A Combinatorial Search of Parameterized Quantum Circuit Learning for Chemical Applications

Grier M. Jones,[†,‡] Nick Taylor,[†] Viki Kumar Prasad,[†,‡] Ulrich Fekl,[*,‡] and Hans-Arno Jacobsen[*,†]

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

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

E-mail: ulrich.fekl@utoronto.ca; 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.

- Abstract
- Introduction
- Methods
- Datasets
- Results and Discussion
  - Function fitting
    * All (5 + 16 qubit)
    * RUD + AL (5 + 16 qubit)
    * Error mitigation on FakeQuebec
  - BSE
    * All 5 qubit
    * Truncated 16 qubit set (cost analysis 5 qubit)
    * RUD + AL (5 + 16 qubit)
    * Real device (without error mitigation and with whatever the best is from function fitting)
  - DDCC (5 qubit only because classically it can be done with 5 features!)
    * Truncated 5 qubit set (based on BSE 5 qubit cost analysis)
    * RUD + AL (5 qubit)
    * Real device (without error mitigation and with whatever the best is from function fitting)
- Conclusion

1 Introduction

- General background information
- Specific background information
- A description of the gap in our knowledge that the study was designed to fill
- A statement of study objective, and (optionally) a brief summary of study

Within the field of chemistry, tools such as machine learning and quantum computing have become popular due to the promise of discoverying new molecules and materials. Machine learning in chemistry has already become a broadly applicable tool for exploring chemical compound space,[1,2] accelerating wave functions,[3,4] and for applications, such as catalysis and drug discovery.[5–9] [Sentence about quantum computing] Another popular avenue, which combines quantum computing and machine learning, is quantum machine learning (QML). QML utilizes quantum algorithms during some portion of the machine learning process in an attempt to outperform classical algorithms.[10] On the current era of quantum computers, parameterized quantum circuits (PQCs) offer an example of QML that is performed using a hybrid quantum-classical feedback loop, where the variational parameters are optimized classical, while the circuit is ran on quantum hardware.[11,12]

PQCs are often composed using two different subcircuits: the encoding layer, which is used to encode the features into the quantum circuit, and variational layers, which are composed of parameterized gates which can be optimized classical.

PQCs are composed of data-encoding and variational subcircuits, which can be repeated multiple times.

While several examples exist for PQCs applied to chemical applications, they are largely relegated to drug discovery[12–19] and experimental properties.[20] Herein, we explore two different chemical datasets: one consisting of bond separation energies of 49 unique bond types, called the BSE49 dataset,[21] and another which utilizes data-driven coupled cluster (DDCC)[3] to predict the

coupled-cluster singles and doubles wave function parameters using lower-level quantum chemistry methods.

In this study, we create a robust Python code base for exploring a set of 14 data-encoding and 12 variational (ansatz) subcircuits for a total of 168 combined PQCs. Our work includes the single and double encoding layers found in the paper of Suzuki and Katouda,[12] including the encoding of Mitarai *et al.*,[22] along with using instantanious quantum polynomial (IQP) as an encoding layer.[23] All of our variational (ansatz) layers can be found in the study of Sim *et al.*,[24] where the expressiblity and entanglement capability of these circuits is analyzed. We also examine the effects of the number of re-upload and ansatz layers, the effects of feature set size, and various error mitigation techniques.

[25]

QML:[10]

- "Quantum machine learning software makes use of quantum algorithms as part of a larger implementation. By analysing the steps that quantum algorithms prescribe, it becomes clear that they have the potential to outperform classical algorithms for specific problems (that is, reduce the number of steps required). This potential is known as quantum speedup."

- "The notion of a quantum speedup depends on whether one takes a formal computer science perspective—which demands mathematical proofs—or a perspective based on what can be done with realistic, finitesize devices—which requires solid statistical evidence of a scaling advantage over some finite range of problem sizes. For the case of quantum machine learning, the best possible performance of classical algorithms is not always known."

- "Determination of a scaling advantage contrasting quantum and classical machine learning would rely on the existence of a quantum computer and is called a 'benchmarking' problem. Such advantages could include improved classification accuracy and sampling of classically inaccessible systems. Accordingly, quantum speedups in machine learning are currently characterized using idealized measures from complexity theory: query complexity and gate

complexity (see Box 1 and Box 1 Table). Query complexity measures the number of queries to the information source for the classical or quantum algorithm. A quantum speedup results if the number of queries needed to solve a problem is lower for the quantum algorithm than for the classical algorithm. To determine the gate complexity, the number of elementary quantum operations (or gates) required to obtain the desired result are counted."

- Input problem: "Classical data must be input before being processed on a quantum computer. This 'input problem' often has little overhead but can present a serious bottleneck for certain algorithms. Likewise, the 'output problem' is faced when reading out data after being processed on a quantum device. Like the input problem, the output problem often causes a noticeable operational slowdown."

- QML challenges: "These hardware challenges are technical in nature, and clear paths exist towards overcoming them. They must be overcome, however, if quantum machine learning is to become a 'killer app' for quantum computers. As noted previously, most of the quantum algorithms that have been identified face a number of caveats that limits their applicability. We can distill the caveats mentioned above into four fundamental problems. (1) The input problem. Although quantum algorithms can provide dramatic speedups for processing data, they seldom provide advantages in reading data. This means that the cost of reading in the input can in some cases dominate the cost of quantum algorithms. Understanding this factor is an ongoing challenge. (2) The output problem. Obtaining the full solution from some quantum algorithms as a string of bits requires learning an exponential number of bits. This makes some applications of quantum machine learning algorithms infeasible. This problem can potentially be sidestepped by learning only summary statistics for the solution state. (3) The costing problem. Closely related to the input/outputproblems, at present very little is known about the true number of gates required by quantum machine learning algorithms. Bounds on the complexity suggest that for sufficiently large problems they will offer huge advantages, but it is still unclear when that crossover point occurs. (4) The benchmarking

problem. It is often difficult to assert that a quantum algorithm is ever better than all known classical machine algorithms in practice because this would require extensive benchmarking against modern heuristic methods. Establishing lower bounds for quantum machine learning would partially address this issue."

PQCs:[11]

- "PQCs are typically composed of fixed gates, e.g. controlled NOTs, and adjustable gates, e.g. qubit rotations. Even at low circuit depth, some classes of PQCs are capable of generating highly non-trivial outputs. 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). The demonstration of quantum supremacy is an important milestone in the development of quantum computers. In practice, however, it is highly desirable to demonstrate a quantum advantage on applications. The main approach taken by the community consists in formalizing problems of interest as variational optimization problems and use hybrid systems of quantum and classical hardware to find approximate solutions. The intuition is that by implementing some subroutines on classical hardware, the requirement of quantum resources is significantly reduced, particularly the number of qubits, circuit depth, and coherence time. Therefore, in the hybrid algorithmic approach NISQ hardware focuses entirely on the classically intractable part of the problem."

PQCs:[12]

- "More recently, quantum machine learning (QML) [26–30] is a rapidly growing research field that combines near-term quantum algorithms and machine learning techniques. In particular, parameterized quantum circuits (PQCs) have been considered as machine learning models with high expressive power within the hybrid quantum–classical framework [31, 32]. PQCs are typically composed of fixed quantum gates (e.g., qubit rotations and entangling

gates) in a shallow circuit layout, with variable parameters optimized in a classical feedback loop."

- "To our knowledge, however, the application of QML to regression tasks has not been fully investigated in the literature. It remains unclear what kinds of quantum states should be used in order to generate the feature map with high expressibility that is suited for real-world data sets."

- "quantitative structure–toxicity relationship (QSTR) models for predicting the toxicity of 221 phenols. While there are a variety of QSAR/QSTR models (e.g., 3D-QSAR [1, 4]),asafirst step we employ QSAR/QSTR models including molecular descriptors such as hydrophobicity, acidity constant, and frontier orbital energies. There have been quantum computations in biochemical and pharmaceutical areas, such as protein folding [43–45], molecular similarity [46], and biological data [47]; yet, there has been no study on quantum application to QSAR modeling, albeit an important part of ligand-based computer-aided drug design."

- "PQCs have been regarded as machine learning models with high expressive power within the framework of the hybrid quantum–classical approach. PQCs are usually composed of one-qubit rotations and two-qubit entangling operations in a shallow circuit layout, with parameters optimized in a feedback loop."

- "Combining near-term quantum algorithms and machine learning, QML using the framework of PQCs is sometimes referred to as quantum circuit learning (QCL)"

- "PQCs consist of three components: the encoder circuit, the variational circuit, and the measurement for the estimation of the loss function"

PQCs:[20]

- What they do: "ntroduce quantum circuit learning (QCL) as an emerging regression algorithm for chemo- and materials-informatics. The supervised model, functioning on the rule of quantum mechanics, can process linear and smooth non-linear functions from small

datasets (<100 records). Compared with conventional algorithms, such as random forest, support vector machine, and linear regressions, the QCL can offer better predictions with some one-dimensional functions and experimental chemical databases"

- "Here, we introduce quantum circuit learning (QCL), an emerging algorithm for supervised learning.15–18 QCL works on the rule of quantum mechanics. It can predict various parameters likewise to classical models. The current quantum systems (noisy intermediate-scale quantum computers: NISQ)19,20 face the problems of calculation noise and the limited processable number of information units (qubits)."

- "Since QCL is a frontier for machine learning, few reports have been published on authentic regression tasks. The success of prediction with a toxicity dataset of organic molecules was reported,17 whereas the study was still conceptual. The prediction processes and advantages have been unclear, especially from chemical and material viewpoints. Here, we conducted a more comprehensive study on QCL with standard datasets of one-dimensional functions and chemical properties. Both simulated and actual quantum computing was undertaken to clarify the challenges of QCL. Various hyperparameters were optimized for higher accuracy. The comparison with conventional models contrasted the benefits of the new approach: capable of learning both linear and non-linear functions even from small datasets. The property was also favorable for predicting the so-called extrapolating data region, which is essential for chemical and material research."

- They cite [12,22,26] as examples of PQCs/quantum circuit learning for regression task

- Justification for starting with 1D regression: "Preliminary optimization of quantum circuits15,17 and our optimization revealed that the following configuration was concise and practical for regression tasks (for details, see results and explanations in Fig. S1"

We introduce *qregress*, a modular Python-based code built on PennyLane for exploring PQCs for regression based learning tasks. We analyze three datasets, a function fitting dataset, used for

model calibration, and two chemically relevant datasets that offer challenging learning tasks for PQCs.

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.[12] In this study, we utilize the Mitarai (M),[22] single- (A1) and double-angle (A2) encoding layers found in Ref.[12], along with the instantaneous quantum polynomial (IQP) circuit found in Refs.[23] and[27]. In the following section, we follow the notations derived from Ref.[12]. Encoding layers work mapping a d-dimensional feature vector, $\mathbf{x} = (x_1, x_2, \ldots, x_d)^T \in \mathbb{R}^d$, 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,

$$U_{\Phi(x)} = \prod_l E^l_{\text{ent}} U_{\phi_l(\mathbf{x})} \tag{1}$$

where, E^l_{ent} 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.[12], we choose $l \in \{1, 2\}$, such that when $l = 1$, E_{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_{A1}, U_{A2}, U_{M}, or U_{IQP}. The single-angle encoding (Fig. 1 **(a)**) is the simplest and takes the following form,

$$U_{\text{A1}} = \prod_{i=0}^{n} R^Y_i(x_i), \tag{2}$$

where R^Y_i 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

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

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

$$U_{\mathrm{M}} = \prod_{i=0}^{n} R_i^Z(\arccos(x_i^2)) R_i^Y(\arcsin(x_i^2)). \tag{4}$$

Following the formulation provided in the PennyLane[28] software package, the IQP encoding layer (Fig. 1 **(d)**) is defined as,

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

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

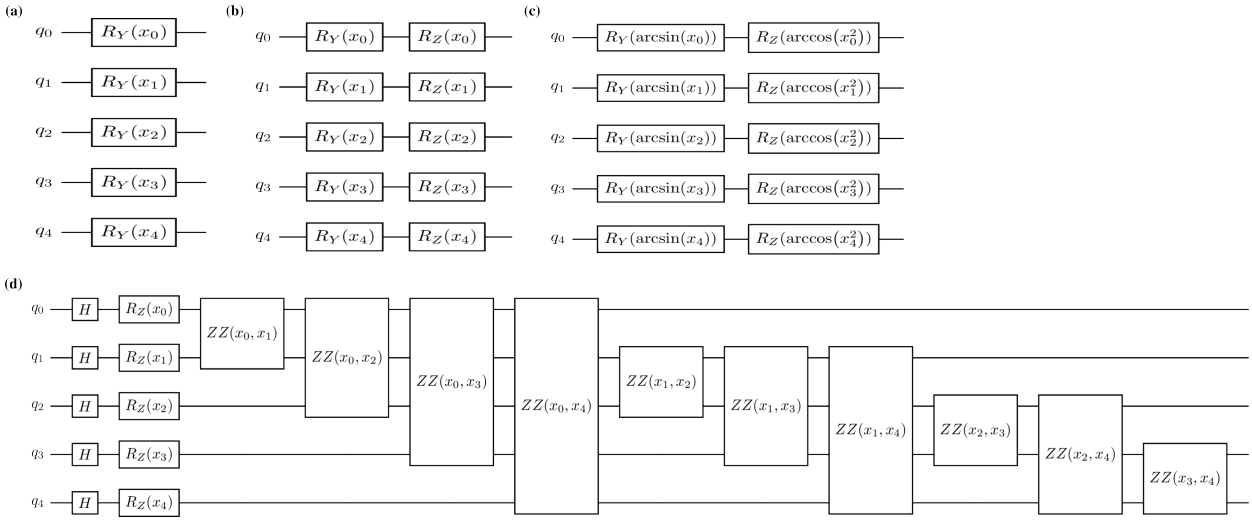


Figure 1: (a) Single angle (A1) encoding, (b) double angle (A2) encoding, (c) Mitarai (M) encoding, and (d) Instantaneous Quantum Polynomial (IQP) encoding

When $l = 2$, like in Ref.[12] we choose entanglement gates, E^1_{ent} and E^2_{ent} to be equivalent, and the encoding layer takes the following form, $U_{\Phi(x)} = E_{\mathrm{ent}} U_{\phi_2(\mathbf{x})} E_{\mathrm{ent}} U_{\phi_1(\mathbf{x})}$. 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.

Table 1: Add something smart

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

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.[12] and[24], since Ref.[24] 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,

$$U(\boldsymbol{\theta}) = \prod_{v} U_v(\boldsymbol{\theta}_v), \tag{6}$$

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, 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)**),

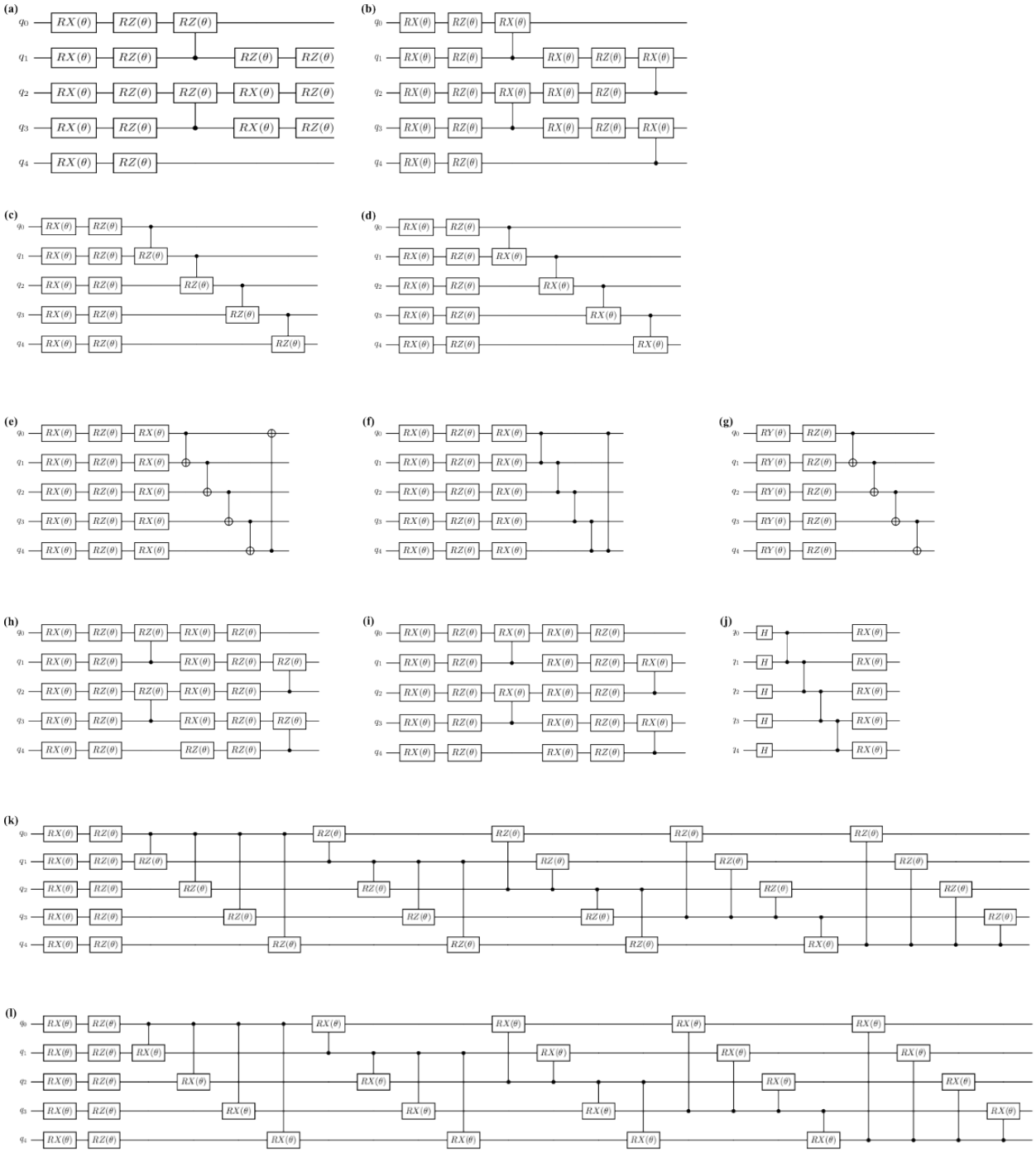


Figure 2: (a) Modified-Pauli-CRZ, (b) Modified-Pauli-CRX, (c) Efficient-CRZ, (d) Efficient-CRX, (e) HWE-CNOT, (f) HWE-CZ, (g) ESU2, (h) Full-Pauli-CRZ, (i) Full-Pauli-CRX, (j) Hadamard, (k) Full-CRZ, and (l)Full-CRX

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,

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

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.[29]

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,

$$\hat{y}_i = \langle\Psi|\, Z_0\, |\Psi\rangle_i\,. \tag{8}$$

The set of predicted target values, $\hat{\boldsymbol{y}} = (\hat{y}_1, \ldots, \hat{y}_N) \in \mathbb{R}^N$, where N is the number of samples, is then passed to the loss function, $\mathscr{L}(\boldsymbol{y}, \hat{\boldsymbol{y}})$, where y_i belongs to the set of true target values $\boldsymbol{y} = (y_1, \ldots, y_N) \in \mathbb{R}^N$. In practice, $\mathscr{L}$ can be any loss function but we choose to use the mean square error loss function denoted as,

$$\mathscr{L}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2. \tag{9}$$

Implementation

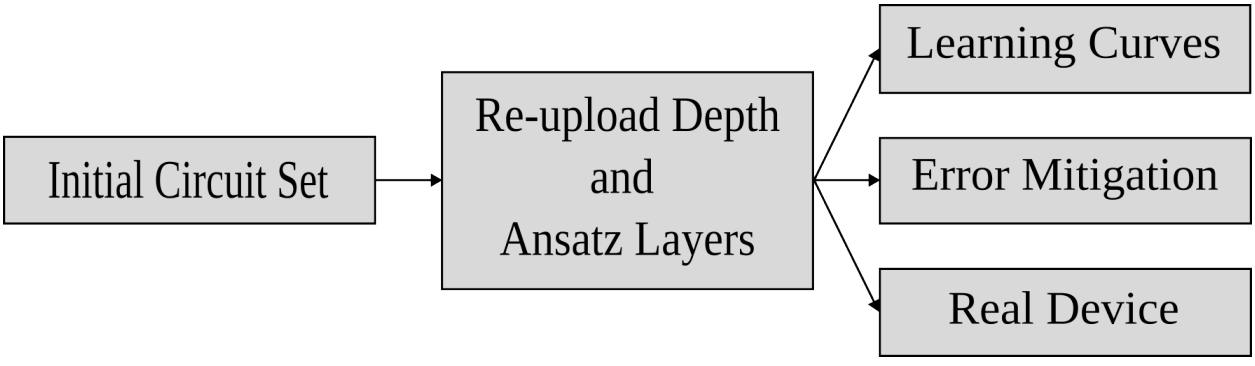


Figure 3: Make this figure highlight the modularity of *qregress*

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

We perform all simulation calculations using PennyLane,[28] either using Qulacs[30] for state vector calcualtions, while noisy calculations were performed using *qiskit-aer* with the *FakeQuebec* backend as implemented in the PennyLane-Qiskit plugin.[31] We perform calculations in the *ibm_quebec* device using circuits implemented using Qiskit,[31] 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.[32] Each optimizer was chosen based on the performance for the given task. All features ($\mathbf{x}$) and target values ($\mathbf{y}$) were scaled using the MinMaxScaler in Scikit-learn,[33] such that all featues 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.

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 three datasets: a function fitting dataset (Figs. 4a,4b, and 4c), consisting of a noisy functions used for model calibration; a dataset of bond separation energies (BSE) of molecules (Figs. 4d and 4e), 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. 4f).[3] 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.

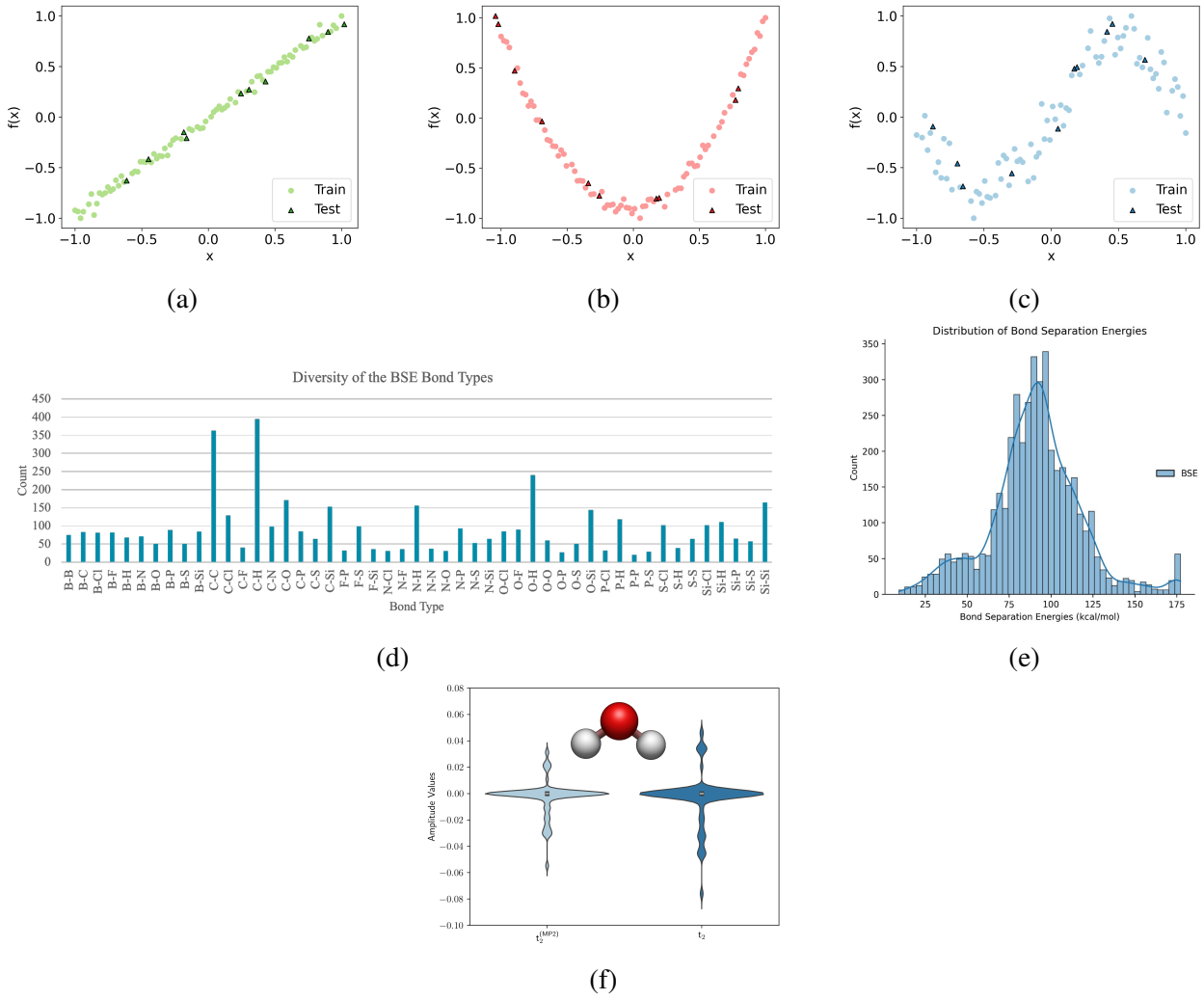


Figure 4: Examples of the datasets explored in this study. The function fitting dataset used for model calibration include (a) linear (green), (b) quadratic (pink), and (c) sine (blue) functions. For the BSE49 dataset the distributions of the (d) bond types and the (e) bond separation energies in kcal/mol. For the DDCC dataset distributions of the initial MP2 t_2-amplitudes and the optimized CCSD t_2-amplitudes are shown in (f).

The function fitting dataset consists of a noisy linear (Fig. 4a), quadratic (Fig. 4b), and sine (4c) function. For each function, 90 points are generated, where 80 are used for training set and 10 for the test set. This dataset is used for calibration of the 168 PQCs, along with providing additional insights into the effects of circuit depth, training set size, and error mitigation on model performance.

Following model calibration using the function fitting dataset, we explore the applicablity of PQCs for complex chemically relevant machine learning tasks. The first chemically motivated

dataset we explore is the BSE49 dataset, which contains the bond seperation energies (BSE) for the homolytic bond cleavage of covalently bonded molecules, such as $A-B \longrightarrow A^{\cdot} + B^{\cdot}$.[21] This dataset consists of 4394 datapoints, 1951 of which are existing and 2443 are hypthoetical 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.[34] Using RDKit[35] we examined three commonly applied graph-based molecular preresentations, Molecular ACCess Systems (MACCS),[36] Morgan or extended-connectivity fingerprints,[37,38] 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.[39–41] We use the implementation from Townsend *et al.*,[40] which uses the Ripser Python package to generate PIs.[42] Lastly, we explre two physics-based representations, Coulomb matrices (CMs)[43] and smooth overlap of atomic positions (SOAPs), that were generated using DScribe.[44] 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.,[45] 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 sckit-learn.[33] Based on our results shown in Fig. 5 we found that Morgan finger-

prints We found that the best molecular representation across all models test, as shown in Fig. 5, was Morgan fingerprints using the *sub* formulation.

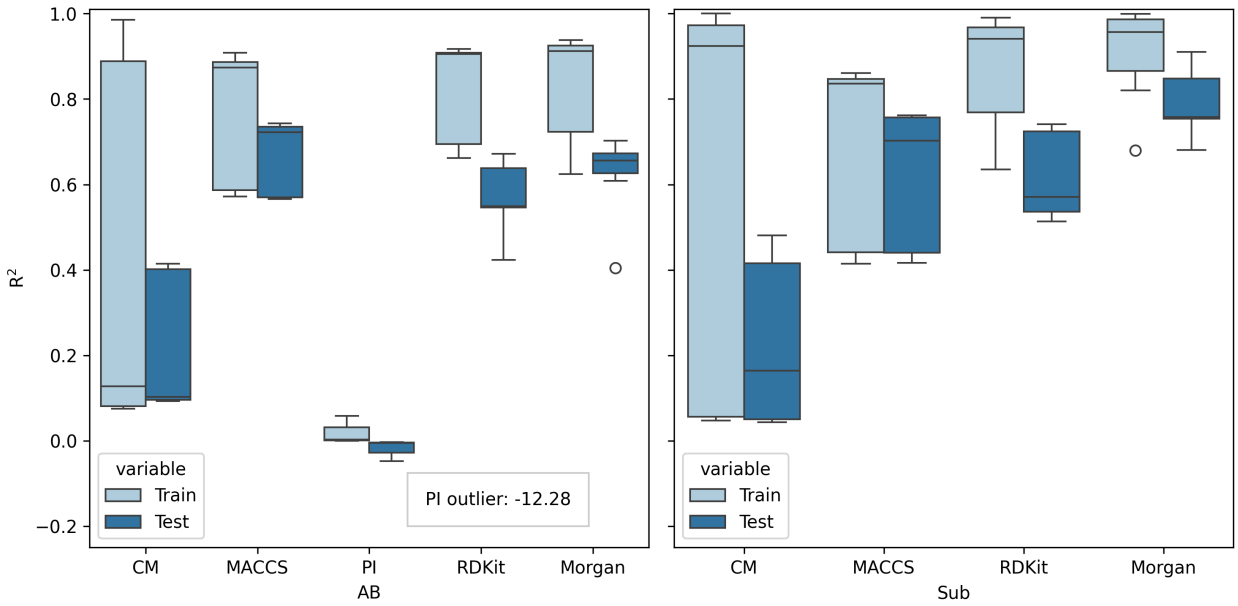
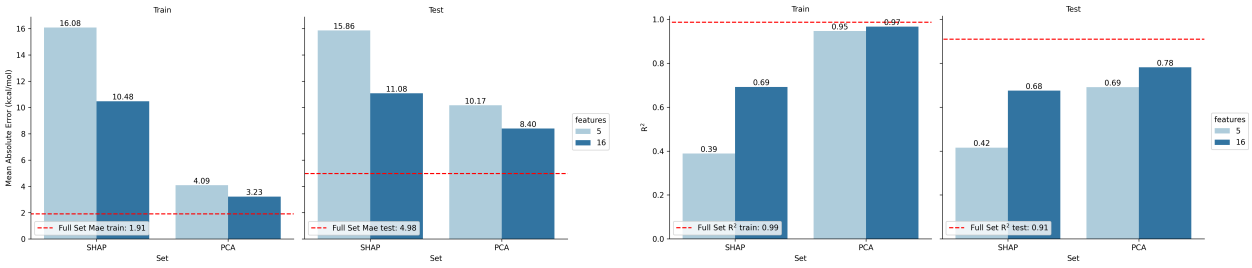


Figure 5: Coulomb matrices (CMs), Molecular ACCess Systems (MACCS), persistence images (PIs), RDKit and Morgan fingerprints. Performance of a diverse set of molecular representations R^2

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 fingerpints 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)[46] and principal component analysis (PCA), as implemented in scikit-learn.[33] 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^2s 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^2s 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^2s for PCA with 5 and 16 features also shows improvement over the reductions using SHAP, with R^2s 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.



(a) (b)

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.[3,4] 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,

$$|\Psi_{CC}\rangle = \exp\left(\hat{T}\right)|\Psi_0\rangle \tag{10}$$

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,

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

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,

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

where ε_i and ε_j denote the orbital energies of the occupied orbitals i and j, while the virtual orbitals a and b are denoted by ε_a and ε_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 Hamiltonain (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 magntudes 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[47] and frozen core orbitals. All data was generated using Psi4[48] and Psi4Numpy.[49] 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 $(N_{occ})^2(N_{virt})^2$, where N_{occ} denotes the number of occupied 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[3]

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^2s 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 ($t^{ab}_{ij(\mathrm{MP2})}$), the magnitude of the MP2 t_2-amplitudes, and the difference in orbital energies ($\varepsilon_i + \varepsilon_j - \varepsilon_a - \varepsilon_b$).

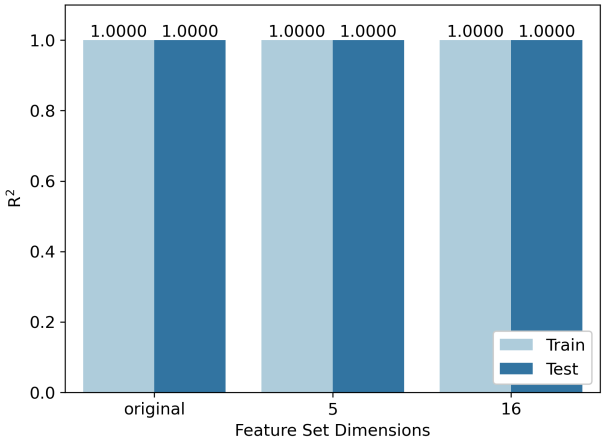


Figure 7

3 Results and Discussion

Function Fitting

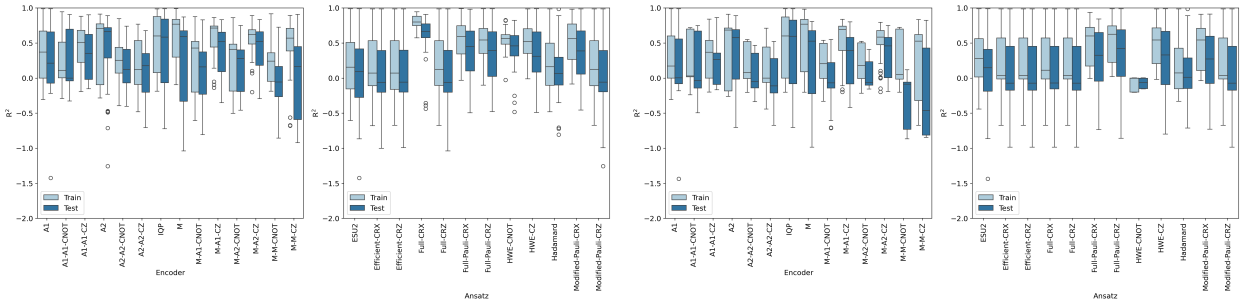
To calibrate the PQCs, we perform an initial study of the 168 encoder-ansatz pairs using both 5 and 16 qubits. Individual heat maps showing the performance of the encoder-ansatz pairs and box plots showing the overall statistics for each encoder and ansatz are shown for the 5 and 16 qubit

models in SI Section S1.

Starting with the five qubit function fitting, we found that out of the 168 encoder-ansatz pairs, for the linear function (SI Fig. S1a) is is A1_ESU2 with a train and test R^2 of 0.9886 and 0.9893, respectively. Box plots for the encoders (left) and variational layers (right) are shown in SI Fig. S1b, where, the best encoder is IQP, with a mean R^2 of 0.8434, and best ansatz layer Full-CRX, with a mean R^2 of 0.7562. For the quadratic function, the best encoder-ansatz pair is A1-A1-CNOT_Full-CRX with a train and test R^2 of 0.8809 and 0.7309, respectively, as shown in SI Fig S1c. On average the best encoder is A1-A1-CNOT with an R^2 of 0.5541 and best variational layer is Full-CRX with an R^2 of 0.5958 (SI Fig. S1d). For the sine function, SI Fig. S1e, M-M-CZ_Full-CRX is the best encoder-ansatz pair with a train and test R^2 of 0.8887 and 0.9081, respectively. As shown in SI Fig. S1f, the best encoder is M with an average R^2 of 0.6456 and best variational layer is Full-CRX with a mean R^2 of 0.7551. Across all three five qubit function fitting datasets, on average, the best encoders are M-A2-CZ with an average R^2 of 0.4776 (Fig. 8a left) and ansatz is Full-CRX with an average R^2 of 0.7023 (Fig. 8a right).

For the sixteen qubit function fitting evaluation of the 168 encoder-ansatz pairs, we found that, similarly to the five qubit data, for the linear function the best pair is A1_ESU2 with train and test R^2s of 0.9886 and 0.9893, respectively, as highlighted in SI Fig. S2a. Like the five qubit data, the best encoder is also IQP with an average R^2 of 0.7800 (SI Fig. S2b left), while the best ansatz varies from the five qubit data, Full-Pauli-CRZ with a mean R^2 of 0.3947 (SI Fig. S2b right). The A1-A1-CZ_Modified-Pauli-CRX encoder-ansatz pair has the best overall model performance, highlighted in SI Fig. S2c, where the training set has an R^2 of 0.8415 and the test set has an R^2 of 0.7722. As demonstrated in SI Fig. S2d, the best encoders and ansätze are A1-A1-CNOT and Full-Pauli-CRX, with mean R^2s of 0.5759 and 0.4474, respectively. For the last sixteen qubit dataset, the sine function data (SI Fig. S2e), the best encoder-ansatz pair is A1-A1-CZ_Full-Pauli-CRZ with an R^2 of 0.8147 for the training set and an R^2 of 0.8626 for the test set. SI Fig. S2f highlights that the best encoder is M with a mean R^2 of 0.5824 and best ansatz, on average, is Full-Pauli-CRZ with an R^2 of 0.4881. Across all three function fitting datasets, the sixteen qubit data varies where

the best encoders are IQP with an average R^2 of 0.4506 and ansätze is Full-Pauli-CRZ with an average R^2 of 0.4393.



(a) (b)

Figure 8: The combined performance of all three function fitting datasets for both the (a) five and (b) sixteen qubit models is evaluated using the R^2 (y-axis) versus the encoders (left) and ansätze (right).

Following the initially analysis of the performance of the 168 PQCs using both five and sixteen qubits, we experimented with the effect of the re-upload depth (RUD) and number of ansatz layers (AL) using the sixteen qubit circuits. We chose to use the sixteen qubit circuits over the five qubit circuits since these circuits have complexity more similar to when the circuits are used for the chemical dataset in terms of the number of parameters and optimization times. From the initial 168 PQCs, for each function fitting dataset, we choose the five best (top row), five median (middle row), and five worst (bottom row) circuits to see the effects of increasing the RUD and AL on model performance as shown in Figs. 9a, 9b, and 9c. For the linear (Fig. 9a), quadratic (9b), and sine (Fig. 9c) data, some generalities arise for the five best circuits. We see that for these circuits, increasing the number of AL from 1 to 3 or 5 improves model performance more than RUD. For the median performers, the RUD improves model performance more than the ansatz layers. And lastly, for the worse performers a general trend does not exist and increasing the RUD and AL depends on the dataset and circuit architecture.

Despite these trends, we found that for the linear function, the best performing model is IQP_Full-Pauli-CRZ with a RUD and AL of 1. This circuit also has a training and test R^2 of 0.9883 and 0.9891, respectively, and a depth of 95. Unlike the best linear circuit, the best circuits

for both the quadratic and sine data, IQP_Full-Pauli-CRX and A2_HWE-CZ, respectively, benefit from an increased RUD, with the best models having a RUD of 5 and AL of 1. These two circuits differ in depth and performance, with the IQP_Full-Pauli-CRX for the quadratic data having a circuit depth of 520, training R^2 of 0.9817, and test R^2 of 0.9609 and the A2_HWE-CZ circuit for the sine data having a depth of 105 , training R^2 of 0.9156, and test R^2 of 0.9576. Following these insights, we applied these circuits for analyzing the effects of the training set size on model performance using learning curves and the effects of error mitigation using noisy simulation.

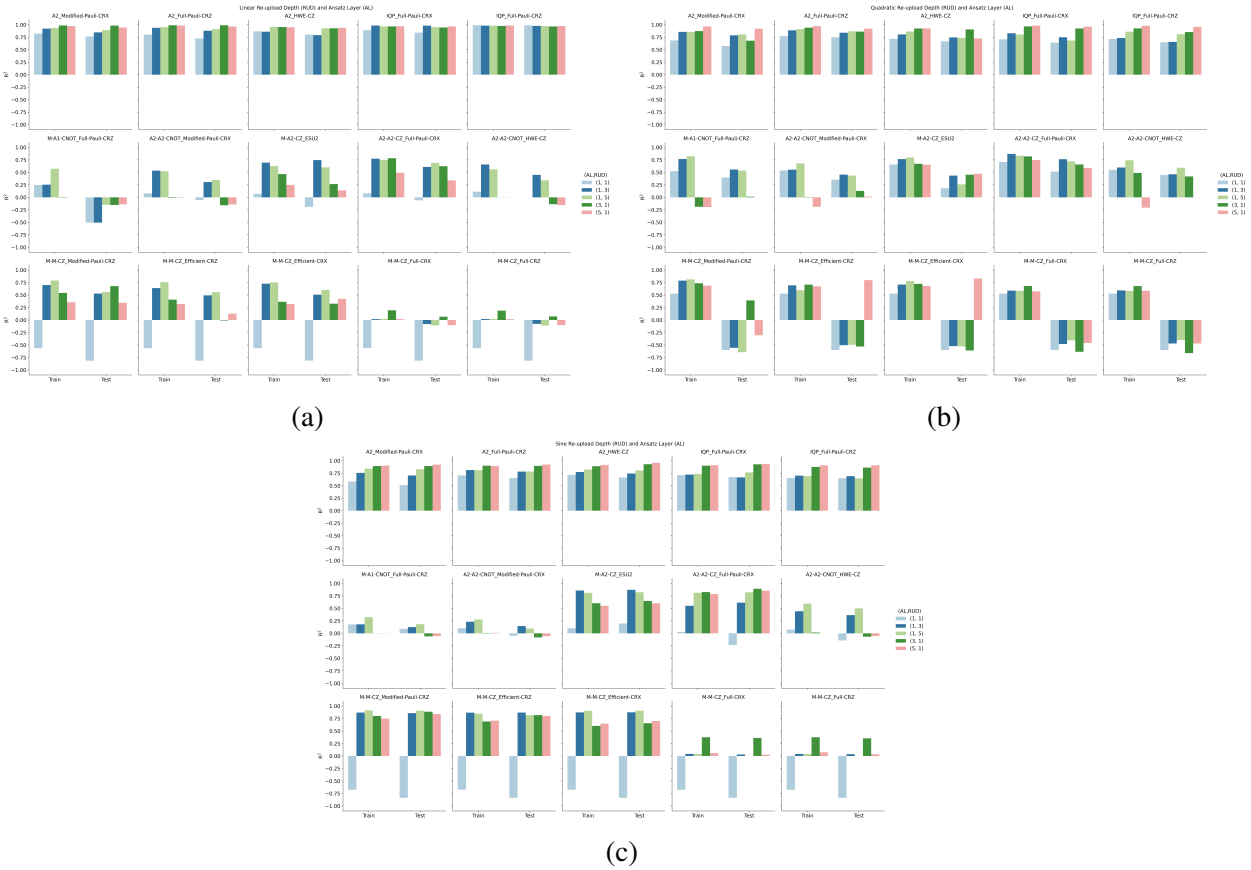


Figure 9: The best five (top row), median five (middle row), and worst five (bottom row) circuits for the 16 qubit (a) linear, (b) quadratic, and (c) sine data show that, in general, an increase in the re-upload depth (RUD) or ansatz layers (AL) increases model performance (R^2 shown on the y-axis).

One proposed advantage of using PQCs for ML tasks is that, in some cases, they can outperform classical ML models using less training data.[20] We examine this phenomena with learning curves using the best performing circuits previously described. The learning curves are generated

by varying the percentage of training data from 10-80% of the available data, while holding the number of test points to 10% of the total data. The three PQCs for each data set are compared to the classical models mentioned in subsection 2 in a statistical manner (left sub-figures in Figs. 10a, 10b, and 10c). Across the full learning curve, IQP_Full-Pauli-CRZ (linear function fitting) outperforms seven out of nine classical models for the training set and all of the classical models on the test set. On the quadratic dataset, using IQP_Full-Pauli-CRX, offers similar performance to the classical models, despite being outperformed on the training set across the learning curve. Similar to the IQP_Full-Pauli-CRZ using the linear data, IQP_Full-Pauli-CRX outperforms seven out of the nine classical models on the test set across the full learning curve. On the sine data, A2_HWE-CZ is consistent across the learning curve, outperforming six of the nine classical models on the training set and all of the models on the test set. For all three datasets, this implies that the PQCs offer similar performance to classical models that do well on these datasets but provide better results on the test set.

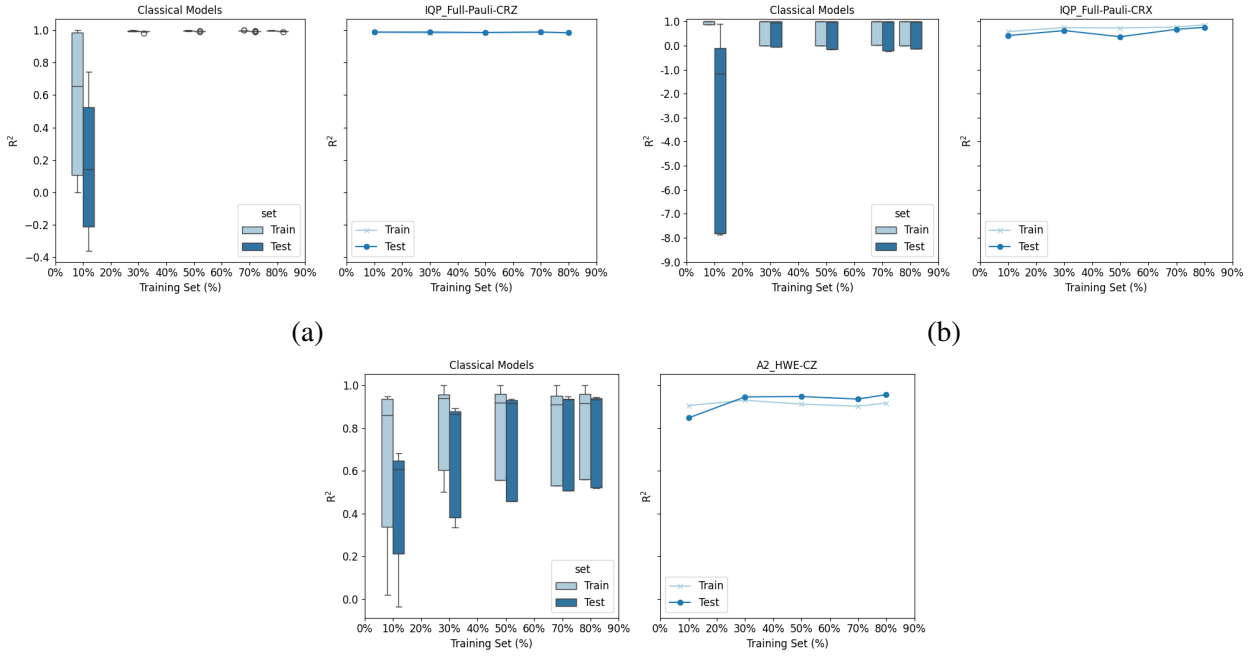


Figure 10: Learning curves of the best PQCs (right side in the subfigures) versus various classical models (left side in the subfigure) for the (a) linear, (b) quadratic, and (c) sine function datasets.

optimization level needs to be set to 0!

Lastly, using the function fitting datasets we analyze the effects of error mitigation on the model accuracy by comparing state vector simulations with noisy and error mitigated simulations using the *fake_quebec* backend. Initially, we explored two different methods of error mitigation zero-noise extrapolation (ZNE) using MITIQ[50] and TREX, as implemented in Qiskit. We found that both linear and Richardson ZNE were too costly to simulate on classical computers using 16 qubits for all three function fitting datasets and chose only to use TREX for error mitigation. Our analysis revealed that the state vector, TREX, and unmitigated IQP_Full-Pauli-CRZ (linear function fitting data) models all have R^2 between 0.98 and 0.99, for both the training and test sets, as highlighted in Fig. 11a. While the performance of the linear data can be attributed to the simiplicity of the dataset, for the quadratic function fitting data using IQP_Full-Pauli-CRX (Fig. 11b), we found that the model with unmitigated error has an R^2 of 0.65 for the training set and 0.45 for the test set. This model is improved slightly by the incorporation of error mitigation using TREX, where the training set R^2 is 0.69 and test set R^2 is 0.60. Overall, the model using TREX offers results more comparable to the state vector simulations, which has a R^2s of 0.80 and 0.68 for the training and test set, respectively. On the sine function fitting data, using A2_HWE-CZ (Fig. 11c), for the state vector, TREX, and unmitigated models, the training sets have R^2s of 0.92, 0.87, and 0.87, respectively, while the test sets have R^2s of 0.96, 0.89, and 0.91, respectively. Overall, while these results are for simple function fitting data, we concluded that TREX offers a computational efficient error mitigation method, with results similar to the state vector calculations.0

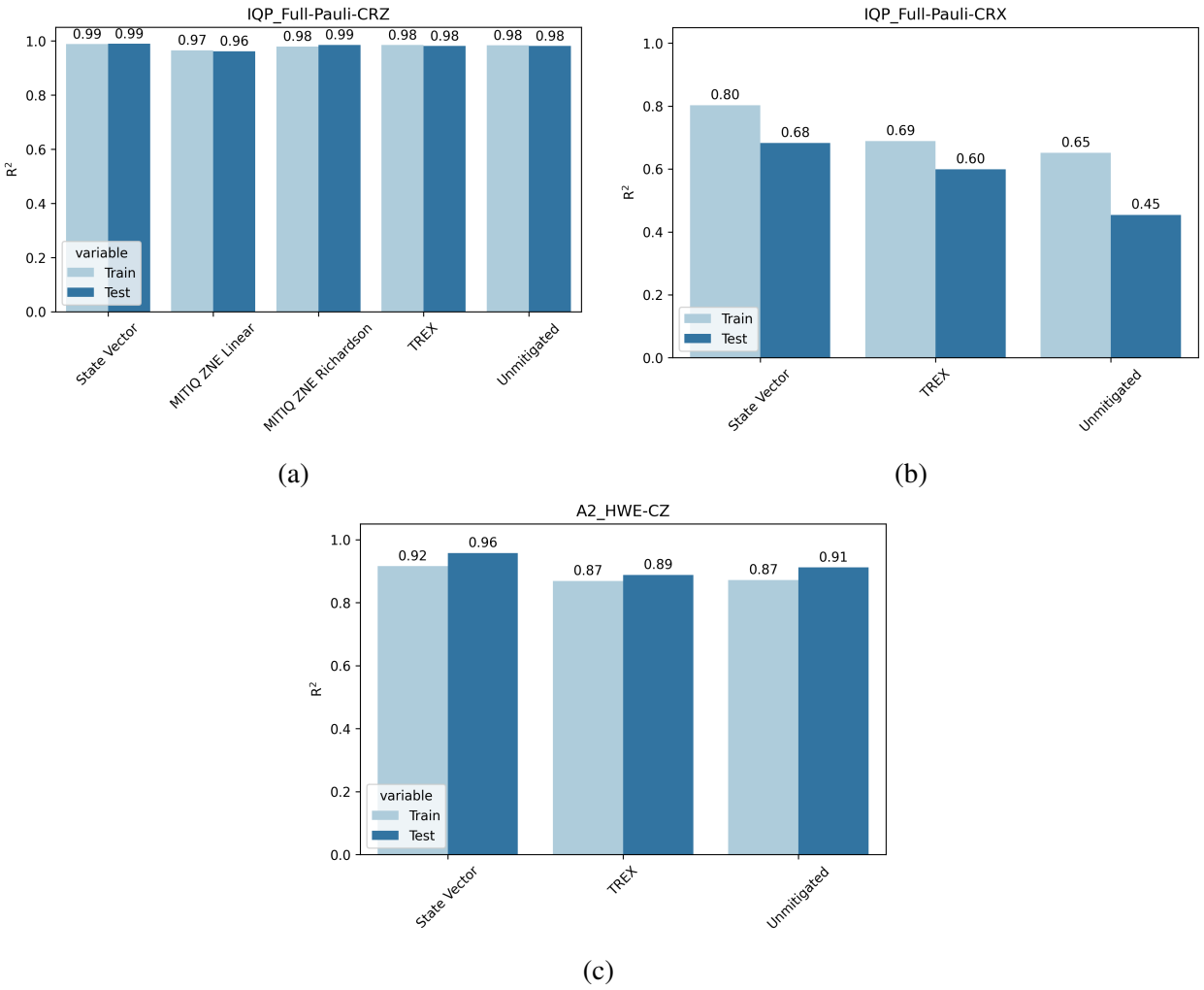


Figure 11: Error mitigation benchmarking for the (a) linear, (b) quadratic, and (c) sine function comparing unmitigated and mitigated error using TREX with state vector simulations.

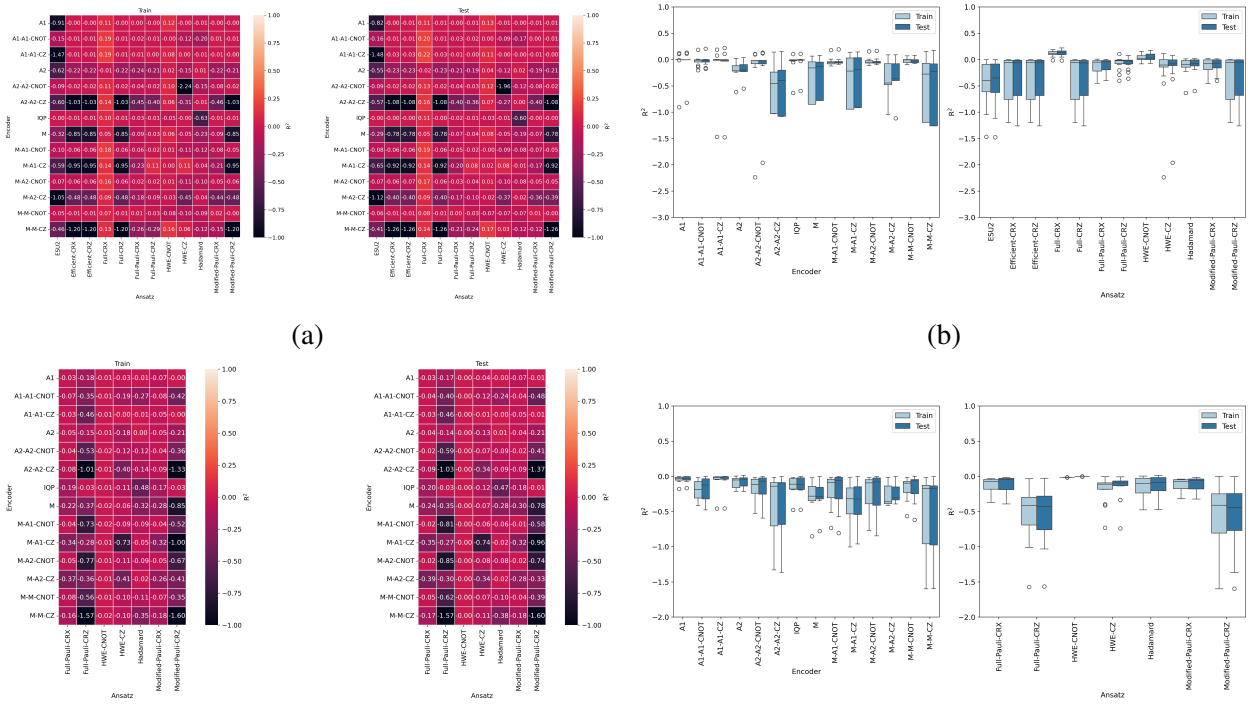
BSE

The first chemically relevant dataset we explore is the BSE49 dataset. As highlighted in Figs. 4d and 4e, this dataset consists of a diverse set of representative chemical bonds and bond separation energies. Herein, we follow a similar analysis to the function fitting data, starting off with an analysis of a large set of PQCs using five and sixteen qubits with RUDs and ALs of 1. From these two sets of data, we choose the best two models to study how an increase in the RUD and AL effects model performance. [learning curves] [Conclude we do not want to run this on real hardware due to the lack of promising results using simulation]

five qubit 12a 12b best encoder-ansatz pair: train R^2/test R^2 Best encoder on average train R^2/test R^2 Best ansatz on average train R^2/test R^2

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

sixteen qubit 12c 12d removed the really bad ones from five qubit best encoder-ansatz pair: train R^2/test R^2 Best encoder on average train R^2/test R^2 Best ansatz on average train R^2/test R^2



(a) (b) (c) (d)

Figure 12

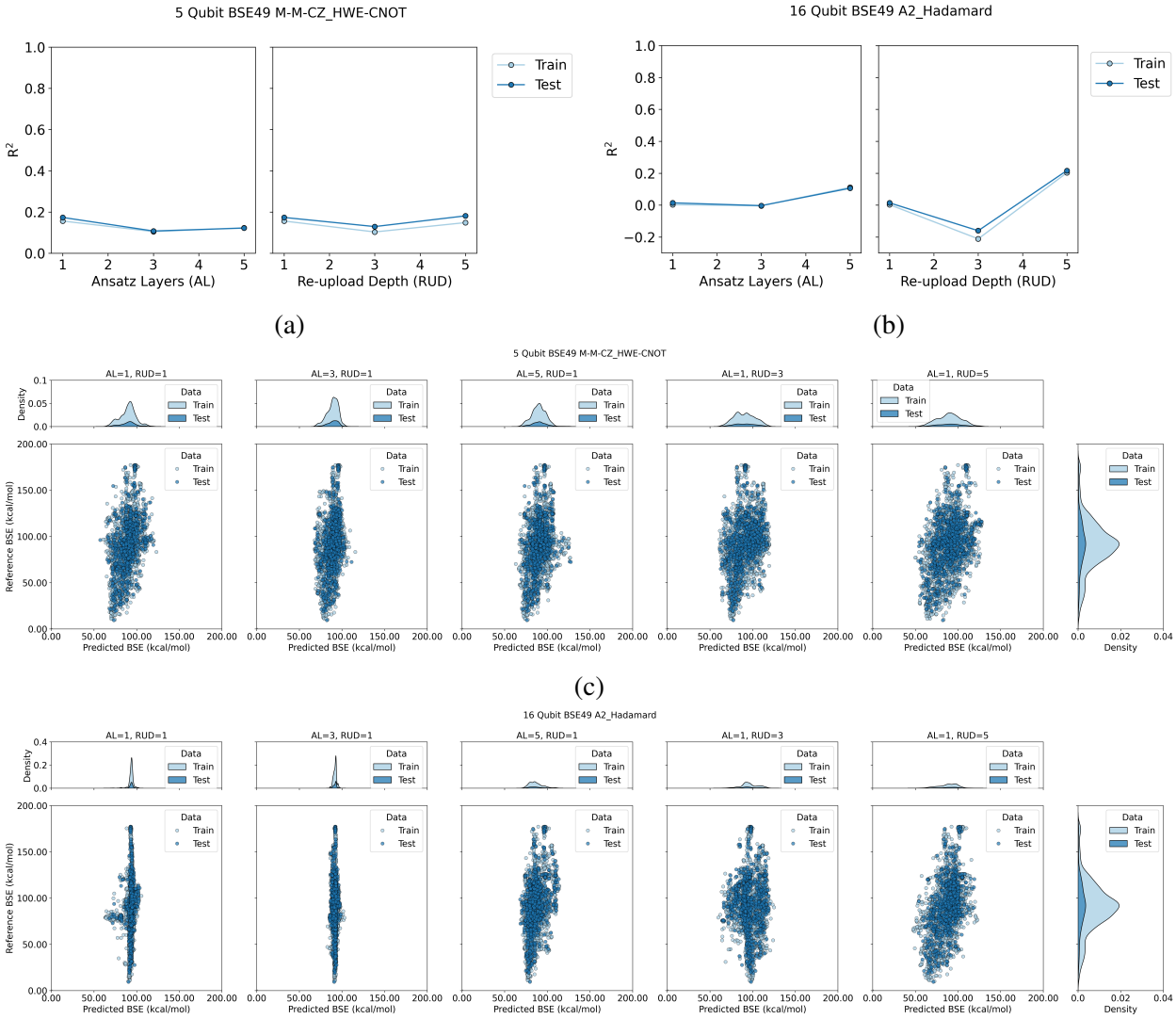
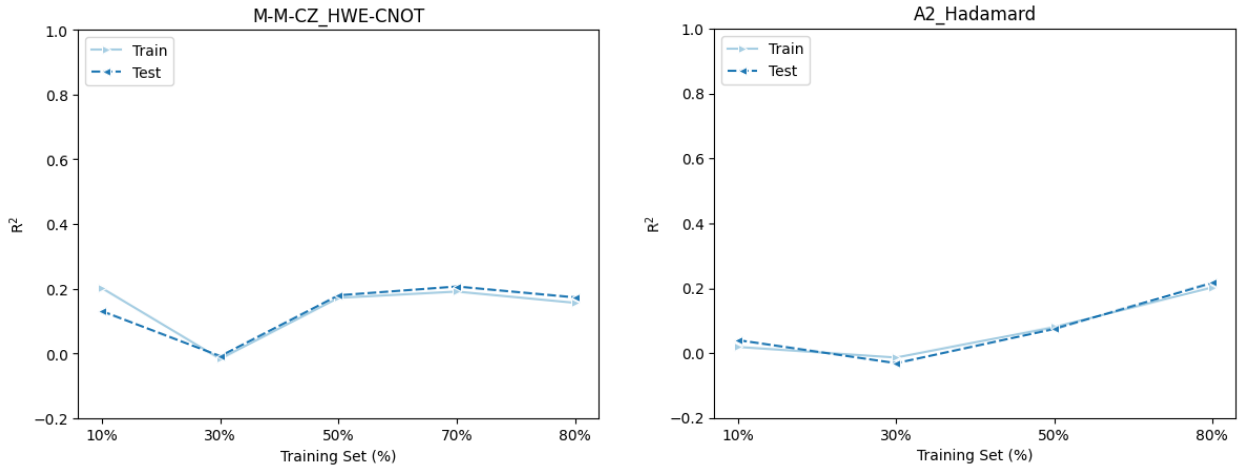

5 Qubit BSE49 M-M-CZ_HWE-CNOT
16 Qubit BSE49 A2_Hadamard
Train
Test
R^2
Ansatz Layers (AL)
Re-upload Depth (RUD)
(a)
(b)
5 Qubit BSE49 M-M-CZ_HWE-CNOT
16 Qubit BSE49 A2_Hadamard
AL=1, RUD=1
AL=3, RUD=1
AL=5, RUD=1
AL=1, RUD=3
AL=1, RUD=5
Data
Density
Reference BSE (kcal/mol)
Predicted BSE (kcal/mol)
(c)
(d)

Figure 13

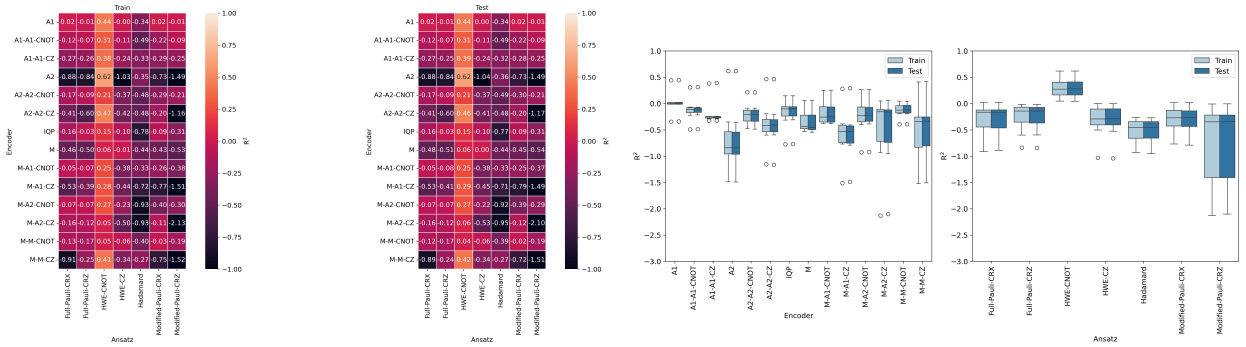


(a) (b)

Figure 14

DDCC

A2_HWE-CNOT Train R^2 0.62/test R^2 0.62 Best encoder average Train R^2 X/test R^2 Y Best ansatz HWE-CNOT average Train R^2 X/test R^2 Y



(a) (b)

Figure 15

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

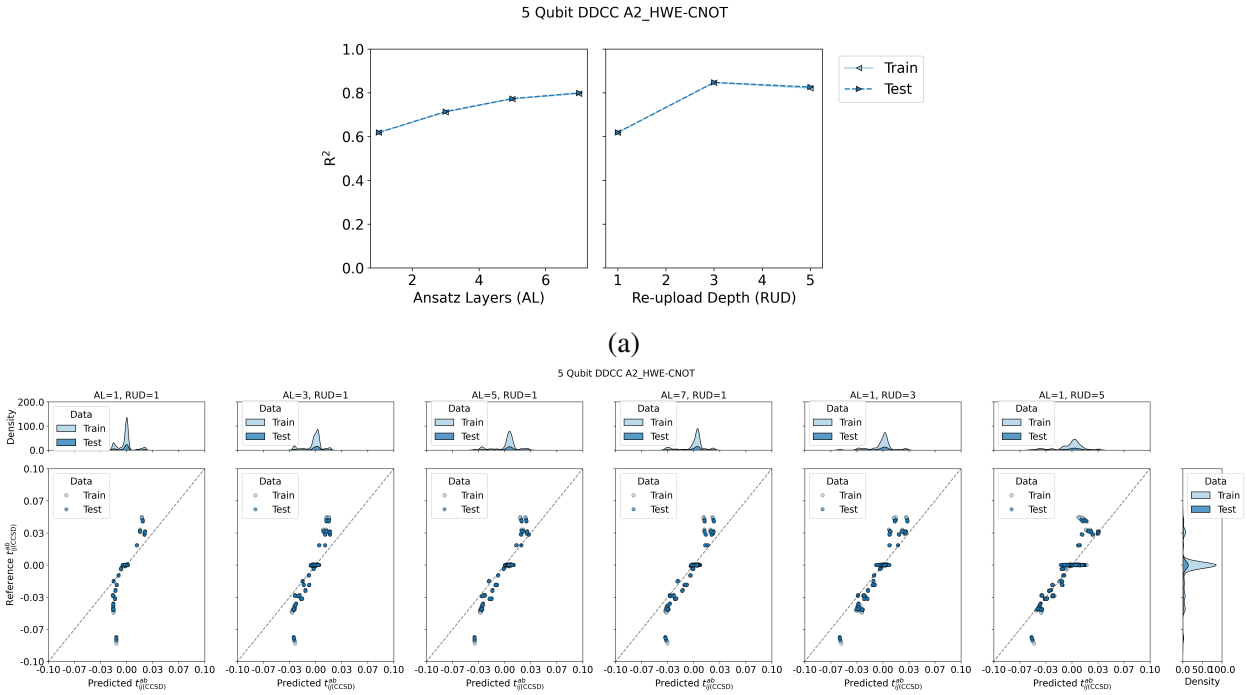


Figure 16: Model evaluation, using R^2 (y-axis), of re-upload depths (RUD) and ansatz layers (AL) of 1, 3, and 5 for the A2_HWE-CNOT using the DDCC dataset. The left side of the plot denotes the training set and the right side the test set.

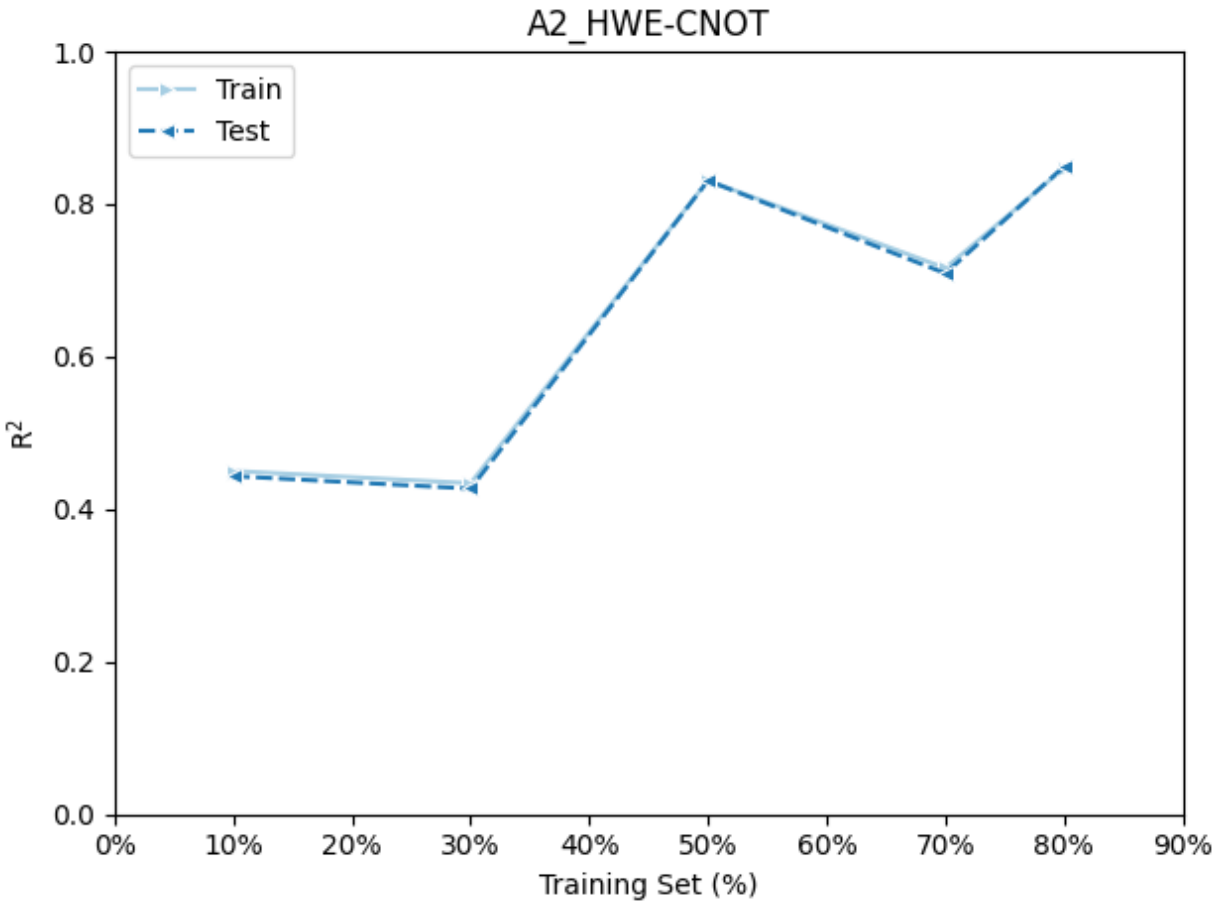


Figure 17

Ansaetze analysis[24] “In particular, a substantial improvement in performance of two-qubit gates in a ring or all-to-all connected arrangement, compared to thatof those on a line, is observed.”

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

“investigated howexpressibility “saturates” with increased circuit depth, finding that the rateand saturated value appear to be distinguishing features of a PQC”

4 Conclusion

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) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *SCIENTIFIC DATA* **2014**, *1*.

(2) Ramakrishnan, R.; Dral, P. O.; Rupp, M.; von Lilienfeld, O. A. Big Data Meets Quantum Chemistry Approximations: The Δ-Machine Learning Approach. *JOURNAL OF CHEMICAL THEORY AND COMPUTATION* **2015**, *11*, 2087–2096.

(3) 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.

(4) 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.

(5) Behler, J. Perspective: Machine learning potentials for atomistic simulations. *JOURNAL OF CHEMICAL PHYSICS* **2016**, *145*.

(6) Goh, G. B.; Hodas, N. O.; Vishnu, A. Deep learning for computational chemistry. *JOURNAL OF COMPUTATIONAL CHEMISTRY* **2017**, *38*, 1291–1307.

(7) Butler, K. T.; Davies, D. W.; Cartwright, H.; Isayev, O.; Walsh, A. Machine learning for molecular and materials science. *NATURE* **2018**, *559*, 547–555.

(8) Sanchez-Lengeling, B.; Aspuru-Guzik, A. Inverse molecular design using machine learning: Generative models for matter engineering. *SCIENCE* **2018**, *361*, 360–365.

(9) Janet, J. P.; Kulik, H. J. *Machine Learning in Chemistry*; ACS In Focus; American Chemical Society, 2020.

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

(11) Benedetti, M.; Lloyd, E.; Sack, S.; Fiorentini, M. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.* **2019**, *4*, 043001.

(12) Suzuki, T.; Katouda, M. Predicting toxicity by quantum machine learning. *J. Phys. Commun.* **2020**, *4*, 125012.

(13) 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.

(14) 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.

(15) Ranga, D.; Rana, A.; Prajapat, S.; Kumar, P.; Kumar, K.; Vasilakos, A. V. Quantum Machine Learning: Exploring the Role of Data Encoding Techniques, Challenges, and Future Directions. *Mathematics* **2024**, *12*, 3318, Number: 21 Publisher: Multidisciplinary Digital Publishing Institute.

(16) Alam, M.; Ghosh, S. QNet: A Scalable and Noise-Resilient Quantum Neural Network Architecture for Noisy Intermediate-Scale Quantum Computers. *Front. Phys.* **2022**, *9*, Publisher: Frontiers.

(17) 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.

(18) 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.

(19) 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.

(20) 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.

(21) 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.

(22) Mitarai, K.; Negoro, M.; Kitagawa, M.; Fujii, K. Quantum circuit learning. *Phys. Rev. A* **2018**, *98*, 032309.

(23) 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].

(24) 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.

(25) Krenn, M.; Landgraf, J.; Foesel, T.; Marquardt, F. Artificial intelligence and machine learning for quantum technologies. *Phys. Rev. A* **2023**, *107*, 010101.

(26) Takaki, Y.; Mitarai, K.; Negoro, M.; Fujii, K.; Kitagawa, M. Learning temporal data with a variational quantum recurrent neural network. *Phys. Rev. A* **2021**, *103*, 052414.

(27) 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].

(28) Bergholm, V. et al. PennyLane: Automatic differentiation of hybrid quantum-classical computations. 2022; `http://arxiv.org/abs/1811.04968`, arXiv:1811.04968 [quant-ph].

(29) 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.

(30) Suzuki, Y. et al. Qulacs: a fast and versatile quantum circuit simulator for research purpose. *Quantum* **2021**, *5*, 559, arXiv:2011.13524 [quant-ph].

(31) 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.

(32) Virtanen, P. et al. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods* **2020**, *17*, 261–272.

(33) Pedregosa, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.

(34) Jones, G. M.; Story, B.; Maroulas, V.; Vogiatzis, K. D. *Molecular Representations for Machine Learning*; ACS In Focus; American Chemical Society, 2023.

(35) RDKit. `https://www.rdkit.org/`.

(36) 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.

(37) 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.

(38) Rogers, D.; Hahn, M. Extended-connectivity fingerprints. *Journal of Chemical Information and Modeling* **2010**, *50*, 742–754, Publisher: American Chemical Society.

(39) 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.

(40) 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.

(41) 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.

(42) Tralie, C.; Saul, N.; Bar-On, R. Ripser.py: A Lean Persistent Homology Library for Python. *Journal of Open Source Software* **2018**, *3*, 925.

(43) 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*.

(44) 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.

(45) 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.

(46) Lundberg, S. M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. Advances in Neural Information Processing Systems. 2017.

(47) 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.

(48) 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.

(49) 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.

(50) LaRose, R. et al. Mitiq: A software package for error mitigation on noisy quantum computers. *Quantum* **2022**, *6*, 774, Publisher: Verein zur Forderung des Open Access Publizierens in den Quantenwissenschaften.