

What Does Entanglement Do in a Projected Quantum Kernel?

An Exact Closed Form for the IQP Embedding

Author

June 8, 2026

Abstract

The projected quantum kernel is one of the most cited near-term proposals for a quantum advantage in machine learning: each input is embedded in a quantum state, and the kernel is built from local measurements of that state. We show, for the standard depth-one IQP embedding, that this kernel has an *exact closed form*—it is, to machine precision, a classical kernel over the explicit trigonometric feature map $\langle X_k \rangle = \cos(x_k) \prod_{j \neq k} \cos(x_j x_k)$. The entanglement in the circuit contributes *only* the fixed product of cosines $\prod_j \cos(x_j x_k)$: a set of classical cross-feature interactions, not a quantum resource. The derivation is elementary and we give it in full. The same one-line formula does double duty: because each added qubit multiplies in another sub-unit cosine, it also explains why the kernel *concentrates*—its features shrink toward zero as the qubit count grows. This note is written as a self-contained tutorial: we build up the quantum feature map, the fidelity kernel and its concentration problem, the projected kernel that was meant to fix it, and the IQP embedding, before deriving and interpreting the closed form. The takeaway is a concrete cautionary tale—a “quantum” kernel that is secretly classical, with its own scaling failure written into the same expression that proves it classical.

1 Introduction

Quantum machine learning often begins with an appealing picture: map a classical input x into the exponentially large Hilbert space of q qubits via a state $|\psi(x)\rangle$, and let a standard kernel method work in that space. Because the embedding can be hard to simulate classically, one hopes the induced kernel captures structure no efficient classical kernel can. The *projected quantum kernel* (PQK) of Huang et al. [1] is the most prominent concrete realization of this hope that remains usable at scale, and it is widely used as a baseline “quantum kernel.”

This tutorial answers a narrow but sharp question about it: *for the standard IQP embedding, what does the quantum part actually contribute?* The answer is clean enough to write on one line. The projected features have an exact closed form,

$$\langle X_k \rangle = \cos(x_k) \prod_{j \neq k} \cos(x_j x_k), \quad \langle Y_k \rangle = -\sin(x_k) \prod_{j \neq k} \cos(x_j x_k), \quad \langle Z_k \rangle = 0, \quad (1)$$

so the PQK is *identical*, to machine precision, to a classical kernel over the explicit feature map (1). The entanglement enters only through the factors $\prod_j \cos(x_j x_k)$ —fixed cross-feature cosines, i.e. ordinary classical feature interactions. There is no quantum resource hiding in this kernel.

We have two goals. First, to *teach*: Sections 2–3 assume only basic linear algebra and build every ingredient—kernels, quantum feature maps, the fidelity kernel and why it fails, the projected kernel, and the IQP circuit—so the derivation in Section 4 is self-contained. Second, to deliver one

payoff with two faces: Equation (1) shows both that the kernel is classical (Section 5) and, because each qubit adds another sub-unit factor, that it *concentrates* as the system grows (Section 6). One formula explains both “it is not quantum” and “it does not scale.”

2 Background: kernels and quantum feature maps

Kernels in one paragraph. A kernel method classifies by comparing inputs through a similarity $K(x, x') = \langle \phi(x), \phi(x') \rangle$, where the *feature map* ϕ sends each input into some (possibly very high-dimensional) vector space. The central trick is that one never needs ϕ explicitly: a support vector machine (SVM) finds a separating hyperplane using only the *Gram matrix* $K_{ij} = K(x_i, x_j)$ of pairwise similarities. Any symmetric, positive-semidefinite K is guaranteed to arise from *some* feature map (Mercer’s theorem), so one is free to design the similarity directly and let the feature space be implicit. The familiar radial basis function (RBF) kernel $K(x, x') = \exp(-\gamma \|x - x'\|^2)$, for instance, corresponds to a fixed, infinite-dimensional smooth feature map; the bandwidth γ sets how quickly similarity decays with distance. A kernel “helps” on a task precisely when its feature map places same-class points near one another—when its geometry aligns with the labels. For the kernel and SVM background assumed here, see Schölkopf and Smola [6] or Shawe-Taylor and Cristianini [7].

Quantum feature maps. A *quantum* feature map takes the feature space to be the state space of a quantum computer. It sends x to a state $|\psi(x)\rangle = U(x)|0\rangle^{\otimes q}$ prepared by a data-dependent circuit $U(x)$ on q qubits, starting from the all-zeros state $|0\rangle^{\otimes q}$. A q -qubit pure state is a unit vector in $\mathbb{C}^{2^q}$, so the feature space dimension grows *exponentially* in the number of qubits—the source of the optimism that a quantum kernel might encode structure no efficient classical feature map can. (Here $\langle A \rangle \equiv \langle \psi | A | \psi \rangle$ denotes the expectation value, or average measurement outcome, of an observable A .) The idea of computing kernels from quantum states is due to Havlíček et al. [4] and Schuld and Killoran [5]; for the broader area see the review/text [8]. Two ways to turn states into a kernel are standard.

The fidelity kernel and its problem. The most direct choice is the *fidelity kernel* $K_{\text{fid}}(x, x') = |\langle \psi(x) | \psi(x') \rangle|^2$, the squared overlap of the two states—a natural “how aligned are these two unit vectors” similarity, equal to 1 when the states coincide and 0 when they are orthogonal. It is a valid kernel and uses the full Hilbert space. Unfortunately it *concentrates*. The intuition is the curse of dimensionality: in an exponentially large space, two generic unit vectors are almost surely nearly orthogonal, so as q grows $\langle \psi(x) | \psi(x') \rangle$ becomes exponentially small for essentially all $x \neq x'$. The Gram matrix then approaches the identity—every distinct pair looks equally (and maximally) dissimilar—and the kernel carries no usable signal [3]. Training on such a kernel is hopeless, and worse, distinguishing its near-equal entries requires exponentially many measurement shots on real hardware.

The projected quantum kernel. Huang et al. [1] proposed to sidestep concentration by reading out only *local* information. For each qubit k one measures the three single-qubit Pauli observables $\langle X_k \rangle, \langle Y_k \rangle, \langle Z_k \rangle$. These are exactly the coordinates of the *Bloch vector* of qubit k ’s reduced state (the 2×2 density matrix obtained by averaging out all other qubits), so the triple $(\langle X_k \rangle, \langle Y_k \rangle, \langle Z_k \rangle)$ is a point in the unit ball $\mathbb{R}^3$ summarizing what qubit k looks like in isolation. Stacking these across qubits gives a classical vector

$$\phi_{\text{PQK}}(x) = (\langle X_k \rangle, \langle Y_k \rangle, \langle Z_k \rangle)_{k=1}^q \in \mathbb{R}^{3q}, \quad (2)$$

and defines the projected quantum kernel as an ordinary RBF over these projected features,

$$K_{\text{PQK}}(x, x') = \exp(-\gamma \|\phi_{\text{PQK}}(x) - \phi_{\text{PQK}}(x')\|^2). \quad (3)$$

Because ϕ_{PQK} has only $3q$ bounded coordinates (a number that grows *linearly*, not exponentially, in q), the kernel cannot collapse to the identity the way the fidelity kernel does—there is simply not enough room for all pairs to look mutually orthogonal. The PQK is thus the natural “robust” quantum kernel, and the question of this note is what its features actually compute.

3 The IQP embedding

To answer that we need to fix a circuit $U(x)$. We use the embedding most common in this setting, the depth-one Instantaneous Quantum Polynomial (IQP) map. It consists of a layer of Hadamards followed by data-dependent *diagonal* phase rotations:

$$|\psi(x)\rangle = \underbrace{\exp\left(i \sum_{(j,k) \in E} x_j x_k Z_j Z_k\right)}_{\text{entangling phases}} \underbrace{\exp\left(i \sum_j x_j Z_j\right)}_{\text{single-qubit phases}} H^{\otimes q} |0\rangle^{\otimes q}, \quad (4)$$

where Z_j and X_j, Y_j are the Pauli matrices acting on qubit j , and E is the set of qubit pairs that are entangled (we take the complete graph, i.e. all pairs, the common default). Reading (4) right to left: the Hadamards put every qubit into an equal superposition; the single-qubit phases inject the features x_j ; and the two-qubit $Z_j Z_k$ phases inject the products $x_j x_k$ and are the *only* entangling part of the circuit. The name “instantaneous” refers to the fact that all the phase gates are diagonal in the computational basis and hence *commute*—a property we will use directly. Despite this simplicity, sampling from the output of IQP circuits is believed to be classically intractable [9]; as we will see, the projected (local) readout discards exactly the global information in which that hardness lives.

To make the diagonal structure concrete, recall two facts. First, the Hadamard layer creates the uniform superposition $H^{\otimes q}|0\rangle = 2^{-q/2} \sum_{z \in \{0,1\}^q} |z\rangle$ over all 2^q computational basis states $|z\rangle$ (bit strings z). Second, a diagonal gate does not move amplitude between basis states; it only multiplies each $|z\rangle$ by a phase. To track those phases, write $s_j = (-1)^{z_j} \in \{+1, -1\}$ for the eigenvalue of Z_j on $|z\rangle$ ($s_j = +1$ if bit j is 0, and -1 if it is 1), since $Z_j|z\rangle = (-1)^{z_j}|z\rangle$. Substituting $Z_j \rightarrow s_j$ and $Z_j Z_k \rightarrow s_j s_k$ into the exponents of (4), the amplitude of $|z\rangle$ in $|\psi(x)\rangle$ is $2^{-q/2} e^{i\Phi(s)}$ with the real phase

$$\Phi(s) = \sum_j x_j s_j + \sum_{(j,k) \in E} x_j x_k s_j s_k. \quad (5)$$

The key observation is that every basis state has the *same* amplitude magnitude $2^{-q/2}$; only the phase $\Phi(s)$ depends on the data. This equal-magnitude structure is what makes the closed form possible, and it is special to embeddings built entirely from Hadamards and diagonal gates.

4 The closed form

Proposition 1. *For the depth-one IQP embedding (4) with entangling set E , the projected features are exactly*

$$\langle X_k \rangle = \cos(x_k) \prod_{j:(j,k) \in E} \cos(x_j x_k), \quad \langle Y_k \rangle = -\sin(x_k) \prod_{j:(j,k) \in E} \cos(x_j x_k), \quad \langle Z_k \rangle = 0.$$

Derivation. Index basis states by their sign vector $s \in \{\pm 1\}^q$ as above, so $|\psi\rangle = 2^{-q/2} \sum_s e^{i\Phi(s)} |s\rangle$ with Φ from (5).

$\langle Z_k \rangle = 0$. Because Z_k is diagonal ($Z_k |s\rangle = s_k |s\rangle$), its expectation depends only on the amplitude *magnitudes*, and those are all equal:

$$\langle Z_k \rangle = \sum_s |2^{-q/2} e^{i\Phi(s)}|^2 s_k = 2^{-q} \sum_s s_k = 0,$$

because exactly half of the sign vectors have $s_k = +1$ and half -1 , so the $+1$ and -1 terms cancel. Geometrically, the equal superposition leaves each qubit's Bloch vector in the XY -plane (the equator), with no Z -component—a fact we did not need to compute coordinate by coordinate.

$\langle X_k \rangle$. The bit-flip X_k sends $|s\rangle$ to $|s^{(k)}\rangle$, where $s^{(k)}$ is s with s_k negated. Hence

$$\langle X_k \rangle = \langle \psi | X_k | \psi \rangle = 2^{-q} \sum_s e^{-i\Phi(s)} e^{i\Phi(s^{(k)})} = 2^{-q} \sum_s e^{i[\Phi(s^{(k)}) - \Phi(s)]}.$$

Only the terms of Φ that contain s_k change under the flip: the single-qubit term $x_k s_k$ and the entangling terms $x_j x_k s_j s_k$ for $j \sim k$. Negating s_k negates each, so

$$\Phi(s^{(k)}) - \Phi(s) = -2 s_k \left(x_k + \sum_{j:(j,k) \in E} x_j x_k s_j \right) \equiv -2 s_k A(s).$$

Summing first over $s_k \in \{\pm 1\}$ (which appears only in the prefactor s_k , since A does not contain s_k) gives $\sum_{s_k} e^{-2i s_k A} = 2 \cos(2A)$, and then averaging over the remaining s_j , $j \sim k$,

$$\langle X_k \rangle = 2^{-(q-1)} \sum_{\{s_j\}_{j \sim k}} \cos \left(2x_k + 2 \sum_{j \sim k} x_j x_k s_j \right).$$

Now use the elementary identity, valid for i.i.d. uniform signs $s_j \in \{\pm 1\}$,

$$\mathbb{E}_s \left[\cos \left(\alpha + \sum_j \beta_j s_j \right) \right] = \cos(\alpha) \prod_j \cos(\beta_j). \quad (6)$$

To see why, note that averaging a single sign gives $\mathbb{E}_{s_j} [e^{i\beta_j s_j}] = \frac{1}{2}(e^{i\beta_j} + e^{-i\beta_j}) = \cos \beta_j$; since the signs are independent the expectation of the product factorizes, $\mathbb{E}[e^{i(\alpha + \sum_j \beta_j s_j)}] = e^{i\alpha} \prod_j \cos \beta_j$, and (6) is the real part. With $\alpha = 2x_k$ and $\beta_j = 2x_j x_k$ this yields $\langle X_k \rangle = \cos(2x_k) \prod_{j \sim k} \cos(2x_j x_k)$. The factors of two are an artifact of the rotation-angle convention in (4); the half-angle convention used by standard implementations (single-qubit gate $\exp(-\frac{i}{2} x_j Z_j)$) replaces $2x_j$ by x_j throughout and gives exactly the stated form. The computation for $\langle Y_k \rangle$ is identical except that Y_k flips $|s\rangle$ with an extra factor $\pm i$, turning the leading $\cos$ into $-\sin$. $\square$

Numerical check. Equation (1) can be verified directly: building the PQK (3) from a state-vector simulation and comparing it to an RBF over the classical features (1) gives identical Gram matrices. Figure 1(A) overlays every Gram entry for $q = 6$; the largest discrepancy is 3×10^{-15} (floating-point noise) and the kernel alignment is 1.0000. They are the same kernel.

Worked example: the two-qubit case by hand. With $q = 2$ and the single entangling pair $E = \{(0, 1)\}$, the formula predicts $\langle X_0 \rangle = \cos(x_0) \cos(x_0 x_1)$. Here is the whole computation. The four sign vectors $s = (s_0, s_1)$ carry phases $\Phi(s) = x_0 s_0 + x_1 s_1 + x_0 x_1 s_0 s_1$ from (5). Flipping qubit 0 changes only the s_0 -dependent terms, by $\Phi(s^{(0)}) - \Phi(s) = -2 s_0 (x_0 + x_0 x_1 s_1)$, so

$$\langle X_0 \rangle = \frac{1}{4} \sum_{s_0, s_1 \in \{\pm 1\}} e^{-2i s_0 (x_0 + x_0 x_1 s_1)}.$$

Summing s_0 over ± 1 turns each exponential into a cosine ($e^{i\theta} + e^{-i\theta} = 2 \cos \theta$):

$$\langle X_0 \rangle = \frac{1}{2} [\cos(2x_0 + 2x_0 x_1) + \cos(2x_0 - 2x_0 x_1)].$$

The two terms combine through the sum-to-product identity $\cos(A+B) + \cos(A-B) = 2 \cos A \cos B$ (the $q = 2$ instance of (6)) with $A = 2x_0$, $B = 2x_0 x_1$, giving $\langle X_0 \rangle = \cos(2x_0) \cos(2x_0 x_1) \rightarrow \cos(x_0) \cos(x_0 x_1)$ in the half-angle convention. As a numerical anchor, at $x_0 = x_1 = 1$ this is $\cos(1) \cos(1) \approx 0.292$, matching a direct simulation. The entangling term $x_0 x_1$ has produced exactly *one* extra classical cosine factor—and with q qubits there are $q - 1$ such factors, which is the whole story of both consequences below.

5 Consequence 1: the kernel is classical

Read (1) as a feature map. Each qubit contributes the two coordinates

$$(\langle X_k \rangle, \langle Y_k \rangle) = m_k(x) (\cos x_k, -\sin x_k), \quad m_k(x) = \prod_{j \neq k} \cos(x_j x_k),$$

that is, a one-dimensional *Fourier feature* $(\cos x_k, -\sin x_k)$ scaled by the scalar $m_k(x)$. The Fourier part is exactly the truncated trigonometric expansion that any data-encoding circuit produces, with frequencies fixed by the encoding [2]. The modulation $m_k(x)$ —the *only* thing entanglement contributes—is a product of cosines of the pairwise products $x_j x_k$: ordinary, fixed, classical cross-feature interactions. Nothing in (1) requires a quantum computer to evaluate; the map is a handful of sines, cosines, and products.

Consequently the PQQ (3) is a *classical RBF kernel over an explicit trigonometric feature map*. Its inductive bias is fixed and low-order: it can help only when the labels happen to align with this particular basis of single-feature sinusoids and pairwise-product cosines, exactly as a classical kernel with that feature map would. This is what we mean by saying the kernel “dequantizes.” It also locates the role of entanglement precisely: not absent (the m_k are genuine data-dependent cross terms), but *classical*—a deterministic interaction map, not a resource a classical model cannot reproduce.

6 Consequence 2: it concentrates as it scales

The same product reveals a scaling pathology. The modulation $m_k(x) = \prod_{j \neq k} \cos(x_j x_k)$ is a product of $q - 1$ factors, each of magnitude at most one and typically strictly less. As the number of qubits grows, $m_k(x) \rightarrow 0$ for all but the most special inputs, so the projected features (1) shrink toward zero and the PQQ Gram matrix drifts toward a constant—the same concentration that afflicts the fidelity kernel [3], now visible directly in a closed form rather than argued abstractly. The local readout mitigates concentration relative to the fidelity kernel but does not escape it: locality buys a milder rate, not immunity.

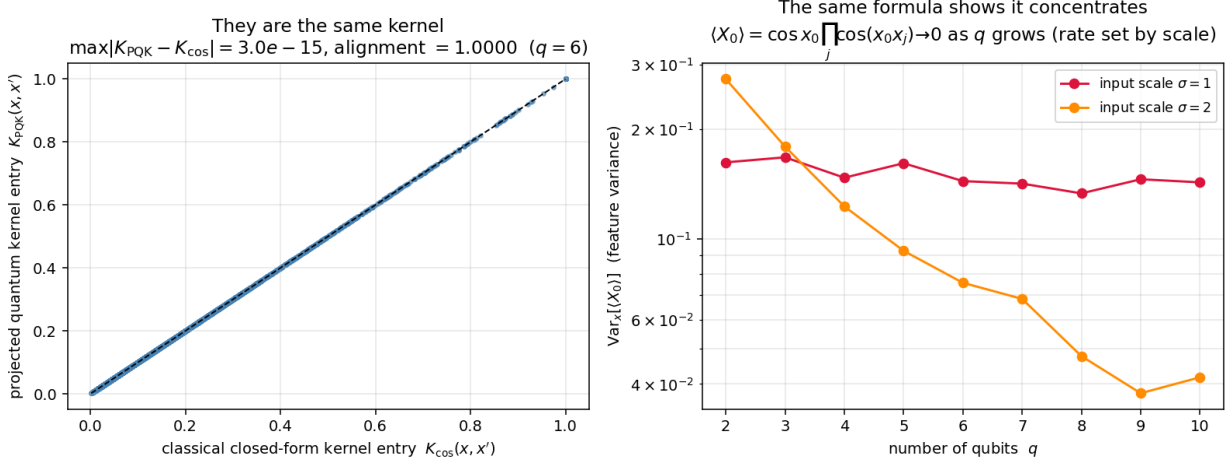


Figure 1: **One formula, two lessons.** (A) The projected quantum kernel built from a simulated IQP circuit and the classical RBF over the closed-form features (1) produce identical Gram matrices (every entry on the diagonal; $\max|\Delta| = 3 \times 10^{-15}$, alignment 1.0000): the kernel is classical. (B) The same closed form predicts concentration: each qubit multiplies in another factor $\cos(x_j x_k) < 1$, so the feature variance $\text{Var}_x[\langle X_0 \rangle]$ falls as q grows, at a rate set by the input scale σ (negligible at $\sigma = 1$, pronounced at $\sigma = 2$).

The rate is set by how spread the encoded angles are. Figure 1(B) plots the feature variance $\text{Var}_x[\langle X_0 \rangle]$ against q for two input scales. At unit scale the cosines stay near one and concentration is slow; at scale $\sigma = 2$ the features lose more than half their variance by $q = 6$ and keep falling. So the very mechanism that the closed form makes transparent—multiplying in one sub-unit cosine per qubit—is also the mechanism that degrades the kernel as the device grows. Scaling this construction up makes it both no less classical and less useful.

7 Discussion

We set out to ask what the quantum part of a projected quantum kernel computes, for the standard IQP embedding, and found that it computes something we can write down by hand: a classical trigonometric feature map whose only entangling content is a fixed product of cosines. The lesson is not that quantum kernels are useless, but that this widely used one is a *classical kernel in disguise*, and that its disguise is thin enough to remove with a page of algebra. The same expression that unmasks it also forecasts its failure to scale.

Two caveats fix the scope. First, the closed form is exact for the *depth-one* IQP map; deeper or non-commuting encodings need not admit it, and whether they hide a genuine resource is a separate question. Second, the concentration we display is for classical inputs of moderate scale; the precise rate depends on the data distribution and the encoding bandwidth. What is robust is the structural point: for this canonical embedding, “quantum kernel” is a classical cosine kernel, and the cosines that make it classical are the cosines that make it concentrate. A practitioner reaching for a PQK on tabular data is, for this embedding, reaching for a fixed classical feature map—and is better served choosing that feature map deliberately.

None of this says quantum kernels cannot help in principle: there are constructed learning problems with a *provable* quantum kernel advantage [10], and a useful kernel must in any case

have an inductive bias matched to the data [11]. The point is narrower and practical—a specific, popular construction is classical—and the method is the transferable part: when in doubt, write the feature map down. Readers wanting to go further may consult the kernel-methods texts [6, 7], the quantum-kernel foundations [4, 5], the encoding/Fourier analysis [2], and the concentration and trainability literature [3] for the obstacles that motivated the projected kernel in the first place.

References

- [1] 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 Communications* **12**, 2631 (2021).
- [2] M. Schuld, R. Sweke, and J. J. Meyer, “Effect of data encoding on the expressive power of variational quantum-machine-learning models,” *Physical Review A* **103**, 032430 (2021).
- [3] S. Thanasilp, S. Wang, M. Cerezo, and Z. Holmes, “Exponential concentration in quantum kernel methods,” *Nature Communications* **15**, 5200 (2024).
- [4] 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* **567**, 209 (2019).
- [5] M. Schuld and N. Killoran, “Quantum machine learning in feature Hilbert spaces,” *Physical Review Letters* **122**, 040504 (2019).
- [6] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (MIT Press, 2002).
- [7] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis* (Cambridge University Press, 2004).
- [8] M. Schuld and F. Petruccione, *Machine Learning with Quantum Computers*, 2nd ed. (Springer, 2021).
- [9] M. J. Bremner, A. Montanaro, and D. J. Shepherd, “Average-case complexity versus approximate simulation of commuting quantum computations,” *Physical Review Letters* **117**, 080501 (2016).
- [10] Y. Liu, S. Arunachalam, and K. Temme, “A rigorous and robust quantum speed-up in supervised machine learning,” *Nature Physics* **17**, 1013 (2021).
- [11] J. M. Kübler, S. Buchholz, and B. Schölkopf, “The inductive bias of quantum kernels,” in *Advances in Neural Information Processing Systems (NeurIPS)* **34** (2021).