\_quebec* utilizes the Constrained Optimization By Linear Approximation (COBYLA) optimizer as implemented ing SciPy.<sup>46</sup> Each optimizer was chosen based on the performance for the given task. All features ( $\mathbf{x}$ ) and target values ( $y$ ) were scaled using the MinMaxScaler in Scikit-learn,<sup>47</sup> such that all fea-  
tures and target values are  $\mathbb{R} \in [-1, 1]$ . For the simulations using *FakeQuebec* and experiments on *ibm\_quebec* we utilize Twirled Readout Error eXtinction (TREX) error mitigation.

Used QisKit for Fake backend

Function fitting 5: all ran with 1000 iterations Function fitting 16: all ran with 1000 iterations  
BSE 5: all ran with 1000 iterations BSE 16: all ran with 1000 iterations DDCC

## Datasets

In this study, we explore two datasets: a dataset of bond separation energies (BSE) of molecules (Figs. 4a and 4b), where the feature set encodes structural information of each molecule; and a dataset consisting of electronic structure features to predict wave functions using the data-driven coupled-cluster scheme of Townsend and Vogiatzis (Fig. 4c).<sup>37</sup> We utilize the BSE49 and DDCC databases for two different reasons: the BSE49 database consists of a hard chemical property to predict using few features, while the DDCC dataset can be predicted easily using few features classically but is data intensive in the number of samples per molecule.

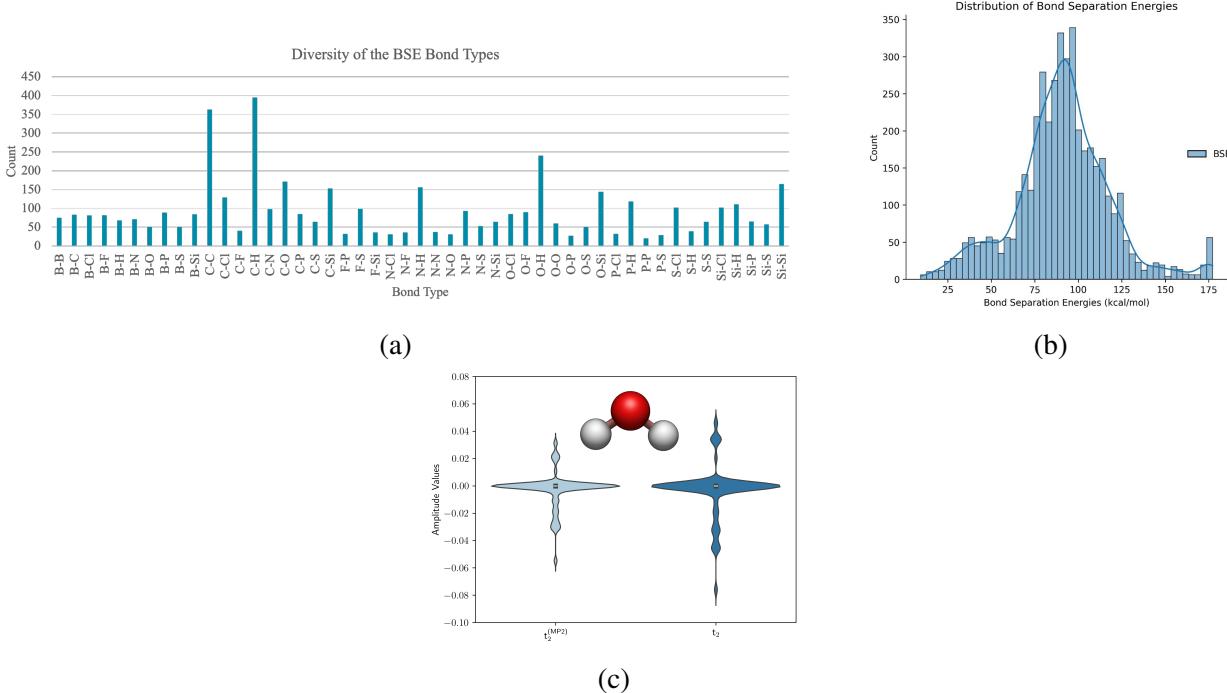


Figure 4: Examples of the datasets explored in this study. For the BSE49 dataset the distributions of the (d) bond types and the (e) bond separation energies in kcal/mol. For the DDCC dataset distributions of the initial MP2  $t_2$ -amplitudes and the optimized CCSD  $t_2$ -amplitudes are shown in (f).

Following model calibration using the function fitting dataset, we explore the applicability of PQCs for complex chemically relevant machine learning tasks. The first chemically motivated dataset we explore is the BSE49 dataset, which contains the bond separation energies (BSE) for the homolytic bond cleavage of covalently bonded molecules, such as  $A-B \longrightarrow A^\cdot + B^\cdot$ .<sup>36</sup> This dataset consists of 4394 datapoints, 1951 of which are existing and 2443 are hypothetical structures, with 49 unique A-B single bond types. In practice, we used 2436 of the hypothetical structures due to issues with valency exceptions when converting to RDKit mol objects which were later used for generating our features for the machine learning models. An important aspect of machine learning in chemistry is the choice of molecular representation, or how the molecule is represented in the machine learning models.<sup>48</sup> Using RDKit<sup>49</sup> we examined three commonly applied graph-based molecular representations, Molecular ACCess Systems (MACCS),<sup>50</sup> Morgan or extended-connectivity fingerprints,<sup>51,52</sup> and RDKit fingerprints. All three of these methods are

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>53–55</sup> We use the implementation from Townsend *et al.*,<sup>54</sup> which uses the Ripser Python package to generate PIs.<sup>56</sup> Lastly, we explore two physics-based representations, Coulomb matrices (CMs)<sup>57</sup> and smooth overlap of atomic positions (SOAPs), that were generated using DScribe.<sup>58</sup> Due to the computational cost of computing the regularized entropy match (REMatch) kernel with the SOAPs representation, we excluded this representation in the overall discussion. We also tested two different methods for representing the components of the bond separation chemical reaction, one where the feature vectors for the products are subtracted from the reactants, denoted by *sub*, similar to the method used in Ref.,<sup>59</sup> and one that is composed of the reactant molecular only, denoted as *AB*.

Since we are analyzing a diverse set of PQCs, we also examine a diverse set of classic regression models, with varying capabilities, such as ridge, lasso, elastic net, *k*-nearest-neighbors, random forest, gradient boosting, support vector, kernel ridge, and gaussian process regression as implemented in scikit-learn.<sup>47</sup> Based on our results shown in Fig. 5 we found that Morgan fingerprints We found that the best molecular representation across all models test, as shown in Fig. 5, was Morgan fingerprints using the *sub* formulation.

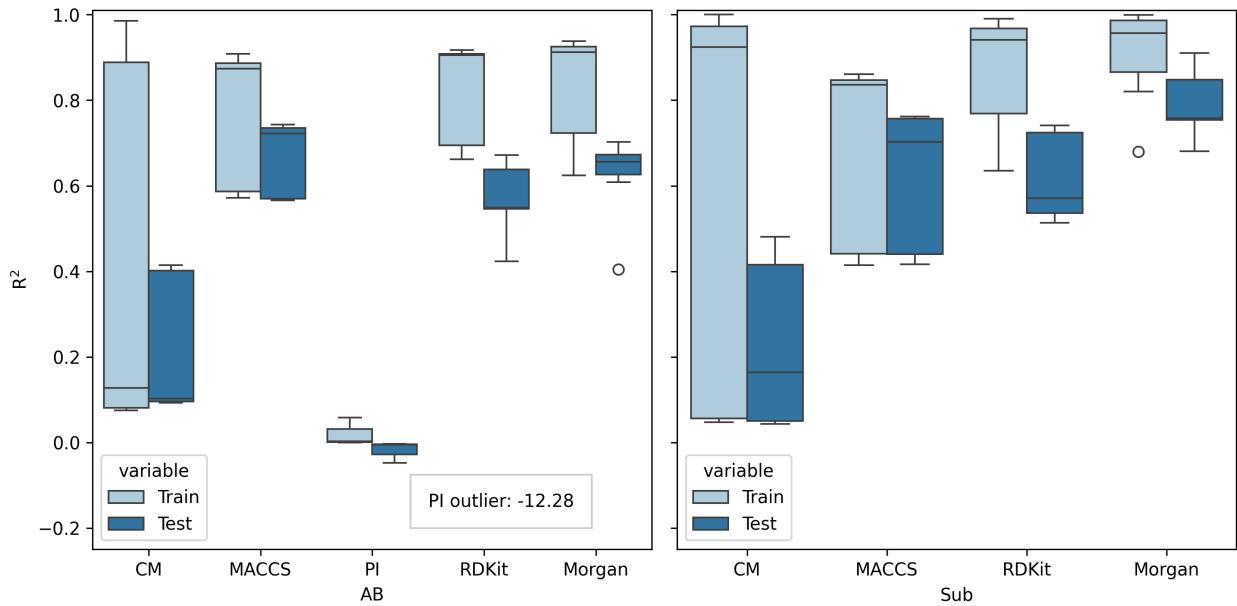


Figure 5: Coulomb matrices (CMs), Molecular ACCess Systems (MACCS), persistence images (PIs), RDKit and Morgan fingerprints. Performance of a diverse set of molecular representations  $R^2$

An additional challenge of applying classical molecular representations for quantum machine learning models is mapping the classical features, often containing hundreds or thousands of features per sample, to the number of qubits used on the quantum device. Initially, the Morgan fingerprints have 2048 features per sample, that need to be reduced down to 5 or 16 qubits. We choose 5 and 16 qubits for two reasons, the first is that these were the standard number of qubits on IBM quantum devices when we started the project and the second is that reducing the number of features reduces the depth of the circuits. To reduce the feature set from 2048 to 5 or 16 features, we explore two different methods, SHapley Additive ExPlanation analysis (SHAP)<sup>60</sup> and principal component analysis (PCA), as implemented in scikit-learn.<sup>47</sup> Figs. 6b and 6a show the results for the reductions using SHAP and PCA for the training and test set of using 5 and 16 features. The initial model using 2048 features has a train and test mean absolute error (MAE) of 1.91 and 4.98 kcal/mol, with train and test  $R^2$ s of 0.99 and 0.91, respectively. When using SHAP to reduce the feature set size to 5 features, we see that the training set has an MAE of 16.08 kcal/mol and an  $R^2$  of 0.39, while for the test set has an MAE of 15.86 kcal/mol and an  $R^2$  of 0.42. When the

number of features is reduced to 16 features using SHAP, we see slight improvements with train and test MAEs of 10.48 and 11.08 kcal/mol with  $R^2$ s of 0.69 and 0.68, respectively. Using PCA, we see an improvement in accuracy for both 5 and 16 features, where the training sets have MAEs of 4.09 and 3.23 kcal/mol and the test sets have MAEs of 10.17 and 8.40 kcal/mol, respectively. The  $R^2$ s for PCA with 5 and 16 features also shows improvement over the reductions using SHAP, with  $R^2$ s of 0.95 and 0.69 for the training and test set, respectively, using 5 features and 0.97 and 0.78 for the training and test set, respectively, using 16 features. Due to the increased performance, despite exhibiting overfitting, we choose to use Morgan fingerprints reduced using PCA for our QML models.

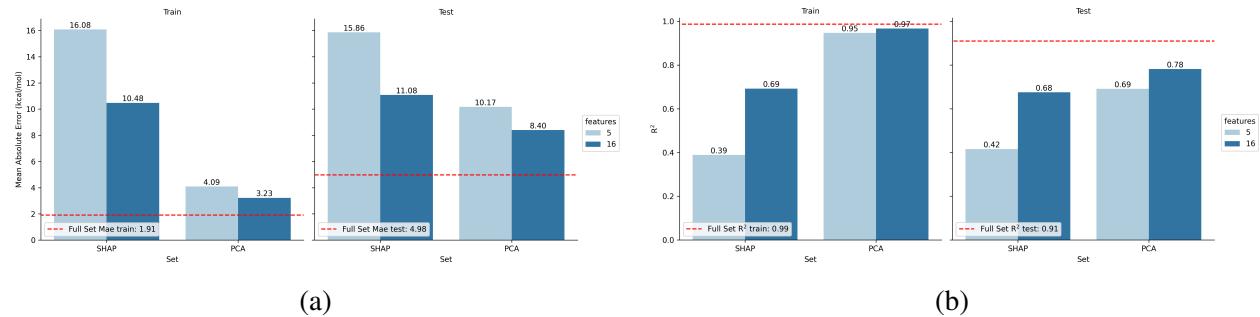


Figure 6: Feature reduction of the BSE dataset represented using

The next chemical dataset explored is based on the data-driven coupled-cluster (DDCC) method, which is a machine learning-based approach for accelerating the convergence of coupled-cluster singles and doubles (CCSD) calculations.<sup>37,61</sup> This method works 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), which are used to initialize CCSD calculations.

The coupled-cluster wave function takes the general form,

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

where the cluster operator is denoted as  $\hat{T}$  and the reference, Hartree-Fock wave function is denoted

as  $|\Psi_0\rangle$ . The CCSD wave function truncates the cluster operator to only include singles and doubles excitations. The CCSD correlation energy is defined 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,  $t_{ij}^{ab}$  are the  $t_2$ -amplitudes which correspond to two-electron excitations,  $t_i^a$  and  $t_j^b$  are  $t_1$ -amplitudes corresponding to one-electron excitations, and  $\langle ij||ab \rangle$  are two-electron integrals.

For each two-electron excitation,  $t_{ij}^{ab}$ , a feature set can be generated from HF and MP2. The feature set includes the MP2  $t_2$ -amplitudes, which are used to initialize the CCSD amplitudes,

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

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  $\varepsilon_b$ . Features related to the MP2  $t_2$ -amplitudes that are also included in the feature set are the numerator ( $\langle ij||ab \rangle$ ) and 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 ( $\varepsilon_i, \varepsilon_j, \varepsilon_a, \varepsilon_b$ ). The feature set also includes terms related to the individual contributes to the orbital energies are also included, such as the one-electron Hamiltonian ( $h$ ), Coulombic matrix ( $J$ ), and exchange matrix  $K$ , and Coulombic and exchange integrals ( $J_a^i, J_b^j, K_i^a, K_j^b$ ). In total, there are 30 features for each  $t_2$ -amplitude due to the addition of features that denote the sign and magnitudes of the previously mentioned features.

Our dataset consists of 199 water molecules from the study by Townsend and Vogiatzis using the STO-3G basis set<sup>62</sup> and frozen core orbitals. All data was generated using Psi4<sup>63</sup> and Psi4Numpy.<sup>64</sup> As previously mentioned, the DDCC method is data intensive regarding the number of samples per molecule, for example, each water molecule has 4 occupied and 2 virtual orbitals. The number of  $t_2$ -amplitudes is equivalent to  $(N_{\text{occ}})^2(N_{\text{virt}})^2$ , where  $N_{\text{occ}}$  denotes the number of oc-

cupied orbitals and  $N_{virt}$  denotes the number of virtual orbitals, so the total number of  $t_2$ -amplitudes per molecule is 64. Further details regarding the feature set and implementation can be found in<sup>37</sup>

Like the BSE dataset, the 30 features from the full DDCC feature set must be reduced to 5 or 16 features using SHAP or PCA. Unlike the BSE dataset, we choose SHAP over PCA for the feature reduction since there is a direct correlation between the input features and output values. As shown in Fig. 7, 5 and 16 features can accurately recover performance of the original model using 30 features, where all three models have train and test R<sup>2</sup>s of 1.00. Due to the computational costs of running QML models, we will then use only 5 features for all DDCC QML models. The 5 most important features are the two-electron integrals ( $\langle ij||ab\rangle$ ), MP2  $t_2$ -amplitudes ( $t_{ij}^{ab}_{(MP2)}$ ), the magnitude of the MP2  $t_2$ -amplitudes, and the difference in orbital energies ( $\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b$ ).

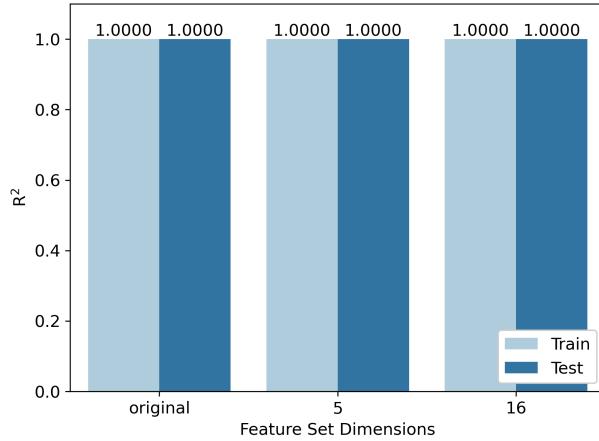


Figure 7

### 3 Results and Discussion

#### BSE

To explore the performance of PQCs for chemically relevant datasets, the dataset we explore is the BSE49 dataset, which consists of a diverse set of bond separation energies and the related structures, as highlighted in Figs. 4a and 4b. Initially, we study the five qubit data, Figs. 8a and 8b, due to computational considerations.

five qubit best encoder-ansatz pair: train R<sup>2</sup>/test R<sup>2</sup> Best encoder on average train R<sup>2</sup>/test R<sup>2</sup>

Best ansatz on average train R<sup>2</sup>/test R<sup>2</sup>

'M-M-CNOT', '-0.0216' 'Full-CRX', '0.1214'

sixteen qubit 8c 8d removed the really bad ones from five qubit best encoder-ansatz pair: train

R<sup>2</sup>/test R<sup>2</sup> Best encoder on average train R<sup>2</sup>/test R<sup>2</sup> Best ansatz on average train R<sup>2</sup>/test R<sup>2</sup>

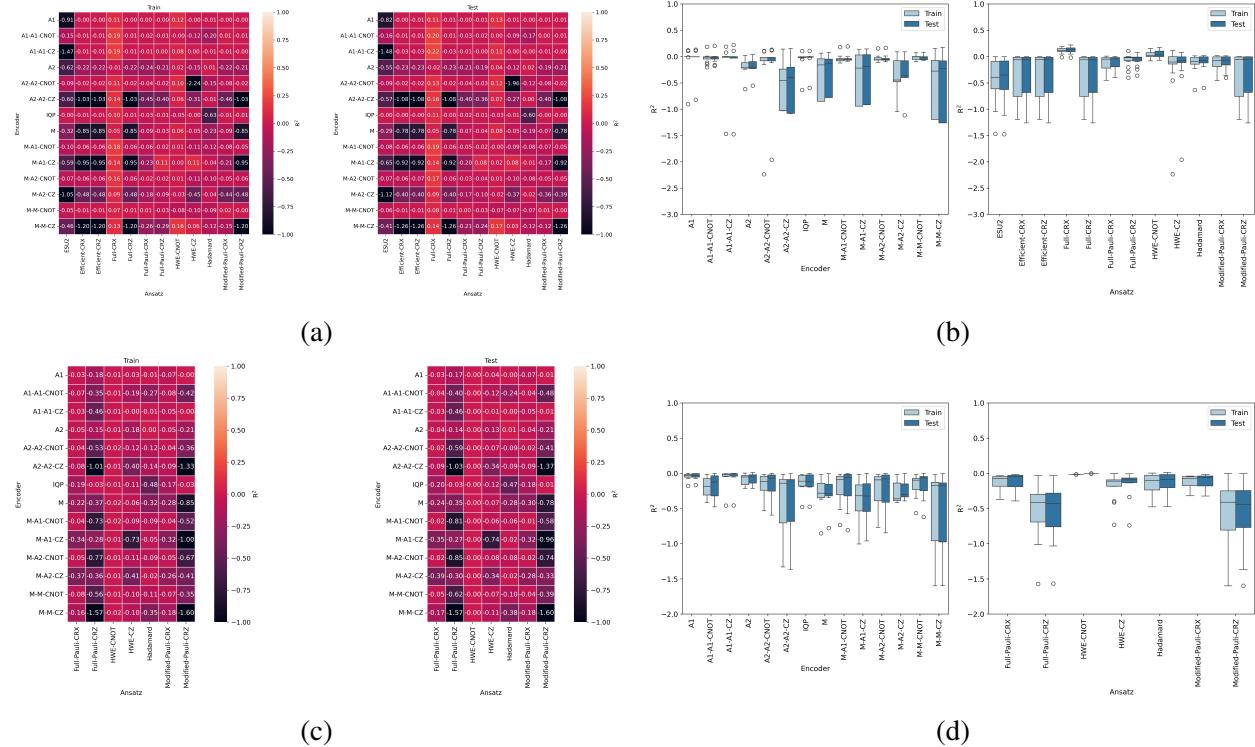


Figure 8

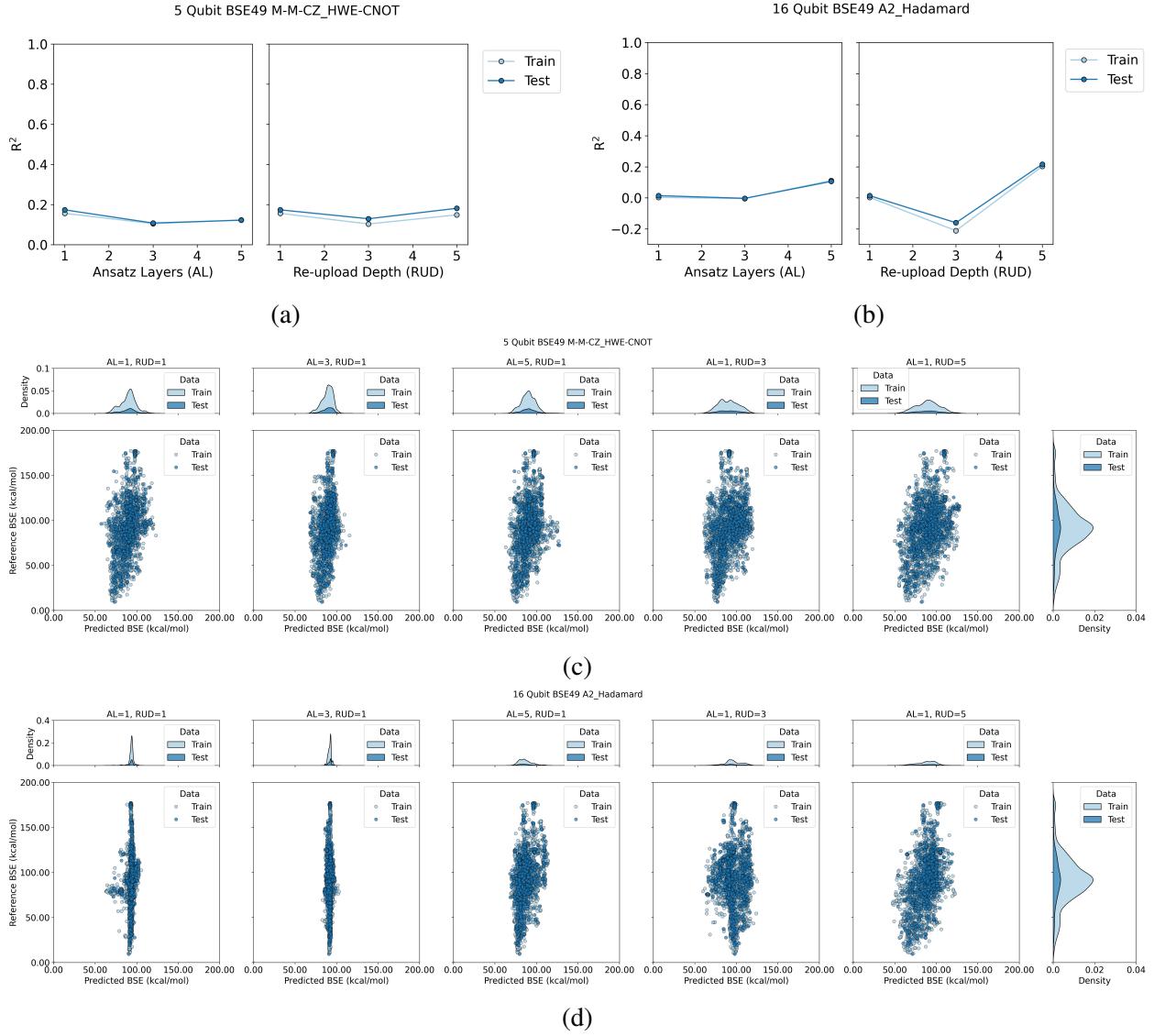


Figure 9

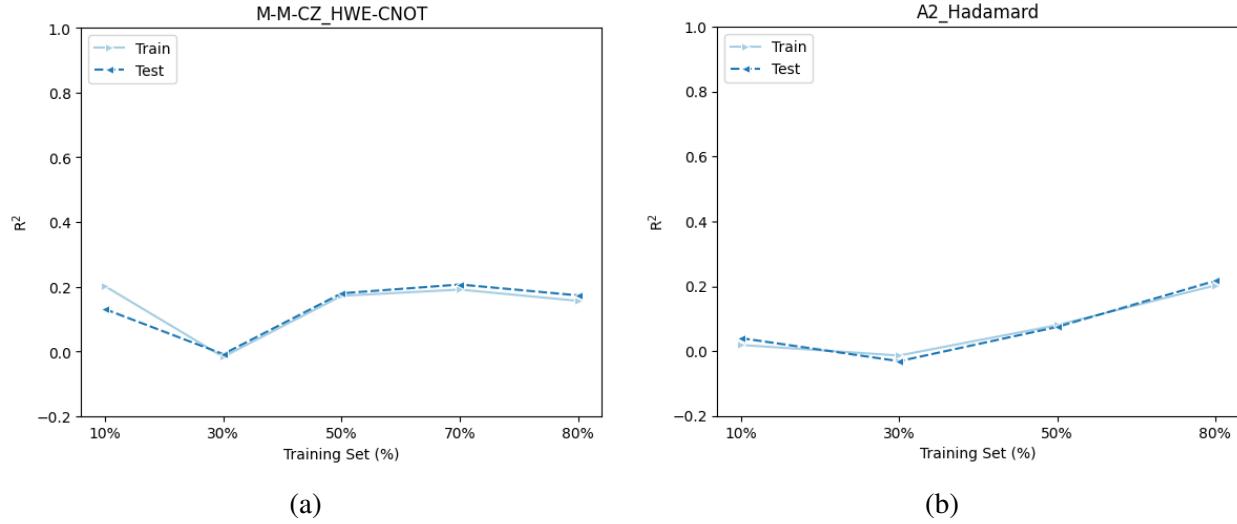


Figure 10

## DDCC

A2\_HWE-CNOT Train R<sup>2</sup> 0.62/test R<sup>2</sup> 0.62 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

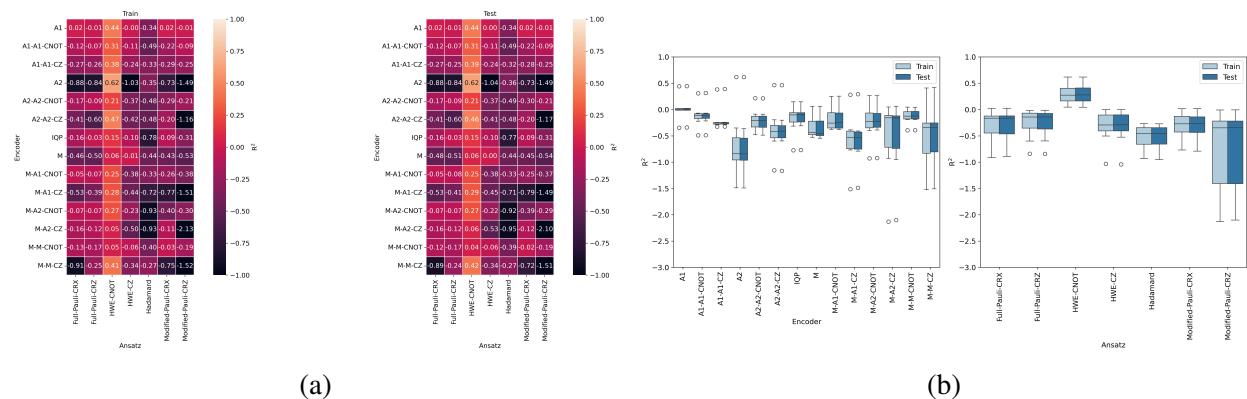
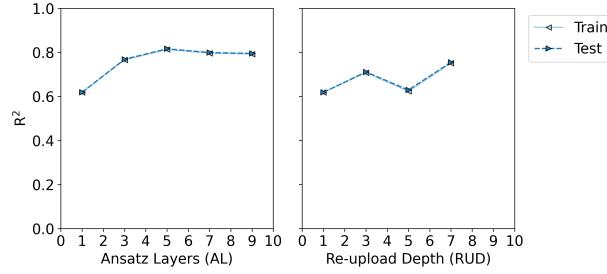


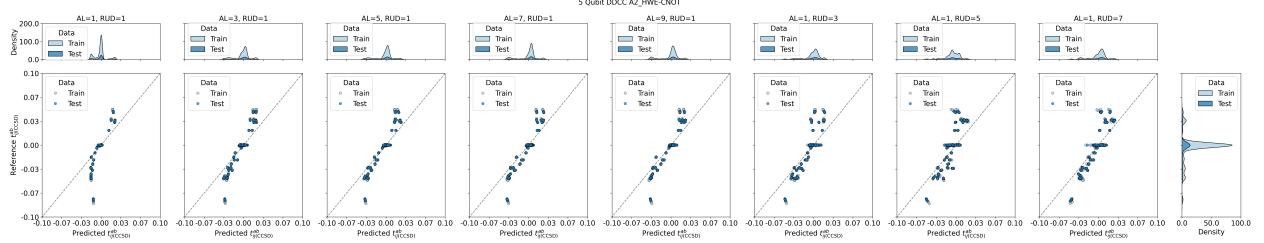
Figure 11

(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

5 Qubit DDCC A2\_HWE-CNOT



(a)



(b)

Figure 12: 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.

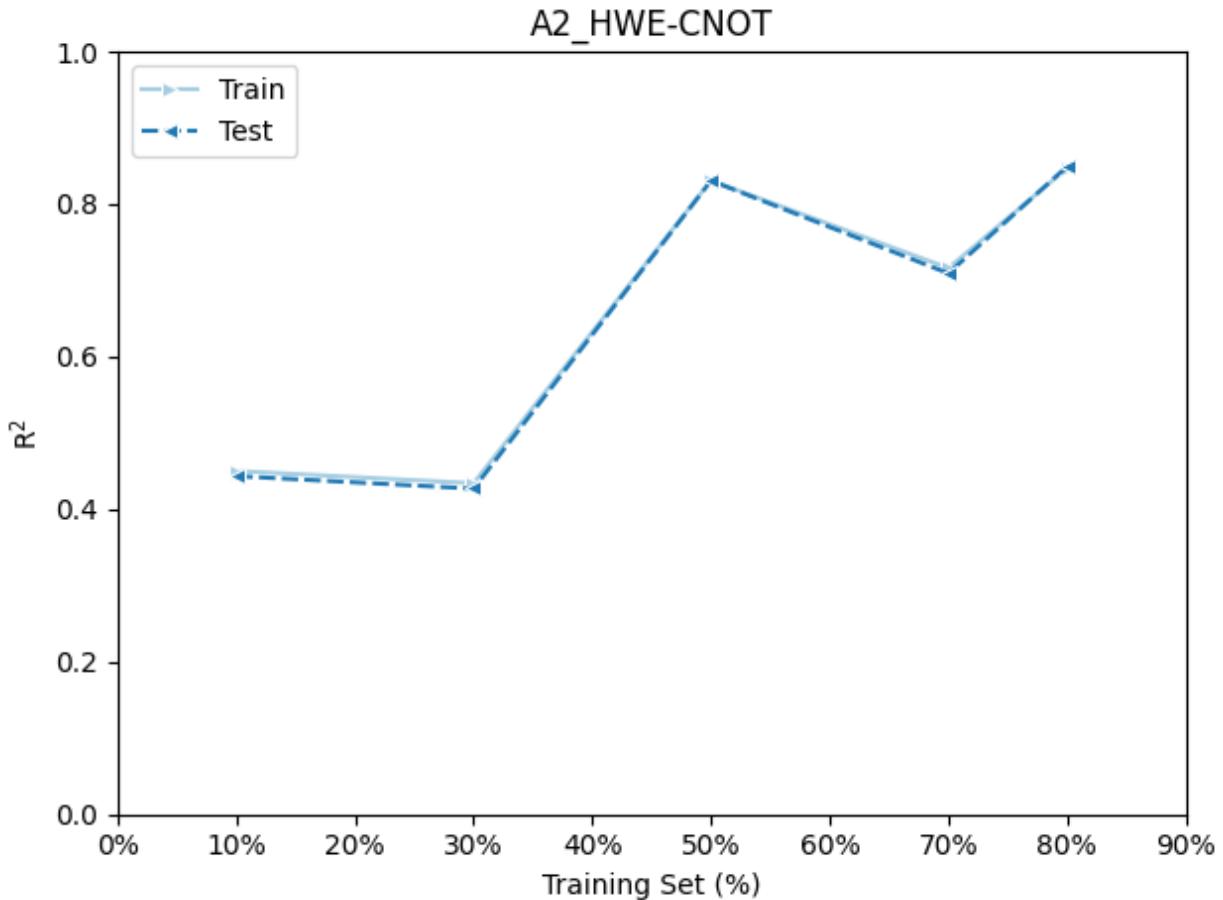


Figure 13

Ansaetze analysis<sup>42</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”

## 4 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...

## References

- (1) Behler, J. Perspective: Machine learning potentials for atomistic simulations. *JOURNAL OF CHEMICAL PHYSICS* **2016**, *145*.
- (2) 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.
- (3) 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 Network Potentials. *J. Chem. Inf. Model.* **2020**, *60*, 3408–3415, Publisher: American Chemical Society.
- (4) 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.
- (5) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *SCIENTIFIC DATA* **2014**, *1*.
- (6) 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.

- (7) 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.
- (8) 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.
- (9) Zhong, M. et al. Accelerated discovery of CO<sub>2</sub> electrocatalysts using active machine learning. *NATURE* **2020**, *581*, 178+.
- (10) 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.
- (11) 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.
- (12) Goh, G. B.; Hodas, N. O.; Vishnu, A. Deep learning for computational chemistry. *JOURNAL OF COMPUTATIONAL CHEMISTRY* **2017**, *38*, 1291–1307.
- (13) 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.
- (14) Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine learning for molecular and materials science. *NATURE* **2018**, *559*, 547–555.
- (15) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *SCIENCE* **2018**, *361*, 360–365.

- (16) 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+.
- (17) Janet, J. P.; Kulik, H. J. *Machine Learning in Chemistry*; ACS In Focus; American Chemical Society, 2020.
- (18) 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.
- (19) 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*.
- (20) McArdle, S.; Endo, S.; Aspuru-Guzik, A.; Benjamin, S.; Yuan, X. Quantum computational chemistry. *REVIEWS OF MODERN PHYSICS* **2020**, *92*.
- (21) 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.
- (22) Biamonte, J.; Wittek, P.; Pancotti, N.; Rebentrost, P.; Wiebe, N.; Lloyd, S. Quantum machine learning. *Nature* **2017**, *549*, 195–202.
- (23) Benedetti, M.; Lloyd, E.; Sack, S.; Fiorentini, M. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.* **2019**, *4*, 043001.
- (24) Suzuki, T.; Katouda, M. Predicting toxicity by quantum machine learning. *J. Phys. Commun.* **2020**, *4*, 125012.

- (25) 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.
- (26) 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.
- (27) 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.
- (28) 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.
- (29) 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.
- (30) 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.
- (31) 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.
- (32) 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.

- (33) 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].
- (34) 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.
- (35) 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.
- (36) 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.
- (37) 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.
- (38) Mitarai, K.; Negoro, M.; Kitagawa, M.; Fujii, K. Quantum circuit learning. *Phys. Rev. A* **2018**, *98*, 032309.
- (39) 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].
- (40) 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].

- (41) Bergholm, V. et al. PennyLane: Automatic differentiation of hybrid quantum-classical computations. 2022; <http://arxiv.org/abs/1811.04968>, arXiv:1811.04968 [quant-ph].
- (42) 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.
- (43) 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.
- (44) Suzuki, Y. et al. Qulacs: a fast and versatile quantum circuit simulator for research purpose. *Quantum* **2021**, *5*, 559, arXiv:2011.13524 [quant-ph].
- (45) 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.
- (46) Virtanen, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **2020**, *17*, 261–272.
- (47) Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
- (48) Jones, G. M.; Story, B.; Maroulas, V.; Vogiatzis, K. D. *Molecular Representations for Machine Learning*; ACS In Focus; American Chemical Society, 2023.
- (49) RDKit. <https://www.rdkit.org/>.
- (50) 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.

- (51) 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.
- (52) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling* **2010**, *50*, 742–754, Publisher: American Chemical Society.
- (53) Adams, H.; Emerson, T.; Kirby, M.; Neville, R.; Peterson, C.; Shipman, P.; Chepush-tanova, 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.
- (54) 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.
- (55) 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.
- (56) Tralie, C.; Saul, N.; Bar-On, R. Ripser.py: A Lean Persistent Homology Library for Python. *Journal of Open Source Software* **2018**, *3*, 925.
- (57) 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*.
- (58) 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.
- (59) García-Andrade, X.; García Tahoces, 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.

- (60) Lundberg, S. M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems. 2017.
- (61) 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.
- (62) 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.
- (63) 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.
- (64) Smith, D. G. A. et al. Psi4NumPy: An Interactive Quantum Chemistry Programming Environment for Reference Implementations and Rapid Development. *J. Chem. Theory Comput.* **2018**, *14*, 3504–3511, Publisher: American Chemical Society.