Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Improved normalization in quantum-classical hybrid models for anti-cancer drug response prediction

Takafumi Ito
The University of Tokyo
Artem Lysenko
The University of Tokyo
Tatsuhiko Tsunoda
tsunoda@bs.s.u-tokyo.ac.jp

The University of Tokyo

Article

Keywords:

Posted Date: November 16th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11587693/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Improved normalization in quantum-classical hybrid models for anti-cancer drug response prediction

Takafumi Ito[1,*], **Artem Lysenko**[2,3], **and Tatsuhiko Tsunoda**[1,2,3,*]

[1]Laboratory for Medical Science Mathematics, Department of Computational Biology and Medical Sciences, Graduate School of Frontier Sciences, The University of Tokyo, Tokyo, Japan
[2]Laboratory for Medical Science Mathematics, Department of Biological Sciences, Graduate School of Science, The University of Tokyo, Tokyo, Japan
[3]Mathematical Drug Discovery Platform Project, Program for Drug Discovery and Medical Technology Platforms, RIKEN TRIP Headquarters, Yokohama, Japan
[*]Corresponding authors: ito-takafumi0906@g.ecc.u-tokyo.ac.jp, tsunoda@bs.s.u-tokyo.ac.jp

ABSTRACT

Quantum-classical Hybrid Machine Learning (QHML) models are AI systems that combine elements of classical algorithms, like neural networks, and quantum circuits to perform supervised inference tasks. They are recognized for their ability to perform comparably to classical models while reducing the number of trainable parameters. These qualities may offer unique advantages in a scenario of anti-cancer drug response prediction, where the number of available samples is typically small. However, we observed that QHML models appear to be very sensitive to the data encoding used at the interface between the neural networks and the quantum circuits, with suboptimal choices leading to stability issues. To address this problem, we propose a novel strategy that uses a hyperbolic tangent function with moderated gradient to normalize quantum circuit inputs. This method transforms the outputs of the neural networks without concentrating them at the extreme ends of the value range. The proposed normalization gave the best test AUC for three of the four tested anti-cancer drugs from the GDSC panel and we successfully confirmed that simulator-based performance for docetaxel response prediction (AUC$\simeq$0.87) can be reproduced on a 20-qubit trapped-ion quantum computer. The results confirmed that the introduction of quantum circuits can degrade performance unless the normalization is chosen appropriately. This study emphasizes the importance of designing normalization functions to fill the gap between quantum and classical parts in QHML.

Introduction

The extreme heterogeneity of cancers necessitates personalization of anti-cancer drug selection because none of the currently existing therapies are universally effective. Computational models are becoming increasingly essential for streamlining such choices because relevant biomedical data, such as gene expression levels, are often high-dimensional and difficult for experts to interpret directly[1]. Among possible approaches, deep learning-based methods have advantages over conventional machine learning, particularly in their superior feature extraction ability[2,3].

On the other hand, deep learning models face the "curse of dimensionality" as a major challenge. Gene expression data are an extremely valuable source of information for predicting the anti-cancer drug response[4], but they typically possess around 20,000 features, which usually far exceed the number of samples available for prediction (typically around 1,000 or fewer). The scarcity of these data points makes it challenging for deep learning models to correctly distinguish intrinsic relationships from noise. As a result, they may suffer from a loss of generalization abilities.

In this study, Quantum-classical Hybrid Machine Learning (QHML) refers to a class of models that combine trainable Parameterized Quantum Circuits (PQC) and conventional neural networks to leverage both the expressivity offered by quantum machine learning (QML) and the superior feature extraction inherent to deep learning models. Previous studies have confirmed that such QHML models can achieve performance comparable to, or even exceeding, classical models of similar parameter scale in specific cases[5,6]. QML models process input data without relying on activation functions, and instead the nonlinearity is realized via an encoding map itself, which determines the accessible frequencies of the underlying truncated-Fourier series. Consequently, they may be able to achieve superior generalization performance in scenarios where observation coverage is sparse in some parts of the training set[7]. However, at present, practical applications of quantum models are stymied by their inability to directly handle high-dimensional data due to current hardware constraints, such as small quantum register sizes and short decoherence time. This issue can be effectively mitigated by using a neural network that converts input data into sufficiently small and compact embeddings.

These characteristics of QHML have potential appeal for the challenge of predicting anti-cancer drug response. However,

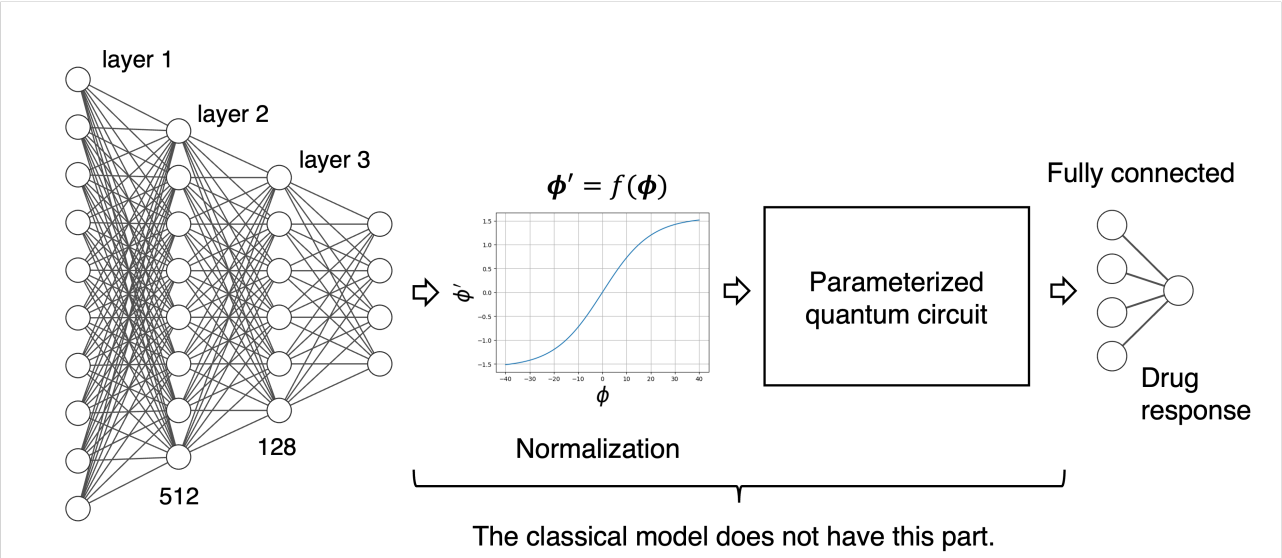


Figure 1. Model architecture overview. The model consists of a three-layer MLP neural network, a normalization function, a PQC, and a linear layer. It predicts the drug response as a scalar value. The classical model used as the baseline does not include the normalization function and the PQC, and its hyperparameter and parameter values including the number of output dimension of layer 3 were optimized independently of the QHML models. Since the majority of the parameters are in the MLP, the number of parameters in the QHML model is comparable to that in the Classical model.

in our experiments, the initial model was unable to adequately capture the relationship between gene expression levels and drug response during training, and the performance of the trained QHML model was unsatisfactory. Among the possible causes, we have examined the information flow between the deep learning component and the PQC as one of the primary factors. In the proposed architecture (Figure 1), the embeddings extracted by the neural network are passed to the quantum circuit by applying rotation angle encoding. Since the embeddings are used as rotation angles for quantum gates, their value ranges need to be controlled before passing them to the quantum circuit. Previous research has introduced the idea of using the tanh function to normalize the input of the quantum circuit[8]. However, due to the steep gradient near zero, many input values map to the most extreme upper and lower bounds of the scale. This saturation of the tanh function can cause the gradient to vanish, stalling the training at a suboptimal solution.[9]. This issue can be mitigated by introducing a "gradual tanh", which is an enhanced tanh function with reduced step steepness. It expands the non-saturating range and confines values within a specific range. To evaluate this approach, we have conducted response prediction for five different anti-cancer drugs and confirmed that appropriate normalization enhances the performance of the QHML model. Comparison of transformed value distributions with the original ones confirmed that it had avoided the value saturation caused by the conventional normalization. In summary, our work discovered several generalizable insights about how to optimally integrate deep learning models with quantum circuits, while also highlighting the potential of quantum machine learning for many biomedical prediction problems where data are usually extremely valuable and scarce.

Results

Model configuration and performance evaluation

We constructed the classical model and QHML models with different normalization techniques, including our "gradual tanh" model, and compared their predictive performance. Figure 1 shows the architectural overview of the models (detailed in Methods). The difference between the classical model and the QHML models is that the former lacks the normalization step and the quantum circuit. Note that its hyperparameter and parameter values were also optimized independently of the QHML models. The set of QHML models tested had the following four variations: one without normalization (Equation (1)), and one with normalization using nn.Layernorm from PyTorch (Equation (2). $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ are trainable affine transform parameters used in[10]), one with normalization using tanh (Equation (3)), and the last one with "gradual tanh" (Equation (4)), which is the version we propose.

$$\boldsymbol{\phi}' = \boldsymbol{\phi} \tag{1}$$

$$\boldsymbol{\phi}' = \frac{\boldsymbol{\phi} - E[\boldsymbol{\phi}]}{\sqrt{Var[\boldsymbol{\phi}] + 10^{-5}}} \cdot \boldsymbol{\gamma} + \boldsymbol{\beta} \tag{2}$$

$$\boldsymbol{\phi}' = \frac{\pi}{2} \tanh(\boldsymbol{\phi}). \tag{3}$$

$$\boldsymbol{\phi}' = r \cdot \tanh\left(\frac{\boldsymbol{\phi}}{a}\right). \tag{4}$$

The hyperparameters a and r of the proposed method were set at $a = 20$ and $r = \frac{\pi}{2}$ for the current implementation. Dividing the input by 20 expands the range where value saturation does not occur. As the results shown later indicate, for values of a greater than or equal to 10, the performance is stable to within run-to-run noise. One note is that the parameters of the neural network used before normalization should be initialized appropriately to prevent value saturation from beginning just after training starts (Methods). In this study, all quantum computations were performed on a noiseless simulator, with the exception of the section "Execution on a quantum device". The simulator is software that replicates the behavior of a quantum computer and can calculate quantum states exactly without being affected by noise.

In this study, we trained a regression model using cell line data from the Genomics of Drug Sensitivity in Cancer (GDSC) collection[11,12] to predict pre-processed IC50 values based on gene expression levels. For evaluating the model's performance, we calculated the AUC values with the binary response labels. The analysis focused on five anti-cancer drugs (cetuximab, cisplatin, docetaxel, erlotinib, and gemcitabine) with sufficient sample sizes (Methods). First, we split the entire data set into training-validation data (90%) and test data (10%). These test data were reserved for model evaluation and were not used for the hyperparameter tuning and subsequent training stages. Next, within the training-validation data, we performed a stratified 5-fold cross-validation (5-fold CV) 10 times to search for the optimal hyperparameters (Figure 2). Stratification was performed on normalized drug response values by dividing them into four equal-width bins using the scikit-learn "StratifiedKFold" function[13]. Optimal hyperparameter values were explored using grid search (Supplementary Table S1, S2, Supplementary Figure S1, S2, S3, S4, S5) and each combination of parameters was evaluated with 10 repetitions of a 5-fold CV (total of 50 runs). For each combination, we calculated the AUC for each epoch and averaged them over 50 runs to obtain the average CV-AUC curve across epochs. We selected the best set of hyperparameters according to the maximum value of this curve. We chose this approach to reduce the impact of random performance fluctuations in individual runs and to select robust hyperparameters that perform consistently well regardless of data allocation. After completion of the hyperparameter search, we evaluated the performance of the model trained with the best hyperparameters on the test data (Figure 2, below). Training-validation data were divided into training and validation split in an 8:2 ratio (corresponding to 72% and 18% of the total) using a stratified approach, as with hyperparameter tuning, to train and validate the representative model. We trained the models for up to 100 epochs using the optimum hyperparameters, calculating the models' AUC on the validation split at the end of each epoch. To avoid overfitting, the training was terminated when performance in the validation data set ceased to improve for three consecutive epochs. In each case, the model from the best epoch before the plateau was adopted as the final model.

We evaluated the models' performance using the cell line test data that were set aside. Specifically, we performed 10,000 resamples of the test data using bootstrapping and calculated the mean $\pm$ standard deviation (SD) of the AUCs (Table 1). The proposed method demonstrated the highest performance for cetuximab, docetaxel, and gemcitabine. For cisplatin, Layernorm performed best, but the proposed method still achieved the second highest performance. For erlotinib, the AUC of all methods was below 0.5, which is worse than random models. Therefore, we do not interpret the differences between the methods for this drug. However, the "tanh" method performed clearly worse than the classical model for cetuximab and cisplatin. These results indicate that the choice of normalization before the quantum circuit can affect the performance of the QHML model.

Effect of normalization function hyperparameters on prediction performance

Next, we have investigated the impact of the hyperparameters a and r on the performance of the method. For each drug, we examined how performance changed when different values of a and r were applied, and the best models were selected using CV as described in the performance evaluation section. Here, we also trained and evaluated the models using a stratified 5-fold CV with early stopping and compared their performance on the test data. When changing a, r was fixed at $\frac{\pi}{2}$; when changing r, a was fixed at 20. For a, when a was small (0.5, 1), performance was lower than the default (a=20) in all cases except for gemcitabine (Table 2). For gemcitabine, its performance was best at $a = 0.5$. Regarding r, since the optimal value varies for each anticancer drug, it is suggested that this parameter requires a dataset-specific adjustment (Table 2). $r = 4\pi$ and $r = 8\pi$ exceed π and can cause periodicity problems (Methods). In this regard, at $r = 8\pi$, the performance in drugs other than erlotinib was lower than the default value ($r = \frac{\pi}{2}$), while at $r = 4\pi$, the performance was comparable.

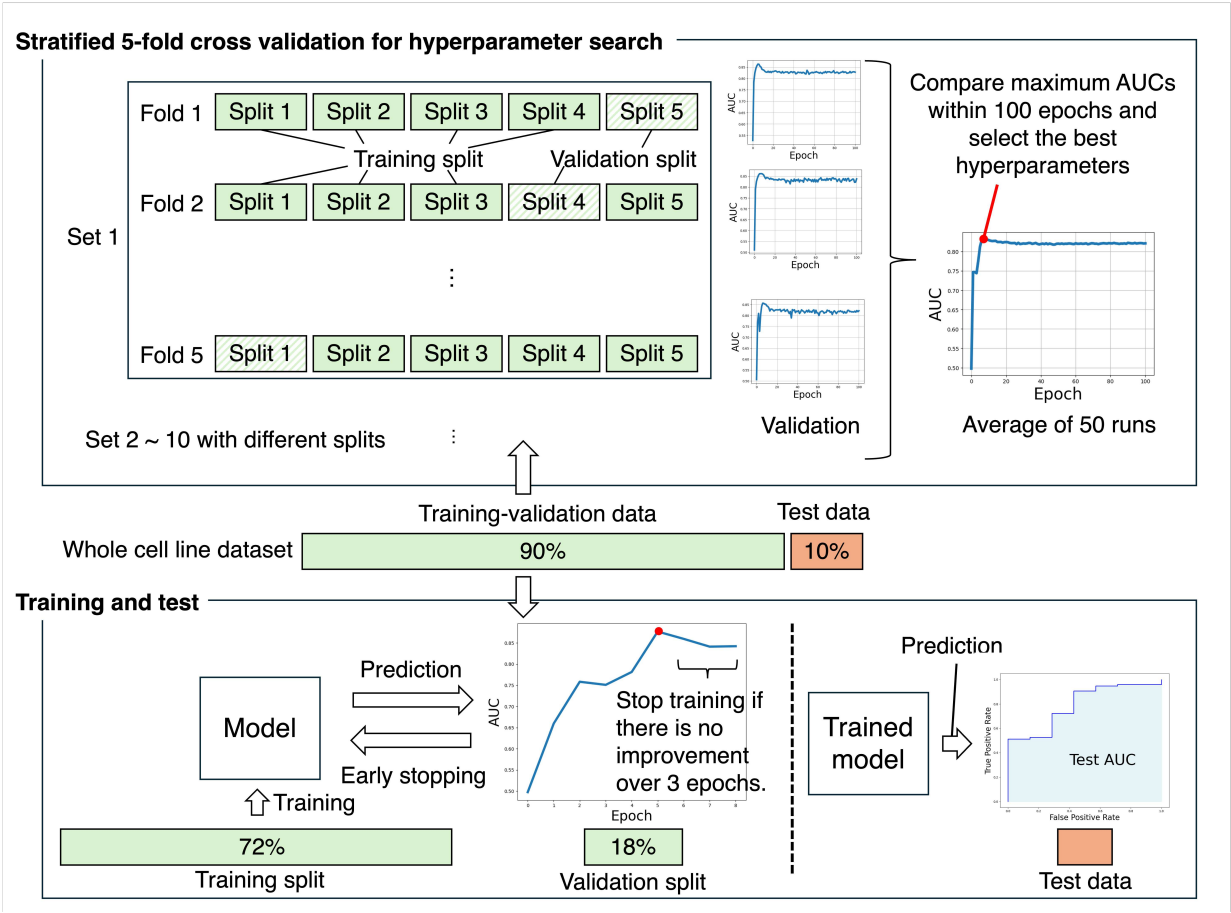


Figure 2. Performance evaluation design. First, we determined the optimal hyperparameters using stratified cross-validation. Then, we trained each model with these optimal hyperparameters using early stopping, and evaluated its performance on the independent test data.

Table 1. The values show the mean ± SD of the AUC from 10,000 bootstrap resamplings of the test set. Bold text indicates the highest value for each drug. "Classic" refers to the neural network model that replaces quantum circuits with a linear layer (optimized independently of hybrid models). "Hybrid" refers to the QHML model with different normalization layers. "Identity", "Layernorm", "tanh", and "Gradual tanh" correspond to Equations (1), (2), (3), and (4), respectively. All values for erlotinib are less than 0.5 and therefore are not considered.

Type	Normalization	Cetuximab	Cisplatin	Docetaxel	Erlotinib	Gemcitabine
Classic		0.707 ± 0.080	0.821 ± 0.070	0.790 ± 0.084	0.297 ± 0.120	0.482 ± 0.175
Hybrid	Identity	0.673 ± 0.081	0.782 ± 0.060	0.806 ± 0.098	0.383 ± 0.123	0.632 ± 0.134
	Layernorm	0.618 ± 0.074	$\mathbf{0.836 \pm 0.060}$	0.841 ± 0.064	0.426 ± 0.154	0.579 ± 0.125
	tanh	0.432 ± 0.078	0.583 ± 0.110	0.794 ± 0.114	0.427 ± 0.158	0.138 ± 0.067
	Gradual tanh (proposed)	$\mathbf{0.709 \pm 0.078}$	0.828 ± 0.068	$\mathbf{0.867 \pm 0.070}$	0.444 ± 0.129	$\mathbf{0.677 \pm 0.148}$

Table 2. Performance with different a or r values. When investigating a, r was fixed at $\frac{\pi}{2}$, and when investigating r, a was fixed at 20. Bold text indicates the highest value for each drug.

	Cetuximab	Cisplatin	Docetaxel	Erlotinib	Gemcitabine
$a = 0.5$	0.577	0.495	0.539	0.371	**0.790**
$a = 1$	0.432	0.580	0.795	0.429	0.139
$a = 10$	0.700	0.716	**0.885**	0.514	0.620
$a = 20$	0.709	**0.827**	0.868	0.443	0.679
$a = 100$	**0.734**	0.727	0.872	0.448	0.614
$r = \frac{\pi}{4}$	0.716	0.782	0.883	0.386	0.660
$r = \frac{\pi}{2}$	0.709	**0.827**	0.868	0.443	0.679
$r = \frac{3\pi}{4}$	0.668	0.730	0.883	0.476	0.645
$r = \pi$	0.667	0.722	**0.885**	0.324	0.454
$r = \frac{3\pi}{2}$	**0.742**	0.766	0.857	0.429	0.596
$r = 2\pi$	0.711	0.749	0.816	0.352	0.620
$r = 4\pi$	0.675	0.802	0.855	0.414	**0.685**
$r = 8\pi$	0.601	0.584	0.812	0.538	0.404

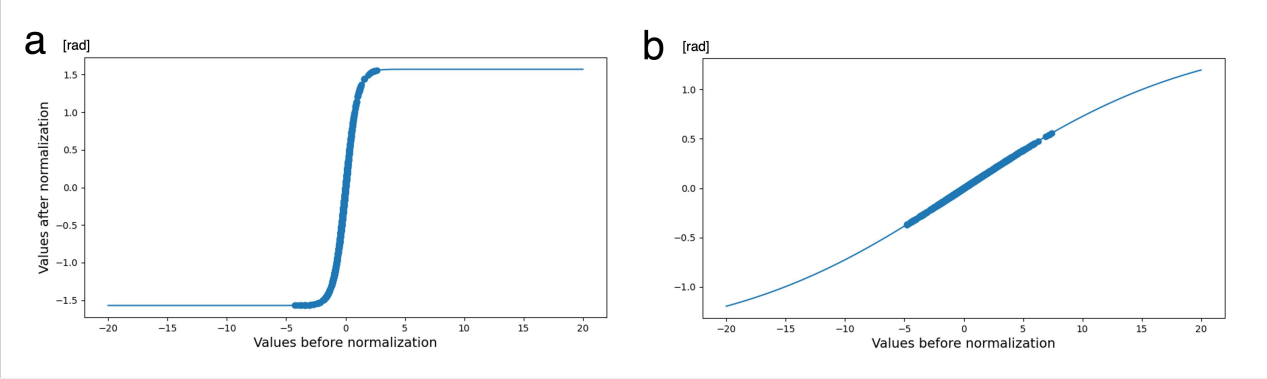


Figure 3. Distributions of embedding values before and after normalization. (**a**) Conventional method described in Equation (3). (**b**) "Gradual tanh" (our proposal) described in Equation (4). ($a = 20$ and $r = \frac{\pi}{2}$).

Distribution of the values transmitted from the neural network to the quantum circuit

To understand how the "gradual tanh" normalizes the data, we examined the overall distribution of the transformed values. For all subsequent analyses, we chose the trained response prediction model for docetaxel with the optimum hyperparameters, due to its superior performance. We examined the values returned by the embedding layer, after their transformation using the conventional tanh method (Equation (3)), and "gradual tanh" (Equation (4)) with $a = 20$ and $r = \frac{\pi}{2}$. We extracted 100 training samples, passed them through these trained models and plotted the distributions before and after the normalizations were applied (Figure 3). With the conventional method, many of the converted values were placed at the limits of the function's range $-\frac{\pi}{2}$ or $\frac{\pi}{2}$, resulting in value saturation. In contrast, the proposed method normalizes the values without this kind of saturation.

Integrated gradient analysis for biological interpretation of the model

To evaluate the biological validity of the model, we have used integrated gradients (IG) analysis[14] to identify key input genes that substantially influenced the prediction. IG analysis is a method to explain the attribution of each input element to the output expressed as a single accumulated gradient value for each feature (Methods). We calculated the mean absolute values of the IG scores for the train set and selected the top 5% of genes (706 genes) exhibiting the highest values as key genes for docetaxel. We performed enrichment analysis using Metascape[15], incorporating annotations from GO processes[16], KEGG pathways[17], Reactome gene sets[18], canonical pathways[19] and CORUM complexes[20] (Methods). The terms most abundantly enriched were "cell migration" and "negative regulation of the apoptotic process". Cell migration is promoted during the epithelial-to-mesenchymal transition, which has been reported to be associated with resistance to decetaxel[21]. In addition, since docetaxel induces apoptosis, the enrichment of a "negative regulation of the apoptotic process" is consistent with resistance to docetaxel. Related to this, "Extracellular matrix organization" is enriched as well. It is known that adhesion to the extracellular

matrix via integrins inhibits taxanes-induced apoptosis, and the enrichment of these terms likely reflects a cell state resistant to apoptosis[22]. IG analysis identified gene sets in the model's predictions that align with known biological mechanisms of docetaxel resistance, suggesting that this model has both predictive ability and biological validity. This indicates that the model is not only a predictive tool, but also has the potential to contribute to a deeper understanding of drug sensitivity and resistance mechanisms. Furthermore, these results demonstrate that IG analysis is an effective framework for explaining systems incorporating quantum circuits, at least on the simulator.

Execution on a quantum device

We executed the QHML model for docetaxel, which was trained using the simulator, on both an emulator and a real quantum device. For this, we have used Quantinuum Reimei H1 and its emulator. Quantinuum Reimei H1 is a 20-qubit trapped-ion quantum computer that commenced operation at the RIKEN Wako Campus in February 2025, notable for its low error rates for 2-qubit gates (infidelity = 1.3×10^{-3}) and full connectivity. The emulator is a simulator that replicates the characteristics of a real quantum device, such as its noise profiles and topology. By comparing results from the simulator, emulator, and real device, we can separately evaluate the impact of modeled noise and unpredictable physical noise on performance. We ran the trained model with "gradual tanh" on the emulator and the real device to make predictions on the test set. Using the "start_compile_job" function of Quantinuum Nexus[23], we compiled the quantum circuit, setting the optimization level to 2. The number of shots was set to 100 and 1,000 for execution with the emulator and the real device. The quantum part of this model used 4 qubits with 4 encoding layers and 4 variational layers. The circuit depth was 44. Each of the 83 test samples was encoded in the same circuit structure and the resulting 83 circuits were executed. The simulator model achieved an AUC of 0.868. The emulator model demonstrated an AUC of 0.829 with 100 shots and 0.878 with 1,000 shots. The real device model achieved an AUC of 0.628 with 100 shots and 0.867 with 1,000 shots. The performance of our model using the emulator and the real device was almost the same as that using the simulator when the number of shots was sufficiently large (Supplementary Figure S7). This implies that current quantum computers possess sufficient error tolerance to execute quantum circuits that constitute the QHML models for drug response prediction.

Nonlinearity of the quantum circuit

To understand the role of quantum circuits in the docetaxel response prediction model with "gradual tanh", we conducted the following detailed analysis using our trained model. Specifically, we visualized the decision landscape in the embedding space used as input to the quantum circuit. First, we applied principal component analysis (PCA) to the embedding vectors taken just before the quantum circuit. The first two components explained 94.2% of the variance (Methods). In the obtained 2D principal component space, we generated a grid of pseudo-data points covering the region around the test data distribution and mapped these back to the original embedding space using the inverse mapping of PCA. Then, we passed these remapped data points through the quantum circuit of the trained model and plotted the predicted values as a contour plot to visualize the model's response to input changes. The result of the analysis is shown in Figure 4. In areas where test data are densely concentrated, the prediction model showed almost linear behavior. In contrast, in areas where the data distribution is sparse, notable nonlinearity was observed.

Discussion

In our application case of prediction of anti-cancer drug responses using cell line data, we confirmed that the integrated models made up of quantum circuits and neural networks demonstrated the highest AUC for three of the four drugs (cetuximab, docetaxel, and gemcitabine; Erlotinib was excluded because the AUC for all models was less than 0.5.). For cisplatin, Layernorm performed best, while our proposed method was second-best. However, these differences compared to the classical model were unclear within the bootstrap uncertainty. A more certain finding is that the integration of quantum circuits can greatly degrade performance when using the conventional tanh function. Since unbounded functions such as Identity and Layernorm were also competitive, it is reasonable to interpret that gradient vanishing due to value saturation during training is the primary cause of the performance reduction. At $a = 20$, the proposed function is nearly linear within the observed range. It does not rely on the curvature of the tanh function but acts as a bounded rescaling mechanism to avoid saturation.

Another notable contribution of this study lies in verifying the biological validity of the QHML predictions using the IG analysis. The IG method was developed for deep learning and, therefore, requires the model to be differentiable. Since in this case quantum computations were performed using the simulator that supports gradient calculation, this method could be applied to our QHML model. The results indicate that drug resistance-associated genes identified in previous studies substantially contributed to the predictions of the model, implying that the model is biologically plausible. In addition, the IG analysis demonstrated that our QHML model is not merely a black box but a tool with a degree of interpretability. In the fields of drug discovery and precision medicine, model interpretability is particularly important. This is expected to not only enhance end-user confidence in the model, but also reveal new mechanisms of drug sensitivity and resistance. Furthermore, our analysis

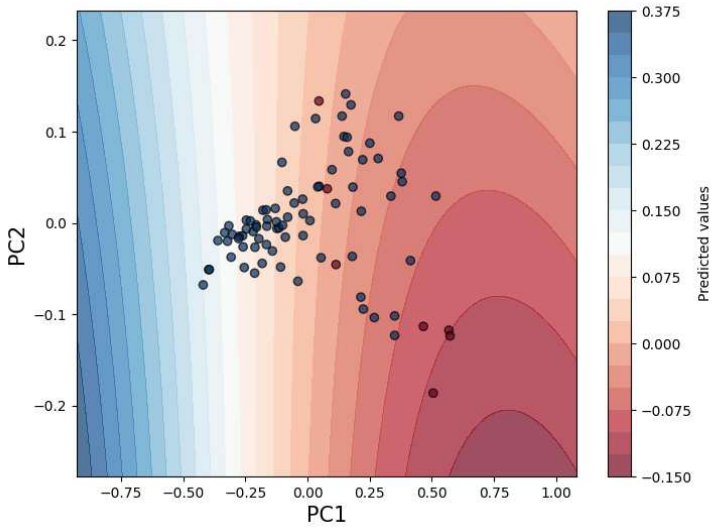


Figure 4. The decision landscape in the embedding space preceding the quantum circuit. The x-axis represents PC1, and the y-axis represents PC2. The contour plot represents predicted values at the points, and the dots in the figure represent test data points; blue dots indicate non-responders, and red dots indicate responders.

of the decision landscape in the latent space revealed additional insights about the prediction mechanism of the QHML model. We confirmed that while the model predicts almost linearly in regions where data is densely distributed, it shows nonlinear behavior in regions where data are sparse. This suggests that the QHML model maintains stable and robust prediction within regions with high data density while utilizing the nonlinearity of quantum circuits to handle complex regions where data are scarce. This characteristic may be especially beneficial for capturing the complex interactions between biomarkers, which are difficult to model using classical approaches.

Importantly, we have also been able to show that when a sufficiently large number of shots were used for measurement, the proposed QHML model performed well on the real quantum computing device. This result demonstrates the feasibility of implementing QML models even with the current generation of quantum devices. Quantum computers are expected to improve in scale and error rates in the near future, therefore proven ability for the predictive models to run on real devices is critical for informing the future advances in the field of QML.

More detailed analysis revealed that our proposed method is sensitive to the selection of relevant hyperparameters, and sufficient care must be taken to select appropriate value ranges. We found that a small a (0.5, 1) and a large r (8π) were unsuitable for the proposed "gradual tanh" normalization function across the data set in our current setting. The former is likely due to value saturation. For the latter, it is unlikely that periodicity is the only reason. This is because the performance with $r = 4\pi$ was comparable to that of the default value. In particular, gemcitabine was an exception, which performed best at $r = 4\pi$. Therefore, we examined the range of normalized values for gemcitabine and found that at $r = 4\pi$, the range was -3.37 to 1.30, and at $r = 8\pi$, it was -3.36 to 1.72. This means that the range for this drug was less than 2π, and different values were not mapped to the same angle. Since the normalization function scales the angle input to the circuit depending on the values of a and r, a better explanation is that this scaling changes the scale of the gradient update, although verifying this would require additional experiments. These results suggest that the optimal values for a and r may differ, and these parameters must be carefully determined. It is likely to be preferable to treat a and r as optimizable parameters and adjust them during the training process. It is also noted that the prediction model for erlotinib achieved only random performance with an AUC around 0.5, regardless of the settings for a or r. Since the response to erlotinib is related to EGFR mutations, performance may be improved by incorporating mutation data[24].

This study has several limitations. First, we split the cell line data into training and test datasets to evaluate the models' performance. This means that we were not yet able to confirm performance on fully independent datasets, and particularly the wider relevance of our models in an *in vivo* setting. In the future, we would like to extend our analysis and conduct additional validations using real patient data, as our ultimate goal. Second, since we compared our model with a simple neural network without a quantum circuit, and the performance differences were within bootstrap uncertainty, we do not claim a general performance advantage over classical models in the field of anti-cancer drug response prediction. We recognize that

various deep learning architectures optimized for omics data have been proposed[2,3]. Evaluating all these options requires a more comprehensive and extensive study. Furthermore, regularization techniques are likely to be effective in enhancing generalization ability[25]. We aim to investigate more advanced QHML anti-cancer drug response prediction models using the proposed normalization method and compare them with existing deep learning approaches in our future research. The applicability of the proposed normalization method to other tasks beyond predicting responses to anti-cancer drugs is one of the issues that should be examined in the future. We expect that it could be beneficial in many cases where the curse of dimensionality is an issue. Additionally, other highly promising applications of this normalization method will be in the field of computer vision, where QHML models have already demonstrated success[5,10]. It is important to note that this normalization method mitigates the vanishing gradient problem caused by value saturation, a typical issue in conventional deep learning. In addition to this classical vanishing gradient problem, quantum circuits are also vulnerable to another type of vanishing gradient problem known as the "barren plateau", caused by the circuit architecture and the vast search space it generates[26,27]. Therefore, to effectively explore the optimization landscape, it is crucial to consider both the appropriate selection of normalization and its relationship with the chosen circuit architecture.

In conclusion, we found that appropriate input value normalization for the quantum circuit improved performance of the QHML model in predicting anti-cancer drug responses using gene expression data, compared to the conventional tanh normalization. The proposed strategy enables stable training by avoiding value saturation caused by the existing normalization methods. Advances in quantum computing have accelerated remarkably in recent years, and practical applications are expected to become increasingly important in the biomedical field in the future. Based on the methodology we proposed, further advances in omics data analysis are anticipated through research combining deep learning and quantum computing.

Methods

Data collection and preprocessing

For both training and test datasets, we have used the cell line data obtained from the GDSC database. The GDSC database stores omics data and response values for various drugs for cancer cell lines[11,12]. These data were used in a previous study[2], and were obtained in a preprocessed form from the Zenodo repository managed by the authors (https://zenodo.org/record/4036592). We randomly split the data into training and test data at a 9:1 ratio. The input data consisted solely of Affymetrix microarray gene expression data, which had already undergone RMA-normalization[28] and log-transformation. In addition, standardization and homogenization were applied to ensure data compatibility across different experiments[4]. To exclude non-informative genes, we only used genes with variance exceeding 0.1 in the training data. The range of values after preprocessing was 0.93 to 20.8. For drug response, we have used continuous log(IC50) values for training and binary response labels from previous research[11] when conducting validation. For training, we normalized log(IC50) values to the range $-1 \sim 1$ using equation (5):

$$DrugResponse_{norm} = \tanh\left(\frac{log(IC50) - mean(log(IC50))}{std(log(IC50))}\right) \tag{5}$$

The analysis focused on drugs with at least 50 responder samples, selecting the following five anti-cancer drugs: cetuximab (Non-Responder (NR): 735 samples, Responder(R): 121 samples), cisplatin (NR: 752 samples, R: 77 samples), docetaxel (NR: 764 samples, R: 65 samples), erlotinib (NR: 298 samples, R: 64 samples), and gemcitabine (NR: 790 samples, R: 54 samples).

QHML model architecture

We have adopted a QHML model consisting of three parts: a classical neural network encoder, a normalization function, and PQC (Figure 1). As previously mentioned, all quantum computation except for those described in the "Execution on a quantum device" section was performed on a noiseless simulator ("default.qubit" from Pennylane), and quantum states were calculated exactly. We have used a combination of PyTorch[29](v2.3.1) and Pennylane[30](v0.40.0) to implement the combined model. All models were trained under identical settings, with the mean squared error loss function and Adam[31] optimizer. The batch size was set to 128. The learning rate was determined during the the hyperparameter search and set at the value 1×10^{-4}, 1×10^{-5} or 1×10^{-6}.

Classical neural network part for data encoding

The model first transforms input gene expression values into a low-dimensional embedding suitable for input to the quantum circuit. This transformation uses a neural network encoder, though the data still requires additional preprocessing as described before. The encoder neural network has three fully connected layers and uses the SiLU[32] activation function, with batch normalization applied between the linear layer and the activation function. The output dimensions of layers 1 and 2 are fixed at 512 and 128, respectively, while that of layer 3 is adjusted to match the size of the subsequent quantum circuit. For the classical model without quantum circuits, we similarly defined the dimension of layer 3. Specifically, we introduced the hyperparameters n_1 and n_2 (described later), which are not required in the classical model. Then, we defined the output dimension of layer 3

as the product of these hyperparameters ($n_1 \times n_2$), identical to the QHML model. We initialized the weights and biases of all linear layers using the default Kaiming Uniform algorithm in PyTorch. Specifically, when the input dimension of a linear layer is d, the weights and biases are initialized by randomly sampling from a uniform distribution $U(-\frac{1}{\sqrt{d}}, \frac{1}{\sqrt{d}})$. This ensures that the output values stay within a certain range, independent of the dimensions of each layer. For the model with "gradual tanh", which was used in the performance evaluation section, the distribution of the output of the classical neural network prior to training was within the range of -2.0 to 2.0 (Supplementary Table S3). If the linear layer is not initialized properly, value saturation may occur in the subsequent normalization function even during the early training stages, and there is a risk that the model will converge to a suboptimal solution. Therefore, if a different initialization method is used, the hyperparameters of the subsequent normalization function will need to be adjusted to prevent value saturation.

Normalization function for passing neural network output to parameterized quantum circuit

In our model, the encoder outputs are used as angles of the Z-rotation gates of the quantum circuit. Since each rotation gate has a 2π periodicity, multiple values can result in the same quantum state, potentially causing unstable training. The previous study[8] used a normalization function with tanh (Equation (3)) in its implementation:

$$\boldsymbol{\phi'} = \frac{\pi}{2}\tanh(\boldsymbol{\phi}), \tag{3}$$

where $\boldsymbol{\phi}$ is the output vector of the classical part, and $\boldsymbol{\phi'}$ is the converted vector passed to the quantum circuit. This function converts values between $-\frac{\pi}{2}$ and $\frac{\pi}{2}$, thereby avoiding the periodicity problem. One drawback of using equation (3) is that the gradient becomes excessively steep near zero. This causes many converted inputs to take values extremely close to $-\frac{\pi}{2}$ or $\frac{\pi}{2}$, destabilizing gradient updates and hindering the model's convergence to the optimal solution. The relationship between tanh function and latent issues during training has already been reported in the field of traditional deep learning[9]. Therefore, we propose the following transformation (Equation (4)):

$$\boldsymbol{\phi'} = r \cdot \tanh\left(\frac{\boldsymbol{\phi}}{a}\right). \tag{4}$$

This function generalizes Equation (3) and maps any real number into the range of $-r \sim r$. By setting a sufficiently large, the input range does not fall into the saturation range of tanh. In this case, the function behaves almost linearly within the observation range and acts as a bounded rescaling. This study also examines the effects of the choice for the value of r. There exists a trade-off between setting a large r and a small r. A larger r expands the reachable search space, enabling more complete coverage for better solutions. However, it simultaneously makes finding the optimal solution more difficult. Conversely, a smaller r narrows the search space, potentially making optimization easier, but there is no guarantee that a good solution exists within it. In addition, if r exceeds π, the periodicity issue may arise in principle, but this is limited to cases where the transformed values actually spread over a range greater than 2π.

Parameterized quantum circuit

After normalization, the features are passed to the PQC. Figure 5 shows the PQC design in our model. This circuit is nearly identical to the one used in[33]. The PQC consists of three parts: the encoding layer, the variational layer, and the measurement layer. We implemented the PQC using n_1 qubits. In the encoding layer, normalized embeddings are passed to the PQC as quantum states. This encoding method was proposed in[34] and[35]. Repeating the encoding layer n_2 times allows efficient data encoding with a relatively small number of qubits. It should be noted that the output dimension of the neural network is set to $n_1 \times n_2$, accordingly. Since the qubit is initialized to $|0\rangle$, the first Rz gates do not change the quantum state and therefore were removed. The next part is the variational layer, where the parameterized gates change the quantum state. The angles of these gates were randomly initialized by sampling values from a z-distribution. To achieve sufficient expressivity, this part is repeated n_3 times. The last part measures the state of the output qubits using a set of Pauli Z observables. After measuring these qubits, a single conventional neural network layer ("Linear" layer from PyTorch[29]) is used to obtain scalar drug response values.

Feature importance analysis

In this study, we employed the Integrated Gradient (IG) method for feature importance analysis. IG is a method for interpreting machine learning models that quantifies the attribution of each input feature to the output[14]. The IG score for the i-th dimension of the input x is calculated as shown in Equation 6:

$$IG_i = (x_i - x_i') \times \int_{\alpha=0}^{1} \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha, \tag{6}$$

where x_i is the ith component of x, x_i' is the ith component of the baseline, and F represents the prediction model. We used the mean expression values of the training data as the baseline. Since directly solving the integral was difficult, we calculated

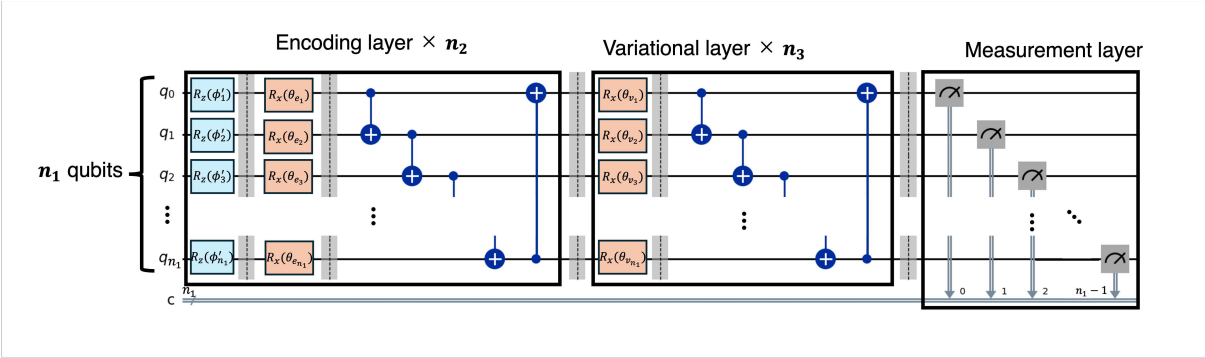


Figure 5. Parameterized quantum circuit architecture. The parameterized quantum circuit consists of three parts and is characterized by 3 hyperparameters (n_1,n_2, and n_3). The blue gates are Z-rotation gates, whose rotation angles are determined by the inputs of the quantum circuit. The red gates are X-rotation gates, whose angles are trainable parameters. All qubits are measured in the measurement layer.

approximate IG values. We used Captum[36] library to perform the calculations using the default Gauss-Legendre quadrature rule. Using this IG score, we identified "key genes" that greatly contribute to the prediction. Specifically, we calculated the average absolute value of the IG score for each gene and defined genes with scores in the top 5% as key gene. These were then used in a subsequent enrichment analysis using Metascape[15], which uses ontologies such as GO processes[16], KEGG pathways[17], Reactome gene sets[18], canonical pathways[19], and CORUM complexes[20] to statistically identify which known biological functions and pathways are associated with input genes.

Decision landscape in embedding space

To understand the details of quantum circuits' function in prediction models, we visualized the decision landscape in the embedding space just before the quantum circuit. We applied the test data to the trained docetaxel response prediction model with "gradual tanh" and extracted the embeddings just before the quantum circuit. We then applied PCA to these embeddings, using scikit-learn in order to perform dimensionality reduction, retaining only the first two principal components for analysis. The total explained variance proportion for these two axes reached 0.942 (PC1: 0.885, PC2: 0.057), verifying that most of the original structure is preserved even after the data are projected into the lower dimensional space. Then, we calculated the minimum, maximum and standard deviation (σ) of the dimensionally reduced test data for each of the PC1 and PC2 axes. For the visualization range, we set an area centered on the average of the maximum and minimum values, covering $\pm 4\sigma$ for each axis, generated a 200×200 grid of data points within this region, and then mapped them back to the original embedding space using the inverse mapping of PCA. Finally, we input all these mapped data points into the quantum circuit and plotted the predicted values as a contour plot using matplotlib's "contourf" function.

Data availability

All datasets used in this paper can be downloaded from the Zenodo repository (https://zenodo.org/record/4036592).

Code availability

All codes used in this paper are on the github repository (https://github.com/Taka-bio/GradualTanh).

References

1. Weinshilboum, R. M. & Wang, L. Pharmacogenomics: Precision Medicine and Drug Response. *Mayo Clin. Proc.* **92**, 1711–1722, DOI: 10.1016/j.mayocp.2017.09.001 (2017).
2. Sharifi-Noghabi, H., Zolotareva, O., Collins, C. C. & Ester, M. MOLI: multi-omics late integration with deep neural networks for drug response prediction. *Bioinformatics* **35**, i501–i509, DOI: 10.1093/bioinformatics/btz318 (2019).
3. Sharma, A., Lysenko, A., Boroevich, K. A. & Tsunoda, T. DeepInsight-3D architecture for anti-cancer drug response prediction with deep-learning on multi-omics. *Sci. Reports* **13**, 2483, DOI: 10.1038/s41598-023-29644-3 (2023).
4. Geeleher, P., Cox, N. J. & Huang, R. S. Clinical drug response can be predicted using baseline gene expression levels and in vitro drug sensitivity in cell lines. *Genome Biol.* **15**, R47, DOI: 10.1186/gb-2014-15-3-r47 (2014).

5. Senokosov, A., Sedykh, A., Sagingalieva, A., Kyriacou, B. & Melnikov, A. Quantum machine learning for image classification. *Mach. Learn. Sci. Technol.* **5**, 015040, DOI: 10.1088/2632-2153/ad2aef (2024).

6. Bischof, L., Teodoropol, S., Füchslin, R. M. & Stockinger, K. Hybrid quantum neural networks show strongly reduced need for free parameters in entity matching. *Sci. Reports* **15**, 4318, DOI: 10.1038/s41598-025-88177-z (2025).

7. Caro, M. C. *et al.* Generalization in quantum machine learning from few training data. *Nat. Commun.* **13**, 4919, DOI: 10.1038/s41467-022-32550-3 (2022).

8. Mari, A., Bromley, T. R., Izaac, J., Schuld, M. & Killoran, N. Transfer learning in hybrid classical-quantum neural networks. *Quantum* **4**, 340, DOI: 10.22331/q-2020-10-09-340 (2020).

9. Glorot, X. & Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In Teh, Y. W. & Titterington, M. (eds.) *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, vol. 9 of *Proceedings of Machine Learning Research*, 249–256 (PMLR, Chia Laguna Resort, Sardinia, Italy, 2010).

10. Alam, M., Kundu, S., Topaloglu, R. O. & Ghosh, S. Quantum-Classical Hybrid Machine Learning for Image Classification (ICCAD Special Session Paper). In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, 1–7, DOI: 10.1109/ICCAD51958.2021.9643516 (2021).

11. Iorio, F. *et al.* A Landscape of Pharmacogenomic Interactions in Cancer. *Cell* **166**, 740–754, DOI: 10.1016/j.cell.2016.06.017 (2016).

12. Yang, W. *et al.* Genomics of Drug Sensitivity in Cancer (GDSC): a resource for therapeutic biomarker discovery in cancer cells. *Nucleic Acids Res.* **41**, D955–D961, DOI: 10.1093/nar/gks1111 (2012).

13. Pedregosa, F. *et al.* Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

14. Sundararajan, M., Taly, A. & Yan, Q. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, 3319–3328 (JMLR.org, 2017).

15. Zhou, Y. *et al.* Metascape provides a biologist-oriented resource for the analysis of systems-level datasets. *Nat. Commun.* **10**, 1523, DOI: 10.1038/s41467-019-09234-6 (2019).

16. Ashburner, M. *et al.* Gene Ontology: tool for the unification of biology. *Nat. Genet.* **25**, 25–29, DOI: 10.1038/75556 (2000).

17. Kanehisa, M. & Goto, S. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.* **28**, 27–30, DOI: 10.1093/nar/28.1.27 (2000).

18. Fabregat, A. *et al.* The Reactome Pathway Knowledgebase. *Nucleic Acids Res.* **46**, D649–D655, DOI: 10.1093/nar/gkx1132 (2017).

19. Subramanian, A. *et al.* Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proc. Natl. Acad. Sci.* **102**, 15545–15550, DOI: 10.1073/pnas.0506580102 (2005).

20. Ruepp, A. *et al.* CORUM: the comprehensive resource of mammalian protein complexes—2009. *Nucleic Acids Res.* **38**, D497–D501, DOI: 10.1093/nar/gkp914 (2009).

21. Puhr, M. *et al.* Epithelial-to-Mesenchymal Transition Leads to Docetaxel Resistance in Prostate Cancer and Is Mediated by Reduced Expression of miR-200c and miR-205. *The Am. J. Pathol.* **181**, 2188–2201, DOI: https://doi.org/10.1016/j.ajpath.2012.08.011 (2012).

22. Aoudjit, F. & Vuori, K. Integrin signaling inhibits paclitaxel-induced apoptosis in breast cancer cells. *Oncogene* **20**, 4995–5004, DOI: 10.1038/sj.onc.1204554 (2001).

23. Quantinuum nexus (2024).

24. Rosell, R. *et al.* Erlotinib versus standard chemotherapy as first-line treatment for European patients with advanced EGFR mutation-positive non-small-cell lung cancer (EURTAC): a multicentre, open-label, randomised phase 3 trial. *The Lancet Oncol.* **13**, 239–246, DOI: 10.1016/S1470-2045(11)70393-X (2012).

25. Tian, Y. & Zhang, Y. A comprehensive survey on regularization strategies in machine learning. *Inf. Fusion* **80**, 146–166, DOI: 10.1016/j.inffus.2021.11.005 (2022).

26. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **9**, 4812, DOI: 10.1038/s41467-018-07090-4 (2018).

27. Gelman, M. A survey of methods for mitigating barren plateaus for parameterized quantum circuits. *arXiv preprint arXiv:2406.14285* (2024).

28. Irizarry, R. A. *et al.* Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res.* **31**, e15–e15, DOI: 10.1093/nar/gng015 (2003).

29. Paszke, A. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703* (2019).

30. Bergholm, V. *et al.* Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968* (2018).

31. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

32. Hendrycks, D. & Gimpel, K. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415* (2016).

33. Sagingalieva, A. *et al.* Hybrid Quantum Neural Network for Drug Response Prediction. *Cancers* **15**, DOI: 10.3390/cancers15102705 (2023).

34. Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E. & Latorre, J. I. Data re-uploading for a universal quantum classifier. *Quantum* **4**, 226, DOI: 10.22331/q-2020-02-06-226 (2020).

35. Schuld, M., Sweke, R. & Meyer, J. J. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Phys. Rev. A* **103**, 032430, DOI: 10.1103/PhysRevA.103.032430 (2021).

36. Kokhlikyan, N. *et al.* Captum: A unified and generic model interpretability library for PyTorch (2020). 2009.07896.

Acknowledgements

We would like to thank Drs. Mitsuhisa Sato, Maxence Vandromme, Miwako Tsuji, Kengo Nakajima, Yuetsu Kodama and Kazuya Yamazaki for their helpful discussions.

Funding

This study was supported by the World-leading Innovative Graduate Study Program in Proactive Environmental Studies (WINGS-PES), The University of Tokyo. This work was also partially funded by the JSPS KAKENHI Grant Numbers JP25K02261 and JP24K15175, and Cabinet Office BRIDGE (programs for bridging the gap between R&D and the ideal society (Society 5.0) and generating economic and social value), Japan.

Author contributions statement

T.I. conceived of the idea, carried out the experiments, and wrote the first draft of the manuscript. T.I., A.L. and T.T. planed the detail of analysis. A.L. and T.T. contributed in the subsequent versions of the manuscript. T.T. provided resources and supervised the project. All authors read and approved the final manuscript.

Additional information

Competing interests

The authors declare no competing interests.

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- Supplementaryinformation.pdf