Adaptive Quantum Kernel Learning with Alignment-Trained Feature Maps for High-Dimensional ECG Segment Classification

Data-dependent projections, bandwidth optimisation and noise-robust fidelity kernels evaluated on MIT-BIH-derived arrhythmia segments

[1]Chandrasekhar Panda, [2]Pravat Satpathy, [3]Sonia Mohapatra
[1]Researcher, [2]Researcher, [3]Researcher
[1,2,3]Research and Development Division,
Tekstein Solutions Private Limited, Bhubaneswar, Odisha, India
[1]contact@chandrasekharpanda.com, [2]analytics.pravat@gmail.com, [3]smctc98@gmail.com

***Abstract*—Quantum fidelity kernels embed classical data in the state space of a parameterised circuit and are attractive for small biomedical datasets, but the encoding is normally fixed in advance, forcing a hand-crafted dimensionality reduction and an arbitrary angle scaling, and the resulting kernel can concentrate towards a constant in a way that finite-shot estimation cannot resolve. This paper develops an adaptive quantum kernel for five-class (N, S, V, F, Q) classification of 10-s single-lead ECG segments in which a 77-dimensional feature vector, produced by a classical conditioning, QRS-detection and feature-extraction stage, is mapped to rotation angles through a trainable projection and per-qubit bandwidth and encoded by a layered Hadamard–rotation–ring-ZZ circuit. The projection, bandwidths and entangling strengths are learned by maximising the centred kernel–target alignment with gradient back-propagation through a state-vector simulator. Three properties are proved and numerically verified: the centred alignment is invariant to global depolarising noise, the expected squared Frobenius error of a finite-shot kernel estimate has a closed form bounded by $N^2/4M$, and an exact-kernel SVM on a globally depolarised Gram matrix coincides with a clean-kernel SVM whose box constraint is scaled by $(1 - p)$. On 500 segments derived from the MIT-BIH Arrhythmia Database under 3×5-fold stratified cross-validation, alignment training raises the training alignment from 0.54 to 0.96 and the accuracy from 89.4 % (fixed map) to 95.4 % at eight qubits, where it matches scalar bandwidth tuning, and exceeds bandwidth tuning by 3.4 points at four qubits; a tuned RBF-SVM on the same features remains higher at 97.9 %, so no quantum advantage is claimed. The adaptive kernel is, however, insensitive to sampling noise: with 128 shots per entry and depolarising strength 0.4 its accuracy stays at 94.9 %, whereas the bandwidth-tuned kernel loses up to 2.1 points and post-processing recovers only part of that, an effect explained by the within-/between-class kernel contrast that alignment training produces and by the strongly regularised SVM it supports. Limitations of segment-level evaluation and future hardware experiments are discussed.**

***Index Terms*—Quantum kernel methods, kernel–target alignment, quantum feature map, ECG arrhythmia classification, MIT-BIH, support vector machine, depolarising noise, shot noise, quantum machine learning, biomedical signal processing.**

I. Introduction

Automatic interpretation of the electrocardiogram (ECG) underpins ambulatory monitoring, intensive-care alarms and the growing class of wearable rhythm monitors. The standard beat taxonomy of ANSI/AAMI EC57 groups beats into non-ectopic (N), supraventricular ectopic (S), ventricular ectopic (V), fusion (F) and unclassifiable or paced (Q) classes [1], [2], and four decades of work on the MIT-BIH Arrhythmia Database [3] has produced a mature classical pipeline of filtering, QRS detection, morphological and rhythm features and a kernel or tree-based classifier [4], [5], later complemented by end-to-end convolutional networks [6], [7]. Within this landscape the practical limits are well known: the S and F classes are rare and morphologically close to N, inter-patient generalisation is far harder than intra-patient evaluation, and deep models need volumes of labelled data that single-institution studies rarely have.

Quantum kernel methods offer a different inductive bias for exactly this small-data regime. A feature map prepares a quantum state from a classical input and the kernel is the squared overlap of two such states [8], [9]; the classifier is a conventional support vector machine (SVM) over this kernel, so convexity and the generalisation theory of kernel machines are retained while the feature space itself is a Hilbert space of dimension 2^n for n qubits. Rigorous separations exist for contrived problems [10], and for generic data the kernel's usefulness is governed by how well its geometry matches the labels [11], [12]. Two obstacles stand between this promise and a biomedical signal task. First, the encoding must be chosen: a 10-s segment at 360 Hz has 3600 samples, hand-crafted features number in the tens, and a near-term circuit has single-digit qubits, so some projection is unavoidable and the standard choice of principal components followed by an arbitrary angle scale is rarely optimal [13], [14]. Second, fidelity kernels of random or deep circuits concentrate exponentially around a constant [15], and the shot noise of estimating an entry near zero with a few hundred measurements then swamps the signal, so a kernel that is

accurate in a state-vector simulator can be unusable on hardware [16], [17].

This paper addresses both obstacles together by making the feature map data-dependent and trained. The projection from the classical feature space to the encoding angles, the per-qubit bandwidth and the strength of the entangling layer are treated as parameters and optimised by kernel–target alignment [18], [19], an objective that measures how closely the Gram matrix resembles the ideal block structure of the labels and that can be back-propagated through a differentiable state-vector simulation. The contribution is deliberately evaluated against strong classical baselines on the same features, and the negative parts of the result — that a tuned radial-basis-function (RBF) SVM remains more accurate and that one regularisation variant did not help — are reported as found.

The main contributions are:

- An adaptive quantum feature map whose projection W, bandwidth vector λ and entangling strengths β are learned jointly by centred kernel–target alignment, initialised from a principal-component projection and an alignment-selected scalar bandwidth so that training starts from, and can only raise the alignment above, the two standard fixed maps.
- A complete, MATLAB-reproducible classical front end (zero-phase conditioning, energy-based QRS detection, 77 rhythm, morphology, spectral, statistical and wavelet features) whose detector is validated against the reference annotations of a raw MIT-BIH record.
- Three propositions, each verified numerically: invariance of centred alignment to global depolarising noise; a closed form and bound for the expected squared Frobenius error of a finite-shot kernel estimate; and the equivalence of an exact-kernel SVM under global depolarising noise to a clean-kernel SVM with a rescaled box constraint.
- A repeated stratified cross-validation study against linear, polynomial and RBF SVMs, a random forest and a one-dimensional convolutional network, with ablations over qubit count, circuit depth and the three adaptive components, and a noise study over depolarising strength, shot budget and two post-processing corrections, with the simulator cross-checked against Qiskit to 10^{-15}.

II. Related Work

Table I summarises the studies most directly relevant to this work; the text below groups them by theme.

Classical ECG classification. de Chazal et al. [4] established the inter-patient protocol on MIT-BIH with RR-interval and morphology features and linear discriminants, and most subsequent feature-based work, including wavelet and higher-order-statistic descriptors with SVMs [5], follows that template. Deep models trained on large private corpora reach cardiologist-level rhythm detection [6], and compact convolutional networks on beat windows [7] are now the usual deep baseline on MIT-BIH. These methods set the accuracy bar that any quantum method must be measured against on the same data.

Quantum kernels and their training. Havlíček et al. [8] and Schuld and Killoran [9] introduced the fidelity kernel and the ZZ-type feature map that is the default in Qiskit Machine Learning. Hubregtsen et al. [19] trained the parameters of an embedding circuit by kernel–target alignment on a trapped-ion device and proposed the diagonal-rescaling correction for depolarising noise that is adopted here. Shaydulin and Wild [13] showed that a scalar bandwidth on the encoding angles is decisive for generalisation, and Canatar et al. [14] gave the theoretical account. Thanasilp et al. [15] proved that kernels of expressive maps concentrate exponentially with qubit count, and Wang et al. [16] analysed how finite shots and noise then degrade the SVM. Kübler et al. [12] characterised the inductive bias of quantum kernels through the spectrum of the embedding, and Gentinetta et al. [20] bounded the shot complexity of training a quantum SVM. Glick et al. [21] obtained covariant kernels for data with group structure, and Suzuki et al. [22] used alignment as an indicator of noisy-versus-ideal kernel quality on IonQ hardware.

Quantum methods for ECG. Ozpolat and Karabatak [23] compared a quantum SVM with a classical SVM on a 12-lead rhythm dataset after PCA of clinical features and found the two comparable, with the quantum variant much slower; the present study differs in using a trainable map rather than a fixed one, AAMI beat classes from MIT-BIH-derived segments, a 77-dimensional feature vector and an explicit noise analysis. Variational quantum classifiers for ECG have also been reported, but they are not kernel machines and do not address kernel concentration.

Gap. No published ECG study trains the quantum feature map on the data, measures the resulting kernel's resistance to sampling noise, or reports the classical ceiling on identical features. Those three elements define the present work.

TABLE I
Summary of Related Work

Ref.	Year	Method / algorithm	Data	Advantage	Limitation
[4]	2004	RR + morphology features, linear discriminants	MIT-BIH, inter-patient	Defines the inter-patient protocol	Linear model; S-class recall low
[7]	2018	1-D CNN on beat windows, transfer learning	MIT-BIH, PTB	High intra-patient accuracy	Needs large labelled sets; class imbalance
[8]	2019	Fidelity kernel, ZZ feature map, QSVM	Synthetic	First hardware QSVM	Fixed map; no bandwidth control
[19]	2022	Alignment-trained embedding kernel	Synthetic / UCI	Trainable map; noise correction	Few qubits; no biomedical data
[13]	2022	Bandwidth-scaled fixed map	Multiple benchmarks	Restores generalisation at scale	Scalar scale only; fixed projection
[15]	2024	Theory of kernel concentration	—	Explains untrainability	No remedy beyond shallow maps

[23]	2023	QSVM vs SVM on PCA features	12-lead rhythm set	ECG application	Fixed map; no noise study; QSVM slower
This work	2026	Adaptive projection + bandwidth + entangling, KTA-trained, noise analysis	MIT-BIH-derived AAMI segments	Trainable map; noise-robust; classical ceiling reported	Segment-level CV; simulation only

III. Problem Formulation

Let $x_i \in \mathbb{R}^D$ be a feature vector computed from the i-th ECG segment and $y_i \in \{\mathrm{N}, \mathrm{S}, \mathrm{V}, \mathrm{F}, \mathrm{Q}\}$ its class, $i = 1, \ldots, N$. A quantum feature map $x \mapsto |\psi(x)\rangle$ on n qubits defines the kernel $k(x, x') = |\langle\psi(x')|\psi(x)\rangle|^2$, and an SVM over the Gram matrix $K_{ij} = k(x_i, x_j)$ provides the classifier. Four difficulties must be dealt with at the same time.

- *Dimensional mismatch.* $D = 77$ exceeds any near-term qubit budget, so a projection $\mathbb{R}^D \to \mathbb{R}^n$ must be chosen; principal components preserve variance, not class separation.
- *Angle scale.* The kernel of an angle-encoded map depends on the scale of the angles. Too small a scale yields a nearly constant kernel, too large a scale yields a nearly diagonal one; neither generalises [13].
- *Concentration and sampling.* On hardware each entry is a Bernoulli estimate from M shots. If the true entries are small and close together the estimate is dominated by sampling variance, and the Gram matrix may lose positive semidefiniteness [15], [16].
- *Honest comparison.* Any improvement must be measured against the best classical kernel on the identical features, with the same cross-validation folds and hyper-parameter search.

The task is therefore to learn the map parameters Θ so that the kernel is well aligned with the labels, to characterise what the resulting kernel loses under depolarising and shot noise, and to quantify the whole against classical and deep baselines.

IV. Proposed Methodology

A. System Architecture

Fig. 1 shows the three stages. A classical front end converts each 10-s segment into the feature vector. An adaptive encoder maps standardised features to rotation angles, a layered circuit prepares the state, and the fidelity kernel is formed from all pairs; an alignment optimiser adjusts the encoder and circuit parameters on the training fold. A noise model and optional post-processing sit between the kernel and the SVM so that the same trained map can be evaluated under ideal, depolarised and finite-shot conditions. The training loop is given in Fig. 2 and the circuit in Fig. 3.

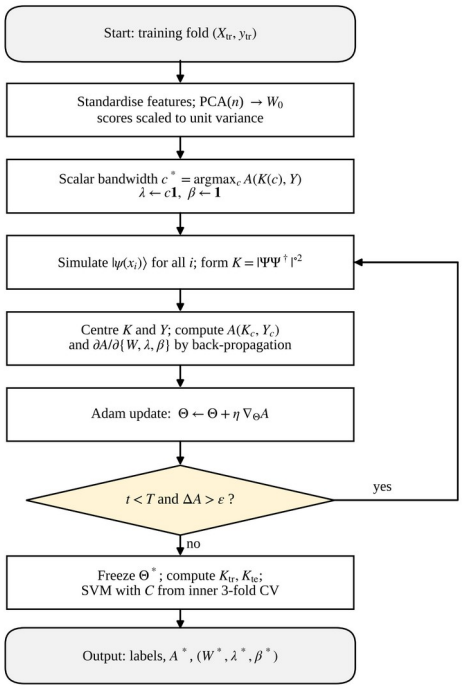


Fig. 2. Flowchart of alignment training and inference for one cross-validation fold.

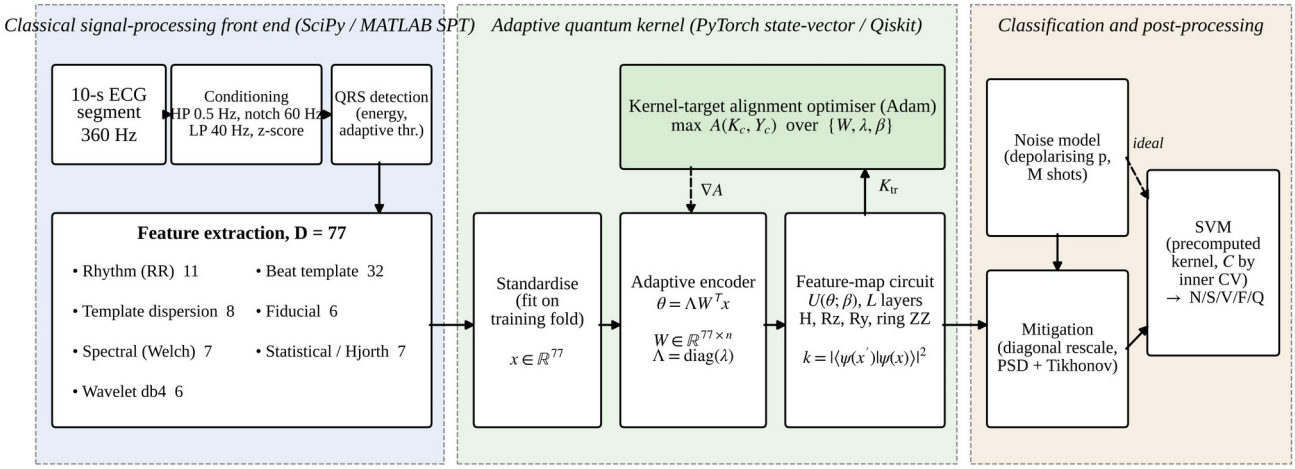


Fig. 1. System architecture. Left: classical conditioning, QRS detection and feature extraction ($D = 77$). Centre: adaptive encoder $\theta = \Lambda W^T x$, layered feature-map circuit and fidelity kernel, with the kernel–target alignment optimiser acting on $\{W, \lambda, \beta\}$. Right: depolarising and shot-noise model, mitigation and precomputed-kernel SVM.

B. Signal Conditioning and QRS Detection

Each segment x (3600 samples, 360 Hz) is conditioned by a cascade of zero-phase filters: a fourth-order Butterworth high-pass at 0.5 Hz for baseline wander, a second-order notch at 60 Hz (quality factor 30) for mains interference, and a fourth-order Butterworth low-pass at 40 Hz for muscle noise, followed by segment-wise standardisation,

(1)

where H_{hp}, H_{nt} and H_{lp} denote the forward–backward filters and $\mu_{\tilde{y}}$, $\sigma_{\tilde{y}}$ the mean and standard deviation of the filtered trace. Forward–backward application doubles the effective order and cancels group delay, so fiducial latencies are unaffected. QRS complexes are located by an energy detector in the Pan–Tompkins family [24]: a 5–15 Hz band-pass isolates the QRS band, the derivative is squared and integrated over a 150-ms window,

(2)

and local maxima of E_n separated by at least 250 ms are accepted when they exceed 0.35 times the median of the largest third of candidate peaks. Each accepted index is refined to the extremum of $|y|$ within ±50 ms. Detections inside a 100-ms guard band at either edge of the segment are discarded, because the zero-phase filters leave a start-up transient there; this guard was added after an early build in which such transients were occasionally detected as beats and corrupted the RR statistics, and the correction is reported rather than hidden. On the first 10 minutes of raw MIT-BIH record 100, processed in the same 10-s windows, the detector recovered 746 of 746 annotated beats outside the guard bands with 0 false positives (sensitivity 100.0 %, positive predictivity 100.0 %); 14 reference beats fell inside guard bands and are excluded by construction. Record 100 is a clean record, so this figure validates the implementation rather than its noise tolerance.

C. Feature Extraction

Seven feature groups are concatenated into $x \in \mathbb{R}^{77}$ (Table II). Rhythm features are computed from the RR series; for example the root-mean-square of successive differences is

(3)

with B detected beats. Morphology is captured by the mean beat template over a window of −250 to +400 ms around each R index, resampled to 32 points, together with its beat-to-beat standard deviation resampled to 8 points and six fiducial descriptors (R, Q and S amplitudes, a slope-based QRS width, T-wave amplitude and latency). Spectral features are relative Welch band powers,

(4)

over the bands 0.5–3, 3–8, 8–15, 15–25 and 25–40 Hz, plus the spectral centroid and the normalised spectral entropy H_{s}. Statistical features are skewness, kurtosis, histogram entropy, zero-crossing rate, crest factor and the Hjorth mobility and complexity (Hjorth activity is identically one after standardisation and is replaced by the crest factor). Wavelet features are the relative energies of a five-level Daubechies-4 decomposition,

(5)

where c_j are the approximation ($j = 0$) and detail coefficient vectors. All features are standardised with the mean and standard deviation of the training fold only.

TABLE II
Feature Groups and Dataset Summary

Group	Dim.	Content
Rhythm	11	mean, s.d., RMSSD, min, max, range/mean, pNN50-type ratio, heart rate, CV, beat count, median RR
Template	32	mean beat (−250 to +400 ms) resampled
Dispersion	8	beat-to-beat s.d. of the template, resampled
Fiducial	6	R, Q, S amplitudes; QRS width; T amplitude, latency
Spectral	7	five Welch band powers, centroid, spectral entropy
Statistical	7	skewness, kurtosis, histogram entropy, ZCR, crest factor, Hjorth mobility, complexity
Wavelet	6	db4 level-5 relative energies
Dataset: 500 segments × 10 s × 360 Hz, single lead; 100 per class N, S, V, F, Q; derived from MIT-BIH [3], [25]		

D. Adaptive Quantum Feature Map

The standardised vector x is mapped to n encoding angles through a trainable projection $W \in \mathbb{R}^{D^{*}\times^{*}n}$ and a positive bandwidth vector $\lambda \in \mathbb{R}^{n}$,

(6)

where the log-parametrisation ℓ keeps every bandwidth positive without constraints. The state is prepared by L layers of a Hadamard–rotation–entangling block,

(7)

(8)

in which $R_z(\theta) = \bigotimes_i R_z(\theta_i)$ and $R_y(\theta)$ similarly, the ring couples qubit i to qubit $i + 1 \bmod n$, and β_l is a trainable entangling strength per layer. The products in (7) and (8) are ordered products of operators. With $L = 2$ the data are re-uploaded, which raises the expressivity of the map [26] without additional qubits. The kernel is the fidelity

(9)

which on hardware is the probability of the all-zero outcome of the compute–uncompute circuit $U^{\dagger}(x')U(x)$. The parameter set is $\Theta = \{W, \ell, \beta\}$; the standard fixed map is recovered by freezing W to the leading n principal components (scores scaled to unit variance) with $\lambda = \mathbf{1}$ and $\beta = \mathbf{1}$, and the bandwidth-tuned map by $\lambda = c_{\mathrm{opt}}\mathbf{1}$ with a scalar c_{opt} chosen on the training fold. Both are used as baselines and as the initialisation of the adaptive map.

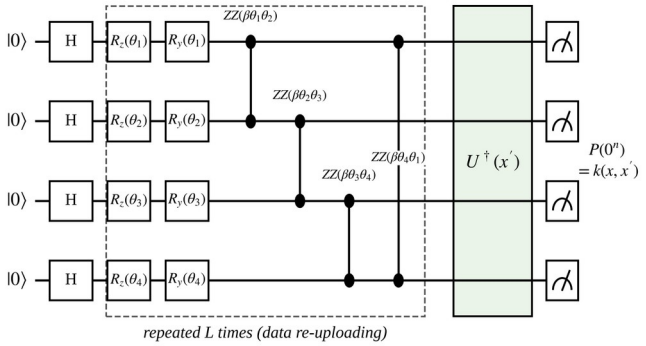


Fig. 3. Feature-map circuit for $n = 4$ (one layer shown; the dashed block is repeated L times). The compute–uncompute overlap circuit appends $U^\dagger(x')$ and the all-zero probability is the kernel entry.

E. Kernel–Target Alignment Objective

Let $Y_{ij} = 1$ if $y_i = y_j$ and 0 otherwise, and let $H = I - N^{-1}\mathbf{1}\mathbf{1}^{\mathrm{T}}$ be the centring matrix. The centred alignment [18] between the Gram matrix and the target is

(10)

which equals one when the kernel is a positive multiple of the centred label structure and is invariant to positive scaling and additive constants (a property used in Section V). The feature map is trained by

(11)

with $\mu = 0$ for the main method and $\mu > 0$ only in the regularised variant of Section VII-D. Because every gate is a differentiable function of Θ, the gradient of (11) is obtained by reverse-mode automatic differentiation through the state-vector simulation; for a single entry

(12)

and the chain rule through (10) is handled by the framework. On hardware the same gradient could be assembled from parameter-shift evaluations, at a cost of two circuit evaluations per parameter per entry; the simulator route is used here because the objective is to learn the map, not to measure it. Adam with step size 0.05 is run for 200 iterations on the full training Gram matrix ($N_{\mathrm{tr}} = 400$).

F. Classifier, Noise Model and Mitigation

With the trained map frozen, a one-versus-one SVM is fitted on the precomputed Gram matrix. For each binary sub-problem the dual is

(13)

with decision function $f(x) = \Sigma_i\ \alpha_i y_i k(x_i, x) + b$. The box constraint C is chosen from {0.1, 1, 10, 100} by an inner three-fold cross-validation on the training fold. Two noise mechanisms are modelled. A global depolarising channel of strength p acting on the compute–uncompute circuit maps the all-zero probability to

(14)

and finite sampling with M shots replaces $\tilde{k}$ by the Bernoulli mean

(15)

Section VII-A verifies (14) against a Qiskit Aer density-matrix simulation with gate-local depolarising errors. Two post-processing steps are evaluated. The diagonal of the ideal Gram matrix equals one, so the depolarising strength can be estimated from the measured diagonal and the affine map inverted [19],

(16)

and the negative eigenvalues produced by sampling are removed by projection onto the positive semidefinite cone followed by a Tikhonov floor, $\bar{K}^{+} = V \max(\Lambda_K, 0)V^{\mathrm{T}} + \tau I$ with $\tau = 10^{-3}$, where $\bar{K} = V\Lambda_K V^{\mathrm{T}}$. The cross-kernel between test and training points has no diagonal and is corrected with the $\hat{p}$ estimated on the training Gram matrix.

G. Algorithm and Complexity

Algorithm 1 summarises training and inference. Preparing N states costs $O(N\,L\,n\,2^n)$ on a state-vector simulator and the Gram matrix $O(N^2\,2^n)$; the alignment and its gradient add $O(N^2)$ per iteration, so training costs $O(T\,N^2\,2^n)$ for T iterations and is dominated by the Gram product for the sizes used here. On hardware, every Gram entry needs M shots of a depth-$O(L\ n)$ circuit, so a kernel costs $O(N^2\ M)$ circuit executions at training and $O(N_{\mathrm{sv}}\,M)$ per test point, where N_{sv} is the number of support vectors. Memory is $O(N\,2^n)$ complex amplitudes plus $O(N^2)$ for the Gram matrix. SVM training on a precomputed kernel is $O(N^2)$ to $O(N^3)$ [20].

ALGORITHM 1
Adaptive Quantum Kernel Training and Inference

Input: training set $\{(x_i, y_i)\}$, qubits n, layers L, iterations T, step η, grids $\mathscr{C}$ (bandwidth), $\{C\}$ (SVM)
Output: map parameters $\Theta^* = \{W, \ell, \beta\}$, SVM (α, b)
standardise features with training-fold statistics
$W_0 \leftarrow$ leading-n PCA loadings, scaled to unit-variance scores
$c_{\mathrm{opt}} \leftarrow \arg\max_{c^*c\in\mathscr{C}} A(K(W_0, c_{\mathrm{opt}}\mathbf{1}, \mathbf{1}), Y)$ // bandwidth baseline
$W \leftarrow W_0$; $\ell \leftarrow \ln(c_{\mathrm{opt}})\,\mathbf{1}$; $\beta \leftarrow \mathbf{1}$
for $t = 1 \ldots T$ **do**
  $|\psi_i\rangle \leftarrow$ Eq. (7) for all i; $K \leftarrow |\Psi\,\Psi^{\dagger}|^{\circ 2}$
  $A \leftarrow$ Eq. (10); compute $\nabla_\Theta A$ by back-propagation
  $\Theta \leftarrow \mathrm{Adam}(\Theta, -\nabla_\Theta A, \eta)$
end for
$K_{\mathrm{tr}} \leftarrow$ kernel of training set; $C \leftarrow$ inner 3-fold CV over $\{C\}$
fit one-vs-one SVM on K_{tr} // Eq. (13)
Inference: $K_{\mathrm{te}} \leftarrow$ kernel(test, train); optional noise model (14), (15); optional mitigation (16); predict

V. Theoretical Properties

Proposition 1 (alignment is invariant to global depolarising noise). *For any Gram matrix K, target Y and p* $\in [0, 1)$, *the centred alignment of the depolarised matrix* (14) *satisfies* $A(\tilde{K}, Y) = A(K, Y)$.

Proof. $\tilde{K} = (1 - p)K + p2^{-*n*}\mathbf{1}\mathbf{1}^{\mathrm{T}}$. Since $H\mathbf{1} = \mathbf{0}$, $H\tilde{K}H = (1 - p)HKH$, and (10) is invariant to positive scaling of K_c. ∎

The practical consequence is that a map trained by alignment in a noiseless simulator remains optimal, in the alignment sense, under any global depolarising strength; only sampling noise and the SVM margin are affected.

Proposition 2 (expected error of a finite-shot kernel estimate). *If every entry of the symmetric Gram matrix is estimated independently from M shots as in* (15), *then*

(17)

and the smallest eigenvalue of $\hat{K}$ *can lie below that of* $\tilde{K}$ *by at most* $\|\hat{K} - \tilde{K}\|_2$.

Proof. Each entry is an unbiased Bernoulli mean with variance $\tilde{k}_{ij}(1 - \tilde{k}_{ij})/M$; summing over the N^2 entries (the symmetric estimator counts each off-diagonal pair twice and the diagonal once) gives the equality, and $\tilde{k}(1 - \tilde{k}) \le 1/4$ the bound. The eigenvalue statement is Weyl's inequality for the symmetric perturbation $\hat{K} - \tilde{K}$. ∎

The bound is loose when entries are near 0 or 1, which is exactly the regime the alignment objective drives the kernel towards (Section VII-E); the PSD projection in (16) is justified by the eigenvalue statement.

Proposition 3 (depolarising noise rescales the SVM box constraint). *With exact ($M \to \infty$) kernels, the one-versus-one SVM trained on $\tilde{K}$ with box constraint C produces the same predictions as the SVM trained on K with box constraint $C(1 - p)$.*

Proof. Substituting (14) into (13), the constant term contributes $p2^{-*n^*}(\Sigma_i \alpha_i y_i)^2 = 0$ by the equality constraint. The remaining objective is $\Sigma\alpha_i - \tfrac{1}{2}(1 - p)\Sigma\alpha_i\alpha_j y_i y_j K_{ij}$; with $\alpha' = (1 - p)\alpha$ it equals $(1 - p)^{-1}$ times the clean objective with box $0 \le \alpha'_i \le C(1 - p)$. The decision function $\Sigma\alpha_i y_i \tilde{k}(x_i, x) + \tilde{b} = \Sigma\alpha'_i y_i k(x_i, x) + \tilde{b}$, with the constant again vanishing, and the bias is fixed by the same support-vector conditions. ∎

Hence, in the absence of sampling noise, the diagonal-rescaling correction in (16) cannot change the classifier beyond what a different C would, and its value lies entirely in restoring the scale of noisy entries before the PSD projection. All three propositions are checked numerically in Section VII.

VI. Simulation Setup

Software. Python 3 with NumPy 2.5, SciPy 1.18, scikit-learn 1.9, PyWavelets 1.8, PyTorch 2.14 (CPU), Qiskit 2.6 [27], Qiskit Aer 0.18 and Qiskit Machine Learning 0.9. The signal-processing stage uses SciPy designs with one-to-one MATLAB Signal Processing Toolbox equivalents (butter, iirnotch, filtfilt, pwelch, findpeaks, wavedec), recorded in the source; MATLAB itself was not available in the execution environment, which is stated plainly. The state-vector kernel and its gradient are implemented in PyTorch; Qiskit is used for independent verification (Statevector, FidelityQuantumKernel) and for the Aer noise simulation. *Hardware.* Two CPU cores and 8 GB of memory; no quantum hardware was used.

Dataset. PhysioNet could not be reached from the execution environment, so the raw 48-record database was unavailable. The study uses a publicly redistributed compilation [25] of 500 non-overlapping 10-s single-lead segments extracted from MIT-BIH at 360 Hz, 100 per AAMI class, in which each segment is reported by its curators to contain beats of a single class. Three consequences follow and are treated as limitations rather than hidden: the compilation carries no patient identifiers, so patient-wise (inter-patient) splitting is impossible and the cross-validation is segment-level; the class purity of segments could not be re-verified against beat annotations; and the sample is small (N = 500). One raw record (100) was available through the WFDB package distribution and is used only to validate the QRS detector.

Protocol. Stratified five-fold cross-validation repeated three times (15 test folds of 100 segments) with seeds 0, 1, 2; standardisation, PCA initialisation, bandwidth selection, alignment training and SVM hyper-parameter search are all performed inside the training fold. The qubit/depth sweep and the noise study use a single five-fold repeat. The noise study draws three independent shot-noise realisations per fold and reuses the C selected in the noiseless run so that only the kernel changes. All random seeds are fixed and listed in the reproducibility bundle. Table III lists the parameters.

TABLE III
Simulation Parameters

Parameter	Value
Sampling rate / segment length	360 Hz / 3600 samples (10 s)
Conditioning	HP 0.5 Hz (4th-order Butterworth), notch 60 Hz (Q = 30), LP 40 Hz (4th-order), zero-phase
QRS detector	5–15 Hz band-pass, 150-ms integration, refractory 250 ms, threshold 0.35 × median of top third, 100-ms edge guard
Feature dimension D	77 (Table II)
Qubits n / layers L (main)	8 / 2; sweep n ∈ {4, 6, 8}, L ∈ {1, 2}
Entangling topology	ring ZZ, trainable β_l
Bandwidth grid $\mathscr{C}$	{0.1, 0.2, 0.3, 0.5, 0.7, 1.0, 1.5}
Alignment optimiser	Adam, η = 0.05, T = 200, full batch (400)
Regularised variant	80/20 split, early stopping (patience 20), μ ∈ {0, 0.03, 0.3} by validation alignment
SVM	one-vs-one, precomputed kernel, C ∈ {0.1, 1, 10, 100} (inner 3-fold)
Classical baselines	linear, poly-3, RBF (γ ∈ {scale, 0.003, 0.01, 0.03}) SVMs; random forest (300 trees); 1-D CNN (4 conv blocks, 16–64 channels, 180 Hz input, Adam 10^{-3}, early stopping)
Noise study	p ∈ {0, 0.1, 0.2, 0.3, 0.4}; M ∈ {128, 512, 2048, 8192, ∞}; 3 realisations; $\tau = 10^{-3}$
Cross-validation	stratified 5-fold × 3 repeats; seeds 0–2



Fig. 4. Representative conditioned segments of the five classes with detected R indices (triangles). The V and F examples show the polarity effect discussed in Section VIII.

VII. Results and Discussion

A. Verification of the Simulator and the Noise Model

For random maps at $(n, L) = (4, 1)$, (6, 2) and (8, 2) the PyTorch kernel agrees with Qiskit's exact Statevector overlap to a maximum absolute difference of 2.0e-15, 2.7e-15 and 3.8e-15 respectively, and with Qiskit Machine Learning's sampler-based FidelityQuantumKernel (2×10^5 shots) to 1.7e-3, 1.4e-3 and 4.3e-4, which is the expected sampling error. The same random maps show the concentration predicted in [15]: the mean off-diagonal entry falls from 0.246 to 0.006 as the map grows, which is why the bandwidth and the alignment objective matter.

The global depolarising model (14) was tested against an Aer density-matrix simulation of the four-qubit, one-layer compute–uncompute circuit with depolarising errors after every gate (one-qubit error p_1, two-qubit error p_2) on a 16×16 Gram matrix. For $(p_1, p_2) = (0.002, 0.01)$, (0.005, 0.03), (0.01, 0.06) the least-squares global strength was $p = 0.085$, 0.224, 0.398, the diagonal estimate (16) gave 0.088, 0.231, 0.409, and the model explained the gate-local result with $R^2 = 0.9999$, 0.9994, 0.9974 and maximum residual 0.006, 0.015, 0.024. The diagonal estimator overshoots by about 3 % of p because the compute–uncompute circuit has depth-dependent, not strictly global, errors; the residual grows with noise strength and is the reason the model is described as an approximation.

B. Alignment Training

Fig. 5(a) shows a representative fold. The fixed map has an alignment of 0.54 ± 0.01 on the training fold (mean ± s.d. over 15 folds); selecting the scalar bandwidth raises it to 0.68 ± 0.01, with $c_{\text{opt}} = 0.3$ chosen in every fold, and Fig. 5(b) shows the single, well-defined maximum of the bandwidth curve together with the collapse of the mean off-diagonal entry at large c. Training W, ℓ and β jointly then reaches 0.96 ± 0.01 within 200 iterations, most of the gain arriving in the first 30. Fig. 5(c)–(d) show the effect on the Gram matrix of one fold with rows sorted by class: the bandwidth kernel has strong N and Q blocks but diffuse S, V and F blocks, whereas the adaptive kernel is close to block-diagonal with a mean within-class entry of 0.75 against 0.03 between classes, compared with 0.50 and 0.10 for the bandwidth kernel and 0.19 and 0.02 for the fixed map (five-fold means). The trained bandwidths are not uniform across qubits and the entangling strengths move away from one, which is the data-dependence the method is designed to supply.

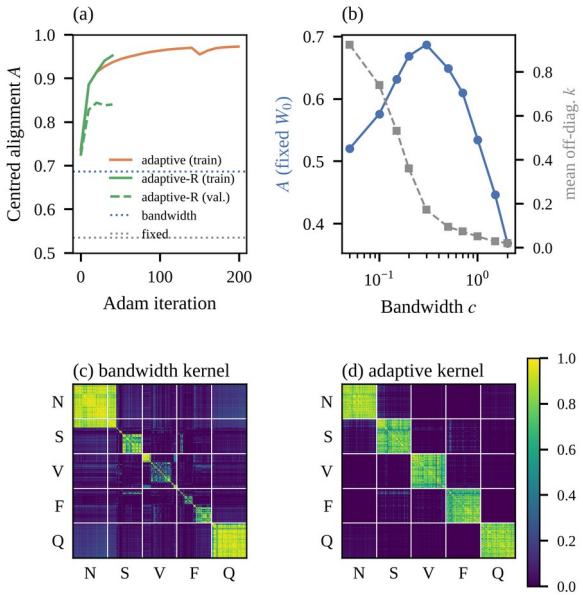


Fig. 5. Alignment training (rep. 0, fold 0, $n = 8$, $L = 2$). (a) Centred alignment against Adam iteration for the adaptive map (training set), the regularised variant (training and 20 % validation split) and the two fixed baselines. (b) Alignment and mean off-diagonal entry of the fixed-projection kernel as a function of the scalar bandwidth c. (c), (d) Training Gram matrices of the bandwidth and adaptive kernels with rows and columns sorted by class.

C. Classification Performance

Table IV and Fig. 6(a) give the main comparison. Among the quantum kernels the fixed map is clearly the weakest (89.4 ± 2.6 %); bandwidth selection alone adds 6.1 points and wins in 15 of 15 folds (Wilcoxon $p = 0.0006$); the adaptive map reaches 95.4 ± 1.5 % and macro-F1 95.4 ± 1.5 %, statistically indistinguishable from the bandwidth kernel at this qubit count (6 wins, 3 ties, 6 losses). The per-class recalls in Table IV show that all kernels classify N and Q almost perfectly and that the errors concentrate in the S–F and S–N confusions, as in the classical literature [4]; the confusion matrix of the adaptive kernel over the 15 folds is in Fig. 6(b).

The classical ceiling on the same 77 features is higher: the RBF-SVM reaches 97.9 ± 1.8 % (macro-F1 97.9 ± 1.8 %) and beats the adaptive kernel in 14 of 15 folds (Wilcoxon $p = 0.0007$); the linear and polynomial SVMs and the random forest are all within a point of it. The features are therefore nearly linearly separable at the segment level, and an eight-qubit kernel built on an eight-dimensional projection of them cannot be expected to exceed a 77-dimensional RBF kernel. No quantum advantage is claimed. The one-dimensional CNN trained on the raw conditioned waveform is the weakest baseline at 93.5 ± 2.3 %, as expected for 400 training segments, with early stopping after 24–60 epochs.

Two details of the selected hyper-parameters are informative. The inner search chose C = 0.1 in 12, C = 1.0 in 3 folds for the adaptive kernel but C = 10.0 in 11, C = 100.0 in 4 folds for the bandwidth kernel and C = 1.0 in 6, C = 10.0 in 8, C = 100.0 in 1 folds for the fixed map: the aligned kernel supports a wide-margin classifier with strong regularisation, which anticipates its behaviour under noise. The RBF search chose $\gamma \in \{0.003, 0.01\}$ or the scikit-learn default $(1/D)$ and $C \in \{10, 100\}$.

TABLE IV
Cross-Validation Results (Mean ± s.d. over 15 Folds; Recall per Class in %; Training Time per Fold)

Method	Accuracy (%)	Macro-F1 (%)	N	S	V	F	Q	Train (s)	Align. A
Linear SVM	96.9 ± 2.2	96.9 ± 2.2	10 0.0	96. 3	96. 3	92. 0	10 0.0	0.1	—
Poly-3 SVM	97.3 ± 1.8	97.3 ± 1.8	99. 0	96. 3	96. 7	94. 7	10 0.0	0.1	—
RBF SVM	97.9 ± 1.8	97.9 ± 1.8	10 0.0	97. 7	96. 0	95. 7	10 0.0	0.3	—
Random forest	97.2 ± 1.4	97.2 ± 1.4	99. 7	97. 0	97. 7	91. 7	10 0.0	0.6	—
1-D CNN (raw)	93.5 ± 2.3	93.4 ± 2.4	95. 3	81. 3	98. 0	93. 0	10 0.0	8.8	—
QK fixed (PCA, λ = 1)	89.4 ± 2.6	89.5 ± 2.5	97. 3	80. 3	87. 3	85. 0	97. 0	<0.1	0.54
QK bandwidth (λ = c*1)	95.5 ± 1.6	95.5 ± 1.6	99. 7	90. 7	96. 7	90. 7	10 0.0	0.3	0.68
QK adaptive-Λ (λ, β only)	94.7 ± 2.3	94.6 ± 2.3	10 0.0	90. 0	93. 3	90. 3	99. 7	23.6	0.71
QK adaptive (proposed)	95.4 ± 1.5	95.4 ± 1.5	10 0.0	89. 0	95. 3	92. 7	10 0.0	24.4	0.96
QK adaptive-R (regularised)	94.9 ± 1.8	94.8 ± 1.9	99. 3	87. 3	95. 3	92. 3	10 0.0	16.9	0.93

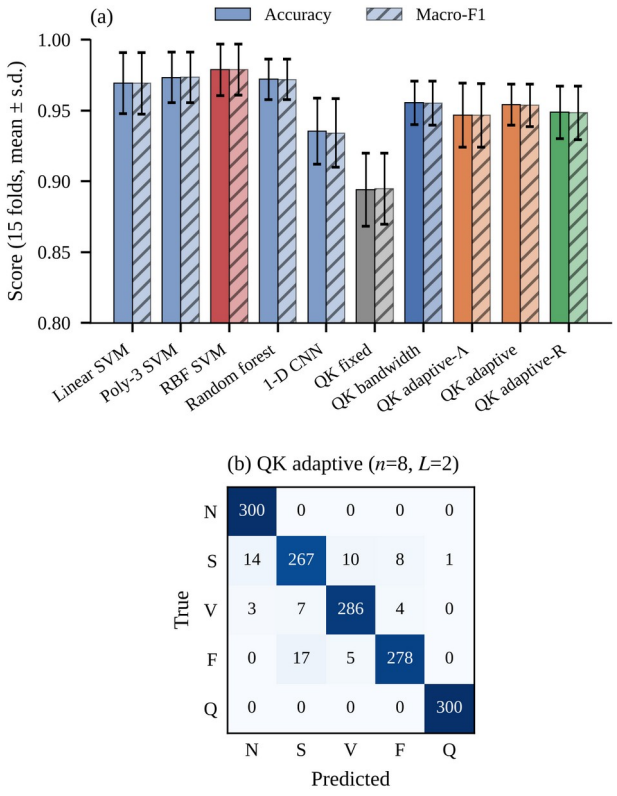


Fig. 6. Classification results. (a) Accuracy and macro-F1 over 15 folds for all methods. (b) Confusion matrix of the adaptive kernel ($n = 8$, $L = 2$) summed over the 15 test folds.

D. Ablation over Adaptive Components, Qubits and Depth

Training only the bandwidths and entangling strengths with W frozen to the PCA loadings (adaptive-Λ) raises the alignment slightly (0.71) but lowers the accuracy to 94.7 ± 2.3 %: per-qubit scaling of an uninformative projection sharpens the kernel along directions that separate variance, not classes. The gain of the full method therefore comes from learning W. The regularised variant, which holds out 20 % of the training fold to stop training on validation alignment and selects a drift penalty μ (chosen as 0 in 2, 0.3 in 10, 0.03 in 3 folds; 31–101 iterations), was introduced after two early folds showed the plain method trailing the bandwidth kernel, on the hypothesis that a 77×8 projection over-fits 400 labels. The hypothesis was not borne out at eight qubits: adaptive-R scores 94.9 ± 1.8 %, 0.5 points below the plain method (7 losses in 15), and its validation alignment (dashed line in Fig. 5(a)) plateaus at about 0.85 while the training alignment keeps rising, so the gap between training and validation alignment is real but does not translate into test error. The plain method is retained as the proposed one and the negative result is kept.

The qubit/depth sweep (Table V, Fig. 7) shows where the learned projection matters. With four qubits the fixed and bandwidth maps are limited to 91.2 and 91.6 % ($L = 2$), whereas the adaptive map reaches 95.0 % and the regularised variant 95.2 % — a gain of 3.4 points over bandwidth tuning at $L = 2$ and 1.8 points at $L = 1$. At six qubits the bandwidth kernel has caught up (94.8 % against 94.4 % at $L = 2$) and at eight the two are equal within the fold-to-fold scatter, with the regularised variant matching the bandwidth kernel at $(n, L) = (8, 1)$ and $(6, 2)$ but not elsewhere. The explanation is that the leading eight principal components already carry 88.9 % of the feature variance and, once the bandwidth is tuned, enough class information; with four components they do not, and a learned projection recovers the missing directions. The second layer hurts the fixed map (92.6 → 89.0 % at eight qubits) because re-uploading an unscaled encoding deepens the concentration, and leaves the tuned kernels essentially unchanged. For a qubit-limited device the adaptive projection is the component that matters; for a simulator with qubits to spare a tuned bandwidth is almost sufficient.

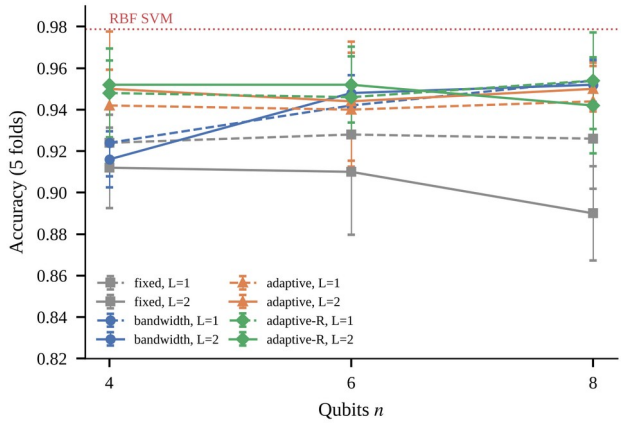


Fig. 7. Qubit/depth sweep for the four quantum kernels (five folds, mean ± s.d.), with the RBF-SVM mean as reference.

TABLE V
Qubit and Depth Sweep: Accuracy (%) over Five Folds (Mean ± s.d.)

n	L	QK fixed	QK bandwidth	QK adaptive	QK adaptive-R
4	1	92.4 ± 1.4	92.4 ± 1.6	94.2 ± 1.7	94.8 ± 2.1
4	2	91.2 ± 1.9	91.6 ± 1.4	95.0 ± 2.8	95.2 ± 1.2
6	1	92.8 ± 1.7	94.2 ± 1.5	94.0 ± 2.8	94.6 ± 2.0

6	2	91.0 ± 3.0	94.8 ± 0.4	94.4 ± 2.9	95.2 ± 1.8
8	1	92.6 ± 2.4	95.4 ± 1.0	94.4 ± 1.9	95.4 ± 2.3
8	2	89.0 ± 2.3	95.2 ± 1.2	95.0 ± 1.1	94.2 ± 2.3

E. Robustness to Depolarising and Shot Noise

Fig. 8 and Table VI report the noise study at $(n, L) = (8, 2)$. Proposition 1 is confirmed to machine precision: the alignment of the depolarised training kernel differs from the clean value by at most 4.9e-15 for p up to 0.4 in every fold and kernel. Proposition 3 is confirmed exactly: for $p \in \{0.1, 0.3\}$ and $C \in \{1, 10\}$ the SVM on the depolarised exact kernel and the SVM on the clean kernel with $C(1 - p)$ made identical predictions on every test segment of every fold (agreement 100 %). Consistent with this, at infinite shots the accuracy depends on p only through the effective box constraint: it moves by 1.4 points for the fixed map, whose selected C is small enough for the rescaling to matter, and by 0.2 and 0.0 points for the bandwidth and adaptive kernels.

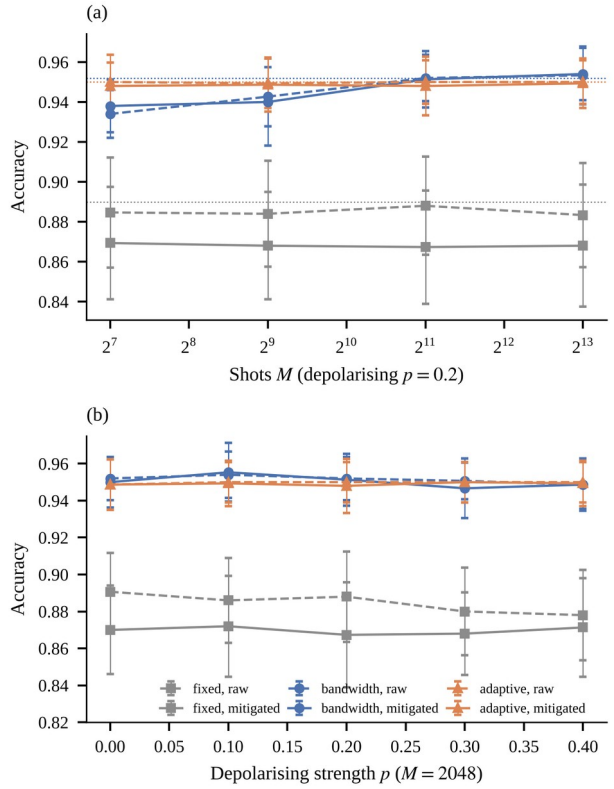


Fig. 8. Noise robustness ($n = 8$, $L = 2$; five folds × three realisations). (a) Accuracy against shot budget at depolarising strength $p = 0.2$ for the fixed, bandwidth and adaptive kernels with and without mitigation; dotted lines mark the infinite-shot values. (b) Accuracy against p at 2048 shots.

Finite sampling is where the kernels separate, and not in the way a concentration argument alone would predict. Fig. 8(a) plots accuracy against the shot budget at $p = 0.2$. The fixed map is poor but almost insensitive to shots (88.5 % at 128 shots against 88.4 % at infinite shots): its entries lie near zero, where the Bernoulli variance $\tilde{k}(1 - \tilde{k})/M$ in (15) is small, so sampling perturbs little. Post-processing hurts it at every shot budget (86.9 % at 128 shots), because dividing small entries by $(1 - \hat{p})$ and adding the Tikhonov floor both distort a kernel whose off-diagonal structure is itself of order 10^{-2}. The bandwidth kernel is the one that degrades: from 95.2 % at infinite shots to 93.4 ± 1.2 % at 128 shots (93.3 % at $p = 0.4$), with mitigation recovering 0.4 points at $p = 0.2$ and costing 1.6 points at $p = 0.4$, where the diagonal estimate of $\hat{p}$ from 128 shots is itself too noisy. Its entries are mid-range (within-class mean 0.50, between-class 0.10), exactly where the sampling variance peaks, and the SVM it supports selects $C = 10$ or 100, so small perturbations move the decision boundary. The adaptive kernel is flat: 95.0 ± 1.4 % at 128 shots against 95.0 % at infinite shots, with or without mitigation, and Fig. 8(b) shows the same insensitivity to p up to 0.4 at 2048 shots. Two properties act together: alignment training pushes within-class entries to about 0.75 and between-class entries to about 0.03, so the sampling standard deviation at 128 shots (at most 0.044) is an order of magnitude below the gap, and the aligned Gram matrix supports a strongly regularised SVM ($C = 0.1$ in 12 of 15 folds) whose decision function averages over many support vectors and is correspondingly insensitive to entry-wise noise.

Proposition 2 is confirmed in Fig. 9: the empirical squared Frobenius error of the shot estimate matches the closed form to within 0.2 % on average (ratio 0.980–1.024) and lies far below the $N^2/4M$ bound because most entries of the aligned kernel are near 0 or 1. The minimum eigenvalue of the raw 400 × 400 estimate is -0.85 at 128 shots and -0.10 at 8192, so the raw estimate is indefinite and the raw-mode results rely on the SVM solver's tolerance of mild indefiniteness rather than on positive semidefiniteness; the PSD projection removes this without changing the adaptive kernel's accuracy.

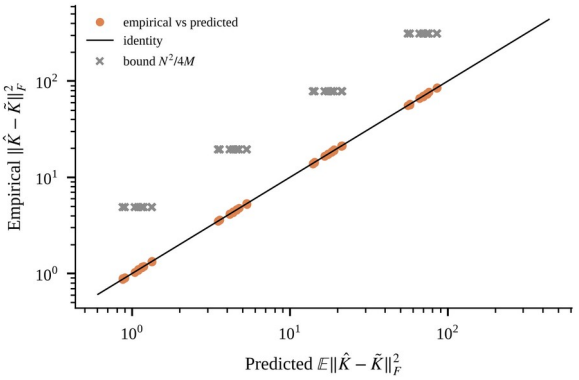


Fig. 9. Numerical check of Proposition 2: empirical against predicted expected squared Frobenius error of the finite-shot kernel estimate (adaptive kernel, $p \in \{0, 0.2\}$, $M \in \{128, 512, 2048, 8192\}$), with the $N^2/4M$ bound.

TABLE VI
Accuracy (%) under Depolarising Strength p and Shot Budget M (Five Folds × Three Realisations; M = 512 in Fig. 8)

Kernel	Mode	p	M = 128	M = 2048	M = 8192	M = ∞
QK fixed	raw	0.0	89.0 ± 2.5	89.1 ± 2.1	88.9 ± 2.4	89.0 ± 2.3
QK fixed	raw	0.2	88.5 ± 2.8	88.8 ± 2.5	88.3 ± 2.6	88.4 ± 2.2
QK fixed	raw	0.4	87.5 ± 3.2	87.8 ± 2.5	87.7 ± 2.5	87.6 ± 2.7
QK fixed	mitigated	0.0	87.7 ± 3.6	87.0 ± 2.4	86.6 ± 3.0	86.6 ± 2.8
QK fixed	mitigated	0.2	86.9 ± 2.8	86.7 ± 2.8	86.8 ± 3.1	86.6 ± 2.8

QK fixed	mitigated	0.4	86.3 ± 2.9	87.1 ± 2.7	86.9 ± 2.8	86.6 ± 2.8
QK bandwidth	raw	0.0	94.2 ± 1.6	95.2 ± 1.2	95.3 ± 1.5	95.2 ± 1.2
QK bandwidth	raw	0.2	93.4 ± 1.2	95.2 ± 1.2	95.3 ± 1.4	95.2 ± 1.5
QK bandwidth	raw	0.4	93.3 ± 1.9	94.9 ± 1.3	95.1 ± 0.8	95.4 ± 1.2
QK bandwidth	mitigated	0.0	94.8 ± 1.7	95.0 ± 1.4	95.1 ± 1.5	94.8 ± 1.2
QK bandwidth	mitigated	0.2	93.8 ± 1.3	95.1 ± 1.4	95.4 ± 1.3	94.8 ± 1.2
QK bandwidth	mitigated	0.4	91.7 ± 2.2	94.9 ± 1.4	95.0 ± 1.4	94.8 ± 1.2
QK adaptive	raw	0.0	94.9 ± 1.2	94.9 ± 1.4	94.9 ± 1.2	95.0 ± 1.1
QK adaptive	raw	0.2	95.0 ± 1.4	95.0 ± 1.1	95.0 ± 1.1	95.0 ± 1.1
QK adaptive	raw	0.4	94.9 ± 1.4	95.0 ± 1.1	95.0 ± 1.1	95.0 ± 1.1
QK adaptive	mitigated	0.0	95.0 ± 1.3	94.9 ± 1.4	94.9 ± 1.2	95.0 ± 1.1
QK adaptive	mitigated	0.2	94.8 ± 1.2	94.8 ± 1.5	94.9 ± 1.2	95.0 ± 1.1
QK adaptive	mitigated	0.4	94.9 ± 1.4	94.9 ± 1.2	95.0 ± 1.1	95.0 ± 1.1

F. Computational Cost

Feature extraction costs 4.6 ms per segment on one core. Fixed and bandwidth kernels are evaluated in well under a second per fold; alignment training of the adaptive map takes 24 s per fold at eight qubits (200 iterations, 400 × 256 complex amplitudes) and the SVM on the precomputed kernel a few milliseconds, so the whole quantum pipeline is an order of magnitude cheaper than the CNN (9 s) on the same hardware but two orders slower than the RBF-SVM. On a device the same training run would need $2|\Theta|$ parameter-shift circuits per Gram entry per iteration, which is impractical for $|\Theta| = 626$; the intended deployment is simulator-side training followed by hardware inference, for which the shot study is the relevant cost model: at 128 shots per entry and 400 support-vector candidates, a test segment costs 5×10^4 circuit executions.

VIII. Comparative Analysis, Key Outcomes and Limitations

Table VII places the method alongside the closest published alternatives on the axes required of an engineering comparison. Its accuracy is below the strongest classical kernel on the same features and in the same folds, which is also the finding of [23] for a fixed-map QSVM; its distinguishing properties are the 6-point advantage over the fixed map, the 3.4-point advantage over bandwidth tuning at four qubits, and a shot-noise tolerance that the bandwidth kernel does not have. Cost is dominated by simulator-side training; reliability under depolarising noise is provably unaffected in the alignment sense (Proposition 1); scalability in qubits is limited by the 2^n simulator cost but, since the projection is learned, the method degrades gracefully as qubits are removed rather than added.

TABLE VII
Comparative Analysis with Representative Approaches

Criterion	RBF-SVM on features [4], [5]	1-D CNN [7]	Fixed-map QSVM [8], [23]	Bandwidth QK [13]	Adaptive QK (this work)
Accuracy (this data, n = 8)	97.9 ± 1.8 %	93.5 ± 2.3 %	89.4 ± 2.6 %	95.5 ± 1.6 %	95.4 ± 1.5 %
Accuracy at n = 4 (L = 2)	—	—	91.2 %	91.6 %	95.0 %
Training cost	seconds	minutes (GPU-friendly)	none (kernel only)	grid over c	~24 s simulator
Complexity	$O(N^2D)$	O(epochs·N·T·ch)	$O(N^2M)$ circuits	$O(N^2M\cdot\lvert\mathcal{C}\rvert)$	$O(TN^2 2^n)$ train, $O(N^2M)$ infer
Robustness to shot noise	n/a	n/a	insensitive but low accuracy	loses up to 2 points at 128 shots	flat to 128 shots (large contrast, small C)
Reliability under depolarising	n/a	n/a	C-rescaling only	C-rescaling only	alignment invariant (Prop. 1)
Scalability	excellent	needs data	degrades with n (concentration)	degrades below n = 8	degrades gracefully with fewer qubits
Interpretability	moderate	low	low	low	learned W, λ inspectable

Key outcomes. (i) A trainable encoder that learns the projection, bandwidth and entangling strength by alignment, initialised so that it dominates the standard fixed and bandwidth maps; (ii) a measured, not assumed, accounting of where it helps (few qubits; sampling noise) and where it does not (eight qubits in a noiseless simulator; against an RBF-SVM); (iii) three propositions that explain the noise results and show that, without sampling noise, diagonal rescaling is equivalent to a change of C; (iv) a reproducible pipeline whose classical stage can be re-implemented in MATLAB and whose quantum stage has been cross-checked against Qiskit. For engineering practice the result argues for alignment-trained maps whenever kernel entries must be estimated from a small shot budget, and for a classical kernel whenever they need not be.

Limitations and threats to validity. The evaluation is segment-level on a third-party compilation without patient identifiers, so segments from the same subject can appear in training and test folds; the accuracies of all methods, classical and quantum, are therefore intra-patient figures and are expected to fall substantially under the inter-patient protocol of [4], which could not be run. The compilation's class purity was not re-verified, and the representative traces in Fig. 4 show that some V and F segments contain beats whose polarity causes the detector to select the S-trough rather than the R-peak; this is consistent within a segment but mixes

fiducials across beats and is one reason the dispersion features carry information. The sample is small (500), the QRS validation used a clean record, the noise model is global depolarising with independent shot noise rather than a calibrated device model, and no hardware execution was performed. The QRS-width descriptor is slope-based and underestimates clinical QRS duration. Finally, the superiority of RBF-SVM on these features means that any claim for the quantum kernel rests on the noise and low-qubit results, not on peak accuracy.

IX. Conclusion and Future Scope

This paper has shown that the feature map of a quantum fidelity kernel can be learned from ECG data rather than fixed in advance, and has quantified what that buys. On 500 MIT-BIH-derived segments, joint alignment training of the projection, bandwidths and entangling strengths raised the training alignment from 0.54 to 0.96 and the five-class accuracy from 89.4 % to 95.4 % at eight qubits, matched the bandwidth-tuned kernel there and exceeded it by 3.4 points at four qubits, while remaining 2.5 points below a tuned RBF-SVM on identical features. The aligned kernel's large within-/between-class contrast, together with the strongly regularised SVM it supports, made it insensitive to sampling noise down to 128 shots and to depolarising strengths up to 0.4, where the bandwidth kernel lost up to 2.1 points, and three propositions — alignment invariance to global depolarising noise, the closed-form shot-noise error, and the C-rescaling equivalence — were proved and verified. A validation-based regularisation variant did not improve the plain method and is reported as such.

Future work follows from the limitations. Inter-patient evaluation on the full 48-record database with beat-level annotations is the first requirement, together with class-imbalanced training that reflects the clinical prevalence of S and F beats. Hardware inference on a trapped-ion or superconducting device with a calibrated noise model would test whether the shot-noise tolerance measured here survives correlated errors, and parameter-shift or SPSA training of a reduced parameter set would test whether the map can be adapted on the device itself. Encoding the beat template directly with amplitude or re-uploading schemes, multi-lead extension, and alignment objectives that weight the rare classes are natural extensions. Finally, a theoretical account of why alignment-trained maps resist concentration — the empirical contrast reported here suggests a connection to the spectrum of the trained embedding — would turn an observation into a design rule.

Acknowledgment

The authors thank the curators of the MIT-BIH Arrhythmia Database and of the segment-level compilation used in this study. [Sponsor acknowledgments, if any, to be inserted.]

References

[1] ANSI/AAMI EC57:2012, Testing and Reporting Performance Results of Cardiac Rhythm and ST Segment Measurement Algorithms. Arlington, VA, USA: Association for the Advancement of Medical Instrumentation, 2012.

[2] A. L. Goldberger, L. A. N. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals,” Circulation, vol. 101, no. 23, pp. e215–e220, Jun. 2000.

[3] G. B. Moody and R. G. Mark, “The impact of the MIT-BIH Arrhythmia Database,” IEEE Eng. Med. Biol. Mag., vol. 20, no. 3, pp. 45–50, May–Jun. 2001.

[4] P. de Chazal, M. O’Dwyer, and R. B. Reilly, “Automatic classification of heartbeats using ECG morphology and heartbeat interval features,” IEEE Trans. Biomed. Eng., vol. 51, no. 7, pp. 1196–1206, Jul. 2004.

[5] E. J. da S. Luz, W. R. Schwartz, G. Cámara-Chávez, and D. Menotti, “ECG-based heartbeat classification for arrhythmia detection: A survey,” Comput. Methods Programs Biomed., vol. 127, pp. 144–164, Apr. 2016.

[6] A. Y. Hannun, P. Rajpurkar, M. Haghpanahi, G. H. Tison, C. Bourn, M. P. Turakhia, and A. Y. Ng, “Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network,” Nature Med., vol. 25, no. 1, pp. 65–69, Jan. 2019.

[7] M. Kachuee, S. Fazeli, and M. Sarrafzadeh, “ECG heartbeat classification: A deep transferable representation,” in Proc. IEEE Int. Conf. Healthcare Informat. (ICHI), New York, NY, USA, Jun. 2018, pp. 443–444.

[8] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, “Supervised learning with quantum-enhanced feature spaces,” Nature, vol. 567, no. 7747, pp. 209–212, Mar. 2019.

[9] M. Schuld and N. Killoran, “Quantum machine learning in feature Hilbert spaces,” Phys. Rev. Lett., vol. 122, no. 4, Art. no. 040504, Feb. 2019.

[10] Y. Liu, S. Arunachalam, and K. Temme, “A rigorous and robust quantum speed-up in supervised machine learning,” Nature Phys., vol. 17, no. 9, pp. 1013–1017, Sep. 2021.

[11] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, “Power of data in quantum machine learning,” Nature Commun., vol. 12, Art. no. 2631, May 2021.

[12] J. M. Kübler, S. Buchholz, and B. Schölkopf, “The inductive bias of quantum kernels,” in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 34, 2021, pp. 12661–12673.

[13] R. Shaydulin and S. M. Wild, “Importance of kernel bandwidth in quantum machine learning,” Phys. Rev. A, vol. 106, no. 4, Art. no. 042407, Oct. 2022.

[14] A. Canatar, E. Peters, C. Pehlevan, S. M. Wild, and R. Shaydulin, “Bandwidth enables generalization in quantum kernel models,” Trans. Mach. Learn. Res., 2023.

[15] S. Thanasilp, S. Wang, M. Cerezo, and Z. Holmes, “Exponential concentration in quantum kernel methods,” Nature Commun., vol. 15, Art. no. 5200, Jun. 2024.

[16] X. Wang, Y. Du, Y. Luo, and D. Tao, “Towards understanding the power of quantum kernels in the NISQ era,” Quantum, vol. 5, p. 531, Aug. 2021.

[17] E. Peters, J. Caldeira, A. Ho, S. Leichenauer, M. Mohseni, H. Neven, P. Spentzouris, D. Strain, and G. N. Perdue, “Machine learning of high dimensional data on a noisy quantum processor,” npj Quantum Inf., vol. 7, Art. no. 161, Nov. 2021.

[18] C. Cortes, M. Mohri, and A. Rostamizadeh, “Algorithms for learning kernels based on centered alignment,” J. Mach. Learn. Res., vol. 13, pp. 795–828, Mar. 2012.

[19] T. Hubregtsen, D. Wierichs, E. Gil-Fuster, P.-J. H. S. Derks, P. K. Faehrmann, and J. J. Meyer, “Training quantum embedding kernels on near-term quantum computers,” Phys. Rev. A, vol. 106, no. 4, Art. no. 042431, Oct. 2022.

[20] G. Gentinetta, A. Thomsen, D. Sutter, and S. Woerner, “The complexity of quantum support vector machines,” Quantum, vol. 8, p. 1225, Jan. 2024.

[21] J. R. Glick, T. P. Gujarati, A. D. Córcoles, Y. Kim, A. Kandala, J. M. Gambetta, and K. Temme, “Covariant quantum kernels for data with group structure,” Nature Phys., vol. 20, no. 3, pp. 479–483, Mar. 2024.

[22] T. Suzuki, T. Hasebe, and T. Miyazaki, “Quantum support vector machines for classification and regression on a trapped-ion quantum computer,” 2023, arXiv:2307.02091.

[23] Z. Ozpolat and M. Karabatak, “Performance evaluation of quantum-based machine learning algorithms for cardiac arrhythmia classification,” Diagnostics, vol. 13, no. 6, Art. no. 1099, Mar. 2023.

[24] J. Pan and W. J. Tompkins, “A real-time QRS detection algorithm,” IEEE Trans. Biomed. Eng., vol. BME-32, no. 3, pp. 230–236, Mar. 1985.

[25] dhiah-dev (GitHub user), “Dataset-Compiled-ECG-MIT-BIH-Arrhythmia: Compiled MIT-BIH Arrhythmia ready for researchers and programmers,” GitHub repository, 2023. [Online]. Available: https://github.com/dhiah-dev/Dataset-Compiled-ECG-MIT-BIH-Arrhythmia

[26] A. Pérez-Salinas, A. Cervera-Lierta, E. Gil-Fuster, and J. I. Latorre, “Data re-uploading for a universal quantum classifier,” Quantum, vol. 4, p. 226, Feb. 2020.

[27] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, “Quantum computing with Qiskit,” 2024, arXiv:2405.08810.