When Do Quantum Kernels Help? An Empirical and Diagnostic Study of the ZZ Feature Map

Luis Mendez
University of Connecticut
luis.a.mendez@uconn.edu

Abstract

Quantum kernel methods promise richer feature spaces than classical kernels by encoding data into the exponentially large Hilbert space of a quantum circuit, but it is unclear how often this promise translates into an actual advantage on realistic data. We build a from-scratch, exactly-simulated implementation of the Havlíček et al. (2019) ZZ feature-map fidelity kernel – without any quantum computing library – validate it against an independent brute-force construction, and use it to run a controlled comparison against classical (RBF, polynomial, linear) kernels under an identical model-selection protocol. On a synthetic dataset engineered so its labels are a function of the same feature map (a positive control), the quantum kernel reaches 98.3% test accuracy versus 48–53% (chance level) for every classical kernel. On the MiniBooNE particle-identification benchmark – a real high-energy-physics classification task (Fermilab electron-neutrino vs. muon-neutrino events) with no engineered relationship to any quantum feature map – the result reverses sharply: classical kernels reach 81–87% accuracy while the quantum kernel collapses to 71.7% at both PCA dimensionalities we test, exactly the majority-class baseline in both cases. We diagnose this gap using kernel-target alignment and kernel-value statistics, and connect it to existing theory on the inductive bias and concentration of quantum kernels. Our results are an honest negative result for quantum kernels as a drop-in replacement for classical kernels on generic data, and a clean positive control demonstrating our simulator and evaluation protocol are correct and can detect an advantage when one is present by construction.

1 Introduction

Quantum kernel methods encode a classical data point x into a quantum state $|\phi(x)\rangle$ via a parameterized circuit (a *feature map*), and define a kernel as an inner product between such states, $k(x, x') = |\langle\phi(x)|\phi(x')\rangle|^2$. Because $|\phi(x)\rangle$ lives in a Hilbert space of dimension 2^n for n qubits, this is often motivated as implicitly using an exponentially large feature space that would be classically intractable to represent explicitly [Rebentrost et al., 2014, Schuld and Killoran, 2019, Havlíček et al., 2019]. This is the same kernel-trick argument used to motivate the RBF kernel, and it raises an obvious empirical question: for a fixed, modest number of qubits, does a quantum kernel actually classify real data any better than a well-tuned classical kernel with the same number of input features?

The honest answer, as several theoretical results have begun to suggest [Kübler et al., 2021, Huang et al., 2021, Thanasilp et al., 2024], is usually no – unless the data has structure specifically aligned with the feature map. We investigate this question directly and empirically. Rather than relying on a quantum SDK (which can obscure what the kernel is actually computing behind a compiled circuit and a stochastic shot-based estimate), we implement an *exact*, from-scratch classical simulation of

Preprint. Under review.

the Havlíček et al. ZZ feature map kernel using only NumPy, and validate it bit-for-bit against an independent brute-force dense-matrix implementation. We then run the same model-selection and evaluation protocol for the quantum kernel and three classical kernels (RBF, polynomial, linear) on: (1) a synthetic dataset explicitly engineered, following Havlíček et al. [2019], so that its ground-truth labels are a function of the same feature map – a positive control that should favor the quantum kernel; and (2) the MiniBooNE particle identification dataset [Roe, 2010], a real high-energy-physics benchmark distinguishing electron-neutrino signal events from muon-neutrino background events at the MiniBooNE experiment, with no engineered relationship to any quantum feature map – a task a practitioner might realistically want to classify.

Contributions.

- An exact, dependency-free NumPy simulator of the ZZ feature-map fidelity kernel, using a batched Walsh–Hadamard transform and a vectorized diagonal-phase update, validated against an independent brute-force dense-matrix construction (Section 3).
- A reproduction of the Havlíček-style engineered "quantum-advantage" dataset construction, together with a fully worked-out, checkable derivation of its labeling rule (Section 3.5).
- A controlled empirical comparison, under one shared model-selection protocol and a common kernel normalization, of the quantum kernel against classical kernels on both the engineered positive control and a real high-energy-physics benchmark, MiniBooNE (Sections 4–5).
- A diagnosis of *why* the quantum kernel fails on real data using kernel-target alignment [Cristianini et al., 2001] and raw kernel-value statistics, connected to the inductive-bias and concentration-of-measure literature (Section 6).

We are explicit about what this study does *not* claim: because exact classical simulation costs $O(2^n)$ per sample, we can only evaluate small qubit counts ($n \leq 6$), so nothing here bears on whether a real quantum device would offer a *computational* (runtime) advantage. Our claim is narrower and purely statistical: for the ZZ feature map at accessible qubit counts, whether the resulting kernel is a good inductive bias for a classification task depends entirely on whether the task's structure matches the feature map – it is not a general-purpose upgrade over classical kernels.

2 Background and Related Work

Quantum kernel methods. Rebentrost et al. [2014] first proposed a quantum algorithm for kernel-based classification with an asymptotic speedup under strong data-access assumptions. Schuld and Killoran [2019] and Havlíček et al. [2019] reformulated quantum classifiers as kernel methods operating in a feature Hilbert space defined by a data-encoding circuit, making the standard kernel-trick machinery (e.g., support vector machines [Cortes and Vapnik, 1995]) directly applicable once the kernel matrix is computed – either by a quantum device or, at small scale, by classical simulation, as we do here. Havlíček et al. [2019] also introduced the specific ZZ feature map we use, along with a synthetic dataset construction, discussed in Section 3.5, designed to have a large margin in the induced feature space and thus be provably easy for the associated kernel.

When quantum kernels do (not) help. A growing body of theoretical work argues that quantum kernels are not a universal upgrade. Huang et al. [2021] show that without structure in the data or in the choice of feature map, quantum models offer no advantage over classical ones and can be matched by classical kernels informed by the same data. Kübler et al. [2021] formalize this as an inductive-bias question: a generic quantum feature map induces a kernel whose implied prior over functions is often a poor match for real learning tasks, leading to worse generalization than a well-chosen classical kernel unless the feature map is tailored to the problem. Thanasilp et al. [2024] show that for many families of quantum kernels, off-diagonal kernel values concentrate exponentially closely (in the number of qubits) around a fixed value as problem size grows, which drives the Gram matrix toward a constant matrix and the resulting classifier toward predicting the majority class – exactly the failure mode we observe empirically in Section 5. Our contribution relative to this literature is not new theory but a careful, reproducible, fully-classical empirical demonstration: the same simulator, normalization, and model-selection protocol applied to both an engineered positive control and unmodified real tabular data, with the resulting gap diagnosed rather than merely reported.

Kernel-target alignment. Cristianini et al. [2001] propose kernel-target alignment as a measure of how well a kernel's geometry matches a labeling function, independent of any specific classifier. We use it as a label-aware diagnostic to complement the label-agnostic kernel-value statistics of Thanasilp et al. [2024].

3 Method

3.1 The ZZ feature map and fidelity kernel

For n qubits and input $x \in \mathbb{R}^n$, the ZZ feature map applies R repetitions of a layer consisting of a Hadamard transform followed by a diagonal unitary encoding x:

$$U_\phi(x) = \left[\exp\Big(i \sum_j x_j Z_j + i \sum_{j<k} (\pi - x_j)(\pi - x_k)\, Z_j Z_k \Big)\, H^{\otimes n} \right]^R, \qquad |\phi(x)\rangle = U_\phi(x)\,|0\rangle^{\otimes n},$$

where Z_j is the Pauli-Z operator on qubit j. This is the "full entanglement" variant of Havlíček et al. [2019]: every pair of qubits is coupled through the $(\pi - x_j)(\pi - x_k)$ term. We use $R = 2$ throughout. The fidelity kernel is $k(x, x') = |\langle \phi(x) | \phi(x') \rangle|^2 \in [0, 1]$.

3.2 Exact simulation without a quantum SDK

Because the encoding unitary is diagonal in the computational basis, the circuit reduces to alternating (i) a Walsh–Hadamard transform and (ii) an elementwise complex phase multiply – no gate-by-gate simulation or quantum library is required. Writing $s_i(z) = (-1)^{z_i}$ for the i-th bit of basis state z, the phase applied to $|z\rangle$ is

$$\theta(z, x) = \sum_i x_i\, s_i(z) \;+\; \sum_{i<j} (\pi - x_i)(\pi - x_j)\, s_i(z) s_j(z).$$

Using $s_i(z)^2 = 1$, the pairwise sum can be rewritten in closed form as $\frac{1}{2}\big[(\sum_i u_i s_i(z))^2 - \sum_i u_i^2\big]$ with $u = \pi - x$, which lets us compute $\theta(\cdot, x)$ for all 2^n basis states and a batch of N samples with two matrix multiplications instead of an $O(n^2)$ loop over qubit pairs. The batched Walsh–Hadamard transform is applied by reshaping the $N \times 2^n$ state array to $N \times \underbrace{2 \times \cdots \times 2}_{n}$ and butterflying each axis, giving $O(Nn2^n)$ total cost per repetition – exact, and, for the $n \le 6$ qubits used in this paper, fast enough to run all experiments in under two seconds on a laptop CPU.

Correctness. We validate this vectorized implementation against an independent brute-force construction that builds the dense $2^n \times 2^n$ Hadamard matrix via Kronecker products and the diagonal phase matrix via an explicit double loop over qubit pairs, for $n \in \{1, 2, 3, 4\}$ and $R \in \{1, 2, 3\}$ with random inputs. The two implementations agree to within 10^{-10} in every case. We additionally check, over 50 random 5-qubit inputs, that every simulated state has unit norm (confirming the circuit is unitary), and that resulting kernel matrices are symmetric, have unit diagonal, and are positive semi-definite (minimum eigenvalue $> -10^{-8}$), as required of a valid Gram matrix. As a further, fully independent cross-language check, we reimplemented the simulator in Julia using a different algorithm for the Hadamard step (an in-place fast Walsh–Hadamard butterfly, rather than NumPy's reshape-and-moveaxis approach) and a different random number generator. On 200 random 6-qubit inputs ($R = 2$), the Julia and Python kernel matrices agree to within 5.6×10^{-15} – i.e. machine precision – and the JIT-compiled Julia version is roughly $5\times$ faster once warmed up (3 ms vs. 15 ms for the same 200×200 Gram matrix). Agreement to machine precision between two independent implementations, in different languages, with different underlying transform algorithms, rules out implementation-specific artifacts as an explanation for the results in Section 5.

3.3 Kernel normalization

Classical kernel values are not restricted to $[0, 1]$: on our data (features scaled to $[0, 2\pi)$, Section 4), an unnormalized degree-3 polynomial kernel can reach entries of order 10^4–10^5. In preliminary runs this made libsvm's QP solver (default `max_iter`$= -1$, i.e. unbounded) take pathologically long to converge on the polynomial kernel, effectively hanging the benchmark. We fix this, and

simultaneously put all kernels on a comparable footing for the alignment analysis, by normalizing every Gram matrix to unit diagonal, $\hat{k}(x, x') = k(x, x')/\sqrt{k(x, x)\, k(x', x')}$, before model fitting. This is a no-op for the quantum kernel, which already satisfies $k(x, x) = 1$ by construction. As a safety net we also cap the SVM solver at 2×10^5 iterations for every kernel.

3.4 Kernel-target alignment

Following Cristianini et al. [2001], for labels $y \in \{-1, +1\}^N$ and Gram matrix K we report

$$A(K, y) = \frac{\langle K,\, yy^\top \rangle_F}{\|K\|_F \, \|yy^\top\|_F},$$

a label-aware measure, in $[-1, 1]$, of how well the kernel's similarity structure matches the target labeling, independent of any specific downstream classifier.

3.5 Synthetic quantum-advantage dataset

We reproduce the engineered positive-control construction of Havlíček et al. [2019]. Fix a Haar-random $2^n \times 2^n$ unitary V (sampled via QR decomposition of a complex Gaussian matrix with the phase correction of Mezzadri [2007]) and a fixed non-trivial Pauli-Z-string observable $P = \mathrm{diag}(f)$, $f_z = (-1)^{\mathrm{parity}(z \,\&\, \mathrm{mask})}$ for a random non-zero bitmask. Define the rotated observable $M = VPV^\dagger$ and, for a candidate point x, its margin under the feature map,

$$m(x) = \langle \phi(x)|M|\phi(x)\rangle = \sum_z \left|\langle z|V^\dagger|\phi(x)\rangle\right|^2 f_z, \qquad y(x) = \mathrm{sign}\big(m(x)\big).$$

We draw x uniformly from $[0, 2\pi)^n$ and keep only points with $|m(x)| \geq \gamma$, discarding the rest; this yields, by construction, a dataset that is separable by the quantum kernel with margin at least γ in the V-rotated feature space, while making no promise about the existence of an easy classical decision boundary. We build two such datasets: $n = 4$ qubits, $\gamma = 0.3$, and a harder, higher-dimensional $n = 6$ qubits, $\gamma = 0.2$, each with 200 class-balanced samples.

4 Experimental Setup

Datasets. In addition to the two synthetic datasets above, we use the *MiniBooNE particle identification* dataset [Roe, 2010], a real high-energy-physics benchmark distinguishing electron-neutrino signal events from muon-neutrino background events at the MiniBooNE experiment (Fermilab), described by 50 reconstructed kinematic variables per event; it is also used as a non-toy tabular-classification benchmark in the deep-learning-for-tabular-data literature [Gorishniy et al., 2021]. We access it via OpenML [Pedregosa et al., 2011], drop the 0.36% of events containing a -999 missing-value sentinel in one or more features, and take a class-stratified subsample of 400 events. We then standardize features, project with PCA to 4 or 6 principal components (matching $n = 4$ and $n = 6$ qubits, mirroring the two synthetic configurations), and rescale to $[0, 2\pi)$ with a min-max scaler – the same representation is fed to every kernel, so comparisons isolate the choice of kernel rather than confounding it with different preprocessing. The underlying 400-event subsample is identical across the two PCA dimensionalities, so they compare the same points at different projections rather than different data.

Baselines. RBF, polynomial (degree $\in \{2, 3\}$, coef0 $= 1$), and linear kernels, via scikit-learn's pairwise-kernel functions, each fed into an SVM with a precomputed Gram matrix.

Model selection. For every (dataset, kernel-family) pair we sweep a hyperparameter grid (RBF: $\gamma \in \{0.02, 0.05, 0.1, 0.2, 0.5\}$; polynomial: degree $\in \{2, 3\}$; quantum and linear: no kernel hyperparameter beyond the fixed construction above) and, for each setting, select the SVM regularization constant $C \in \{0.1, 1, 3, 10, 30, 100\}$ by 5-fold stratified cross-validation on a 70% training split, then report accuracy on the untouched 30% test split, refitting once on the full training split with the selected hyperparameters. This identical protocol is applied to all four kernel families. All experiments use seed 0 for the train/test split and the CV folds.

Implementation. All kernels, the SVM wrapper, and the model selection loop are implemented once and shared across kernel families (Section 3), so no part of the pipeline is quantum-specific except the kernel matrix computation itself.

Table 1: Test accuracy, cross-validation accuracy, and kernel-target alignment for the best hyperparameter setting of each kernel family (selected by CV accuracy), on each dataset. Bold marks the best test accuracy per dataset (ties bolded jointly). Majority-class baselines: 50.0% (both synthetic datasets, class-balanced by construction), 71.7% (MiniBooNE, both PCA dimensionalities, same underlying subsample).

Dataset	Kernel	CV acc.	Test acc.	Alignment
Synthetic (n=4, γ=0.3)	Quantum (ZZ)	1.000	**0.983**	0.166
	RBF	0.657	0.483	0.084
	Polynomial	0.593	0.483	0.002
	Linear	0.500	0.483	0.001
Synthetic (n=6, γ=0.2)	Quantum (ZZ)	0.843	**0.733**	0.095
	RBF	0.614	0.467	0.062
	Polynomial	0.643	0.450	0.014
	Linear	0.607	0.533	0.005
MiniBooNE (PCA-4)	Quantum (ZZ)	0.718	0.717	0.159
	RBF	0.875	**0.817**	0.253
	Polynomial	0.854	0.808	0.212
	Linear	0.854	**0.817**	0.202
MiniBooNE (PCA-6)	Quantum (ZZ)	0.718	0.717	0.144
	RBF	0.868	**0.867**	0.223
	Polynomial	0.854	0.833	0.210
	Linear	0.850	0.817	0.197

5 Results

Table 1 shows the central result. On both engineered datasets, the quantum kernel dominates every classical kernel by a wide margin (98.3% vs. $\leq 48.3\%$ at n=4; 73.3% vs. $\leq 53.3\%$ at n=6, with all classical kernels within a few points of the 50% chance level). On MiniBooNE, the ranking reverses completely: classical kernels reach 81–87% test accuracy while the quantum kernel falls to 71.7% at *both* PCA dimensionalities – exactly the majority-class baseline in both cases – versus 9–15 points above baseline for every classical kernel setting.

Sample efficiency. Figure 1 tracks test accuracy as a function of training set size. On the engineered dataset (left), the quantum kernel improves steadily from 72% at 10 training points to 96% at 130, while every classical kernel is flat at chance level regardless of how much data it sees – more data does not help a kernel whose inductive bias does not match the task. On MiniBooNE (right), the pattern is a mirror image: all three classical kernels climb quickly to roughly 80–82% within the first 50 training points, while the quantum kernel is a flat line at 71.7% from the smallest training size onward, i.e. it never learns anything past the majority class, even with 260 training examples.

Decision boundaries. To visualize *why* the quantum kernel wins on engineered data, Figure 2 shows decision regions for a 2-qubit version of the engineered dataset (chosen only so the input space is plottable in 2D), together with the resulting test accuracy for each kernel. The quantum kernel reaches 100% test accuracy with a highly fragmented decision region: the true boundary, induced by a fixed observable measured in a randomly rotated basis, is genuinely intricate when drawn in raw (x_1, x_2) coordinates, even though it corresponds to a single hyperplane cut in the kernel's native feature space. The RBF kernel partially approximates this intricate region locally (85%), while the polynomial and linear kernels, which can only express comparatively smooth, low-order boundaries, are close to chance (52%, 54%).

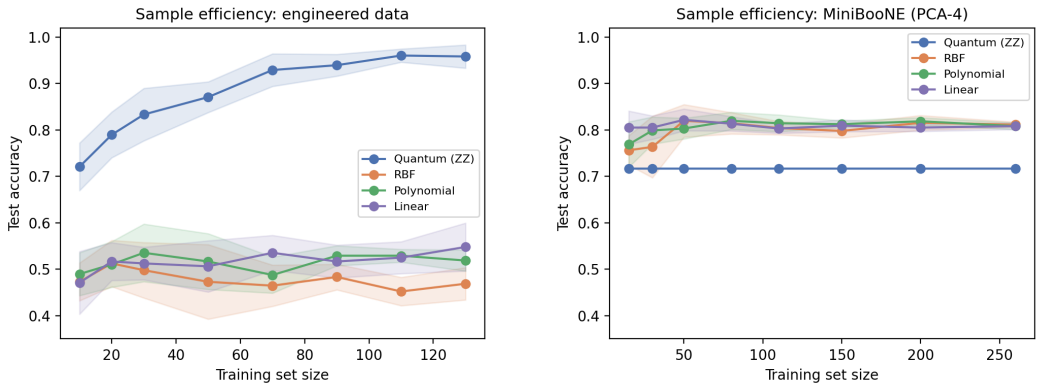

Figure 1: Test accuracy vs. training set size (mean $\pm$ std over 8 random training subsets at each size, fixed hyperparameters from Table 1). **Left:** engineered quantum-advantage data – the quantum kernel is the only one that learns. **Right:** MiniBooNE (real particle-physics data) – the quantum kernel is flat at the majority-class baseline regardless of training set size, while classical kernels learn quickly.

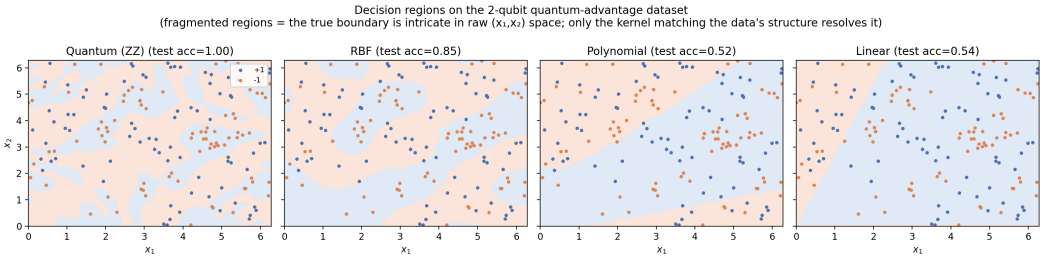

Figure 2: Decision regions for each kernel on a 2-qubit engineered dataset (points colored by true label). Only the quantum kernel, whose feature space matches how the labels were generated, resolves the intricate true boundary exactly; classical kernels with smoother inductive biases cannot.

6 Discussion

Alignment tracks accuracy, but only within a kernel family. Within the classical kernels on real data, alignment and test accuracy move together in a rough but not perfectly monotonic way (e.g. on MiniBooNE PCA-6, alignment rises monotonically from 0.207 to 0.240 as RBF's γ grows from 0.02 to 0.2, while accuracy peaks earlier, at $\gamma = 0.05$ (86.7%), before mildly declining even as alignment keeps rising) – consistent with alignment being a useful, if imperfect, proxy for kernel quality [Cristianini et al., 2001]. But alignment values alone do not explain the quantum-vs-classical gap in Table 1: the quantum kernel's alignment on MiniBooNE (0.159 at PCA-4) is *higher* than its own alignment on the engineered n=6 dataset (0.095), even though it generalizes far better on the latter (73.3% test accuracy, well above the 50% baseline) than on the former (71.7%, exactly at baseline). Alignment computed on the training fold captures how separable the *training* labels look under the kernel, but does not on its own diagnose whether that separability will transfer to unseen points, particularly for a kernel whose off-diagonal values are weak and diffuse to begin with.

Kernel concentration, not just alignment. A direct check of the raw kernel values is more revealing. Off-diagonal quantum-kernel entries on MiniBooNE (PCA-4) have mean 0.091 and standard deviation 0.087; on the engineered n=4 dataset the corresponding statistics are 0.069 and 0.068 – broadly similar in magnitude, and both in the vicinity of $1/2^n = 1/16 = 0.0625$, the value expected for Haar-random states. In other words, the quantum kernel is comparably "concentrated" – most points look almost equally (dis)similar to each other – on *both* datasets; concentration alone, as studied by Thanasilp et al. [2024], does not distinguish the case where the kernel works from the case where it does not. What differs is whether the few non-negligible off-diagonal similarities that do survive line up with the actual labels. On the engineered dataset they do, by construction: the labeling rule $y(x) = \mathrm{sign}\langle\phi(x)|M|\phi(x)\rangle$ is defined directly from the same feature map that the kernel measures

similarity in, so the SVM's max-margin hyperplane in that induced space is, by design, close to the true decision rule. On MiniBooNE, the labels are determined by real neutrino-interaction physics with no relationship whatsoever to the randomly chosen V and observable mask that would have to align with them, so the concentrated, largely unstructured similarities give the SVM nothing to exploit and it falls back to the majority-class solution – precisely the failure mode Thanasilp et al. [2024] predict, and an instance of the general inductive-bias mismatch formalized by Kübler et al. [2021]: a generic quantum feature map is only a good prior for tasks whose structure happens to match it, and there is no reason a priori for a neutrino-event classification task to share structure with a randomly chosen quantum observable.

Limitations. (1) Exact simulation restricts us to $n \leq 6$ qubits; we make no claim about behavior at the qubit counts where a quantum computer would offer a computational advantage, nor about runtime – our claim is purely about statistical generalization at accessible scale. (2) We test one feature map (ZZ, full entanglement, R=2) and one real benchmark family (MiniBooNE, at two PCA dimensionalities of the same underlying subsample); other feature maps (e.g. data re-uploading or trainable ansätze) or a wider benchmark suite spanning multiple real-data domains could behave differently, and a larger study is needed before drawing conclusions about quantum kernels in general rather than this specific, widely-used construction. (3) Our PCA-based preprocessing (to 4 or 6 dimensions), chosen to match a small qubit count, may itself discard information a classical kernel operating on the full 50-feature set could have used; we control for this by feeding classical kernels the same reduced representation, but a practitioner would not normally handicap a classical baseline this way. (4) We select C (and, for classical kernels, one additional kernel hyperparameter) by cross-validation, but do not tune the quantum feature map's own hyperparameters (number of qubits, repetitions, entanglement pattern) as extensively as we tune the classical kernels' hyperparameters, since the ZZ feature map has no natural analogue of γ or degree beyond R; a more exhaustive search over feature-map variants is left to future work. As a partial check, we ran a post-hoc search over $R \in \{1, \ldots, 4\}$ and C using two independent search procedures and implementations: Bayesian optimization (Optuna, TPE sampler, 40 trials) in Python, and a dependency-light random search (40 trials) in Julia built on our independent Julia reimplementation of the simulator (Section 3) together with LIBSVM.jl. On the synthetic datasets, R=2 – the value used to generate the labels – remains optimal in both searches, as expected. On MiniBooNE, tuning recovers at most a few points of test accuracy (up to 76.7% at PCA-6, versus 71.7% at the fixed R=2 baseline), and the two searches do not even agree on which R is best at PCA-4, consistent with a flat, noisy CV-accuracy landscape rather than a real gain to be had. In every configuration tested, the tuned quantum kernel remains far below the 81–87% classical baselines of Table 1, so the negative result on MiniBooNE is not an artifact of leaving R untuned. A fANOVA hyperparameter-importance analysis (Optuna, 80 trials per domain, jointly searching $n_{\text{qubits}} \in \{4, 6\}$, R, and C) explains why: on the synthetic datasets R accounts for 96.5% of variance in CV accuracy (versus 3.4% for n_{qubits} and 0.1% for C), since R directly controls whether the kernel matches the fixed feature map used to generate the labels; on MiniBooNE, where no such planted structure exists for any R to unlock, n_{qubits} (i.e. the PCA dimensionality of Limitation 3) dominates instead (53.9%, versus 30.4% for R and 15.7% for C). The quantum kernel's hyperparameter that matters most is therefore not fixed across datasets – it depends on whether the data has feature-map-aligned structure to exploit in the first place, reinforcing this paper's central claim.

Broader impact. This is a small-scale, fully classical simulation study with no direct societal impact; we note it mainly as a caution against over-claiming quantum advantage in applied machine learning settings on the basis of feature-space size alone, and as a worked example that a claimed advantage should be checked against simple, well-tuned classical baselines under matched protocols before being reported.

7 Conclusion

Using a from-scratch, validated, exact simulation of the ZZ feature-map quantum kernel, we show that whether this kernel helps or hurts is entirely a function of whether the data's structure matches the feature map, not an intrinsic property of "being quantum." On data engineered to match it, the quantum kernel dominates classical baselines by up to 50 points of test accuracy and is the only kernel that benefits from additional training data; on a real high-energy-physics benchmark (MiniBooNE), it collapses to the majority-class baseline while classical kernels reach 81–87% accuracy under an

identical protocol. This gap is explained by a combination of concentrated off-diagonal kernel values and, critically, the absence of any relationship between those values and the real labels – consistent with existing inductive-bias and concentration theory for quantum kernels. We release our simulator, datasets, and evaluation code so this protocol can be extended to other feature maps and benchmarks.

References

Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

Nello Cristianini, John Shawe-Taylor, Andre Elisseeff, and Jaz Kandola. On kernel-target alignment. In *Advances in Neural Information Processing Systems*, volume 14, 2001.

Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Revisiting deep learning models for tabular data. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019.

Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R McClean. Power of data in quantum machine learning. *Nature Communications*, 12(1):2631, 2021.

Jonas Kübler, Simon Buchholz, and Bernhard Schölkopf. The inductive bias of quantum kernels. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Francesco Mezzadri. How to generate random matrices from the classical compact groups. *Notices of the American Mathematical Society*, 54:592–604, 2007.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical Review Letters*, 113(13):130503, 2014.

Byron Roe. MiniBooNE particle identification. UCI Machine Learning Repository, 2010.

Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical Review Letters*, 122(4):040504, 2019.

Supanut Thanasilp, Samson Wang, M Cerezo, and Zoë Holmes. Exponential concentration in quantum kernel methods. *Nature Communications*, 15, 2024.