

Robust Quantum Reservoir Computing for Molecular Property Prediction

Daniel Beaulieu,^{1,*} Milan Kornjača,² Zoran Krunić,³ Michael Stivaktakis,⁴ Thomas Ehmer,⁵ Sheng-Tao Wang,² and Anh Pham¹

¹*Deloitte Consulting LLP*

²*QuEra Computing Inc., 1284 Soldiers Field Road, Boston, MA, 02135, USA*

³*Amgen, Thousand Oaks, CA 91320 USA*

⁴*Technical University of Darmstadt, Darmstadt, Germany*

⁵*Merck KGaA, Darmstadt, Germany*

(Dated: December 10, 2024)

Machine learning has been increasingly utilized in the field of biomedical research to accelerate the drug discovery process. In recent years, the emergence of quantum computing has been followed by extensive exploration of quantum machine learning algorithms. Quantum variational machine learning algorithms are currently the most prevalent but face issues with trainability due to vanishing gradients. An emerging alternative is the quantum reservoir computing (QRC) approach, in which the quantum algorithm does not require gradient evaluation on quantum hardware. Motivated by the potential advantages of the QRC method, we apply it to predict the biological activity of potential drug molecules based on molecular descriptors. We observe more robust QRC performance as the size of the dataset decreases, compared to standard classical models, a quality of potential interest for pharmaceutical datasets of limited size. In addition, we leverage the uniform manifold approximation and projection technique to analyze structural changes as classical features are transformed through quantum dynamics and find that quantum reservoir embeddings appear to be more interpretable in lower dimensions.

I. INTRODUCTION

Predicting drug activities is an important and difficult task in pharmaceutical discovery. This task has traditionally been performed using a labor-intensive process of repeated trials and errors within a laboratory setting [1, 2]. Recently, machine learning techniques have been employed to accelerate the discovery process [3–5]. Within a machine learning workflow, molecules can be mapped to features using different molecular descriptors, such as physiological, biochemical properties or molecular fingerprints, which can then be used to predict a target variable such as therapeutic efficacy [6–9]. However, an immediate challenge is how to extract the relevant molecular features to improve the overall performance of the model, since the structure-property relationship for molecules is complex [1, 3, 10]. Recently, the advent of quantum computing has given rise to a novel machine learning paradigm [11–14]. Various quantum machine learning (QML) versions of classical ML models such as generative, support vector machines (SVMs), and neural networks have been explored [15–18]. Quantum kernel methods, in particular, have shown potential advantage on synthetic datasets of quantum origin [19–23]. As a result, there are early attempts to apply QML algorithms for predicting the properties of molecules and pharmacological activity [24–26].

In this manuscript, we explore the performance of quantum reservoir computing (QRC) for molecular property prediction. Reservoir computing [27] addresses the potential drawback of widely employed variational

QML models. There, the need to estimate gradients on quantum hardware leads to fundamental trainability issues due to entanglement and noise-induced barren plateaus [28–30]. QRC has been proposed as an alternative approach, in which the gradient estimation can be fully off-loaded to classical post-processing [23, 31–34]. The QRC algorithm we employ in this study leverages entangled quantum dynamics of neutral atom arrays, a platform that has already shown promise for quantum simulation, optimization, and machine learning [23, 35–41].

We applied the QRC algorithm to the dataset within the Merck Molecular Activity challenge (MMACD) [42], with the task of predicting pharmacological activity based on molecular fingerprints [43]. We systematically report the performance of the classical and QRC models across different dataset sample sizes and feature choices generated by the quantum reservoir dynamics. We then compare classical features and embedded quantum reservoir features by employing the uniform manifold approximation and projection (UMAP) technique. Our results reveal a slower performance decay with decreasing training dataset size when QRC embeddings are used compared to classical features. The UMAP analysis reveals more structure for QRC-embedded data when projected to low dimensional spaces. Together, the results indicate that the QRC approach has the potential to build a more robust and interpretable model with less training data, which could be relevant for health-related applications.

* dabeaulieu@deloitte.com

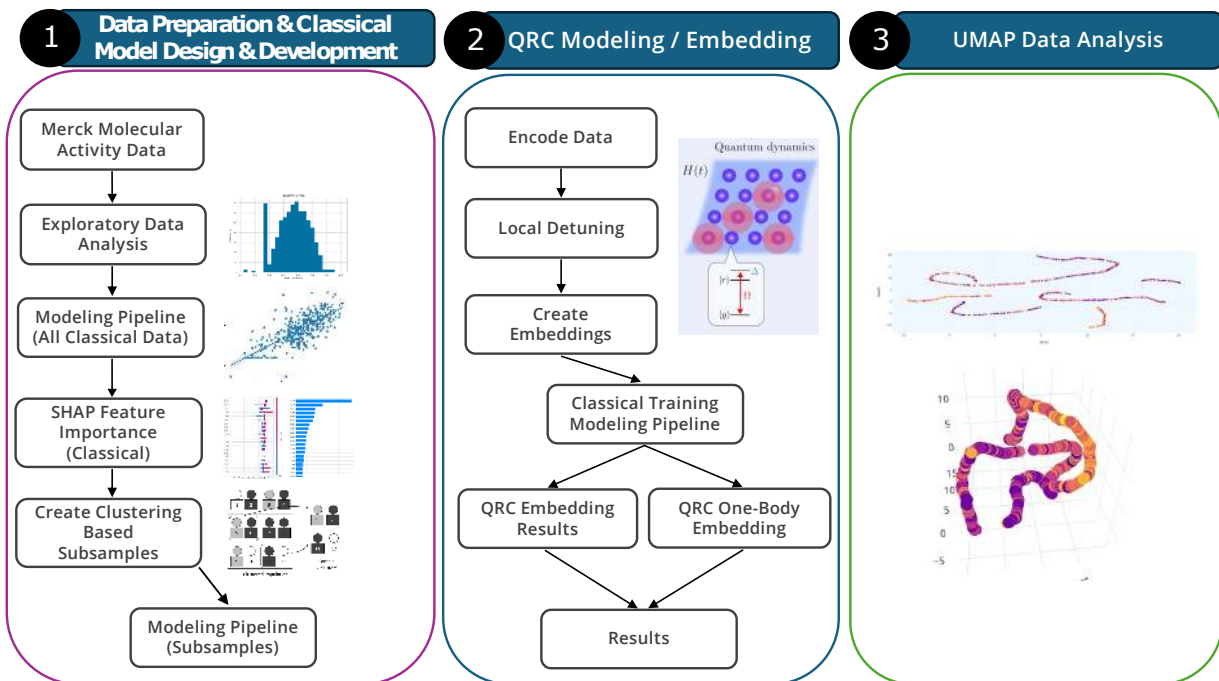


FIG. 1: **Machine learning modeling workflow and analysis of the different embeddings.** (1) A schematic workflow used to determine the optimal classical model, the feature selection process with the SHAP method to enable QRC simulation, and the subroutine to perform dataset subsampling. (2) An outline of the quantum reservoir computing algorithm. Both two-body observable embeddings and one-body observable embeddings are estimated and utilized to train the classical models. (3) UMAP analysis is used to interpret the difference between classical and quantum features.

II. METHODS

A. Molecular dataset

To explore how quantum reservoir computing can be utilized to make predictions of molecular activities, we applied the QRC technique to predict the activity (ACT) values based on molecular descriptors as presented in the MMACD [42]. The dataset contains 15 different biological activity prediction problems, each represented by a unique set of molecular descriptors. The MMACD has been used extensively in the machine learning community [44–46]. In this manuscript, we perform general modeling and the QRC pipeline to five of the smallest MMACD datasets, 4, 5, 9, 14, and 15, out of the entirety of the MMACD. For our UMAP projection and more detailed analysis, we focus our study on one of the 15 ACT datasets, MMACD 4. MMACD 4 was selected as the ideal candidate for probing the robustness of QRC due to having the smallest sample size. In addition, we chose MMACD 14 to perform our 25 subsample QRC modeling on analysis so as to reduce potential dataset bias for our 25 subsample results.

B. Research Modeling Workflow Overview

As shown in Fig. 1, the modeling process starts with data preparation and exploratory data analysis, as described in Appendix B 1. We then model all available data to select the best candidate model for the raw classical data, as described in Appendix B 2. The best model is used as a basis for SHAP feature importance selection to select the top 18 features. Clustering based sampling is used to create five subsamples of 100, 200, and 800 records. The modeling pipeline is applied to each such subsampled dataset for several subsample sizes. We feed the subsamples into the Julia QRC simulation pipeline [23] and create QRC embeddings. The QRC embeddings are fed into the classical modeling pipeline for all subsamples. Subsequent analysis of the embeddings is performed using the UMAP technique [47].

C. Machine Learning Pipeline

In this study, we implemented a machine learning pipeline to evaluate the performance of various regression algorithms on the MMACD, as shown in Fig. 1. The first step in our pipeline is the pre-processing step that involves inputting missing values using a data standardization procedure described in Appendix B 1. The next

step is to conduct an exploratory data analysis to determine salient features and transform data as necessary, also described in more detail in Appendix B 1. Subsequently, we binned the pre-processing steps and the specified regression modeling algorithms for fitting the models to the training data, which is described in Appendix A. The next step in the machine learning pipeline illustrated in Fig. 1 is the use of the SHAP method [48, 49] to choose the most important features and reduce the feature dimension to 18. We choose 18 data features to balance computational feasibility (of simulating the QRC dynamics) with maintaining the most relevant molecular information. We find that the difference in classical model performance with the top 18 predictors and all predictors is less than 1 percent in modeling MMACD 4 dataset. The biological relevance of our selected features is implicitly addressed through our use of established molecular descriptors from the MMACD dataset.

We then proceed to assign data to clusters of different sizes. These cluster values serve two purposes. First, they ensure a representative subsample of the ensemble from which the data is sampled. Second, they allow us to test the performance of QRC-embedded data versus raw features across various sample sizes. Optionally, we employ the QRC algorithm to create the QRC embeddings, which are used as input training data for our ensemble of classical models. Overall, the goal of this pipeline is to pass the data, either the raw features or QRC embedded features, to a series of regression models, and compare their predictive abilities. In this procedure, we ensure that the data is processed in the same way for a fair comparison at different subsample sizes (100, 200, and 800).

In our pipeline, we evaluated ten different regression algorithms, described in Appendix A: Random Forest Regressor, AdaBoost Regressor, Gradient Boosting Regressor, Bagging Regressor, SVR, Decision Tree Regressor, Extra Tree Regressor, Linear Regression, SGD Regressor, and KNeighbors Regressor as implemented in the Scikit-learn package [50]. For each algorithm, we compare their predictive ability using the mean squared error (MSE). We iterate over the list of algorithms, fitting each model to the training data, predicting on the test data, and computing the performance metrics. We then decide to report the results for two models – random forest regressor [51], as it consistently performed the best across sample sizes and embedding types, and Gaussian radial basis function (RBF) kernel [52] since it is a representative classical kernel algorithm.

D. Quantum Reservoir Computing

The QRC method was applied to the 18 classical features selected from our SHAP analysis, with the numerical simulations of neutral atom dynamics performed with the Bloqade simulator [53, 54]. The neutral-atom QRC protocol, described in [23], encodes the data features into

the dynamics of the Rydberg Hamiltonian that governs the analog dynamics of neutral atom arrays [39]:

$$H(t) = \frac{\Omega(t)}{2} \sum_j (|g_j\rangle \langle r_j| + |r_j\rangle \langle g_j|) + \sum_{j < k} V_{jk} n_j n_k - \sum_j [\Delta_g(t) + f_j \Delta_1(t)] n_j. \quad (1)$$

Here, $\Omega(t)$ represents the global Rabi drive amplitude between a ground ($|g_j\rangle$, j indexes atoms) and an excited state of an atom ($|r_j\rangle$), $n_j = |r_j\rangle \langle r_j|$, while $V_{jk} = C/\|\mathbf{r}_j - \mathbf{r}_k\|^6$ describes the interactions between atoms. The detuning is split into the global term, $\Delta_g(t)$ and the site-dependent term $\Delta_1(t)$, with site modulation $f_j \in [-1, 1]$.

The data features were normalized and encoded into the local detuning pattern, f_j , with constant $\Delta_1 = 6 \text{ rad}/\mu\text{s}$. We set $\Omega = 2\pi \text{ rad}/\mu\text{s}$, set the atoms in the linear chain with interatomic distance of $10 \mu\text{m}$, and probe the quantum dynamics starting from the all-ground state for $4.3 \mu\text{s}$, with a step-size of $0.4 \mu\text{s}$. The expectation values of the local observables in the computational basis are collected, including one-body $\langle Z_i \rangle$ and two-body $\langle Z_i Z_j \rangle$ observables. These expectation values form feature embeddings, which are then fed further into the classical model-based postprocessing pipeline.

E. Uniform Manifold Approximation and Projection (UMAP) 2D Projection

To try to understand the embeddings and have better interpretability, we utilized UMAP to project the high dimensional data to low dimensional spaces [47]. The UMAP algorithm is applied to the dataset features to generate 2D embeddings. These embeddings are used to visualize the molecular descriptors and, optionally, to facilitate the training of machine learning models. Using UMAP for dimensionality reduction offers an interpretation of the data’s structure in a low-dimensional space, enabling us to identify patterns that are not immediately apparent in high-dimensional space, as demonstrated in previous studies [55]. Specifically, we projected the data features to 2D with UMAP, to compare the QRC generated feature embeddings with raw features. UMAP projection was guided by several parameters: the number of neighbors was set to 200 to maximize global structure vs local structure in the data; the minimum distance between points in the low-dimensional space was set to 0.9 to control the tightness of UMAP’s embedding and maximize global UMAP structure; and the metric for distance calculation chosen is Minkowski, a generalization of both Euclidean distance and Manhattan distance. The same parameters were used to perform the UMAP projection for both the quantum and classical features.

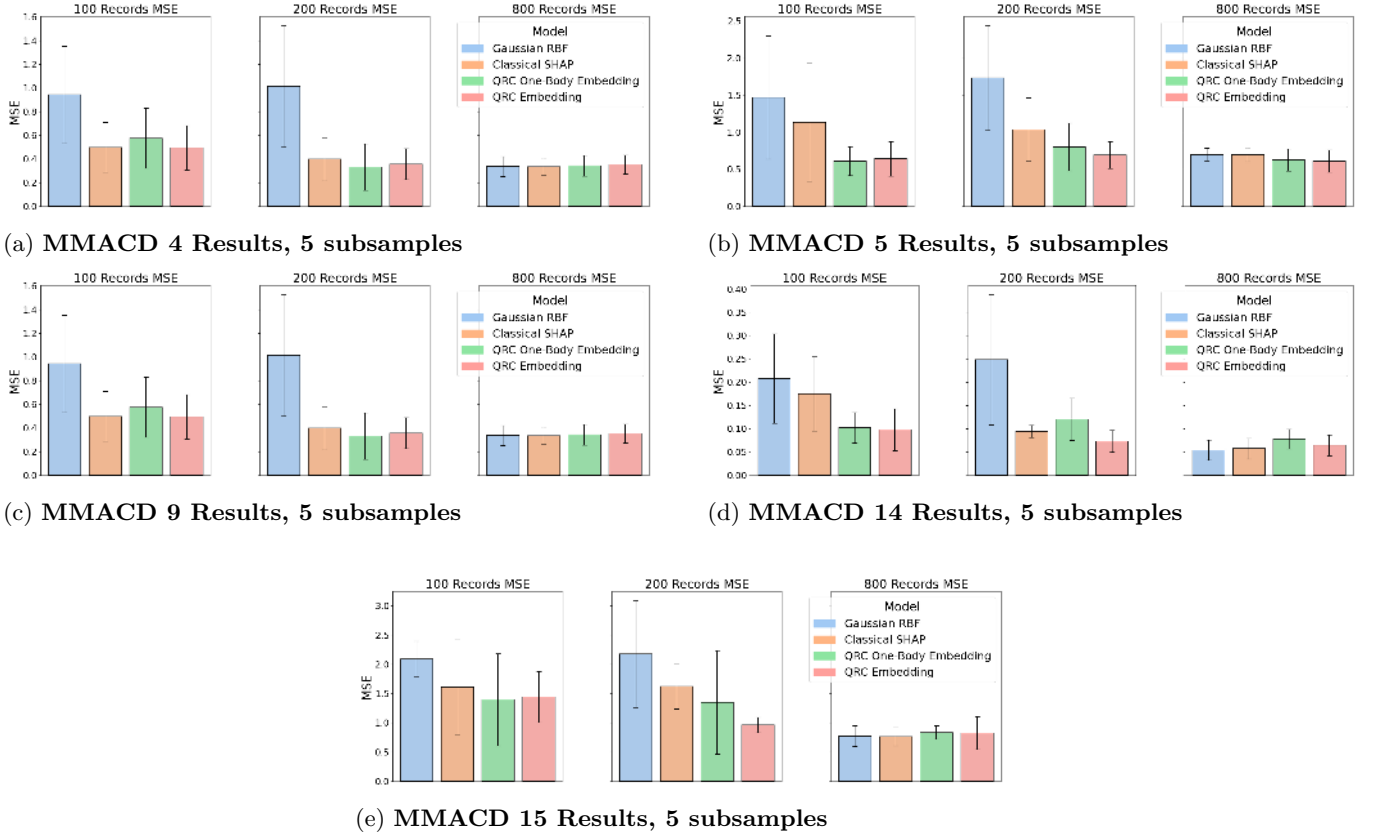


FIG. 2: The performance of the classical and quantum methods on the MMACD datasets. (a)-(e) Test mean square error (smaller is better) is reported for the five smallest datasets in MMACD and for two classical and two quantum feature methods. In all cases, we report averages across five dataset subsamples, with standard deviation over that sample reported as uncertainty estimate.

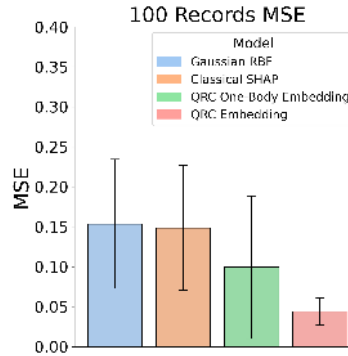


FIG. 3: The performance of the classical and quantum methods on MMACD 14 with 25 subsamples of 100 records. QRC, raw data, and Gaussian RBF results for the MMACD 14 data. Here, we report averages across twenty-five dataset subsamples, with standard deviation over that sample as uncertainty estimate.

III. RESULTS AND DISCUSSION

A. QRC and classical model performance

Based on the results of the initial pass of classical data through the modeling pipeline, the random forest regres-

sion was chosen as the classical model of choice since it outperformed other models in the workflow regardless of the training data. Our main results on the performance of classical and QRC-enhanced pipelines are summarized in Fig. 2 [56]. Taking Fig. 2(d) as a representa-

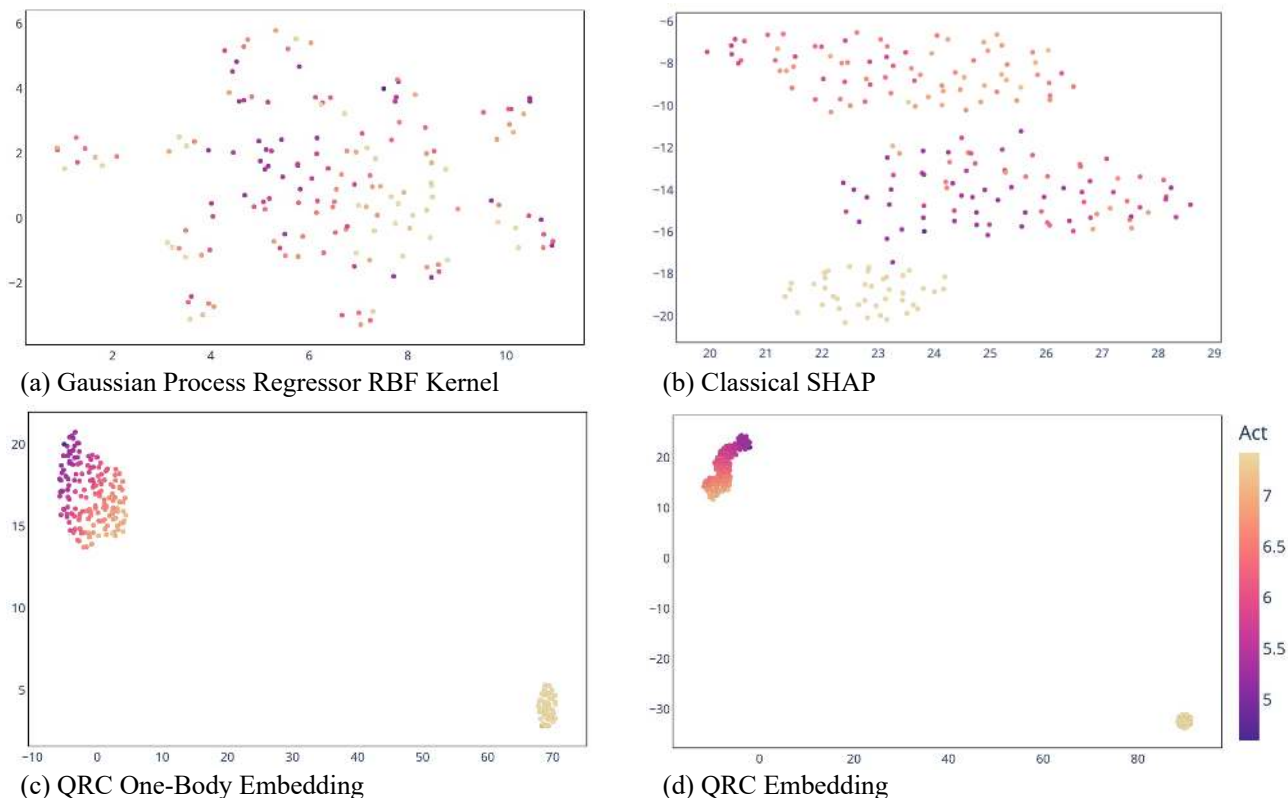


FIG. 4: UMAP 2D projections for 200 record samples from MMACD 4 dataset and sub-sample 4.

tive example, the QRC embeddings appear to result in an improvement upon the classical model performance for small training data sizes (e.g., 100 and 200 records). The average MSE is better for the QRC-enhanced models versus purely classical models across an average of five samples for 100 records. When compared to the classical Gaussian RBF kernel, we find that the QRC embedding encodes the data in a manner that outperforms the classical Gaussian RBF kernel in both the entire data sample and the five subsample averages for 100, 200, and 800 records. In addition, we also observe that QRC embeddings that include two-body observables outperformed one-body-observable embeddings. For larger subsamples consisting of 800 records, the classical input features yielded similar performance as the QRC embedded features across an average of 5 runs.

Furthermore, Fig. 2 also reveals a consistent trend where QRC embeddings perform equally or better than classical methods across the five different MMACD data sets. QRC two-body observable embeddings generally match or outperform QRC one-body observable embeddings across these five data sets as well. In contrast, the 800 data records results consistently show that QRC embedded data no longer has the advantage at this sample size. We note, however, that the QRC performance advantage size is generically comparable to the performance error estimates. This is, in large part, a direct consequence of the small sample size of the datasets consid-

ered. To check the robustness of our results against limited dataset subsampling, we also ran simulations with 25 subsamples of 100 records for MMACD 14, as shown in Fig. 3. While the performance estimates and the sizes of error bars for classical methods and the one-body embedding QRC remain similar to the 5 subsample case, the two-body QRC embedding performance estimate improves and becomes more certain. As a result, we observe QRC performance improvements over classical methods that are outside of the error estimates in spite of the small dataset size. Overall, the consistent QRC performance improvements that we observe across multiple molecular activity datasets at small sample sizes indicate that the feature engineering process utilizing QRC techniques could help improve the stability of the classical machine learning model, reducing the need for large training datasets. This could make QRC potentially valuable in application areas involving small datasets, which are common in chemistry and health applications where data collection is expensive and complex.

B. UMAP results

In this section, we present the findings from our UMAP analysis. UMAP was utilized to project high-dimensional data into a two-dimensional space, enabling the visual-

TABLE I: **MMACD activity classification performance on 2D UMAP embeddings.** 2D UMAP embedding data for MMACD dataset 4 is converted to binary classification task by high/low median cut. Performance of support vector machine (SVM) on 100, 200, and 800 record subsamples is reported with both QRC and classical embedding features. A five-subsample average is used to estimate the uncertainty as the standard deviation over the sample.

	Metric	100 records		200 records		800 records	
		Mean	Std. Dev.	Mean	Std. Dev.	Mean	Std. Dev.
Classical	Accuracy	0.660	± 0.095	0.737	± 0.041	0.709	± 0.027
	Precision	0.672	± 0.115	0.784	± 0.098	0.727	± 0.065
	Recall	0.653	± 0.073	0.680	± 0.143	0.689	± 0.077
	F1	0.660	± 0.084	0.716	± 0.061	0.702	± 0.019
QRC	Accuracy	0.747	± 0.077	0.760	± 0.042	0.742	± 0.033
	Precision	0.766	± 0.068	0.776	± 0.061	0.763	± 0.063
	Recall	0.707	± 0.101	0.747	± 0.139	0.713	± 0.062
	F1	0.734	± 0.085	0.752	± 0.062	0.734	± 0.028

ization and identification of clusters. This dimensionality reduction technique preserves the local and global structure of the data [47, 55], facilitating the interpretation of data patterns and relationships. We summarize representative results for the case of 200-sized subsampled datasets derived from MMACD 4 in Fig. 4. Although there is a clear concentration of high activity values for raw classical features, the correlation of data geometry and molecular activity is manifestly improved for QRC embeddings. The QRC one-body-observable embedding shows a similar pattern to the regular QRC embedding but with a tighter clustering of low ACT values. QRC data clustering in low dimensions also has manifest advantages when compared with the classical non-linear Gaussian kernel. Thus, we hypothesize that the improved QRC clustering in the low dimensional space is not purely an effect of the QRC kernel non-linearity but is an intrinsic feature of the QRC embeddings.

The distinct clustering patterns observed in the UMAP visualization (Fig. 4) after SHAP-based feature selection reveal fundamental differences in how classical and quantum methods process molecular information. The scattered distribution in the classical approach indicates a more continuous representation of molecular features, suggesting linear or gradual transitions between different molecular activities. In contrast, the emergence of two clear clusters in the QRC visualization suggests that the quantum processing has identified distinct underlying patterns in the molecular activity space, potentially leveraging quantum effects to better distinguish between different molecular behavior classes. This enhanced clustering capability of QRC on low dimensional UMAP projection suggests the potential to capture complex, nonlinear relationships between molecular features with a more interpretable and therefore also more robust model.

In order to quantify the relationship between the interpretability in UMAP and the model performance, we used the 2D UMAP manifold embeddings in a readily interpretable binary classification task. The classification task was derived from the original by splitting the target activity variable at the median into high and low-value

classes. The classical and QRC UMAP embeddings were input for a Support Vector Machine, whose performance we calculate from the five-run average for the 100, 200, and 800 record subsamples. As can be seen in Table I, the QRC UMAP embedding outperforms the classical embedding for all record sizes, thus quantifying the difference between UMAP embeddings in Fig. 4. However, we note again that the performance improvements are comparable to the error estimates, in large part due to the small sample sizes in the dataset. We conjecture that the improved interpretability of the QRC embeddings in low dimensions might lead to the better performance of the QRC method for small sample sizes observed throughout MMACD datasets previously. While the abundance of data can overcome suboptimal data clustering in a high-dimensional data representation, the interpretability of the data in terms of simpler low-dimensional structures might lead to performance gains when data is scarce.

IV. CONCLUSIONS AND OUTLOOK

In conclusion, our study presented a systematic exploration of QRC method on biomedical data exemplified by MMACD. We investigated the performance of both classical and QRC models at different sample sizes as well as the interpretability of the data at low feature dimensions using the UMAP technique. Our results suggest that QRC embeddings result in comparatively more robust, stable model performance for smaller datasets. The stability in QRC performance for small training data is further augmented by the manifestly more interpretable nature of QRC-derived features when projected by UMAP to low-dimensional feature space. Consequently, this study suggests a potential for QRC-enhanced models in the domain of biomedical data science for use cases that necessitate small training set sizes and readily interpretable, robust models.

We believe the results presented in this paper will motivate future studies across several directions. Firstly, the use of quantum hardware could allow for effective

QRC embedding generation for more classical input features than the cost of numerical simulations allows for – in fact, recent QRC experiments have been effective at 100 feature (qubit) scale [23]. Further increase in qubit counts and improved coherence times could enable QRC to handle larger datasets and more complex molecular systems, facilitating more extensive comparisons with classical methods. In fact, such extensions to more varied and feature-rich datasets could be a route to overcome the main caveat of our results and the large uncertainty estimates that result from modeling on small dataset sizes. Furthermore, a wide range of alternative feature selection techniques, such as genetic algorithms, could be relevant for selecting relevant molecular descriptors for the QRC pipeline and validating their biological relevance. Alternatively, a QRC pre-processing could

help in learning the correlations using Bayesian network learning methods [57, 58]. Finally, collaborative efforts with experimental laboratories will be crucial for empirically validating QRC’s computational predictions, bridging the gap between theoretical advantages and practical clinical applications [26].

ACKNOWLEDGEMENTS

We would like to thank Gopal Karemore, Scott Buchholz, Fangli Liu, Yuval Boger, and Chen Zhao for the fruitful discussion and support during this research project.

-
- [1] R. Liu, H. Wang, K. Glover, M. Feasel, and A. Walqvist, *Journal of Chemical Information and Modeling* **59** (2018), 10.1021/acs.jcim.8b00348.
 - [2] S. Q. Pantaleão, P. O. Fernandes, J. E. Gonçalves, V. G. Maltarollo, and K. M. Honório, *ChemMedChem* **17**, e202100542 (2022).
 - [3] J. Vamathevan, D. Clark, P. Czodrowski, I. Dunham, E. Ferran, G. Lee, B. Li, A. Madabhushi, P. Shah, M. Spitzer, and S. Zhao, *Nature Reviews Drug Discovery* **18**, 463 (2019).
 - [4] J. Deng, Z. Yang, H. Wang, I. Ojima, D. Samaras, and F. Wang, *Nature Communications* **14**, 6395 (2023).
 - [5] X. Zeng, H. Xiang, L. Yu, J. Wang, K. Li, R. Nussinov, and F. Cheng, *Nature Machine Intelligence* **4**, 1004 (2022).
 - [6] R. Team, “Rdkit: Open-source cheminformatics,” <http://www.rdkit.org> (), accessed: 2024-06-04.
 - [7] A. E. Comesana, T. T. Huntington, C. D. Scown, K. E. Niemeyer, and V. H. Rapp, *Fuel* **321**, 123836 (2022).
 - [8] A. Cereto-Massagué, M. J. Ojeda, C. Valls, M. Mulero, S. Garcia-Vallvé, and G. Pujadas, *Methods* **71**, 58 (2015), virtual Screening.
 - [9] C. Mellor, R. Marchese Robinson, R. Benigni, D. Ebbrell, S. Enoch, J. Firman, J. Madden, G. Pawar, C. Yang, and M. Cronin, *Regulatory Toxicology and Pharmacology* **101**, 121 (2019).
 - [10] T.-S. Vu, M.-Q. Ha, D.-N. Nguyen, V.-C. Nguyen, Y. Abe, T. Tran, H. Tran, H. Kino, T. Miyake, K. Tsuda, and H.-C. Dam, *npj Computational Materials* **9**, 215 (2023).
 - [11] M. C. Caro, H.-Y. Huang, M. Cerezo, K. Sharma, A. Sornborger, L. Cincio, and P. J. Coles, *Nature Communications* **13** (2022), 10.1038/s41467-022-32550-3.
 - [12] M. Schuld, I. Sinayskiy, and F. Petruccione, *Contemporary Physics* **56**, 172–185 (2014).
 - [13] G. Carleo, I. Cirac, K. Cranmer, L. Daudet, M. Schuld, N. Tishby, L. Vogt-Maranto, and L. Zdeborová, *Reviews of Modern Physics* **91** (2019), 10.1103/revmodphys.91.045002.
 - [14] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, *Nature* **549**, 195–202 (2017).
 - [15] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, “Quantum embeddings for machine learning,” (2020), [arXiv:2001.03622](https://arxiv.org/abs/2001.03622) [quant-ph].
 - [16] M. Schuld, “Supervised quantum machine learning models are kernel methods,” (2021), [arXiv:2101.11020](https://arxiv.org/abs/2101.11020) [quant-ph].
 - [17] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, *Physical Review Research* **1** (2019), 10.1103/physrevresearch.1.033063.
 - [18] P. Huembeli, G. Carleo, and A. Mezzacapo, “Entanglement forging with generative neural network models,” (2022), [arXiv:2205.00933](https://arxiv.org/abs/2205.00933) [quant-ph].
 - [19] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, *Nature* **567**, 209–212 (2019).
 - [20] M. Schuld and N. Killoran, *Physical Review Letters* **122** (2019), 10.1103/physrevlett.122.040504.
 - [21] Y. Liu, S. Arunachalam, and K. Temme, *Nature Physics* **17**, 1013–1017 (2021).
 - [22] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, *Nature Communications* **12** (2021), 10.1038/s41467-021-22539-9.
 - [23] M. Kornjača, H.-Y. Hu, C. Zhao, J. Wurtz, P. Weinberg, M. Hamdan, A. Zhdanov, S. H. Cantu, H. Zhou, R. A. Bravo, K. Bagnall, J. I. Basham, J. Campo, A. Choukri, R. DeAngelo, P. Frederick, D. Haines, J. Hammett, N. Hsu, M.-G. Hu, F. Huber, P. N. Jepsen, N. Jia, T. Karolyshyn, M. Kwon, J. Long, J. Lopatin, A. Lukin, T. Macrì, O. Marković, L. A. Martínez-Martínez, X. Meng, E. Ostroumov, D. Paquette, J. Robinson, P. S. Rodriguez, A. Singh, N. Sinha, H. Thoreen, N. Wan, D. Waxman-Lenz, T. Wong, K.-H. Wu, P. L. S. Lopes, Y. Boger, N. Gemelke, T. Kitagawa, A. Keesling, X. Gao, A. Bylinskii, S. F. Yelin, F. Liu, and S.-T. Wang, “Large-scale quantum reservoir learning with an analog quantum computer,” (2024), [arXiv:2407.02553](https://arxiv.org/abs/2407.02553) [quant-ph].
 - [24] K. Batra, K. M. Zorn, D. H. Foil, E. Minerali, V. O. Gawriljuk, T. R. Lane, and S. Ekins, *Journal of Chemical Information and Modeling* **61**, 2641 (2021).
 - [25] A. S. Bhatia, M. K. Saggi, and S. Kais, *Journal of Chemical Information and Modeling* **63**, 6476 (2023).

- [26] M. G. Vakili, C. Gorgulla, A. Nigam, D. Bezrukov, D. Varoli, A. Aliper, D. Polykovsky, K. M. P. Das, J. Snider, A. Lyakisheva, A. H. Mansob, Z. Yao, L. Bitar, E. Radchenko, X. Ding, J. Liu, F. Meng, F. Ren, Y. Cao, I. Stagliar, A. Aspuru-Guzik, and A. Zhavoronkov, “Quantum computing-enhanced algorithm unveils novel inhibitors for kras,” (2024), [arXiv:2402.08210 \[quant-ph\]](#).
- [27] H. Jaeger and H. Haas, *Science* **304**, 78 (2004).
- [28] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babush, and H. Neven, *Nature Communications* **9** (2018), [10.1038/s41467-018-07090-4](#).
- [29] C. O. Marrero, M. Kieferová, and N. Wiebe, “Entanglement induced barren plateaus,” (2021), [arXiv:2010.15968 \[quant-ph\]](#).
- [30] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo, “A review of barren plateaus in variational quantum computing,” (2024), [arXiv:2405.00781 \[quant-ph\]](#).
- [31] K. Fujii and K. Nakajima, *Physical Review Applied* **8** (2017), [10.1103/physrevapplied.8.024030](#).
- [32] R. Martínez-Peña, G. L. Giorgi, J. Nokkala, M. C. Soriano, and R. Zambrini, *Phys. Rev. Lett.* **127**, 100502 (2021).
- [33] R. A. Bravo, K. Najafi, X. Gao, and S. F. Yelin, *PRX Quantum* **3**, 030325 (2022).
- [34] A. Senanian, S. Prabhu, V. Kremenetski, S. Roy, Y. Cao, J. Kline, T. Onodera, L. G. Wright, X. Wu, V. Fatemi, and P. L. McMahon, “Microwave signal processing using an analog quantum reservoir computer,” (2023), [arXiv:2312.16166 \[quant-ph\]](#).
- [35] P. Scholl, M. Schuler, H. J. Williams, A. A. Eberharter, D. Barredo, K.-N. Schymik, V. Lienhard, L.-P. Henry, T. C. Lang, T. Lahaye, A. M. Läuchli, and A. Browaeys, *Nature* **595**, 233 (2021).
- [36] S. Ebadi, T. T. Wang, H. Levine, A. Keesling, G. Semeghini, A. Omran, D. Bluvstein, R. Samajdar, H. Pichler, V. W. Ho, S. Choi, S. Sachdev, M. Greiner, V. Vuletić, and M. D. Lukin, *Nature* **595**, 227 (2021).
- [37] G. Semeghini, H. Levine, A. Keesling, S. Ebadi, T. T. Wang, D. Bluvstein, R. Verresen, H. Pichler, M. Kalinowski, R. Samajdar, A. Omran, S. Sachdev, A. Vishwanath, M. Greiner, V. Vuletić, and M. D. Lukin, *Science* **374**, 1242 (2021).
- [38] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar, X.-Z. Luo, B. Nash, X. Gao, B. Barak, E. Farhi, S. Sachdev, N. Gemelke, L. Zhou, S. Choi, H. Pichler, S.-T. Wang, M. Greiner, V. Vuletić, and M. D. Lukin, *Science* **376**, 1209 (2022).
- [39] J. Wurtz, A. Bylinskii, B. Braverman, J. Amato-Grill, S. H. Cantu, F. Huber, A. Lukin, F. Liu, P. Weinberg, J. Long, S.-T. Wang, N. Gemelke, and A. Keesling, “Aquila: Quera’s 256-qubit neutral-atom quantum computer,” (2023), [arXiv:2306.11727 \[quant-ph\]](#).
- [40] J. Balewski, M. Kornjaca, K. Klymko, S. Darbha, M. R. Hirsbrunner, P. Lopes, F. Liu, and D. Camps, “Engineering quantum states with neutral atoms,” (2024), [arXiv:2404.04411 \[quant-ph\]](#).
- [41] D. Bluvstein, S. J. Evered, A. A. Geim, S. H. Li, H. Zhou, T. Manovitz, S. Ebadi, M. Cain, M. Kalinowski, D. Hangleiter, J. P. Bonilla Ataides, N. Maskara, I. Cong, X. Gao, P. Sales Rodriguez, T. Karolyshyn, G. Semeghini, M. J. Gullans, M. Greiner, V. Vuletić, and M. D. Lukin, *Nature* **626**, 58–65 (2023).
- [42] A. Kulkarni, “Merck molecular activity challenge,” <https://kaggle.com/competitions/MerckActivity> (2012), accessed: 2024-06-04.
- [43] The Competition was sponsored in 2012 by Merck & Co., Inc, located today at 26 E Lincoln Ave, Rahway, NJ 07065, US, USA. Merck KGaA and Merck & Co. are separate entities. Merck KGaA is headquartered in Germany, operates globally as EMD in North America, focusing on pharmaceuticals, life sciences, and performance materials, whereas Merck & Co., headquartered in the USA, operates as MSD outside North America, primarily focusing on pharmaceuticals, vaccines, and animal health.
- [44] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Kondering, and V. Pande, “Massively multitask networks for drug discovery,” (2015), [arXiv:1502.02072 \[stat.ML\]](#).
- [45] G. B. Goh, C. Siegel, A. Vishnu, N. O. Hodas, and N. Baker, “Chemception: A deep neural network with minimal chemistry knowledge matches the performance of expert-developed qsar/qspr models,” (2017), [arXiv:1706.06689 \[stat.ML\]](#).
- [46] Y. Hechtlinger, P. Chakravarti, and J. Qin, “A generalization of convolutional neural networks to graph-structured data,” (2017), [arXiv:1704.08165 \[stat.ML\]](#).
- [47] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” (2020), [arXiv:1802.03426 \[stat.ML\]](#).
- [48] S. M. Lundberg and S.-I. Lee, in *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17* (Curran Associates Inc., Red Hook, NY, USA, 2017) p. 4768–4777.
- [49] S. Team, “Introduction to shap,” (), accessed: 2024-09-13.
- [50] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, *Journal of Machine Learning Research* **12**, 2825 (2011).
- [51] S. L. Team, “Random forest regressor,” <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html> (), accessed: 2024-09-13.
- [52] S. L. Team, “Forecasting of co2 level on mona loa dataset using gaussian process regression (gpr),” https://scikit-learn.org/stable/auto_examples/gaussian_process/plot_gpr_co2.html (), accessed: 2024-09-13.
- [53] “Bloqade.jl: Package for the quantum computation and quantum simulation based on the neutral-atom architecture.” (2023).
- [54] “GitHub - QuEraComputing/QRC-tutorials: A set of tutorials for quantum reservoir learning. — github.com,” <https://github.com/QuEraComputing/QRC-tutorials> (2024), [Accessed 04-10-2024].
- [55] M. W. Dorrity, L. M. Saunders, C. Queitsch, S. Fields, and C. Trapnell, *Nature Communications* **11**, 1537 (2020).
- [56] 800 record samples for MMACD 5, 9, 14, and 15 features were chosen based on SHAP gradient method as compared to MMACD 4 which used SHAP kernel method to save on compute resources, as the change is verified to have minimal impact on the results.

- [57] K. Sachs, O. Perez, D. Pe’er, D. A. Lauffenburger, and G. P. Nolan, *Science (New York, N.Y.)* **308**, 523–529 (2005).
- [58] X. Gao, E. R. Anschuetz, S.-T. Wang, J. I. Cirac, and M. D. Lukin, *Physical Review X* **12** (2022), 10.1103/physrevx.12.021037.
- [59] S. L. Team, “Decision trees,” <https://scikit-learn.org/stable/modules/tree.html> (), accessed: 2024-09-13.
- [60] S. L. Team, “Bagging regressor,” <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingRegressor.html> (), accessed: 2024-09-13.
- [61] S. L. Team, “Gradient boosting regression,” https://scikit-learn.org/stable/auto_examples/ensemble/plot_gradient_boosting_regression.html (), accessed: 2024-09-13.
- [62] S. L. Team, “Decision tree regression with adaboos,” https://scikit-learn.org/stable/auto_examples/ensemble/plot_adaboost_regression.html (), accessed: 2024-09-13.
- [63] S. L. Team, “Linear regression,” https://scikit-learn.org/stable/modules/linear_model.html (), accessed: 2024-09-13.
- [64] S. L. Team, “Support vector regressor,” <https://scikit-learn.org/stable/modules/svm.html> (), accessed: 2024-09-13.
- [65] S. L. Team, “Knearest neighbor,” <https://scikit-learn.org/stable/modules/neighbors.html> (), accessed: 2024-09-13.
- [66] S. L. Team, “Extra tree regressor,” <https://scikit-learn.org/stable/modules/generated/sklearn.tree.ExtraTreeRegressor.html> (), accessed: 2024-09-13.
- [67] S. L. Team, “Sgd regressor,” https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDRegressor.html (), accessed: 2024-09-13.
- [68] J. Schmidhuber, *Neural Networks* **61**, 85–117 (2015).
- [69] P. Chalise, D. C. Koestler, M. Bimali, Q. Yu, and B. L. Fridley, *Transl Cancer Res* **1** (2014), 10.3978/j.issn.2218-676X.2014.06.03.

Appendix A: Classical machine learning models

Multiple regression models are trained on both classical and quantum-embedded training data. The performance of these models is evaluated using metrics such as the Mean Squared Error (MSE). The program compares the performance of classical raw features, Gaussian Process Regressor with RBF kernel embedded data, and data embedded with Quantum Reservoir Computing (QRC) to assess the potential performance benefits of quantum and classical data embedding methods.

Although in the final analysis we used only random forest regressor and Gaussian RBF kernel, we tested many other models to see if they performed better on our data. The following regression algorithms were included in this study:

1. Decision Tree Regressor: A non-parametric supervised learning method used for regression. This

model predicts the value of a target variable by learning simple decision rules inferred from the data features [59].

2. Bagging Regressor: An ensemble meta-estimator that fits base regressors each on random subsets of the original dataset and then aggregates their individual predictions to form a final prediction. This is typically used to reduce variance for algorithms like decision trees [60].
3. Gradient Boosting Regressor: An ensemble technique that builds models sequentially, each new model attempting to correct errors made by the previous ones. Models are built using the gradient descent algorithm to minimize the loss [61].
4. Random Forest Regressor: An ensemble of decision trees, typically trained via the bagging method. It is used for regression by averaging the predictions of each component tree [51].
5. AdaBoost Regressor: An adaptive boosting algorithm that adjusts the weights of incorrectly predicted instances so that subsequent classifiers focus more on difficult cases [62].
6. Gaussian Process Regressor (GPR) with a Radial Basis Function (RBF) kernel: Effective for capturing smooth, non-linear relationships in the data. The RBF kernel with an added level of noise allows the model to provide both predictions and uncertainty estimates [52].
7. Linear Regression: A linear approach to modeling the relationship between a dependent variable and one or more independent variables [63].
8. Support Vector Regression (SVR): An extension of the Support Vector Machine (SVM) that supports linear and non-linear regression [64].
9. K-Neighbors Regressor: A type of instance-based learning or non-generalizing learning that does not attempt to construct a general internal model but simply stores instances of the data [65].
10. Extra Tree Regressor: Implements a meta estimator that fits a number of randomized decision trees on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting [66].
11. SGD Regressor: A linear model fitted by minimizing a regularized empirical loss with SGD, the gradient of the loss is estimated each sample at a time, and the model is updated along the way with a decreasing learning rate [67].
12. Neural Network: We employed a four-layer neural network for regression to approximate the depth of the QRC regression model. The model sequentially

processes input data, extracting and consolidating features through the hidden layers, and employs a rectified linear unit activation function to predict continuous target values in the output layer [68].

Appendix B: Details of the modeling pipeline

1. Data Preparation Steps

Loading and Cleaning Data: The program begins by loading the dataset and performing necessary cleaning operations. This includes removing any unnamed columns that may have been introduced during data export or import processes.

Exploratory Data Analysis: The initial exploratory data analysis steps of the program involve preparing the dataset and selecting relevant features for the analysis. The dataset used in this study is derived from the MMACD challenge, which includes various molecular descriptors and activity labels. We examine the univariate statistics for individual columns in boxplots, scatterplots check for missing values, outlier values, as well as skew and kurtosis of the data distribution.

Data Standardization: Next the dataset undergoes standardization to ensure that all features are on a comparable scale, which is essential for the performance of many machine learning algorithms. The process begins by selecting a subset of the dataset, retaining only the columns deemed consistent and relevant for the analysis, while excluding any non-essential columns such as identifiers or target variables. This selected subset is then standardized so that each feature has a mean of zero and a standard deviation of one using scikit-learn’s standard-scalar function [50]. This normalization step is crucial for mitigating the effects of differing scales among features and ensuring that each feature contributes equally to the model’s performance. We then use scikit-learn’s train-test-split function to divide the data into training and test data sets of the desired size [50].

2. Initial Candidate Model

In the initial candidate model evaluation phase, several key metrics are initialized to systematically assess the performance of the machine learning models. The training dataset is based on raw classical features that are run against all models in the machine learning pipeline (See Fig. A). The model’s performance metrics are compared against one another, including MSE. We select the model with the best performance as our initial candidate model, which we then use as the basis for the SHAP feature selection based on feature importance.

3. SHAP Feature Selection

Our pipeline utilizes SHAP (SHapley Additive exPlanations) values to identify important features. The selected features are then used to create a subset of the original dataset, ensuring that only the most relevant descriptors are included in the subsequent analysis. As seen in Fig. 1, after our initial classical models were created, we applied SHAP to the data and determined the most important features for our champion model. The original MMACD dataset 4 has 4,308 columns which would correspond to 4,308 qubits when encoded with the QRC. Due to the limited computing resources, we were able to perform the QRC simulation for 18 SHAP features out of 4,308, corresponding to 18 qubits. The SHAP kernel method fits a specified model, in our case, the Random Forest regressor, which performed the best on the entire data set, to our data [48]. The SHAP kernel method explains the model’s predictions by computing SHAP values for the training data, which quantify the contribution of each feature to the model’s predictions [48]. Following the computation of SHAP values, we identify the most important features by averaging the absolute SHAP values and ordering them by relative importance. We then select the top features to retain for further analysis.

4. Clustering Subsample Algorithm

Next, in our research modeling and visualization workflow (See Fig. 1), we create clustering-based subsamples. Our sampling methodology helps create subsamples that can be efficiently processed on our quantum simulator while at the same time being representative of the entire dataset without the risk of the same record being in multiple subsamples [69]. To create valid subsamples of the MMACD dataset 4, we created a methodology for subsampling that clusters data in order to improve the representativeness of the subsample to the total population [69]. We remove the target variable, then we apply the KMeans clustering algorithm to the remaining numeric data, predicting cluster assignments for each data point. This clustering step helps in grouping similar data points together. Next, we calculate the sample size per cluster and sample data points from each cluster with no replacement so each record is included in only one subsample [69]. Our sampling methodology separates the features and target values from the sampled subset and splits this subset into training and testing sets, stratified by the cluster predictions. The optional modeling step is creating the QRC embeddings, as described in more detail in the section IID. After that, we return to the modeling pipeline to put the QRC embedded data, Gaussian Process Regressor with RBF kernel, and raw classical features as training and test data into the models as input data.

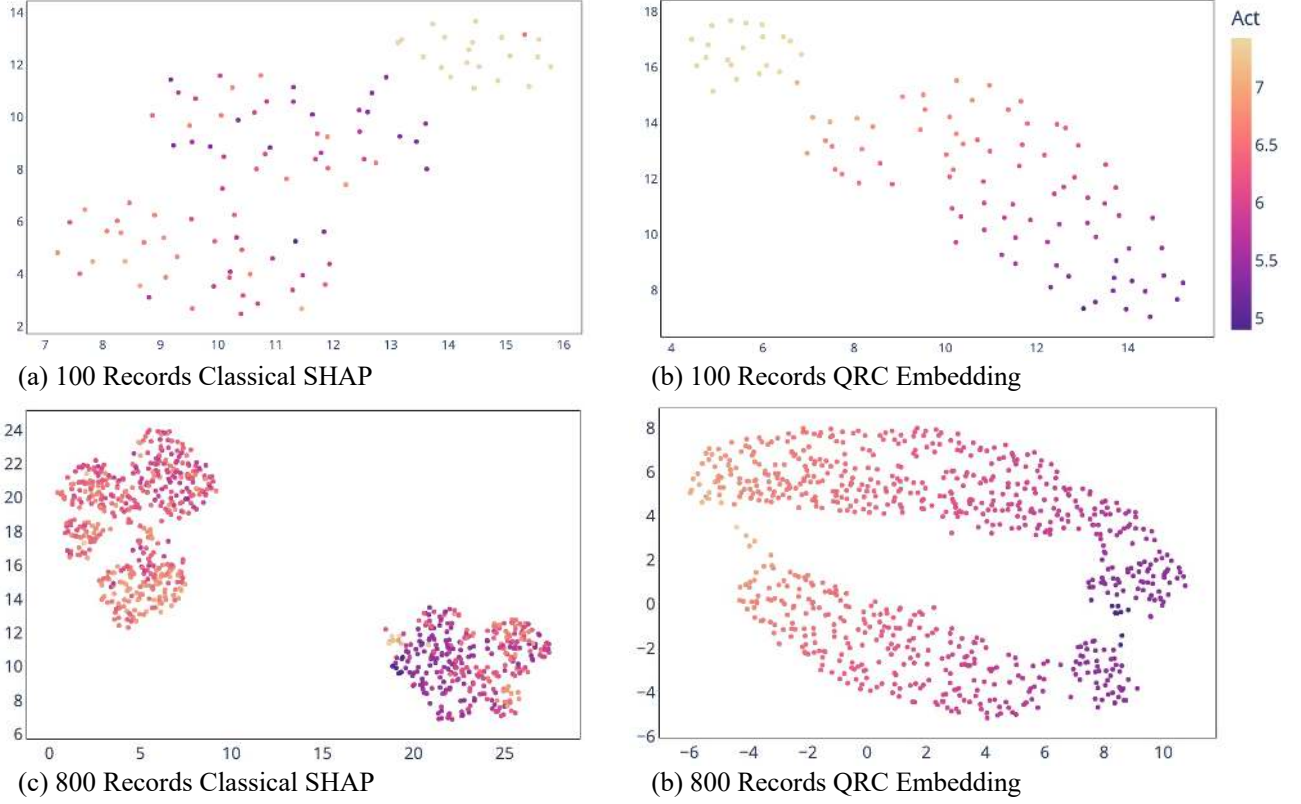


FIG. 5: UMAP 2D projections for 100 and 800 record samples of MMACD dataset 4.

Appendix C: Supplementary UMAP Results

The patterns in the 2D UMAP representation of the data, as presented in Fig. 5 largely mirror the patterns shown in the main text of the article for the 200 record sample, except that the pattern is less manifest in the 800 record samples. For 100 records, as seen in Fig. 4, there is an area of high ACT concentration for the classical data, while for the QRC two-body observable embedded data there is a clearer pattern in the data for all activity values. The same can be seen in the 800 record sample, with classical data having a tendency towards high values clustering to the lower values of the X-axis but otherwise, no other clear pattern. For the 800 record QRC two-body observable embeddings (Fig. 5), there is a clear pattern of the highest values being on the left and a decrease in ACT values towards the higher values of the X-axis, albeit with a more complex data cluster topology. The 100 and 800 record samples reinforce the observation of the clearer structure of the QRC embedded features versus the raw classical features.