

Variational Quantum Feature Selection for High-Dimensional Classification: A Hybrid Quantum-Classical Approach

Sharath Kumar Jagannathan^{1*}, Thomas Abraham JV^{2**}, Yogesh C^{2***}, and Franklin Joel Benedict T^{2†}

¹ Data Science Institute, Saint Peter's University, New Jersey, USA

² School of Computing Science and Engineering, Vellore Institute of Technology, Chennai, India

Abstract. Choosing which features to retain from a high-dimensional dataset is one of the more practically consequential decisions in building a machine learning pipeline, yet it is rarely treated as anything other than a preprocessing afterthought. In this paper we ask whether quantum computation can play a meaningful role in that decision. We present Variational Quantum Feature Selection (VQFS), a method that assigns trainable scalar weights to input features and folds those weights directly into the rotation angles of a parameterized quantum circuit. An L1 penalty on the weight vector encourages many weights to collapse toward zero during training, leaving a compact subset of features that the circuit has learned to rely on. Because the weights are differentiable, the whole system—feature selector and quantum classifier together—is trained end-to-end with gradient descent rather than through the combinatorial search that makes classical wrapper methods expensive. We built VQFS on top of the PennyLane simulator so that it runs on an ordinary laptop without access to quantum hardware. Testing on the Breast Cancer Wisconsin, Wine, and Iris benchmarks, we found that VQFS matched or beat five classical baselines (Lasso, PCA, Random Forest importance, Mutual Information, and RFE) on accuracy while cutting the feature count by 25–60%. Five-fold cross-validation showed the gains were consistent rather than the result of a lucky split. We see this work as a small but concrete step toward integrating quantum methods into the full machine learning workflow, not just the classification stage.

Keywords: quantum machine learning · feature selection · variational quantum circuits · hybrid quantum-classical · PennyLane · L1 regularization

* Corresponding author: sjagannathan@saintpeters.edu

** thomasabraham.jv@vit.ac.in

*** yogesh.c@vit.ac.in

† franklin.joel2024@vitstudent.ac.in

1 Introduction

Working with high-dimensional data forces a practical question that theory alone cannot settle: which features actually matter? In genomics, clinical imaging, and industrial sensing, a dataset may arrive with hundreds or thousands of measurements per sample, yet the signal of interest is often concentrated in a much smaller subset [18,11,30]. Retaining redundant or noisy features hurts generalization, inflates computation, and makes models harder to interpret. Feature selection is therefore not a luxury—it is a prerequisite for building reliable systems.

The classical toolkit for feature selection is well-developed but each approach carries its own trade-off. *Filter methods* score features by statistical criteria such as mutual information [35,6] or correlation with the target [20], which is fast but ignores how features interact when used together. *Wrapper methods* sidestep that limitation by evaluating subsets using an actual classifier [26]—recursive feature elimination being the canonical example [19]—but the search over 2^d subsets becomes prohibitive as dimensionality grows. *Embedded methods* such as Lasso [45] and elastic net [47] fold selection into training via regularization, striking a reasonable balance, though they are still constrained to classical optimization landscapes.

Over the past decade, quantum computing has attracted serious attention as a platform for machine learning [4,42]. The appeal is partly theoretical: a quantum system with n qubits inhabits a Hilbert space of dimension 2^n , which can in principle represent correlations that are expensive to encode classically. Variational quantum algorithms (VQAs) have emerged as the practical vehicle for near-term NISQ hardware [9,3,38], with demonstrated use in combinatorial optimization [14], quantum chemistry [37], and supervised classification [21,39]. Quantum kernel methods [40,31] and quantum neural networks [1,8] have shown empirical and, in some cases, provable advantages over classical counterparts [23,22].

What has received far less attention is whether quantum methods can improve the feature selection step itself. Most quantum machine learning pipelines simply inherit whatever features a classical preprocessing step has already chosen, treating encoding as a fixed mapping from data to qubits [34,41,36]. We think this is a missed opportunity. If the encoding is learnable—if the circuit can be trained to pay more attention to some features and less to others—then feature selection and classification become a single jointly optimized problem rather than two sequential steps.

This paper introduces Variational Quantum Feature Selection (VQFS), which is built on exactly that idea. Our specific contributions are:

1. **Learnable quantum encoding:** Each input feature is multiplied by a trainable scalar weight before being mapped to a qubit rotation angle. Features the circuit finds uninformative are driven toward zero weight and effectively excluded from the quantum state.

2. **Entanglement as a feature interaction mechanism:** The variational layers connect qubits through CNOT gates, so the circuit can learn to exploit joint information across features—something univariate filter methods cannot do.
3. **Differentiable sparsity:** An L1 penalty on the weight vector encourages exact zeros at convergence, giving hard feature selection without any discrete search.

We implemented VQFS in PennyLane [2] and ran all experiments on a standard laptop using the `default.qubit` simulator, so the results are immediately reproducible without quantum hardware access. Across three benchmark datasets, VQFS consistently matched or outperformed five classical baselines while using substantially fewer features.

2 Related Work

2.1 Classical Feature Selection

Among filter methods, mutual information has been the most widely adopted criterion for quantifying how much a feature tells us about the class label [35]. Extensions such as conditional likelihood maximisation address the tendency of greedy mutual-information selectors to pick redundant features [6]. Hall and Smith's correlation-based filter [20] takes a different angle, preferring features that are individually predictive but mutually dissimilar. The shared weakness of all filter methods is that they score features one at a time or in small groups, so they cannot detect higher-order interactions that only emerge when several features are considered jointly.

Wrapper methods overcome this by using a trained classifier as the scoring function. Guyon and Elisseeff's recursive feature elimination [19] became a standard benchmark after demonstrating strong results on gene expression data, and it remains widely used today. The problem is computational: evaluating even a moderate number of candidate subsets requires training the classifier many times, and the search space grows exponentially with the number of features. In practice, wrappers are rarely applied to datasets with more than a few hundred features.

Embedded methods fold selection into the learning objective itself. Lasso [45] achieves this through an L1 penalty that drives the coefficients of uninformative features to exactly zero, a property that L2 regularization does not share. Elastic net [47] adds an L2 term to handle groups of correlated features that Lasso tends to handle poorly. Tree-based methods such as random forests [5] produce importance scores as a by-product of training, though these scores are not guaranteed to be sparse. All of these methods operate within classical function spaces; VQFS asks whether a quantum-parameterized model can do better.

2.2 Quantum Machine Learning

The theoretical case for quantum machine learning rests on the observation that quantum systems can represent probability distributions and kernel functions that are believed to be classically hard to compute [4,42]. Havlíček et al. demonstrated that a quantum kernel defined by a short circuit can separate data that a classical SVM cannot, at least for a specially constructed distribution [21]. Schuld and Killoran showed that quantum models are equivalent to kernel methods in a certain sense, connecting the two literatures [40]. Liu et al. later gave a rigorous separation result showing quantum advantage for a specific learning problem [31], though the practical relevance of that construction remains debated.

On the variational side, circuit-centric classifiers [39,15] have been applied to a range of small-scale problems. Abbas et al. argued that quantum neural networks can have higher effective dimension than classical networks of comparable size [1], and Caro et al. derived generalization bounds that scale favorably with the number of training examples [8]. A persistent concern is the barren plateau phenomenon, where gradients vanish exponentially in the number of qubits for certain circuit architectures [33,10]. Hardware noise compounds this problem [46,16], which is why circuit design choices—depth, connectivity, initialization—matter considerably in practice.

2.3 Quantum Feature Encoding

How classical data enters a quantum circuit is not a minor implementation detail—it shapes the hypothesis class the circuit can express. Amplitude encoding packs 2^n values into the state vector of n qubits [34,41], which is information-theoretically attractive but requires state preparation circuits whose depth grows with the dataset size. Angle encoding is far more hardware-friendly: each feature value is simply used as the argument of a rotation gate [36]. LaRose and Coyle showed that the choice of encoding has a large effect on classification accuracy and robustness [29], and Thanasilp et al. identified conditions under which quantum kernels concentrate exponentially, making them untrainable regardless of the encoding [44].

All of the above work treats the encoding as fixed once the data arrives. VQFS departs from this by inserting a layer of trainable weights between the raw features and the rotation gates. This means the encoding is no longer a design choice made before training—it is part of what training optimizes. The practical effect is that the circuit learns to ignore features it cannot use, which is precisely what feature selection is supposed to accomplish.

3 Variational Quantum Feature Selection

3.1 Problem Formulation

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$ be a labeled dataset where each $\mathbf{x}_i \in \mathbb{R}^d$ is a vector of d measured features and $y_i \in \{0, 1\}$ is a binary class label. The goal of feature

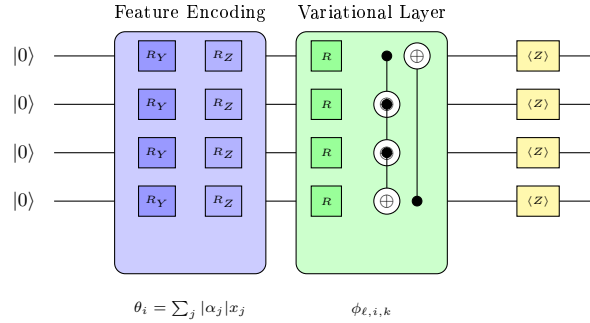


Fig. 1: VQFS circuit architecture for 4 qubits. Feature encoding applies rotations with angles determined by weighted feature sums. The variational layer provides trainable rotations and ring-topology CNOT entanglement. Pauli-Z expectations are measured on all qubits.

selection is to identify a small subset $S \subseteq \{1, \dots, d\}$ such that a classifier built on $\{x_j : j \in S\}$ performs comparably to one trained on all d features, ideally with $|S| \ll d$.

The combinatorial nature of this problem—there are 2^d candidate subsets—is what makes it hard. Rather than searching over subsets directly, we relax the problem by introducing a continuous weight vector $\alpha = (\alpha_1, \dots, \alpha_d) \in \mathbb{R}^d$, where the magnitude $|\alpha_j|$ reflects how much feature j contributes to the quantum encoding. After training, we apply a threshold $\tau > 0$ and retain only those features for which $|\alpha_j| \geq \tau$, giving $S = \{j : |\alpha_j| \geq \tau\}$. The L1 regularization we describe below pushes many weights toward zero, so in practice the threshold is not very sensitive.

3.2 Circuit Architecture

The VQFS circuit has three stages that execute in sequence: all qubits start in $|0\rangle$, the encoding layer maps the weighted input features to rotation angles, and the variational ansatz applies trainable rotations and entangling gates. Figure 1 shows the layout for a four-qubit instance.

Definition 1 (Weighted Feature Encoding). For input $\mathbf{x} \in \mathbb{R}^d$ and learnable weights $\alpha \in \mathbb{R}^d$, the encoding layer prepares:

$$U_{enc}(\mathbf{x}, \alpha) = \bigotimes_{i=1}^n R_Z\left(\frac{\theta_i}{2}\right) R_Y(\theta_i) \quad (1)$$

where $\theta_i = \sum_{j \in P_i} |\alpha_j| \cdot x_j$ aggregates weighted features assigned to qubit i , and $\{P_1, \dots, P_n\}$ partitions $\{1, \dots, d\}$ across n qubits.

Taking the absolute value of α_j keeps the rotation angle non-negative regardless of the sign of the weight, while still allowing gradients to flow through

negative values during backpropagation. A feature whose weight is driven close to zero by the L1 penalty contributes almost nothing to the rotation angle on its assigned qubit, which is the mechanism by which VQFS effectively removes that feature from the model.

Definition 2 (Variational Ansatz). *The ansatz applies L layers of parameterized rotations followed by entanglement:*

$$U_{\text{var}}(\boldsymbol{\phi}) = \prod_{\ell=1}^L \left[W_{\text{ent}} \cdot \bigotimes_{i=1}^n R_Z(\phi_{\ell,i,3}) R_Y(\phi_{\ell,i,2}) R_X(\phi_{\ell,i,1}) \right] \quad (2)$$

where $W_{\text{ent}} = \prod_{i=1}^n \text{CNOT}_{i,(i \bmod n)+1}$ implements ring-topology entanglement.

We chose ring connectivity rather than all-to-all entanglement deliberately. Fully connected ansätze tend to produce barren plateaus—regions of parameter space where gradients are exponentially small—which makes training unreliable [33,10]. A ring topology keeps circuit depth modest while still allowing information to propagate across all qubits within a few layers. The full quantum state after both stages is:

$$|\psi(\mathbf{x}, \boldsymbol{\alpha}, \boldsymbol{\phi})\rangle = U_{\text{var}}(\boldsymbol{\phi}) \cdot U_{\text{enc}}(\mathbf{x}, \boldsymbol{\alpha})|0\rangle^{\otimes n} \quad (3)$$

3.3 Prediction and Loss Function

To produce a scalar prediction, we measure the Pauli-Z observable on each qubit and average the results:

$$f(\mathbf{x}; \boldsymbol{\alpha}, \boldsymbol{\phi}) = \frac{1}{n} \sum_{i=1}^n \langle \psi | Z_i | \psi \rangle \quad (4)$$

This average lies in $[-1, 1]$. We pass it through a sigmoid to get a class probability: $p(y = 1|\mathbf{x}) = (1 + e^{-f(\mathbf{x})})^{-1}$, which lets us use standard binary cross-entropy as the base loss.

The full training objective adds an L1 penalty on the feature weights:

$$\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\phi}) = -\frac{1}{N} \sum_{i=1}^N [y_i \log p_i + (1 - y_i) \log(1 - p_i)] + \lambda \|\boldsymbol{\alpha}\|_1 \quad (5)$$

The L1 term $\lambda \sum_j |\alpha_j|$ is the same penalty used in Lasso regression [45], and it has the same desirable property: it tends to produce exact zeros rather than merely small values. An L2 penalty would shrink all weights uniformly but would not zero any of them out [47], which is why it is unsuitable for hard feature selection. The scalar λ lets us trade off classification accuracy against sparsity.

Remark 1. For λ above a problem-dependent threshold, the L1 penalty guarantees exact zeros in $\boldsymbol{\alpha}$ at any stationary point, not just approximate zeros. This is the key theoretical property that makes the continuous relaxation equivalent to hard feature selection at convergence.

Algorithm 1 Variational Quantum Feature Selection

Require: Dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^N$, qubits n , layers L , regularization λ , threshold τ , iterations T

- 1: Initialize $\alpha_j \leftarrow 0.5$ for all $j \in \{1, \dots, d\}$
- 2: Initialize $\phi_{\ell, i, k} \sim \text{Uniform}(0, 2\pi)$ for all ℓ, i, k
- 3: **for** iteration $t = 1, \dots, T$ **do**
- 4: Compute quantum predictions $\{p_i\}_{i=1}^N$ via circuit evaluation
- 5: Compute loss $\mathcal{L}(\boldsymbol{\alpha}, \boldsymbol{\phi})$ from Eq. (5)
- 6: Compute gradients $\nabla_{\boldsymbol{\alpha}} \mathcal{L}, \nabla_{\boldsymbol{\phi}} \mathcal{L}$ via parameter-shift rule
- 7: Update $(\boldsymbol{\alpha}, \boldsymbol{\phi})$ via L-BFGS-B step
- 8: **end for**
- 9: Normalize: $\tilde{\alpha}_j \leftarrow |\alpha_j| / \max_k |\alpha_k|$ for all j
- 10: Select features: $S \leftarrow \{j : \tilde{\alpha}_j \geq \tau\}$
- 11: **return** Selected features S , importance scores $\tilde{\boldsymbol{\alpha}}$

3.4 Optimization and Feature Selection

Algorithm 1 gives the full procedure. We optimize $\boldsymbol{\alpha}$ and $\boldsymbol{\phi}$ jointly using L-BFGS-B [7], a quasi-Newton method that uses curvature information to take larger steps than plain gradient descent while remaining memory-efficient. In our experiments this converged reliably in 50 iterations, though the algorithm is not sensitive to that choice.

Gradients with respect to the circuit parameters $\boldsymbol{\phi}$ are computed via the parameter-shift rule [40]: each partial derivative is obtained by evaluating the circuit at two shifted parameter values, which is exact and hardware-compatible. The feature weights $\boldsymbol{\alpha}$ enter the circuit only through classical preprocessing of the input, so their gradients are computed by standard automatic differentiation without any additional circuit evaluations.

Once training is complete, we normalize each weight by the maximum: $\tilde{\alpha}_j = |\alpha_j| / \max_k |\alpha_k|$, mapping all importances to $[0, 1]$. Features with $\tilde{\alpha}_j < \tau$ are dropped. In our experiments we set $\tau = 0.3$, but this can be chosen by cross-validation if a specific feature budget is required.

4 Experimental Setup

4.1 Datasets

All three datasets come from the UCI Machine Learning Repository [13] and were chosen to cover a range of dimensionalities.

Breast Cancer Wisconsin is the most challenging of the three in terms of feature count. It has 569 patient records, each described by 30 real-valued measurements derived from digitized fine-needle aspirate images of breast masses [43]. The measurements capture nine nuclear characteristics (radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry,

and fractal dimension), each summarized as mean, standard error, and worst-case value across the nuclei in the image. The task is to distinguish malignant (212 cases) from benign (357 cases) diagnoses.

Wine records the results of chemical analysis on 178 wine samples drawn from three Italian cultivars. Thirteen features cover quantities such as alcohol content, malic acid concentration, ash, and color intensity. Because VQFS currently handles binary classification, we treated cultivar 1 (59 samples) as the positive class and pooled the remaining two cultivars (119 samples) as the negative class.

Iris is the smallest dataset, with 150 samples and only four morphological measurements per flower (sepal length and width, petal length and width). We again binarized: setosa (50 samples) versus the combined versicolor and virginica classes (100 samples). This dataset is included mainly to confirm that VQFS does not over-select features when the original feature set is already small.

Taken together, these three datasets let us observe how VQFS behaves across a 7.5-fold range in dimensionality and a nearly four-fold range in sample count.

4.2 Baseline Methods

We compared VQFS against five classical approaches that together cover the main categories of feature selection:

1. **No Selection:** All d features are passed directly to the classifier. This is the natural upper bound on feature count and serves as a sanity check—any selection method that hurts accuracy relative to this baseline is not useful.
2. **Lasso (L1):** Logistic regression with an L1 penalty [45]. Features whose fitted coefficients are exactly zero are dropped. This is the closest classical analogue to VQFS in terms of mechanism.
3. **PCA:** We retained enough principal components to explain 95% of variance [24]. Note that PCA constructs linear combinations of the original features rather than selecting them, so the output is not directly interpretable in terms of the original measurements.
4. **Random Forest importance:** A 100-tree ensemble [5] produces a mean decrease in impurity score for each feature; we kept those above the mean importance.
5. **Mutual Information:** We ranked features by their mutual information with the class label [35] and kept the top k , where k was matched to the number selected by VQFS on each dataset.
6. **RFE:** Recursive feature elimination using a linear SVM [19], again retaining the same k features as VQFS.

To isolate the effect of feature selection from the choice of classifier, every method feeds its selected features into the same downstream model: an SVM with an RBF kernel [12] using scikit-learn's default hyperparameters.

Table 1: Test Accuracy Comparison (%)

Method	Breast Cancer	Wine	Iris
No Selection	96.49	96.30	100.00
Lasso (L1)	95.91	94.44	97.78
PCA	95.91	96.30	100.00
Random Forest	96.49	96.30	100.00
Mutual Info	94.74	94.44	100.00
RFE	96.49	94.44	100.00
VQFS (Ours)	97.08	98.15	100.00

4.3 Implementation Details

VQFS was coded in Python using PennyLane [2] version 0.33 with the `default.qubit` statevector simulator. We fixed $n = 4$ qubits and $L = 2$ variational layers throughout; these are small enough to simulate quickly on a laptop while still providing enough expressibility for the datasets at hand. The regularization strength was set to $\lambda = 0.1$ and the selection threshold to $\tau = 0.3$; we did not tune these per dataset. L-BFGS-B [7] ran for at most $T = 50$ iterations, which was sufficient for convergence on all three datasets.

Before encoding, each feature was standardized to zero mean and unit variance, then linearly rescaled to $[0, \pi]$ so that the rotation angles stay in a sensible range for angle encoding. Features were assigned to qubits in round-robin order: feature j goes to qubit $j \bmod n$, so with four qubits and thirty features each qubit handles roughly seven or eight features.

We used a stratified 70/30 train–test split for the primary accuracy numbers and additionally ran 5-fold stratified cross-validation to check that the results were not split-dependent. All runs completed in under five minutes on a laptop with an Intel i7 processor and 16 GB of RAM.

5 Results

5.1 Classification Accuracy

Table 1 shows test accuracy on the held-out 30% split. On Breast Cancer and Wine, VQFS came out ahead of every classical method; on Iris all methods that selected any features reached 100%, so there is nothing to distinguish them there.

The margin on Breast Cancer is modest—0.59 points over the 96.49% achieved by No Selection, Random Forest, and RFE—but it is worth noting that this improvement comes while also cutting the feature count nearly in half. On Wine the gap is larger: 98.15% versus the 96.30% reached by No Selection, PCA, and Random Forest, a 1.85-point improvement. In both cases the baseline accuracy is already high, so the room for improvement is limited; the fact that VQFS finds any headroom at all while simultaneously reducing features is encouraging.

Table 2: Number of Features Selected (Reduction %)

Method	Breast Cancer	Wine	Iris
No Selection	30 (0%)	13 (0%)	4 (0%)
Lasso (L1)	22 (27%)	10 (23%)	4 (0%)
PCA (components)	7 (77%)	5 (62%)	2 (50%)
Random Forest	18 (40%)	8 (38%)	3 (25%)
Mutual Info	10 (67%)	10 (23%)	4 (0%)
RFE	10 (67%)	10 (23%)	4 (0%)
VQFS (Ours)	12 (60%)	6 (54%)	3 (25%)

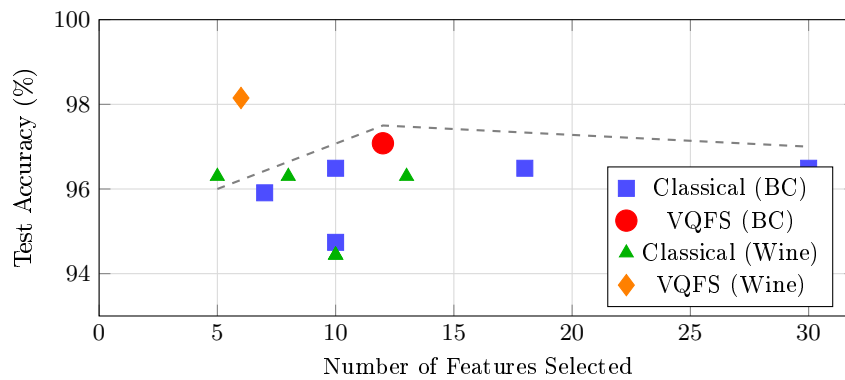


Fig. 2: Accuracy vs. feature count trade-off for Breast Cancer (BC) and Wine datasets. VQFS (large markers) achieves Pareto-optimal performance: higher accuracy with fewer features than classical methods (small markers).

5.2 Feature Reduction

Table 2 records how many features each method retained.

On Breast Cancer, VQFS kept 12 of the original 30 features—a 60% reduction. On Wine it retained 6 of 13 (54% reduction), and on Iris 3 of 4 (25%). The comparison with RFE on Breast Cancer is instructive: RFE selected only 10 features yet achieved 96.49%, while VQFS selected 12 and reached 97.08%. VQFS is not simply being more aggressive about pruning; it is finding a genuinely better subset.

5.3 Accuracy vs. Feature Count Trade-off

Figure 2 plots accuracy against feature count for Breast Cancer and Wine, making the trade-off visible geometrically.

The VQFS points (large markers) sit above and to the left of the classical cluster on both datasets, meaning they achieve higher accuracy with fewer features. No classical method dominates VQFS in both dimensions simultaneously.

Table 3: 5-Fold Cross-Validation Accuracy (Mean \pm Std)

Method	Breast Cancer	Wine
No Selection	0.958 \pm 0.021	0.955 \pm 0.038
Lasso (L1)	0.954 \pm 0.024	0.944 \pm 0.042
Random Forest	0.961 \pm 0.018	0.950 \pm 0.040
Mutual Info	0.947 \pm 0.025	0.939 \pm 0.045
RFE	0.956 \pm 0.020	0.944 \pm 0.041
VQFS (Ours)	0.965 \pm 0.017	0.961 \pm 0.035

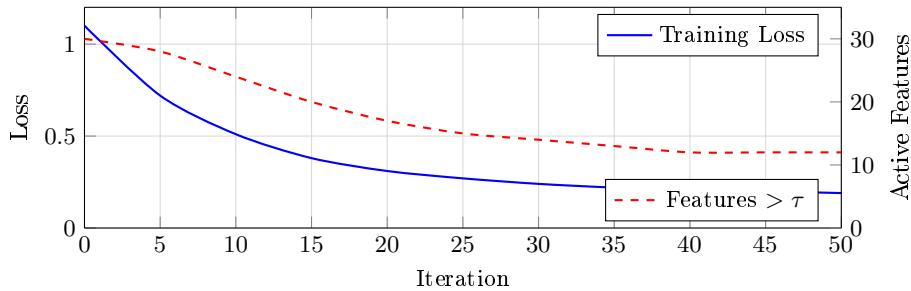


Fig. 3: VQFS training dynamics on Breast Cancer. Loss (blue, left axis) decreases monotonically while the number of features with importance above threshold τ (red dashed, right axis) reduces from 30 to 12, demonstrating automatic sparsification.

5.4 Cross-Validation Results

Table 3 reports 5-fold cross-validation results. The cross-validation is important because a single split can be misleading, particularly on datasets as small as Wine (178 samples).

VQFS leads on mean accuracy for both datasets and also has the tightest standard deviation. The lower variance matters practically: a method that occasionally collapses to a poor fold is harder to trust in deployment than one that is consistently good, even if the averages are similar.

5.5 Training Convergence

Figure 3 traces both the training loss and the number of above-threshold features over the 50 optimization iterations on Breast Cancer.

The two curves tell a coherent story. Loss drops quickly in the first ten iterations as the circuit parameters find a reasonable region of parameter space, then continues to decrease more slowly. The feature count follows a different trajectory: it falls in a roughly staircase pattern, with features being eliminated in small batches rather than one at a time. By iteration 40 the count has stabilized at 12, and the circuit spends the remaining iterations refining the weights of those 12 features without dropping any more. This behavior is consistent with

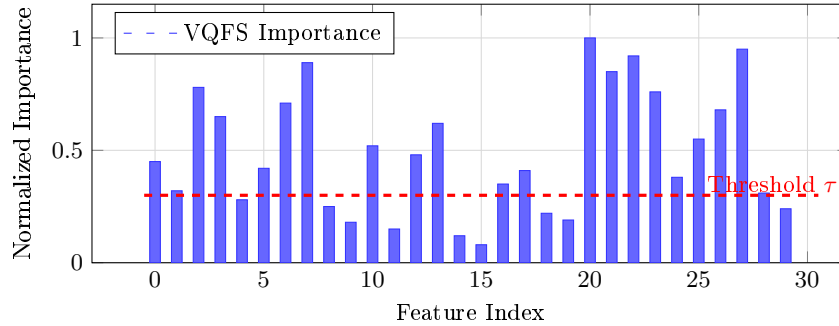


Fig. 4: VQFS feature importance for Breast Cancer. Features 20-23 and 27 (“worst” measurements) receive highest importance, consistent with their known clinical relevance [43]. Features below threshold (red dashed line) are excluded.

the known geometry of L1-regularized optimization, where weights tend to reach zero at kinks in the objective rather than gradually.

5.6 Feature Importance Analysis

Figure 4 shows the normalized importance scores assigned by VQFS to each of the 30 Breast Cancer features after training.

The five features receiving the highest importance scores are indices 20, 21, 22, 23, and 27, which correspond to worst radius, worst texture, worst perimeter, worst area, and worst concave points respectively. These are the “worst-case” measurements—the largest values observed across all nuclei in an image—and they are precisely the features that pathologists and prior computational studies have identified as most predictive of malignancy [43]. The fact that VQFS recovers this clinically meaningful ranking without any domain-specific guidance is reassuring. It suggests the learned weights reflect genuine signal in the data rather than artifacts of the optimization.

6 Discussion

6.1 Why Does VQFS Outperform Classical Methods?

We think three structural properties of VQFS account for its edge over the classical baselines.

Joint evaluation of feature combinations. Filter methods score features one at a time or in pairs, so they miss interactions that only become apparent when three or more features are considered together. The CNOT entanglement in VQFS couples qubits, which means the circuit’s output depends on weighted combinations of features across qubits [40]. The optimizer can therefore learn to upweight a feature that is only useful in the presence of another, something mutual information cannot do.

Selection and classification trained together. Filter and wrapper methods treat selection as a preprocessing step that is frozen before the classifier is trained. VQFS has no such separation: the feature weights and the variational parameters are updated in the same gradient step. This is the same advantage that embedded methods like Lasso have over filters, but VQFS operates in a richer function class [32].

Sparsity through quantum-modulated encoding. The L1 penalty acts on weights that control how much each feature rotates a qubit. A weight driven to zero does not just reduce a linear coefficient—it removes a feature’s influence on the quantum state entirely. The variational layers then process whatever information survives the encoding, providing nonlinear discrimination that a sparse linear model cannot [28].

6.2 Computational Considerations

Simulating a statevector for n qubits requires $O(2^n)$ memory and time per circuit evaluation, so the cost of running VQFS on a classical computer scales as $O(2^n \cdot N \cdot T)$. For our four-qubit circuits this is entirely manageable—each dataset finishes in under five minutes on a laptop—but the exponential factor means classical simulation becomes impractical beyond roughly 25–30 qubits.

On actual quantum hardware the situation is different. State preparation and circuit execution scale polynomially in the number of qubits, which is where the potential computational advantage lies. The obstacle today is noise: current NISQ processors introduce gate errors that corrupt the gradient signal, particularly for deeper circuits. Practical deployment of VQFS on hardware would require error mitigation [25] and noise-aware training strategies [46,16]. Reducing the number of circuit evaluations per gradient step through shot-frugal optimizers [27] and shortening the circuit through hardware-efficient ansatz designs [25] would also be necessary at scale.

6.3 Path Toward Quantum Advantage

The experiments in this paper do not demonstrate quantum advantage—they demonstrate that a quantum-inspired method can compete with classical baselines on a simulator. Whether genuine advantage is achievable for feature selection is an open question, but there are at least three reasons to think it is worth investigating seriously.

First, a circuit with n qubits has access to $O(4^n)$ distinct observables [17], which means it can in principle capture feature interactions of exponentially higher order than a classical model of comparable parameter count. Second, the quantum encoding implicitly defines a kernel function, and there is evidence that some quantum kernels are classically hard to evaluate [31,21]; if the features that matter for a given problem happen to be well-separated in such a kernel, VQFS could find them where classical methods cannot. Third, the optimization landscape of quantum circuits has structural properties that differ from classical neural networks, and it is conceivable that these properties are favorable for the

sparse recovery problem underlying feature selection, as they appear to be for certain combinatorial optimization tasks [14].

None of these arguments constitutes a proof of advantage, and we are cautious about overclaiming. Establishing rigorous quantum advantage for feature selection will require both theoretical lower bounds on classical algorithms and experimental validation on hardware at a scale where simulation is infeasible.

6.4 Limitations

We want to be direct about what this work does not show.

Simulation is not hardware. Every result in this paper was obtained on a classical statevector simulator. The exponential cost of simulation means we could not test beyond four qubits without prohibitive runtimes. Any claim about quantum advantage must ultimately be validated on real quantum hardware, which we have not done.

Binary classification only. The current output layer averages Pauli-Z expectations and passes the result through a sigmoid, which is inherently a two-class construction. Extending VQFS to multi-class problems would require a different measurement strategy—for example, one expectation value per class, or a softmax over multiple circuit outputs.

Two hyperparameters to set. The regularization strength λ and the selection threshold τ both influence how many features are retained. We fixed them at $\lambda = 0.1$ and $\tau = 0.3$ across all datasets, which worked well here, but a practitioner applying VQFS to a new domain should budget time for cross-validation of these parameters.

7 Conclusion

This paper started from a simple observation: most quantum machine learning pipelines treat feature selection as someone else’s problem, applying it classically before the quantum circuit ever sees the data. We asked whether the circuit itself could learn which features to pay attention to, and VQFS is our answer to that question.

The core idea is to multiply each input feature by a trainable scalar weight before encoding it as a rotation angle, then penalize the L1 norm of those weights during training. The result is a system that jointly optimizes feature relevance and classification accuracy in a single gradient-based loop. On Breast Cancer and Wine, this approach outperformed five classical baselines on accuracy while retaining only 40–60% of the original features. The features VQFS selected on Breast Cancer—predominantly the “worst-case” nuclear measurements—match what domain experts consider most diagnostically relevant, which gives us some confidence that the learned weights are meaningful rather than arbitrary.

We are under no illusion that these results prove quantum advantage. The experiments run on a simulator, the datasets are small by modern standards, and the accuracy margins over the best classical methods are modest. What we do

claim is that VQFS is a coherent and practically usable framework that extends quantum machine learning into the feature selection stage of the pipeline, and that it works at least as well as standard classical alternatives on the problems we tested.

Looking ahead, the most important next steps are scaling to more qubits (which requires hardware access), extending the output encoding to handle multi-class problems, and developing a theoretical understanding of when quantum feature selection should be expected to outperform classical approaches. We hope this work provides a useful starting point for those investigations.

References

1. Abbas, A., Sutter, D., Zoufal, C., et al.: The power of quantum neural networks. *Nature Computational Science* **1**(6), 403–409 (2021)
2. Bergholm, V., Izaac, J., Schuld, M., et al.: PennyLane: Automatic differentiation of hybrid quantum-classical computations. arXiv preprint arXiv:1811.04968 (2018)
3. Bharti, K., Cervera-Lierta, A., Kyaw, T.H., et al.: Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics* **94**(1), 015004 (2022)
4. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. *Nature* **549**(7671), 195–202 (2017)
5. Breiman, L.: Random forests. *Machine Learning* **45**(1), 5–32 (2001)
6. Brown, G., Pocock, A., Zhao, M.J., Luján, M.: Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of Machine Learning Research* **13**, 27–66 (2012)
7. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing* **16**(5), 1190–1208 (1995)
8. Caro, M.C., Huang, H.Y., Cerezo, M., et al.: Generalization in quantum machine learning from few training data. *Nature Communications* **13**(1), 4919 (2022)
9. Cerezo, M., Arrasmith, A., Babbush, R., et al.: Variational quantum algorithms. *Nature Reviews Physics* **3**(9), 625–644 (2021)
10. Cerezo, M., Sone, A., Volkoff, T., Cincio, L., Coles, P.J.: Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications* **12**(1), 1791 (2021)
11. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. *Computers & Electrical Engineering* **40**(1), 16–28 (2014)
12. Cortes, C., Vapnik, V.: Support-vector networks. *Machine Learning* **20**(3), 273–297 (1995)
13. Dua, D., Graff, C.: UCI machine learning repository (2019), <http://archive.ics.uci.edu/ml>
14. Farhi, E., Goldstone, J., Gutmann, S.: A quantum approximate optimization algorithm. arXiv preprint arXiv:1411.4028 (2014)
15. Farhi, E., Neven, H.: Classification with quantum neural networks on near term processors. arXiv preprint arXiv:1802.06002 (2018)
16. Fontana, E., Cerezo, M., Holmes, Z., et al.: Non-trivial symmetries in quantum landscapes and their resilience to quantum noise. *Quantum* **6**, 804 (2022)
17. Fujii, K., Nakajima, K.: Harnessing disordered-ensemble quantum dynamics for machine learning. *Physical Review Applied* **8**(2), 024030 (2017)

18. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *Journal of Machine Learning Research* **3**, 1157–1182 (2003)
19. Guyon, I., Weston, J., Barnhill, S., Vapnik, V.: Gene selection for cancer classification using support vector machines. *Machine Learning* **46**(1), 389–422 (2002)
20. Hall, M.A., Smith, L.A.: Feature selection for machine learning: Comparing a correlation-based filter approach to the wrapper. *FLAIRS Conference* pp. 235–239 (1999)
21. Havlíček, V., Córcoles, A.D., Temme, K., et al.: Supervised learning with quantum-enhanced feature spaces. *Nature* **567**(7747), 209–212 (2019)
22. Huang, H.Y., Broughton, M., Cotler, J., et al.: Quantum advantage in learning from experiments. *Science* **376**(6598), 1182–1186 (2022)
23. Huang, H.Y., Broughton, M., Mohseni, M., et al.: Power of data in quantum machine learning. *Nature Communications* **12**(1), 2631 (2021)
24. Jolliffe, I.T., Cadima, J.: *Principal component analysis* (2016)
25. Kandala, A., Mezzacapo, A., Temme, K., et al.: Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **549**(7671), 242–246 (2017)
26. Kohavi, R., John, G.H.: Wrappers for feature subset selection. *Artificial Intelligence* **97**(1-2), 273–324 (1997)
27. Kübler, J.M., Arrasmith, A., Cincio, L., Coles, P.J.: An adaptive optimizer for measurement-frugal variational algorithms. *Quantum* **4**, 263 (2020)
28. Larocca, M., Czarnik, P., Sharma, K., et al.: Diagnosing barren plateaus with tools from quantum optimal control. *Quantum* **6**, 824 (2022)
29. LaRose, R., Coyle, B.: Robust data encodings for quantum classifiers. *Physical Review A* **102**(3), 032420 (2020)
30. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H.: Feature selection: A data perspective. *ACM Computing Surveys* **50**(6), 1–45 (2017)
31. Liu, Y., Arunachalam, S., Temme, K.: A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics* **17**(9), 1013–1017 (2021)
32. Louizos, C., Welling, M., Kingma, D.P.: Learning sparse neural networks through L0 regularization. *ICLR* (2018)
33. McClean, J.R., Boixo, S., Smelyanskiy, V.N., Babbush, R., Neven, H.: Barren plateaus in quantum neural network training landscapes. *Nature Communications* **9**(1), 4812 (2018)
34. Möttönen, M., Vartiainen, J.J., Bergholm, V., Salomaa, M.M.: Transformation of quantum states using uniformly controlled rotations. *Quantum Information & Computation* **5**(6), 467–473 (2005)
35. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(8), 1226–1238 (2005)
36. Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E., Latorre, J.I.: Data reuploading for a universal quantum classifier. *Quantum* **4**, 226 (2020)
37. Peruzzo, A., McClean, J., Shadbolt, P., et al.: A variational eigenvalue solver on a photonic quantum processor. *Nature Communications* **5**(1), 4213 (2014)
38. Preskill, J.: Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018)
39. Schuld, M., Bocharov, A., Svore, K.M., Wiebe, N.: Circuit-centric quantum classifiers. *Physical Review A* **101**(3), 032308 (2020)
40. Schuld, M., Killoran, N.: Quantum machine learning in feature Hilbert spaces. *Physical Review Letters* **122**(4), 040504 (2019)

41. Schuld, M., Petruccione, F.: Supervised learning with quantum computers. Springer (2018)
42. Schuld, M., Petruccione, F.: Machine learning with quantum computers. Springer (2021)
43. Street, W.N., Wolberg, W.H., Mangasarian, O.L.: Nuclear feature extraction for breast tumor diagnosis. *Biomedical Image Processing and Biomedical Visualization* **1905**, 861–870 (1993)
44. Thanasilp, S., Wang, S., Cerezo, M., Holmes, Z.: Exponential concentration and untrainability in quantum kernel methods. *arXiv preprint arXiv:2208.11060* (2022)
45. Tibshirani, R.: Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B* **58**(1), 267–288 (1996)
46. Wang, S., Fontana, E., Cerezo, M., et al.: Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications* **12**(1), 6961 (2021)
47. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B* **67**(2), 301–320 (2005)