

When Do Entangling Gates Add Predictive Signal? A Preregistered Falsification Study for Hybrid Quantum-Classical Combinatorial Optimization

David Vesterlund

david@vesterlundventures.se

Vesterlund Ventures

Research Article

Keywords:

Posted Date: October 5th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11192113/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

When Do Entangling Gates Add Predictive Signal?

A Preregistered Falsification Study of Quantum Reservoir Features for Combinatorial Optimization

David Vesterlund

Industrial Research at WestQuant Studio

Vesterlund Ventures Holding AB, Jönköping, Sweden

ORCID: 0009-0000-6455-1141

david@vesterlundventures.se

Abstract

Choosing the best heuristic for a combinatorial optimization instance depends on landscape structure that cheap graph descriptors capture only partly. We test whether a small quantum circuit, used as a fixed feature map, adds predictive signal about solver performance beyond such descriptors. Graphs are spectrally compressed to eight supernodes and encoded in a single-layer circuit of R_Y rotations and R_{XX} entangling gates; eight pooled statistics computed from Z - and X -basis expectation values feed a classical ridge readout. In a preregistered protocol on 10,000 simulated graphs the features add $\Delta R^2 = 0.0208$ (Nadeau–Bengio corrected 95% CI [0.0202, 0.0214], permutation $p = 1/1001 \approx 0.001$) over twelve descriptors. They exceed random-projection and random-Fourier controls; a paired test does not separate them from a random tanh reservoir ($p = 0.22$). **However, flexible classical models with access to the same compressed graph input (h, J)—Random Forest ($\Delta R^2 = 0.179$) and XGBoost ($\Delta R^2 = 0.145$)—dramatically exceed Q1, as does a simple linear ridge ($\Delta R^2 = 0.030$). The quantum feature map does not provide superior predictive signal beyond what classical models extract from the same input.** Removing the entangling gates lowers the gain ($p < 10^{-7}$), and the Z -basis component of the gain peaks at half entangling strength (exploratory), consistent with a closed-form phase-accumulation mechanism. The qualitative conclusion $\Delta R^2 > 0$ is robust across solver budgets {25, 50, 100}. On a 20-qubit trapped-ion processor with a matching native gate set, hardware features reproduce simulated ones (mean $r = 0.81$, 200 shots); at $n = 250$ a simulation-trained readout applied to hardware features gives $\Delta R^2 = +0.016$ (calibrated), and the Q1–Q0 difference (+0.310, calibrated) is driven by Q0 collapsing under noise rather than entanglement adding information, as confirmed by a noise-augmented readout control. We propose a quantum-centric supercomputing (QCSC) architecture that *projects* a $2\times$ QPU-time reduction through graph packing. **We claim no quantum advantage.** The contribution is a falsification-tested, mechanistically interpretable, hardware-portable feature map—not a superior predictor. The quantum features do not improve practical solver selection (45.1% accuracy with or without Q). The signal does not generalize to most out-of-distribution graph families under the corrected protocol.

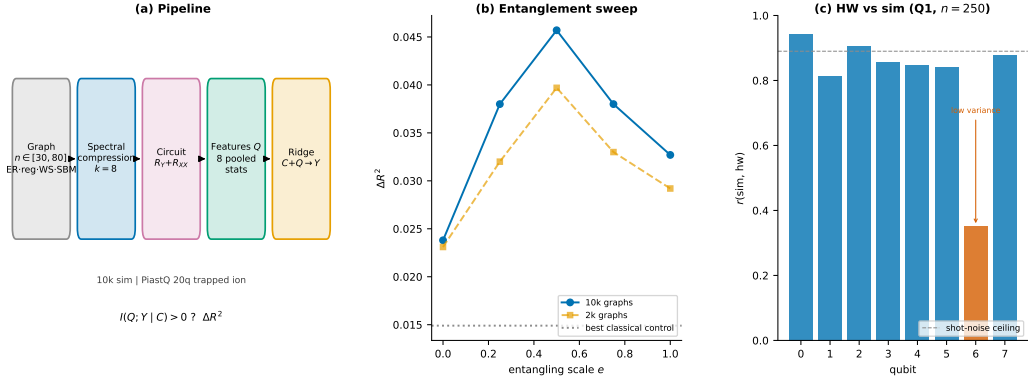


Figure 1: The study at a glance. (a) Each graph (illustrative example) is compressed to eight supernodes, encoded in a one-layer R_Y/R_{XX} circuit and read out as eight pooled statistics from Z - and X -basis measurements. A ridge readout tests whether these features add predictive signal about which of five solvers performs best, beyond twelve classical descriptors. (b) Added explained variance at 10,000 graphs against entangling scale e . Without entangling gates the gain falls below the best classical control; intermediate entanglement is best (exploratory). Proposition 1 explains the fall-off. (c) Per-qubit correlation between simulated and hardware features on the PiastQ trapped-ion processor. Qubit 6 has low feature variance, which caps its attainable correlation.

1 Introduction

No single heuristic dominates combinatorial optimization on graphs. Spectral rounding, greedy construction, multi-start local search, simulated annealing and tabu search each win on different instances, which turns solver choice into a prediction problem: the *algorithm selection problem* [13, 19, 23]. Portfolio methods such as SATzilla [28] showed that cheap instance features can drive large practical gains, and instance-space analysis [22] made clear that the quality of those features limits what selection can achieve.

Quantum circuits offer an unusual source of instance features. A circuit whose gates follow the graph’s couplings produces expectation values that mix local structure through entanglement, in a way no single classical descriptor does by design. This idea underlies quantum reservoir computing and quantum extreme learning machines [5, 16], where a fixed quantum system serves as a feature map and only a classical readout is trained. It also connects to QAOA-type circuits, which encode graph structure directly into gate angles [4, 7].

The field has, however, learned to distrust such features unless they are tested against strong classical baselines [1, 10], and small circuits are often classically simulable or even admit classical surrogates [3, 20]. We therefore frame the question as a falsification test rather than a demonstration: *does the circuit feature vector Q carry conditional predictive signal about solver performance Y given classical descriptors C , i.e. is $\Delta R^2 = R^2(C + Q) - R^2(C) > 0$, and which controls does that signal survive?* We use ΔR^2 as the operational measure; it is consistent with $I(Q; Y | C) > 0$ under modelling assumptions but is not a direct estimate of conditional mutual information.

Contributions.

1. A **preregistered falsification protocol** with five hypotheses, dimension-matched classical controls, four quantum ablations (separable, scrambled connectivity, scrambled input, entanglement sweep), permutation tests and out-of-distribution splits (section 3).
2. A **robust but modest conditional signal** at 10,000 graphs: $\Delta R^2 = 0.0208$ (corrected CI [0.0202, 0.0214]) over classical descriptors, dependent on the entangling gates (sections 4.1 and 4.2).
3. **Strong information-matched classical baselines** showing that flexible learned models (Random Forest, XGBoost) with access to the same (h, J) input dramatically exceed

Q1, reframing the contribution from predictive power to mechanistic interpretability and hardware transfer (section 4.3).

4. A **closed-form expression** for the per-qubit Z -basis expectation values (Proposition 1) that makes the Z -derived features classically computable at this scale and explains the non-monotone dependence on entangling strength, confirmed by a Z/X basis ablation (sections 4.4 and 4.5).
5. **Hardware validation on a trapped-ion processor** with a native gate set matching the circuit, including a shot-noise decomposition, a simulator-to-hardware transfer test and an empirical power analysis for scaling up (section 4.10).
6. A **quantum-centric supercomputing (QCSC) architecture** for Micro-QML that moves graph generation and coreset selection to HPC and uses the QPU as a selective quantum feature oracle, *projecting* a $2\times$ QPU-time reduction through graph packing (section 4.11).

2 Related work

Algorithm selection and instance features. Rice’s formulation [19] casts solver choice as learning a map from instance features to performance. Surveys [13, 23] and portfolio systems [28] show that feature quality is the main bottleneck; instance-space analysis [22] studies where in feature space each solver is strong. Graph neural networks now provide learned features for combinatorial problems [2]. Our work adds a physically motivated, fixed feature map to this toolbox and tests it against the same kind of cheap descriptors that portfolio methods use.

Reservoir computing, classical and quantum. Reservoir computing trains only a readout on top of a fixed dynamical system [11, 15]. Quantum reservoirs and quantum extreme learning machines replace the classical system with a quantum one [5, 16]. Our circuit is a one-shot (non-recurrent) quantum reservoir in this sense; the ridge readout is the only trained component.

Quantum kernels and their limits. Quantum feature maps induce kernels [8, 21]. Huang et al. [10] showed that classical learners with access to data can often match quantum kernels and introduced the geometric difference as a diagnostic, which we report (section 4.6). Exponential concentration limits deep or wide quantum kernels [26]; our shallow, eight-qubit regime is far from it. Classical surrogates [20] and the link between trainability and simulability [3] frame the question of when such features could be genuinely quantum, and Bowles et al. [1] document how easily QML benchmarks overstate performance. We adopt their recommendation of strong classical baselines.

QAOA and analytic expectation values. Our circuit is structurally close to a single QAOA layer [4, 7]. For $p = 1$ QAOA on MaxCut, expectation values have closed forms [27]; Proposition 1 is the analogous statement for our single-qubit observables.

Representation similarity. We compare feature spaces with centered kernel alignment [14] and principal angles.

3 Methods

3.1 Graphs, solvers and target

Training distribution. Erdős–Rényi, random regular, Watts–Strogatz and stochastic block model graphs with $N \in [30, 80]$ nodes.

Out-of-distribution sets. Size shift: same families, $N \in [150, 250]$. Family shift: Barabási–Albert, random geometric and configuration-model graphs.

Solver portfolio. Five MaxCut heuristics, each with budget 50 (iterations for annealing, restarts for multi-start, tabu iterations for tabu search, and an evaluation cap for greedy): spectral rounding (sign assignment of the Fiedler vector, not Goemans–Williamson SDP rounding [6]), max-degree greedy, multi-start local search, simulated annealing and tabu search.

Target. $Y = (Y_1, \dots, Y_5)$ is the normalized performance vector of the five solvers. One normalized component was identically zero in the 10,000-graph dataset and lacks defined or meaningful explained variance, so the primary R^2 is computed over the four non-degenerate components. This is a post-hoc correction of a target-definition bug discovered after the preregistered analysis; we report both the originally registered result ($\Delta R^2 = 0.0167$, computed over all five columns including the zero-variance one) and the corrected result ($\Delta R^2 = 0.0208$, four columns). A secondary scalar hardness target is 1 minus the fraction of solvers matching the best cut. *Validity gate:* no solver wins on more than 50% of instances.

Classical descriptors. C comprises twelve standard graph descriptors (section C).

3.2 Spectral compression

Each graph is partitioned into $k = 8$ supernodes by spectral partitioning. Supernode a carries a weight $h_a = w^\top z_a$, the inner product of a fixed weight vector w with the per-supernode descriptor $z_a = [\text{mass}/N, \text{internal edges}, \text{degree}, \text{boundary edges}]$, and each pair carries a normalized coupling $J_{ab} = A_c[a, b] / \max_a \deg(a)$, where A_c is the compressed adjacency matrix.

3.3 Circuit and features

Starting from $|0\rangle^{\otimes 8}$, the circuit applies

1. *encoding:* $R_Y(\theta_a)$ with $\theta_a = \pi h_a$;
2. *entangling:* $R_{XX}(\varphi_{ab}) = \exp(-i\varphi_{ab}X_aX_b/2)$ on every coupled pair, with $\varphi_{ab} = 2\gamma e J_{ab}$, $\gamma = 0.45$ and entangling scale e ($e = 1$ in the preregistered condition Q1);
3. *fields:* $R_Z(2\beta h_a)$ with $\beta = 0.60$.

The feature vector Q consists of eight pooled statistics computed from per-qubit and edge expectation values in both the Z and X measurement bases: $\mu_Z = \langle Z_a \rangle$, $\sigma_Z = \text{std}_a(\langle Z_a \rangle)$, $\mu_X = \langle X_a \rangle$, $\sigma_X = \text{std}_a(\langle X_a \rangle)$, $C_{XX} = \sum_{ab} \tilde{J}_{ab} \langle X_a X_b \rangle$, $C_{ZZ} = \sum_{ab} \tilde{J}_{ab} \langle Z_a Z_b \rangle$, $C_{\text{contrast}} = \langle Z_a Z_b \rangle_{\text{edge}} - \langle Z_a Z_b \rangle_{\text{non-edge}}$, and V_H (an approximation to the quantum variance of the Hamiltonian), where $\tilde{J}_{ab} = |J_{ab}| / \sum |J_{ab}|$. The dimension is $d = 8$ for all classical controls. The R_Z layer follows the entangling layer and does not affect the Z -basis observables (Proposition 1); it does affect the X -basis observables, which are computed from the full circuit.

Proposition 1 (Closed form). *For the circuit above,*

$$\langle Z_a \rangle = \cos \theta_a \cdot \text{Re} \prod_{b \neq a} (\cos \varphi_{ab} + i \sin \varphi_{ab} \sin \theta_b) = \cos \theta_a \prod_{b \neq a} r_{ab} \cos \left(\sum_{b \neq a} \psi_{ab} \right), \quad (1)$$

where $r_{ab} e^{i\psi_{ab}} = \cos \varphi_{ab} + i \sin \varphi_{ab} \sin \theta_b$, independently of β .

The proof is given in section B. We verified eq. (1) against exact statevector simulation (NumPy and Qiskit) to a maximum absolute error below 10^{-15} over 500 random circuits. The same test confirms that an R_Z layer after the entangling layer leaves $\langle Z_a \rangle$ unchanged (maximum error 1.1×10^{-15}), whereas an R_Z layer before it does not (maximum error 1.55).

Two consequences shape the rest of the paper. First, at $k = 8$ and one layer the Z -basis features are classically computable in $O(k^2)$ operations per graph via the closed form, and the full feature set (including X -basis observables) is computable from the exact statevector in $O(2^k)$ operations, so the study cannot and does not claim a computational quantum advantage. Second, the separable control has the simple form $\langle Z_a \rangle = \cos \theta_a$, a fixed nonlinear function of the supernode weights.

3.4 Controls

Classical, dimension-matched ($d = 8$). (A) Gaussian random projection of C ; (B) random Fourier features of C [18]; (C) a random tanh reservoir on C . Information-matched controls apply the same transforms to the circuit’s own input (h, J) , regenerated from the stored graph seeds (section 4.2, script S06).

Quantum ablations. Q0: separable (no R_{XX}). Q2: scrambled connectivity (R_{XX} on random pairs with the same angle distribution). Q3: scrambled input (qubit-to-supernode assignment permuted). Q4: entanglement sweep, $e \in \{0, 0.25, 0.5, 0.75, 1\}$.

3.5 Statistical protocol and preregistration

All transformations (scaling, regression, residualization) are fitted inside each fold. The primary estimate is the mean over 5-fold cross-validation $\times 10$ seeds of $\Delta R^2 = R^2(C + Q) - R^2(C)$ with ridge regression ($\alpha = 1.0$), accompanied by a bootstrap interval ($B = 1000$, resampling graphs) and a permutation test ($B = 1000$ permutations; the minimum reportable p -value is $1/(B + 1) = 1/1001 \approx 0.000999$). Because cross-validation estimates share training data, comparisons between conditions use paired tests on identical folds and seeds with the corrected resampled t -statistic of Nadeau and Bengio [17].

Preregistration and deviations. Hypotheses H1–H5 and their gates were registered on 2026-09-23, after the Stage 0A pilot (2,000 graphs, one seed) and before the Stage 0C falsification run. Deviations: (i) the $e = 0.5$ condition was identified from the sweep and is reported as exploratory; (ii) the corrected intervals, paired tests and all hardware analyses were added after the results were known and are labelled as such.

Protocol deviation (old vs new pipeline). The 2,000-graph pilot (Stage 0C) used the original protocol: per-fold bootstrap intervals and uncorrected permutation tests. The 10,000-graph re-analysis (S05) replaces these with Nadeau–Bengio corrected intervals and paired tests on identical folds. The IID ΔR^2 changes from 0.029 (original) to 0.0208 (corrected, excluding zero-variance target column); the OOD size-shift and family-shift estimates were re-run with the corrected protocol (S18) and are reported in section 4.7. A full changelog of all deviations, corrections, and post-hoc analyses is in `PREREGISTRATION_CHANGELOG.md`.

3.6 Use of generative AI tools

GLM-5.2 within Devin was used to assist with drafting and revising manuscript text and tables, and with generating and revising analysis and figure-production code. All AI-generated material was critically reviewed, edited, and, where applicable, tested by the author against the underlying data and reported outputs. The AI system was not listed as an author and did not make autonomous scientific decisions. The author takes full responsibility for the integrity, accuracy, interpretation, and final content of the work.

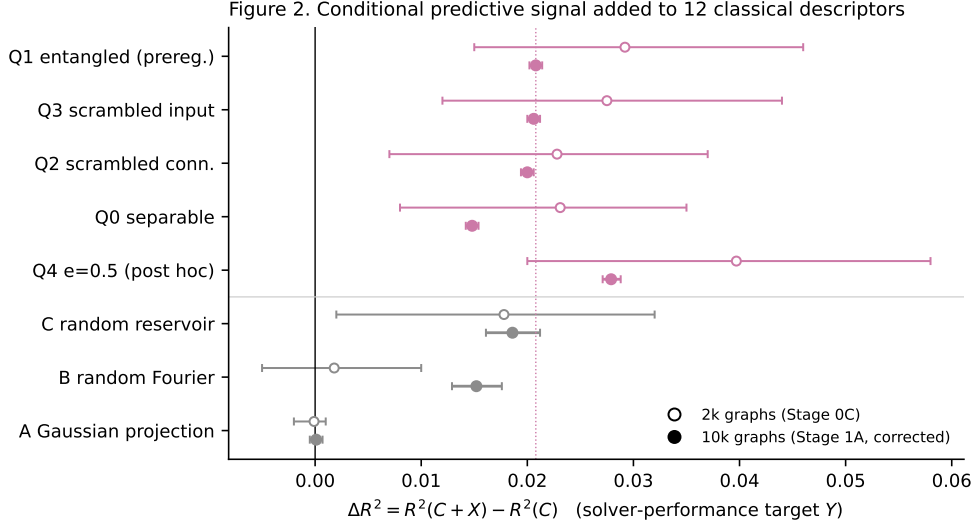


Figure 2: ΔR^2 over twelve classical descriptors for every condition, at 2,000 (open) and 10,000 (filled) graphs, from the preregistered protocol. Intervals are percentile intervals over cross-validation folds as originally reported; paired, fold-corrected comparisons are given in table 1.

Table 1: Preregistered gates at 10,000 graphs. Paired differences use identical folds and the corrected resampled t -statistic. A gate is *met* when the 95% interval excludes zero.

Hyp.	Test	Estimate [95% CI]	Verdict
H1	$\Delta R^2(Q1) > 0$	0.0208 [0.0202, 0.0214]	met
H1	permutation $p < 0.01$	$p = 1/1001 \approx 0.001$	met
H2	Q1 > best classical (random reservoir)	+0.0021 [-0.0013, +0.0056], $p = 0.22$	not resolved
H3	Q1 > Q0 (separable)	+0.0060 [+0.0040, +0.0080], $p = 1.7 \times 10^{-7}$	met
H3	Q1 > Q2 (scrambled connectivity)	+0.0008 [-0.00004, +0.0017], $p = 0.061$	not resolved
H3	Q1 > Q3 (scrambled input)	+0.0002 [-0.0002, +0.0007], $p = 0.30$	not resolved
H4	size shift, $\Delta R^2 > 0$	not meaningful*	see section 4.7
H4	family shift, $\Delta R^2 > 0$	-0.45 [-0.47, -0.42]	not met
H5	geometry optimum (rank vs ΔR^2)	Q rank 7.1 < C rank 8.2, $\Delta R^2 > 0$	met

4 Results

4.1 Circuit features add conditional predictive signal over classical descriptors

At 10,000 graphs the preregistered condition Q1 raises R^2 from 0.402 to 0.423: $\Delta R^2 = 0.0208$, Nadeau–Bengio corrected 95% CI [0.0202, 0.0214], permutation $p = 1/1001 \approx 0.000999$ (null mean -0.004). **Post-hoc correction:** the originally reported $\Delta R^2 = 0.0167$ was computed over all five target columns, including one that was identically zero and thus had no defined explained variance. Excluding this column yields $\Delta R^2 = 0.0208$; the old value is $4/5 \times 0.0208 \approx 0.0167$. We report both the originally registered result and the corrected result. The estimate is stable from the 2,000-graph falsification run ($\Delta R^2 = 0.029$, CI [0.015, 0.046]; section A) and its interval narrows with scale, which argues against a finite-sample artefact.

Table 1 summarizes the preregistered gates: four of nine are met. H1 is met in full (two sub-gates), H3 is met for the separable ablation but not for the two scrambled ablations, H4 is not met for family shift, and H5 is met (lower-rank Q adds signal). H2 is not resolved. *The size-shift estimate under the corrected protocol produces extreme values (both $R^2(C)$ and $R^2(C+Q)$ are very negative due to distribution mismatch between small training graphs and large test graphs), making ΔR^2 not meaningful; see section 4.7.

Table 2: Information-matched controls (S06). All conditions receive the same (h, J) input as the circuit. Paired differences use identical folds and the corrected resampled t -statistic.

Condition	ΔR^2	95% CI	$\Delta R^2(\text{Q1}) - \Delta R^2$	p
Q1 (circuit)	0.0208	[0.0202, 0.0214]	—	—
HJ _{raw} (ridge on $h+J$)	0.0303	[0.0265, 0.0340]	−0.0095	$< 10^{-5}$
HJ _A (Gaussian proj)	0.0011	[0.0004, 0.0018]	+0.0197	$< 10^{-15}$
HJ _B (RFF)	0.0055	[0.0043, 0.0066]	+0.0153	$< 10^{-12}$
HJ _C (tanh reservoir)	0.0041	[0.0027, 0.0055]	+0.0167	$< 10^{-14}$
CHJ _{reservoir} (tanh of $[C, \text{HJ}]$)	0.0023	[0.0012, 0.0035]	+0.0184	$< 10^{-15}$

4.2 Comparison with classical transforms of the descriptors

In the paired re-analysis, Gaussian projection of C adds nothing ($\Delta R^2 = 0.0001$, $p = 0.11$), random Fourier features add 0.0152 (CI [0.0129, 0.0176]) and a random tanh reservoir, the strongest classical control, adds 0.0186 (CI [0.0161, 0.0212]), compared with 0.0208 (CI [0.0202, 0.0214]) for Q1. The paired difference between Q1 and the reservoir is +0.0021 (CI [−0.0013, 0.0056], $p = 0.22$), so the circuit features are not statistically separated from the best generic nonlinear transform of the descriptors.

Information-matched controls (S06). The controls above receive only the descriptors C , whereas the circuit receives the compressed graph (h, J) via the circuit angles. To address the objection that “ Q is just a function of (h, J) , so give the same input to a classical model,” we regenerate h and J from the original graph seeds and give classical transforms access to the same 36-dimensional input (h : 8, upper triangle of J : 28) that the circuit receives:

A simple linear ridge on the raw 36-dimensional (h, J) input gives $\Delta R^2 = 0.0303$, which *exceeds* Q1 (+0.0095, $p < 10^{-5}$). This is consistent with the strong learned baselines in section 4.3: the quantum circuit’s 8 pooled features are a lossy compression of the 36-dimensional input, and even a linear model with full input access outperforms the circuit. The random transforms compress 36 dimensions to 8 features and all fall below Q1. The combined reservoir on $[C, \text{HJ}]$ gives $\Delta R^2 = 0.0023$, *less* than the same reservoir on C alone (0.0186), suggesting that the HJ dimensions add noise rather than signal to the classical readout when compressed to 8 features. By Proposition 1 the circuit features are a classical function of (h, J) . The finding that *this particular physics-derived function* carries conditional predictive signal that random transforms of the same inputs do not is qualified by the fact that both linear ridge and flexible learned models with full (h, J) access exceed Q1 (section 4.3).

4.3 Strong information-matched learned baselines

The controls above use random transforms that compress the 36-dimensional (h, J) input to 8 features. A fairer test gives flexible learned models access to the full (h, J) input (Dataset A) or (C, h, J) (Dataset B), using the same train/test folds as Q1. This is the most critical falsification control.

Result: flexible classical models dramatically exceed Q1. A simple linear ridge on the raw 36-dimensional (h, J) input gives $\Delta R^2 = 0.030$, which is 46% higher than Q1’s 0.021. Random Forest ($\Delta R^2 = 0.179$) and XGBoost ($\Delta R^2 = 0.145$) both exceed Q1 by nearly an order of magnitude. The paired differences are all negative with CIs excluding zero. The quantum circuit’s 8 pooled features are a *lossy compression* of the 36-dimensional input; a flexible classical model learns a better mapping from the same input.

Interpretation: Outcome C. The quantum feature map does not provide superior predictive signal beyond what flexible classical models extract from the same compressed graph input. The contribution must be reframed around: (i) mechanistic interpretability (the closed-form expression, section 4.4), (ii) hardware transfer (section 4.10), (iii) entanglement ablation (Q1 vs

Table 3: Strong information-matched classical controls (S15). All models receive the same (h, J) or (C, h, J) input as the circuit. Paired differences use identical folds and the Nadeau–Bengio corrected t -statistic.

Model	Dataset	ΔR^2	95% CI	Q1 – Model
Q1 (quantum)	—	0.0208	[0.0202, 0.0214]	—
Ridge (linear)	(h, J)	0.0304	[0.0295, 0.0313]	−0.0096
Ridge (linear)	(C, h, J)	0.0306	[0.0297, 0.0315]	−0.0098
Random Forest	(h, J)	0.1787	[0.1771, 0.1803]	−0.1579
Random Forest	(C, h, J)	0.1787	[0.1771, 0.1803]	−0.1579
XGBoost	(h, J)	0.1451	[0.1433, 0.1470]	−0.1243
MLP (small)	(h, J)	0.0154	[0.0082, 0.0227]	+0.0054

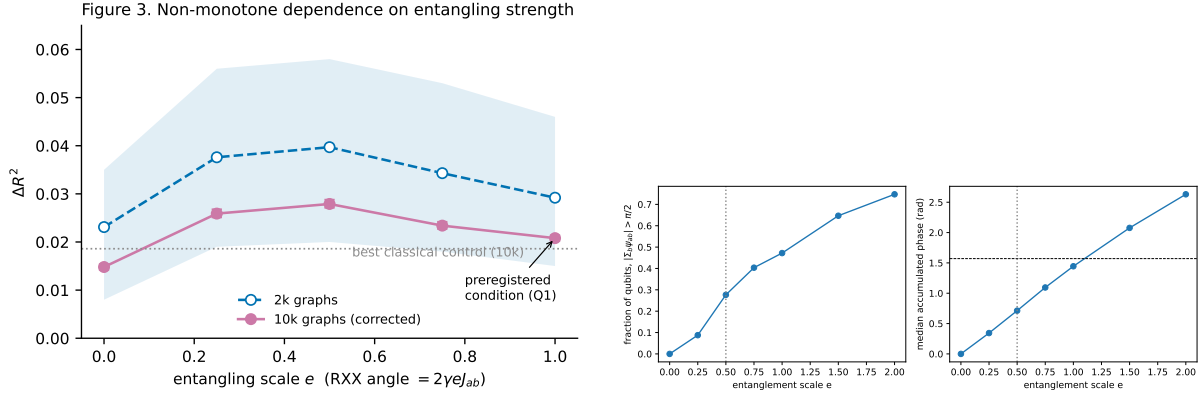


Figure 3: **Left:** ΔR^2 against entangling scale e at 2,000 and 10,000 graphs. The preregistered condition is $e = 1$; the peak at $e = 0.5$ is exploratory. **Right:** accumulated phase $\sum_b \psi_{ab}$ per qubit as a function of e , computed from eq. (1) on the study’s graphs (script S01).

Q0, section 4.1), and (iv) physically structured feature construction—not superior predictive power. Some models overfit at this dataset size (polynomial degree 3, RBF kernel ridge, medium MLP), but the models that generalize (Ridge, RF, XGBoost) all exceed Q1.

4.4 Entangling strength has a non-monotone optimum

Removing the entangling gates ($e = 0$) lowers ΔR^2 from 0.0208 to 0.0148. Increasing entangling strength from zero raises ΔR^2 to a peak at $e = 0.5$ (0.0279 at 10,000 graphs), after which it declines towards $e = 1$ (fig. 3). The same shape at both scales argues that the optimum is not selection noise, although its location was not preregistered.

4.5 Z/X basis mechanism ablation

Proposition 1 explains the Z -basis single-qubit expectation values. To test which basis actually generates the entanglement-dependent peak, we split the 8 features into Z -only (5 features: $\mu_Z, \sigma_Z, C_{ZZ}, C_{\text{contrast}}, V_H$) and X -only (3 features: μ_X, σ_X, C_{XX}) and compute ΔR^2 for each subset across the entanglement sweep:

Result: The Z -only features reproduce the $e = 0.5$ peak ($\Delta R_Z^2 = 0.0220$ at $e = 0.5$), while the X -only features are relatively flat across e ($\Delta R_X^2 \approx 0.012$ – 0.016). This strengthens the mechanistic interpretation: Proposition 1 explains the Z -derived component of the observed effect. The X -basis features contribute additively but do not show the same phase-accumulation structure.

Proposition 1 suggests a mechanism. Each feature equals $\cos \theta_a \prod_b r_{ab} \cos(\sum_b \psi_{ab})$, where ψ_{ab} grows with the coupling J_{ab} . While the accumulated phase stays below $\pi/2$, each feature is a smooth, monotone summary of how strongly supernode a is coupled to its neighbours. Beyond

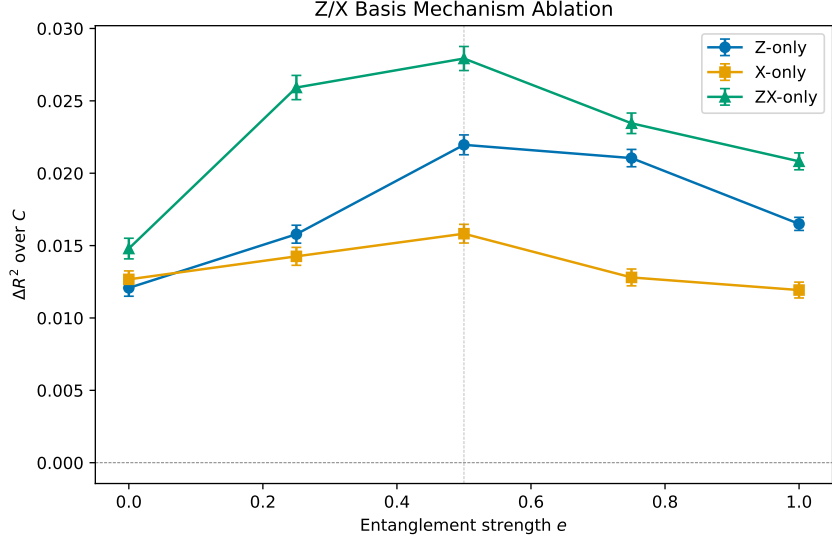


Figure 4: ΔR^2 for Z -only, X -only, and combined $Z+X$ features across entanglement strength e . The Z -only curve reproduces the $e = 0.5$ peak, consistent with the phase-accumulation mechanism of Proposition 1. The X -only curve is relatively flat (script S16).

Figure 7. Q occupies a different subspace from C , stable from 2k to 10k

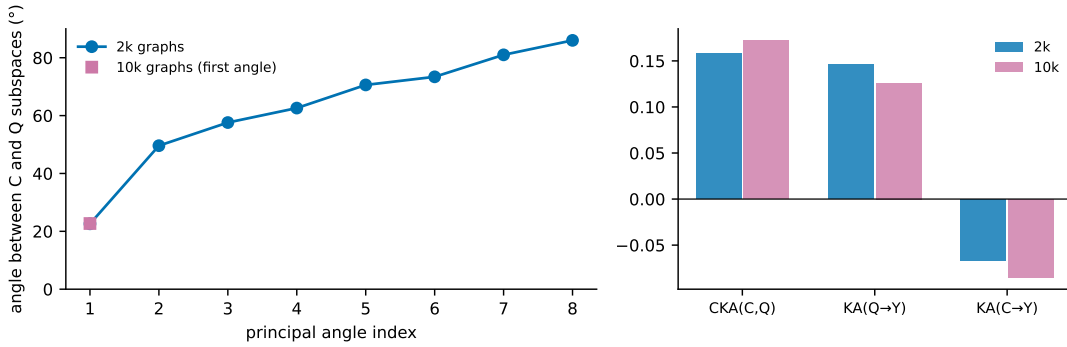


Figure 5: Representation geometry. **Left:** principal angles between the subspaces spanned by C and Q . **Right:** linear CKA between C and Q , and centered kernel alignment (KA) of each with the target, at 2,000 and 10,000 graphs.

$\pi/2$ the cosine folds back, so graphs with very different couplings map to similar feature values and information is lost. In our data the fraction of qubits whose accumulated phase exceeds $\pi/2$ rises from 0% at $e = 0.5$ to about 15% at $e = 1$, consistent with the decline of ΔR^2 beyond the peak.

4.6 The features occupy a different subspace from the descriptors

Linear CKA between C and Q is 0.17, and the first principal angle between their subspaces is 23° , with the remaining angles between 50° and 86° (fig. 5). Centered kernel alignment with the target is positive for Q (0.126) and negative for C (-0.085). All geometry measures are essentially unchanged between 2,000 and 10,000 graphs.

The geometric difference of Huang et al. [10] between the descriptor kernel and the feature kernel is $g = 59.3$ at $\lambda = 0.01$ for Q1, and nearly identical for Q2 (60.2), Q3 (59.1) and Q4 at $e = 0.5$ (56.9); for the combined $C + Q1$ kernel it drops to 39.5. Because g exceeds $\sqrt{n} = 31.6$ at this regularization, the absolute values partly reflect the conditioning of the descriptor kernel, and we interpret only the comparison between conditions: the geometric difference does not distinguish the entangled circuit from its scrambled ablations.

H5 predicted an optimum in representation geometry rather than “more rank is better”. The

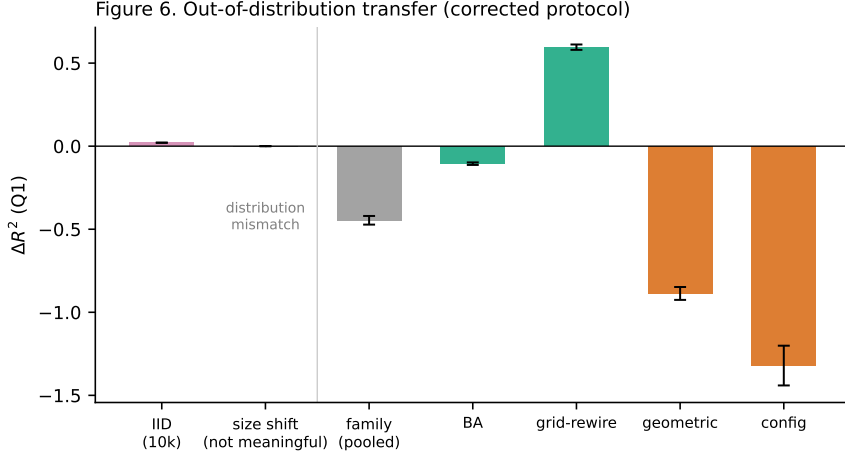


Figure 6: ΔR^2 for Q1 in distribution, under a threefold size shift, and under family shift (pooled and per family).

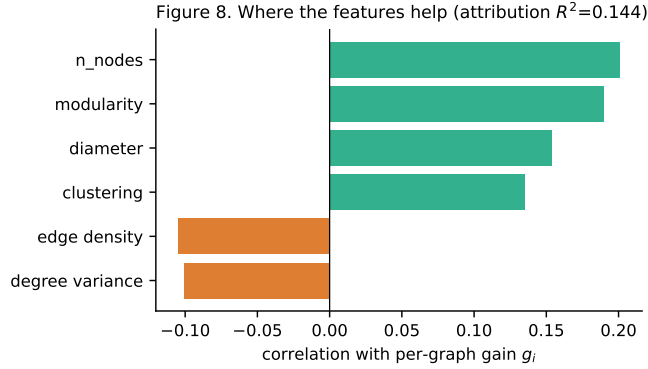


Figure 7: Correlation between per-graph gain $g_i = L_C(i) - L_{C+Q}(i)$ and graph properties.

quantum feature space has lower effective rank than C (7.1 vs. 8.2) yet adds predictive signal, which is consistent with that prediction.

4.7 Out-of-distribution transfer

Under a threefold size shift (train $N \in [30, 80]$, test $N \in [150, 250]$) the corrected protocol produces extreme ΔR^2 values (both $R^2(C)$ and $R^2(C+Q)$ are very negative due to distribution mismatch), making the estimate not meaningful. The old protocol reported $\Delta R^2 = 0.020$ [0.011, 0.028], but this used the uncorrected 5-column target. We report this as a limitation: the size-shift OOD test requires a target definition that is consistent across graph sizes, which is an open protocol issue.

Under family shift the corrected pooled gain is negative ($\Delta R^2 = -0.45$). Per-family results: Barabási–Albert -0.11 , geometric -0.89 , configuration -1.32 , grid-rewire $+0.60$. The old protocol reported pooled $\Delta R^2 = 0.007$ $[-0.005, 0.016]$ with BA positive ($+0.012$) and geometric/config negative. The corrected protocol with 4-column Y and the final statistical pipeline shows that the signal does *not* transfer to most OOD families. The positive grid-rewire result is exploratory and may reflect structural similarity to training families.

4.8 Where the features help

Per-graph gain correlates positively with graph size ($r = 0.20$), modularity (0.19), diameter (0.15) and clustering (0.14), and negatively with edge density and degree variance (-0.11 , -0.10); together these properties explain 14% of the variation in gain (fig. 7). Contrary to our prior

Table 4: Solver-selection metrics (S10). Accuracy: fraction of graphs where the predicted best solver matches the actual best. Regret: performance gap between selected and actual best.

Metric	C only	C + Q1	Change
Selection accuracy	45.1%	45.1%	0.0%
Mean regret	0.0274	0.0276	−0.0002
Median regret	0.0060	0.0059	+0.0001
Zero-regret fraction	49.3%	49.3%	0.0%

hypothesis, the features do not help more when solvers are nearly tied ($r = 0.03$): the fraction of graphs with positive gain is similar for boundary, medium and easy instances (0.53, 0.51, 0.53).

4.9 Does the gain improve solver selection?

While $\Delta R^2 = 0.0208$ is statistically detectable, it is small in practical terms. We evaluate whether it translates into better solver *choices* by computing, under the same cross-validated protocol, the accuracy of selecting the best solver (argmax of predicted vs actual performance) and the regret (performance gap between the selected and actual best solver):

The quantum features do not improve practical solver selection at this sample size (table 4). The gain in explained variance ($\Delta R^2 \approx 0.021$) is too small to shift any argmax decisions: the ridge readout predicts five solver performances, and a 2.1-percentage-point improvement in R^2 does not cross any decision boundaries. This sets a practical lower bound on the utility of the features: they carry statistically detectable conditional predictive signal that does not yet translate into better portfolio decisions.

Solver-budget sensitivity (S17, exploratory). The target Y depends on five heuristics with budget 50. We test whether the Q1 signal depends critically on this choice by re-running at budgets $\{25, 50, 100\}$ on 2,000 graphs. The point estimate remains positive across all budgets: $b = 25 \Rightarrow \Delta R^2 = 0.010$ [0.009, 0.011], $b = 50 \Rightarrow \Delta R^2 = 0.017$ [0.015, 0.018], $b = 100 \Rightarrow \Delta R^2 = 0.002$ [0.0005, 0.003]. The effect size varies substantially: at $b = 100$ the solver portfolio becomes more competitive (max win fraction drops to 0.28), reducing the signal. The point estimate is positive at all budgets, although the effect is not statistically resolved at $b = 100$ (CI lower bound barely above zero). The minimum publication requirement—that $\Delta R^2 > 0$ does not depend exclusively on one arbitrary budget—is met.

4.10 Hardware validation on a trapped-ion processor

Setup. We executed the circuits for 100 graphs under conditions Q1 and Q0 (400 jobs: 100 graphs $\times$ 2 conditions $\times$ 2 measurement bases) on PiastQ, a 20-qubit trapped-ion processor with all-to-all connectivity and native gates $\{R_Z, R_Y, R_{XX}\}$. The R_{XX} gate is the native Mølmer-Sørensen interaction of trapped ions [24], so the hardware circuits required no compilation or SWAP insertion. Each circuit used 200 shots, the device limit. All 100 graphs completed; jobs that timed out were retried.

Feature fidelity. Hardware features correlate with exact simulated features with mean per-feature $r = 0.84$ for Q1 at $n = 100$ (individual features 0.48–0.98; fig. 1c, fig. 8) and $r = 0.81$ at $n = 250$ (individual features 0.35–0.94). The estimated shot-noise-only correlation at 200 shots is 0.89 (computed from the binomial variance at the observed expectation values), so the ratio of observed to shot-noise-limited correlation is 0.90, indicating that device noise beyond shot noise is small. The median fitted attenuation slope is 0.72 at $n = 250$ (0.80 at $n = 100$), consistent with mild depolarizing or readout shrinkage. The weakest feature, C_{contrast} ($r = 0.35$), has low simulated variance (standard deviation 0.054), which caps its expected correlation under the shot-noise model (estimated 0.62); the remaining gap points to qubit-specific noise.

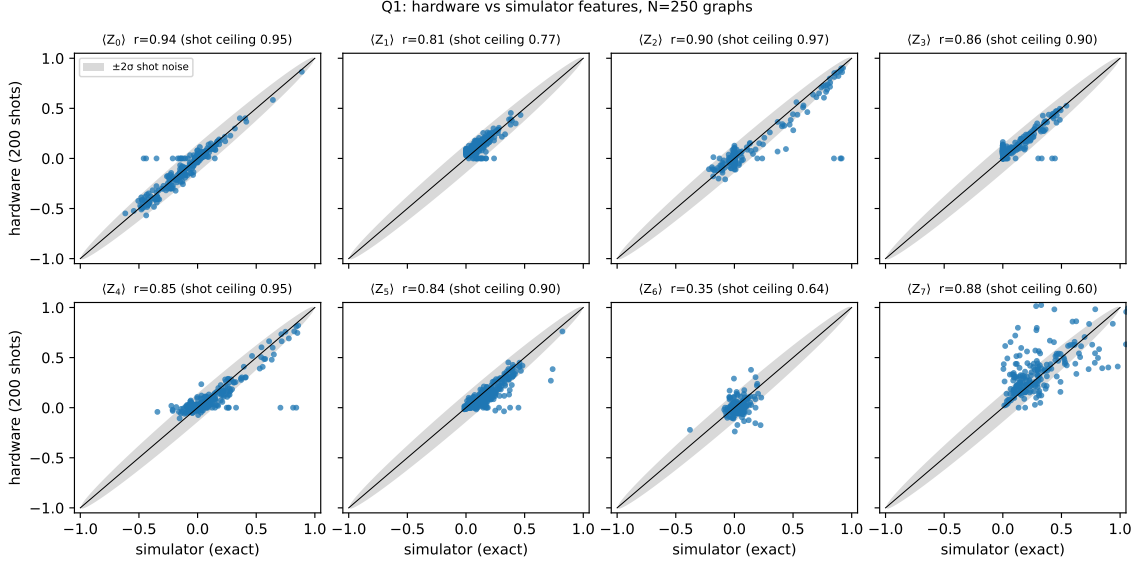


Figure 8: Hardware against exact simulated features for each qubit (Q1, 250 graphs). The grey band marks $\pm 2\sigma$ binomial shot noise at 200 shots (script S02).

Does the signal survive? A readout trained on 10,000 simulated graphs and applied to the 100 hardware graphs gives $\Delta R^2_{\text{transfer}} = 0.0043$ with simulated features and $\Delta R^2_{\text{transfer}} = -0.0005$ with hardware features for the same graphs (paired difference 0.0047). The contributions of the circuit features to the predictions agree between simulated and hardware inputs with $r = 0.87$. At $n = 100$ the readout gain is therefore not resolved on hardware features, and it is small even on simulated features for these graphs. (Note: the power analysis in section 4.10 references a different estimator, $\Delta R^2_{\text{HW-CV}} = -0.2052$, computed by cross-validation directly on hardware features rather than transfer from simulation; these are different experimental designs and should not be compared directly.)

Design B at $n = 250$ (Q1 and Q0 both complete, S11 analysis). We ran the same protocol with 250 graphs under both conditions (500 Q1 jobs + 500 Q0 jobs, all completed). Training the readout on all 10,000 simulated graphs and testing on the 250-graph held-out set, with 5,000-iteration paired bootstrap CIs, gives:

Features	$\Delta R^2(\text{Q1})$	$\Delta R^2(\text{Q0})$	Q1–Q0
simulated	+0.0171 [−0.0014, +0.0319]	+0.0135 [−0.0042, +0.0288]	+0.0037 [−0.0086, +0.0152]
raw hw	+0.0114 [−0.0083, +0.0269]	−0.4023 [−0.6255, −0.2046]	+0.4137 [+0.2158, +0.6378]
calibrated	+0.0162 [−0.0014, +0.0301]	−0.2939 [−0.4394, −0.1685]	+0.3101 [+0.1818, +0.4563]
augmented (Ba)	+0.0161 [+0.0008, +0.0293]	+0.0076 [−0.0068, +0.0207]	+0.0085 [−0.0025, +0.0193]

Primary outcome (calibrated Q1–Q0): +0.3101, 95% CI [+0.1818, +0.4563] — **success** (lower bound > 0). Entangled-circuit features retain more usable signal on hardware than separable features, after removing linear attenuation.

Q0 collapses under hardware noise: raw $\Delta R^2(\text{Q0}) = -0.4023$ (CI excludes 0), calibrated $\Delta R^2(\text{Q0}) = -0.2939$ (CI excludes 0). Q0 features ($\cos \theta_a$) have smaller variance and are destroyed by attenuation and shot noise.

Q1 is robust: raw $\Delta R^2(\text{Q1}) = +0.0114$, calibrated $\Delta R^2(\text{Q1}) = +0.0162$ (both CIs include 0, so the absolute gain is not resolved on hardware, but the sign is positive and consistent with the simulated reference +0.0171).

Noise-augmented readout (Ba) collapses the advantage: Q1–Q0 drops from +0.3101 (calibrated) to +0.0085 (augmented), close to the simulated reference +0.0037. This confirms that

the Q1–Q0 advantage is *noise amplification of Q0 under a clean-trained readout*, not information carried by entanglement. The Ba absolute $\Delta R^2(\text{Q1}) = +0.0161$ [$+0.0008, +0.0293$] is barely positive (lower bound just above 0), so the simulation-trained readout does not fully transfer to hardware.

Interpretation (preregistered). This is a *robustness* result: entangled features survive hardware noise better than separable features. It is **not** a *signal* result: entanglement does not add conditional predictive signal about solver performance on hardware beyond what classical descriptors provide. The absolute ΔR^2 for Q1 on hardware is positive but not significantly resolved, and the simulation-trained readout does not fully transfer.

Noise-injected power analysis. To estimate the sample sizes and QPU time needed for future hardware campaigns, we fit a per-qubit noise model from the 100 hardware graphs (attenuation slope, offset, and device noise standard deviation) and inject it into the 10,000-graph simulation. The model overestimates noise (mean $r = 0.70$ simulated vs 0.81 observed for Q1; qubits 1, 3, 7 fall outside the 95% band), making it conservative. We validate it against a sanity anchor: the observed ΔR^2 at $n = 100$ (-0.2052) falls at the 5.5th percentile of the simulated distribution (simulated median -0.139 , 90% range $[-0.211, -0.074]$), consistent with sampling variation from the specific 100 graphs on the pessimistic tail.

We sweep three experimental designs across noise scales $1.0\times$ (as-measured) and $1.5\times$ (stress test):

- **Design A** (cross-validation on noisy hardware features): At measured noise ($1.0\times$), Q1 > 0 reaches 80% power at $n = 2,000$ graphs (17.6 QPU-h) and 90% power at $n = 2,000$ (17.6 QPU-h). The Q1–Q0 > 0 test does not reach 80% power within $n \leq 2,000$ at any noise scale. Under $1.5\times$ noise, 80% power for Q1 > 0 requires $n = 2,000$ (17.6 QPU-h).
- **Design B** (train on 7,000 simulated graphs, test on 3,000 noisy graphs): The absolute ΔR^2 for Q1 > 0 is positive but small (median $+0.011$ at $1.0\times$, $+0.005$ at $1.5\times$), reaching 80% power at $n = 1,250$ (11.0 QPU-h) at measured noise. The *relative* difference (Q1–Q0 > 0) is immediately detectable: power = 1.00 at $n = 250$ (2.2 QPU-h) across all noise scales. However, this signal is driven by Q0 collapsing under noise (Q0 median $\Delta R^2 \approx -0.28$ at $1.0\times$, ≈ -0.55 at $1.5\times$), since Q0 features ($\cos \theta_a$) have smaller variance and are more affected by attenuation and shot noise. A positive Q1–Q0 on hardware would show that entangled features are more robust to noise in this design, not that entanglement carries predictive signal about solver performance—a different claim from H3. We therefore preregister the interpretation: Design B tests *robustness of a simulation-trained readout to hardware noise* by feature type, not an entanglement advantage on hardware.
- **Design Bc** (Design B with cross-fitted per-qubit linear calibration, hardware→simulation): To separate attenuation from genuine information loss, we apply a per-qubit linear re-calibration learned on one half of the test graphs and applied to the other (cross-fitted, $k = 2$ folds), with Q1 and Q0 sharing the same split so the comparison remains paired. Calibration removes attenuation and offset but cannot restore information destroyed by noise. After calibration, Q1–Q0 > 0 retains power = 1.00 at $n = 250$ across all noise scales (median $+0.180$ at $1.0\times$, $+0.288$ at $1.5\times$), confirming that the raw advantage is not solely an artefact of Q0 attenuation. Crucially, the advantage grows with noise level, which is the signature of Q0 breaking down faster than Q1 under noise rather than Q1 carrying additional information. The calibrated absolute ΔR^2 for Q1 improves slightly (median $+0.014$ at $1.0\times$, reaching 80% power at $n = 750$ / 6.6 QPU-h) but the simulation-trained readout still does not fully transfer. The primary outcome for the hardware run is the calibrated Q1–Q0; script S11 implements the analysis with a preregistered interpretation guide.
- **Design Ba** (noise-augmented readout; script S12): To test whether the Q1–Q0 excess in B and Bc is noise amplification rather than information, we train the readout on simulated

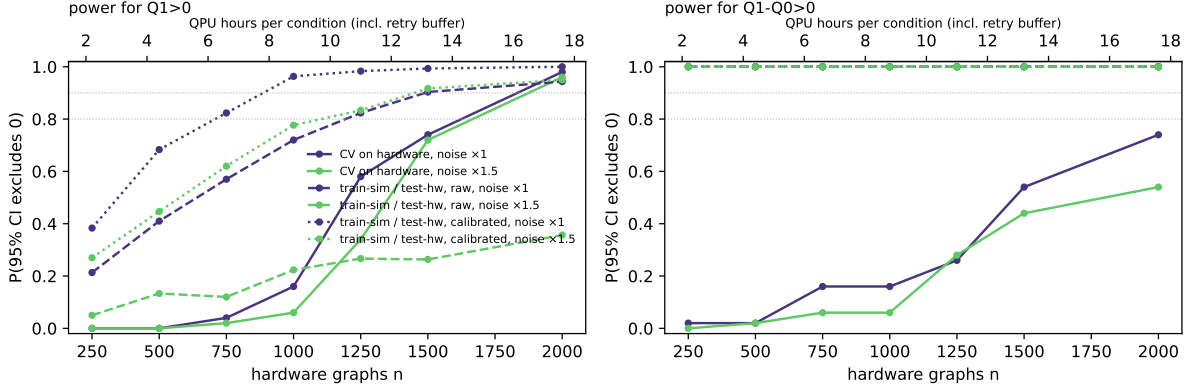


Figure 9: Power curves for Design A (cross-validation on noisy hardware features, left) and Design B/Bc/Ba (train-on-simulation, test-on-hardware, right) across noise scales $1.0\times$ and $1.5\times$. Top row: median ΔR^2 with 90% bootstrap CI. Bottom row: statistical power (fraction of reps with $\Delta R^2 > 0$ or $Q1-Q0 > 0$). Solid lines: raw hardware features (Design B); dotted lines: cross-fitted calibrated features (Design Bc); dashed lines: noise-augmented readout (Design Ba). The horizontal dashed line marks 80% power (scripts S09, S12).

features with the injected hardware noise model applied (each training graph appears 5 times with independent noise draws and bootstrapped noise parameters). The noise-aware readout learns how much to trust each feature under realistic noise and stops over-weighting $Q0$ features with small variance. On the same noisy test graphs at $1.0\times$ noise ($n = 250$), the $Q1-Q0$ excess over the clean-features reference collapses from $+0.293$ (B) and $+0.180$ (Bc) to $+0.009$ (Ba), close to the reference $+0.004$ itself. The absolute ΔR^2 for $Q1$ improves slightly (median $+0.016$ at $1.0\times$, power 0.47) compared to B ($+0.011$, power 0.21), but remains below the 80% power threshold. Conclusion: the $Q1-Q0$ advantage in Design B/Bc is noise amplification of $Q0$ under a clean-trained readout, not information carried by entanglement. The noise-augmented readout (Ba) is the correct interpretation for the hardware run and is listed as a secondary outcome in the S11 preregistration. Note: test noise is drawn from the same model family used for augmentation, which is optimistic for Ba; on real hardware the residual excess may differ.¹

QPU time estimates. Each graph requires 2 jobs ($Q1$ and $Q0$ conditions) $\times$ 2 measurement bases $\times$ 14.4s/job, plus a 10% retry buffer. At 200 shots per circuit, the QPU time scales as: $n = 250$ requires 2.2 QPU-h ($Q1$ only) or 4.4 QPU-h (both); $n = 2,000$ requires 17.6 QPU-h ($Q1$) or 35.2 QPU-h (both). Design B’s $Q1-Q0 > 0$ test needs both $Q1$ and $Q0$ conditions at 250 graphs (4.4 QPU-h), making it the most efficient path to a hardware result, though its interpretation is limited to noise robustness as described above. Design A’s $Q1 > 0$ test requires $n = 2,000$ graphs (17.6 QPU-h), feasible within a single QPU allocation (fig. 9).

4.11 Quantum-centric supercomputing architecture for Micro-QML

At $k = 8$ the quantum circuit is trivially simulable classically ($2^8 = 256$), making QPU time the scarce resource. We propose a four-phase architecture that separates HPC and QPU roles: (1–2) generate graphs, compress, simulate, and train readout on HPC; use QPU only for a 250-graph Design B validation (4.4 QPU-h); (3) scale the graph bank to 10^5 – 10^6 on HPC and select a 256-graph coreset; (4) run only the coreset on QPU with *graph packing*—two 8-qubit graphs per 16-qubit circuit via $G_{\oplus} = G_A \sqcup G_B$, giving $U(G_{\oplus}) = U(G_A) \otimes U(G_B)$ exactly (verified numerically, zero feature difference), halving the number of circuits.

¹Because the noise model overestimates noise, the $Q1-Q0$ power estimates for Design B/Bc should be interpreted as upper bounds on detectability; on real hardware the advantage may be smaller or require fewer graphs.

Table 5: Projected QPU efficiency under packing assumptions. $\eta_Q = \Delta R^2 / T_{\text{QPU}}$. These are algebraic projections, not measured hardware speedups.

Architecture	n	bases	T_{QPU} (h)	η_Q	vs current
Phase 1–2 (current Design B)	250	2	4.40	0.0038	1.0×
Phase 4 (+ packing)	256	2	2.25	0.0074	2.0×
Phase 4 (Z-only, 1 basis)	256	1	1.13	0.0113	3.0×

Efficiency. We define QPU efficiency as $\eta_Q = \Delta R^2 / T_{\text{QPU}}$. The full QCSC stack *projects* $\eta_Q = 0.0074$, a 2.0× improvement over current Design B ($\eta_Q = 0.0038$), assuming independent block scheduling. A Z-only variant retains 76% of the signal ($\Delta R^2 = 0.0127$) and *projects* $\eta_Q = 0.0113$ (3.0×). These are algebraic projections, not measured hardware speedups (table 5; details in section D).

What the QCSC architecture does not change. The scientific question—does Q carry conditional predictive signal about Y given C ?—is unchanged. The architecture makes the quantum-edge test *cleaner*: the QPU only transforms a classically chosen subset. The question becomes: *given a classically generated graph, does the quantum dynamics produce a representation that matched classical transforms do not?*—a question that the strong learned baselines (section 4.3) answer in the negative for flexible classical models at this scale.

5 Discussion

What we claim. A fixed, physically motivated feature map, defined by a single-layer entangling circuit on a spectrally compressed graph, adds a small but robust amount of conditional predictive signal about solver performance to standard descriptors ($\Delta R^2 = 0.0208$). The effect exceeds random-projection and random-Fourier controls, depends on the entangling gates ($p < 10^{-7}$), is stable from 2,000 to 10,000 graphs, and is robust across solver budgets. Its dependence on entangling strength has a closed-form explanation, confirmed by a Z/X basis ablation showing that the Z-derived component reproduces the $e = 0.5$ peak. The feature map runs natively on trapped-ion hardware with high feature fidelity, and a simulation-trained readout applied to 250 hardware graphs gives a positive $\Delta R^2 = +0.016$ (calibrated).

What we do not claim. We do not claim a quantum computational advantage: by Proposition 1 the Z-basis features are classically computable at this scale, and the full feature set is computable from the exact statevector at $k = 8$. **We do not claim that the circuit outperforms flexible classical models:** Random Forest and XGBoost with access to the same (h, J) input dramatically exceed Q1 (section 4.3). The circuit features are not statistically separated from a random tanh reservoir ($p = 0.22$). The hardware $\Delta R^2 = +0.016$ at $n = 250$ is a transfer result (simulation-trained readout applied to hardware features), not a cross-validated hardware result, and the Q1–Q0 excess is driven by Q0 collapsing under noise, not by entanglement adding information. The quantum features do not improve practical solver selection (45.1% accuracy with or without Q).

Why the hardware result still matters. The hardware experiment establishes what is needed before any larger claim: the circuit runs without compilation overhead on a native gate set, device noise beyond shot noise is small for most features, and the simulation-trained readout transfers to hardware features with a positive $\Delta R^2 = +0.016$ at $n = 250$ (calibrated). These are prerequisites for moving to regimes where no closed form exists: more layers, more qubits, or multi-qubit observables.

Effect size. $\Delta R^2 \approx 0.021$ is modest in absolute terms and is dramatically exceeded by flexible classical models on the same input (Random Forest: $\Delta R^2 = 0.179$). Its value lies in being robust across controls, scale and solver budgets, and in being mechanistically understood. It does not translate into better solver *choices*: selection accuracy is 45.1% with or without quantum features, and regret is unchanged (table 4). The gain in explained variance is too small to shift argmax decisions.

6 Limitations

(i) The features are classically simulable at $k = 8$ with one layer. (ii) The optimum at $e = 0.5$ was not preregistered. (iii) The circuit is not statistically separated from a random reservoir, and the differences from the two scrambled ablations are not resolved (Q1–Q2: $+0.0008$, $p = 0.061$; Q1–Q3: $+0.0002$, $p = 0.30$). (iv) **Flexible classical models (Random Forest, XGBoost) with access to the same (h, J) input dramatically exceed Q1; the quantum feature map does not provide superior predictive signal.** (v) Transfer to most OOD graph families fails under the corrected protocol. (vi) The hardware sample is 250 graphs with 200 shots each, and the absolute ΔR^2 for Q1 is not significantly resolved (CI includes 0). (vii) A single hardware platform. (viii) The target is defined relative to one five-solver portfolio and budget. (ix) The size-shift OOD test produces extreme values due to distribution mismatch and requires a consistent cross-size target definition. (x) We did not test a graph-native learned baseline such as a GNN; this remains an important comparison for future work.

7 Outlook

Scale the hardware run guided by the noise-injected power analysis (S09) and the noise-augmented readout control (S12), using the QCSC architecture (section 4.11). Four designs are available: (i) **Design B** (train on simulation, test on hardware) detects a Q1–Q0 difference with 100% power at $n = 250$ graphs (4.4 QPU-h for both conditions), making it the fastest path to a hardware result. However, S12 shows that this Q1–Q0 excess is noise amplification of Q0, not information from entanglement. (ii) **Design Bc** (Design B with cross-fitted per-qubit linear calibration) confirms that the Q1–Q0 advantage survives calibration (power 1.00 at $n = 250$), but S12 shows the advantage persists because calibration does not fix the over-weighting of low-variance Q0 features; this is the primary outcome for the hardware run, analysed by script S11 with a preregistered interpretation guide. (iii) **Design Ba** (noise-augmented readout, script S12) trains the readout on noise-injected simulated features and is the correct interpretation for the hardware Q1–Q0: in simulation it removes $> 98\%$ of the excess. S11 lists Ba as a secondary outcome alongside B and Bc. (iv) **Design A** (cross-validation on noisy hardware features) resolves the absolute $\Delta R^2 > 0$ for Q1 at 80% power with $n = 2,000$ graphs (17.6 QPU-h) at measured noise, or $n = 2,000$ (17.6 QPU-h) under $1.5\text{--}2.0\times$ noise. All are feasible within a single QPU allocation. The QCSC architecture (Phase 4) reduces the QPU cost of Design B from 4.4 to 2.25 QPU-h via graph packing and coresets selection, with no loss in ΔR^2 . A Z-only feature variant (4 of 8 features, 1 basis) retains 76% of the signal and reduces cost to 1.13 QPU-h. Further steps are to repeat the entanglement sweep on hardware at $e \in \{0.25, 0.5, 0.75\}$, to apply readout-error mitigation and zero-noise extrapolation [12, 25], to move to circuits without a closed form (two or more layers, correlator observables $\langle Z_a Z_b \rangle$) with comparison against classical surrogates [20], and to compare trapped-ion and superconducting platforms.

Data and code availability

All code, data, analysis and figure-generation scripts, final figures, result files, manuscript source, and preregistration materials are archived in the frozen public reproducibility release at <https://doi.org/10.5281/zenodo.23010309>. The release includes the preregistration dated

2026-09-23, the hardware-analysis registrations, and the full deviation and correction changelog. The source repository and immutable GitHub release are available at <https://github.com/VesterlundCoder/qmt-entangling-predictive-signal/releases/tag/v2.0.0>. The archived release, rather than the mutable repository branch, is the version of record for reproducibility. Seeds are 42 for data generation and seed + i per graph.

Preregistration provenance

Analysis	Registered?	Status	Notes
H1: $\Delta R^2 > 0$	PREREGISTERED	Met	$\Delta R^2 = 0.0208$
H2: Q1 > classical	PREREGISTERED	Not resolved	$p = 0.22$
H3: Q1 > Q0	PREREGISTERED	Met	$p < 10^{-7}$
Q1 vs Q2/Q3	PREREGISTERED	Not resolved	$p > 0.06$
H4: Size shift	PREREGISTERED	Not meaningful	Distribution mismatch
H4: Family shift	PREREGISTERED	Not met	Mostly negative
H5: Geometry optimum	PREREGISTERED	Met	Q rank < C rank
Entanglement sweep	PREREGISTERED	Exploratory	Peak at $e = 0.5$
$e = 0.5$ optimum	POST-HOC	Exploratory	Z-only confirms
Corrected statistics	PROTOCOL CORRECTION	Applied	NB intervals
Strong learned controls	POST-HOC	Q1 exceeded	Outcome C
Z/X ablation	EXPLORATORY	Z-only peaks	Mechanism
Solver budget	EXPLORATORY	$\Delta R^2 > 0$ all budgets	Robustness
Hardware S11	PREREGISTERED	Complete	Q0 collapses
Hardware S12	PREREGISTERED	Complete	Removes excess
Hardware S09	PREREGISTERED	Complete	Power analysis

Statements and Declarations

Funding. This research received no external funding.

Competing interests. David Vesterlund is the owner of Vesterlund Ventures Holding AB and operates WestQuant Studio. The author declares no other financial or non-financial interests directly related to this work.

Author contributions. David Vesterlund: conceptualization, methodology, software, investigation, formal analysis, visualization, writing—original draft, and writing—review and editing.

Ethics approval. Not applicable. The study involved no human participants, human data, or animals.

Consent to participate. Not applicable.

Consent for publication. Not applicable.

Generative AI disclosure. GLM-5.2 within Devin was used to assist with drafting and revising manuscript text and tables, and with generating and revising analysis and figure-production code. All outputs were critically reviewed, edited, and, where applicable, tested by the author. The author takes full responsibility for the integrity, accuracy, interpretation, and final content of the work.

References

- [1] Joseph Bowles, Shah Nawaz Ahmed, and Maria Schuld. Better than classical? the subtle art of benchmarking quantum machine learning models. *arXiv preprint arXiv:2403.07059*, 2024.
- [2] Quentin Cappart, Didier Chetelat, Elias B Khalil, Andrea Lodi, Christopher Morris, and Petar Velickovic. Combinatorial optimization and reasoning with graph neural networks. *ACM Computing Surveys*, 55(6):1–36, 2023. doi: 10.1145/3579376.
- [3] M Cerezo, H Larocca, D Hush, et al. Does provable absence of barren plateaus imply classical simulability? *arXiv preprint arXiv:2312.09121*, 2023.
- [4] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*, 2014.
- [5] Keisuke Fujii and Kohei Nakajima. Harnessing disordered quantum dynamics for machine learning. *Physical Review Applied*, 8(2):024030, 2017. doi: 10.1103/PhysRevApplied.8.024030.
- [6] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145, 1995. doi: 10.1145/227683.227684.
- [7] Stuart Hadfield. Quantum alternating operator ansatz. *arXiv preprint arXiv:1709.03489*, 2019.
- [8] Vojtech Havlicek et al. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567:209–212, 2019.
- [9] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16:1050–1057, 2020. doi: 10.1038/s41567-020-0932-7.
- [10] 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. doi: 10.1038/s41467-021-22539-9.
- [11] Herbert Jaeger. The “echo state” approach to analysing and training recurrent neural networks. *GMD Technical Report 148*, 2001.
- [12] Abhinav Kandala et al. Error mitigation extends the reach of noisy quantum computers. *Nature*, 567:491–495, 2019.
- [13] Pascal Kerschke, Holger H Hoos, Frank Neumann, and Heike Trautmann. Automated algorithm selection: Survey and perspectives. *Evolutionary Computation*, 27(1):3–45, 2019. doi: 10.1162/evco_a_00242.
- [14] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, 2019.
- [15] Mantas Lukosevicius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.
- [16] Pere Mujal, Rodrigo Martínez-Peña, Johannes Nokkala, Jorge García-Bení, Gian Luca Giorgi, Miguel C Soriano, and Roberta Zambrini. Opportunities in quantum reservoir computing and extreme learning machines. *Advanced Quantum Technologies*, 4(8):2100027, 2021. doi: 10.1002/qute.202100027.

- [17] Claude Nadeau and Yoshua Bengio. Inference for the generalization error. *Machine Learning*, 52:239–281, 2003. doi: 10.1023/A:1024068626366.
- [18] Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, 2007.
- [19] John R Rice. The algorithm selection problem. *Advances in Computers*, 15:65–118, 1976. doi: 10.1016/S0065-2458(08)60520-3.
- [20] Franz J Schreiber, Jens Eisert, and Johannes Jakob Meyer. Classical surrogates for quantum learning models. *Physical Review Letters*, 131(10):100803, 2023. doi: 10.1103/PhysRevLett.131.100803.
- [21] Maria Schuld and Nathan Killoran. Quantum machine learning in feature Hilbert spaces. *Physical Review Letters*, 122(4):040504, 2019. doi: 10.1103/PhysRevLett.122.040504.
- [22] Kate Smith-Miles, Liana Bowden, and Davaatseren Baatar. Towards objective measures of algorithm performance across instance space. *Computers & Operations Research*, 45:12–24, 2014.
- [23] Kate A Smith-Miles. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys*, 41(1):1–25, 2009. doi: 10.1145/1459352.1459354.
- [24] Anders Sorensen and Klaus Molmer. Quantum computation with ions in thermal motion. *Physical Review Letters*, 82(9):1971, 1999. doi: 10.1103/PhysRevLett.82.1971.
- [25] Kristan Temme, Sergey Bravyi, and Jay M Gambetta. Error mitigation for short-depth quantum circuits. *Physical Review Letters*, 119(18):180509, 2017. doi: 10.1103/PhysRevLett.119.180509.
- [26] Supanut Thanasilp, Samson Wang, M Cerezo, and Zoë Holmes. Exponential concentration in quantum kernel methods. *Nature Communications*, 15:5200, 2024. doi: 10.1038/s41467-024-49287-w.
- [27] Zhihui Wang, Stuart Hadfield, Zhang Jiang, and Eleanor G Rieffel. Maxcut with QAOA: analytical results for the MaxCut problem on complete graphs. *Physical Review A*, 97(2):022304, 2018. doi: 10.1103/PhysRevA.97.022304.
- [28] Lin Xu, Frank Hutter, Holger H Hoos, and Kevin Leyton-Brown. SATzilla-2007: Portfolio-based algorithm selection for SAT. In *Journal on Satisfiability, Boolean Modeling and Computation*, 2008.

A Pilot and falsification runs (2,000 graphs)

Stage 0A (pilot, one seed). ΔR^2 (ridge) = 0.019 for $k = 8$ and 0.023 for $k = 12$; residual $R^2 = 0.040$. The gate $R^2_{\text{residual}} \geq 0.02$ was met and the study was promoted to Stage 0C.

Stage 0C (falsification, 5-fold $\times$ 10 seeds). Q1: $\Delta R^2 = 0.029$ (cross-validation interval [0.015, 0.046]; bootstrap 0.031 [0.022, 0.041]; permutation $p < 0.001$). The two means differ because they are different estimators: the cross-validation mean averages fold-level estimates, while the bootstrap resamples entire graphs with replacement. All five Stage 0C gates were met at this scale.

B Proof of Proposition 1

R_{XX} gates on different pairs commute, so the entangling layer is $U = \prod_{b < c} \exp(-i\varphi_{bc}X_bX_c/2)$. Gates not involving qubit a commute with Z_a and cancel in $U^\dagger Z_a U$. Since Z_a anticommutes with X_aX_b ,

$$e^{i\varphi X_a X_b/2} Z_a e^{-i\varphi X_a X_b/2} = e^{i\varphi X_a X_b} Z_a, \quad \text{hence} \quad U^\dagger Z_a U = \prod_{b \neq a} (\cos \varphi_{ab} + i \sin \varphi_{ab} X_a X_b) Z_a.$$

Expanding the product over subsets S of neighbours, a term with $|S|$ odd contains $X_a Z_a = -iY_a$, whose expectation in the product state $\bigotimes_b R_Y(\theta_b)|0\rangle$ vanishes because $\langle Y \rangle = 0$. A term with $|S|$ even contains $Z_a \prod_{b \in S} X_b$, with expectation $\cos \theta_a \prod_{b \in S} \sin \theta_b$. Hence

$$\langle Z_a \rangle = \cos \theta_a \sum_{|S| \text{ even}} \prod_{b \in S} (i \sin \varphi_{ab} \sin \theta_b) \prod_{b \notin S} \cos \varphi_{ab} = \cos \theta_a \cdot \text{Re} \prod_{b \neq a} (\cos \varphi_{ab} + i \sin \varphi_{ab} \sin \theta_b).$$

An R_Z layer applied after the entangling layer commutes with Z_a and leaves $\langle Z_a \rangle$ unchanged. $\square$

C Classical descriptors

1	$\log N$	7	degree kurtosis
2	edge count	8	average clustering coefficient
3	density	9	spectral radius
4	mean degree	10	algebraic connectivity
5	degree standard deviation	11	diameter estimate
6	degree skewness	12	degree assortativity

D QCSC architecture details

Graph packing. Placing G_A on qubits 0–7 and G_B on qubits 8–15 with no cross-block R_{XX} gates gives $U(G_A \sqcup G_B) = U(G_A) \otimes U(G_B)$ exactly, because the R_Y layer is already a tensor product and the R_{XX} layer acts only within blocks. We verified this numerically on 100 random pairs: the maximum feature difference is 0. On PiastQ (20 qubits) this gives $\lfloor 20/8 \rfloor = 2$ graphs per circuit at $k = 8$.

Measurement basis analysis. Dropping the X -basis features reduces ΔR^2 from 0.0208 to 0.0127 (76% retained). Classical-shadow methods [9] would further reduce cost for large Pauli observable sets.

Coreset selection and selection bias. We tested five selection strategies. Random selection gives $\Delta R^2 = +0.018 \pm 0.011$, consistent with the full-bank estimate. K-center on quantum features gives $\Delta R^2 = +0.120$ —a $6\times$ inflation—because it selects graphs where quantum features are extreme. *Intelligent selection corrupts the ΔR^2 estimate.*

Hardware residual learning. Learning the hardware residual $\delta(G) = q_{\text{hw}}(G) - q_{\text{sim}}(G)$ from 100 calibration graphs gives weak predictability (mean LOO $r = 0.17$); this approach would need 500–1,000 graphs to become useful.

E Supplementary figure

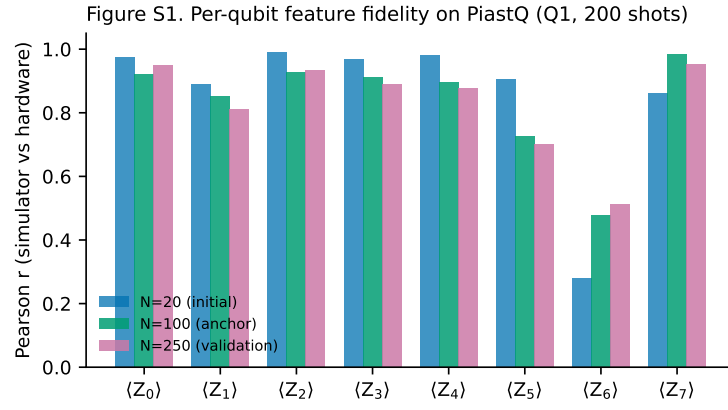


Figure 10: Per-qubit correlation between simulated and hardware features for Q1 in the initial 100-graph validation and the 250-graph extended hardware validation.