AUDITING CLAIMED SPEEDUPS OF LEARNED SURROGATES FOR VARIATIONAL QUANTUM OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

A growing line of work accelerates variational quantum algorithms by learning a classical surrogate of the optimization trajectory and rolling it forward, so that quantum measurements are replaced by classical inference. We observe that the two headline results in this line—a $30\times$ physics-informed speedup and a $>200\times$ Koopman speedup in an over-parameterised regime—are both measured against small-step gradient-descent reference trajectories, at $\eta = 10^{-2}$ and $\eta = 5 \times 10^{-6}$. We show this makes the reported quantity uninformative: for quadratics the reference needs $\Theta(1/\eta)$ iterations in the worst case, so a ratio against a fixed step diverges as $\eta \to 0$. We also show that a rule predicting the next iterate from the step index and the parameter vector alone cannot represent the transition law of an optimizer with internal state, which covers one of the two families we audit. Running both released implementations unmodified through one counter that charges energy and gradient evaluations at their true circuit cost, Adam is more query-efficient in both families; for the physics-informed surrogate it also reaches accuracies the surrogate never attains, at 8, 20 and 40 qubits, on two Hamiltonians, and under finite sampling. On `ibm_kawasaki` we find the sampling model holds to within 7% at the one circuit and parameter point we could afford to repeat and find a device error that, at the configurations tested, is two orders of magnitude larger than the accuracy at which those ratios are defined. Asked of ourselves, the same sensitivity analysis shows our ordering holds over a $5\times$ range of step size—and reverses below it, at exactly the reference value the audited work uses.

1 INTRODUCTION

Variational quantum algorithms (Peruzzo et al., 2014; Kandala et al., 2017) optimize a parameterized circuit $|\psi(\theta)\rangle$ to minimize $E(\theta) = \langle\psi(\theta)| H |\psi(\theta)\rangle$ (Cerezo et al., 2021a). Because the no-cloning theorem precludes backpropagation through the quantum state (Abbas et al., 2023), each gradient entry must be estimated by repeated measurement via the parameter-shift rule (Mitarai et al., 2018; Schuld et al., 2019), and the number of quantum queries is the dominant cost of VQA training. Reducing that count is therefore a central goal.

One increasingly popular strategy is to *predict* the optimization trajectory classically. A model is trained on a short prefix $\{\theta^{(t)}\}_{t=1}^{\tau}$ collected from the device and then rolled forward to produce $\theta^{(\tau+k)}$ without further measurement. Instances of this idea include physics-informed models (Raissi et al., 2019) that treat the training dynamics as a partial differential equation (Huang et al., 2025), and Koopman-operator surrogates that embed the dynamics in a learned linear representation (Luo et al., 2023). Reported headline gains range from $30\times$ to over $200\times$.

This paper asks a narrow question: *relative to what?*

Our starting observation concerns the configurations in which the headline numbers are measured. The physics-informed model is derived by taking the gradient-descent update $\Delta\theta = -\eta\,\partial E/\partial\theta$ and letting $\eta \to 0$, giving the gradient-flow ODE it then models; its reported speedup is measured against that flow at $\eta = 10^{-2}$. The Koopman surrogate is evaluated on a range of optimizers elsewhere in its paper, but its headline $>200\times$ result is obtained in an over-parameterised configuration whose base optimizer is gradient descent at $\eta = 5 \times 10^{-6}$, chosen so that the dynamics in θ-space are

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

approximately linear. In both cases the number that travels is normalised by a reference trajectory whose step size is never varied.

This is not a matter of degree. We show in Section 4 that for quadratics the reference optimizer needs $\Theta(1/\eta)$ iterations to fixed accuracy, so a speedup ratio measured against a fixed step η can be made as large as one likes by shrinking η; the two audited headline configurations use $\eta = 10^{-2}$ and 5×10^{-6}. A second observation limits what one such model can fit: a rule predicting the next iterate from the step index and the current parameter vector alone cannot express the transition law of an optimizer with internal state, so momentum, Adam and quasi-Newton methods lie outside its class. This applies to the physics-informed family and not to the Koopman one, which carries a window of history, so we offer it as a partial rather than a general explanation.

Contributions.

- **Two reasons the comparison cannot settle the question.** Proposition 1: the reference optimizer needs $\Theta(1/(\eta\mu))$ iterations, so a speedup ratio against a fixed step diverges as $\eta \to 0$. Proposition 2: a rule $g(t, \theta)$ cannot express the transition law of a stateful optimizer, placing momentum, Adam and quasi-Newton methods outside what a model of that form can fit. The Koopman family carries history and is outside its scope.
- **An audit on one resource axis.** Both released implementations, unmodified at their defaults, with energy and gradient evaluations charged at their true circuit cost. Adam and L-BFGS reach an accuracy the physics-informed surrogate never attains while spending fewer evaluations, at 8, 20 and 40 qubits on two Hamiltonians; in the over-parameterised regime the Koopman surrogate claims, Adam succeeds in five seeds of five at a third of its cost, the surrogate in three.
- **The boundary of the claim, measured.** Under finite sampling the surrogate does not become competitive, but L-BFGS stops working. On `ibm_kawasaki` the sampling model is accurate to 7%, while device error over eighteen configurations has spread 0.207 and reaches 0.60—far above the 10^{-3} at which speedup ratios are defined. Applied to ourselves, the ordering holds over a $5\times$ band of step size and reverses below it.
- **An instrumented harness** making the comparison a one-line change, with the pitfalls we hit: a two-sided accuracy window that silently records fast optimizers as failures, and a released implementation whose fixed trajectory filename makes concurrent seeds identical.

We are auditing a specific and well-defined family—surrogates that learn the optimizer's trajectory and roll it forward—not the broader question of what optimizer a VQA should use. The VQA optimization literature contains many methods we do not evaluate here, among them SPSA (Spall, 1992), quantum natural gradient (Stokes et al., 2020; Yamamoto, 2019), coordinate schemes (Nakanishi et al., 2020), and shot-adaptive variants (Arrasmith et al., 2020; Moussa et al., 2023). Our claim is correspondingly narrow: it is about the reference point against which this family reports its gains.

The empirical half checks what this costs. We route every energy and gradient evaluation of every method—including the authors' released implementations, run unmodified at their own defaults—through a single counter charging each at its true circuit cost, add the omitted baselines (tuned-step gradient descent, momentum, Adam (Kingma & Ba, 2015), L-BFGS (Liu & Nocedal, 1989)), and compare at matched budgets across 8, 20 and 40 qubits, two Hamiltonians, two surrogate families, finite sampling and hardware. Each surrogate does accelerate the optimizer it is derived from, often by two or three orders of magnitude; Adam and L-BFGS accelerate it more, at a fraction of the circuit evaluations, using no model and no trajectory data. Nothing here says the surrogates are wrong—it says the quantity measuring them is a property of the baseline. We release the harness so the alternative measurement is a one-line change.

2 Background

Variational quantum algorithms. A VQA prepares $|\psi(\theta)\rangle = U(\theta)\,|0\rangle^{\otimes n}$ with a parameterized circuit and minimizes $E(\theta) = \langle\psi(\theta)|\,H\,|\psi(\theta)\rangle$ for a Hamiltonian $H = \sum_j c_j P_j$ written in Pauli terms. The circuit we use throughout is the hardware-efficient ansatz (Kandala et al., 2017) of the implementations we audit: per layer, single-qubit R_Y, R_Z rotations on every wire followed by a

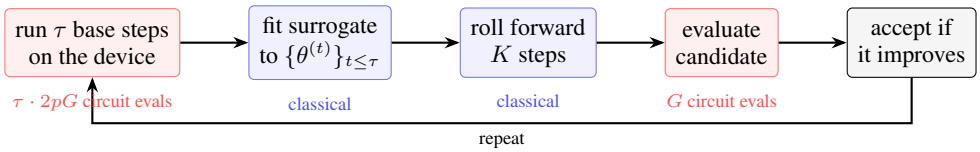


Figure 1: The surrogate protocol and its cost. Red boxes touch the device; blue boxes are classical. The saving comes from replacing base steps, each costing a full gradient, with a classical rollout checked by one energy evaluation. Because a gradient costs $2pG$ circuit evaluations and an energy costs G, the two device operations differ in price by two orders of magnitude at the sizes considered here, which is why we charge them separately.

chain of CZ gates, repeated L times, giving $p = 2nL$ parameters ($p = 120$ at $n = 20$, $L = 3$); Appendix F draws one layer.

Why queries, not iterations, are the resource. The no-cloning theorem rules out backpropagation through the state (Abbas et al., 2023), so each partial derivative is obtained by the parameter-shift rule (Mitarai et al., 2018; Schuld et al., 2019),

$$\frac{\partial E}{\partial \theta_i} = \tfrac{1}{2}\Big[E\big(\theta + \tfrac{\pi}{2}e_i\big) - E\big(\theta - \tfrac{\pi}{2}e_i\big)\Big],$$

which costs two circuit evaluations per parameter, for each of the G commuting groups of H. A full gradient therefore costs $2pG$ circuit evaluations while a single energy costs G. The asymmetry is large—at $n = 20$ a gradient is $240\times$ an energy—and it is what makes the accounting choice in Section 5 matter.

Trajectory surrogates. The methods we audit exploit the fact that the sequence $\{\theta^{(t)}\}$ produced by the optimizer is itself structured. A short prefix is measured on the device, a model is fit to it, and the model is rolled forward to propose a distant iterate, which is then checked with a single energy evaluation and accepted if it improves. Figure 1 shows where the queries go.

3 Related Work

Surrogates for VQA training. Predicting the optimization trajectory classically to save quantum queries has been approached with Koopman operator learning (Luo et al., 2023), building on dynamic mode decomposition (Schmid, 2010), and with physics-informed models of the gradient-flow PDE (Huang et al., 2025; Raissi et al., 2019). Both belong to the wider learning-to-optimize programme (Chen et al., 2022). A related but distinct idea builds a surrogate of the *cost landscape* rather than of the optimizer's trajectory (Gustafson et al., 2025); our propositions concern the trajectory case, where the object being modelled is the optimizer's own dynamics. Both report large speedups over gradient descent with a small fixed step. Our audit takes these methods at their word, runs their released code, and asks what the same query budget buys with a different base optimizer.

Benchmarking practice in quantum machine learning. Bowles et al. (2024) argue that benchmarking quantum models is unusually error-prone and that the field's optimistic picture partly reflects methodological choices rather than quantum effects. Our finding is an instance of that thesis in a specific and checkable form: the choice is the reference optimizer, and the fix costs one line. Abbas et al. (2023) quantify why gradient estimation dominates VQA cost, which is what makes the query axis the right one.

Optimizers for variational circuits. That quasi-Newton and adaptive methods outperform fixed-step gradient descent is not new, and has been established across many optimizers (Bonet-Monroig et al., 2023; Jones et al., 2025); specialised schemes exploit structure generic ones do not (Nakanishi et al., 2020; Stokes et al., 2020; Arrasmith et al., 2020). We add no entry to that comparison. Our point is that this ordinary fact accounts for speedups attributed to learned surrogates, and that the two literatures have not been measured on one axis.

4 TWO OBSERVATIONS ABOUT SPEEDUP RATIOS AND SURROGATE CLASSES

The empirical picture has a simple explanation, which we state in two parts. Both are elementary; we make them explicit because together they predict exactly what we measure.

Proposition 1 (A speedup ratio is a statement about the baseline)**.** *Let $E(\theta) = \frac{1}{2}\theta^\top A\theta$ with $\mu I \preceq A \preceq LI$ and $\mu > 0$, and let gradient descent use a fixed step $\eta \in (0, 1/L]$. Then the* worst-case *number of iterations to reach relative accuracy ε—attained whenever $\theta^{(0)}$ has a non-zero component in the μ-eigenspace—is $N(\eta) = \Theta\big(\log(1/\varepsilon)/(\eta\mu)\big)$, so its circuit-evaluation cost is*

$$C_{\mathrm{GD}}(\eta) = 2pG\,N(\eta) = \Theta\big(2pG\log(1/\varepsilon)/(\eta\mu)\big).$$

Consequently, for any method M whose circuit-evaluation cost C_M does not depend on η, the speedup ratio $\alpha(\eta) = C_{\mathrm{GD}}(\eta)/C_M$ diverges as $\eta \to 0$.

Proofs are in Appendix E. The proviso on C_M is worth stating plainly: a surrogate retrained for each η could in principle have an η-dependent cost, so the proposition should be read as saying that a baseline-normalised speedup is intrinsically a function of the baseline's hyperparameter, not as a proof that any particular surrogate's reported number can be inflated without limit. That the two we audit *are* inflated by this mechanism is an empirical finding, reported below. An unbounded quantity is not evidence about a method. Numerically, on a 20-dimensional quadratic with $\mu = 0.05$, $L = 1$, the product $N(\eta) \cdot \eta$ stays within 8.4–8.5 as η ranges over 0.4 down to 0.005, i.e. $N = \Theta(1/\eta)$ to two digits. The two audited headline configurations use $\eta = 10^{-2}$ and, for the Koopman surrogate, $\eta = 5 \times 10^{-6}$.

Proposition 2 (A state-free map cannot represent heavy-ball)**.** *Write heavy-ball in state form, $v^{(t+1)} = \beta v^{(t)} - \eta\nabla E(\theta^{(t)})$ and $\theta^{(t+1)} = \theta^{(t)} + v^{(t+1)}$ with $\beta > 0$. Call a rule* state-free *if its increment satisfies $\Delta^{(t)} = \theta^{(t+1)} - \theta^{(t)} = g\big(t, \theta^{(t)}\big)$, i.e. it is a function of the iteration index and the current parameter vector only. Gradient descent is state-free, with $g = -\eta\nabla E$. Heavy-ball is not: fix any t and θ and take $v_1 \neq v_2$; the states (θ, v_1) and (θ, v_2) agree in both arguments of g yet give increments differing by $\beta(v_1 - v_2) \neq 0$, so no such g reproduces the transition law.*

The same holds for Adam and quasi-Newton methods, whose increments depend on accumulated moment or curvature estimates. We emphasise what this does *not* say. It is a statement about which dynamics a state-free map can express, proved across states; it is not a claim that two increments must stay far apart along one convergent trajectory. On a convergent run both $\theta^{(t)}$ and $\Delta^{(t)}$ tend to zero, so increments at nearby iterates necessarily become close, and no separation bounded below by a constant is available there.

Two qualifications matter, and we would rather state them than have them found. First, this is a statement about representing an optimizer's *transition law* over its state space. It does not preclude fitting a particular realised trajectory, along which the iteration index can disambiguate states that share a parameter vector; what it rules out is a rule of this form reproducing the optimizer in general. Second, the Koopman surrogate carries a sliding window of past iterates and lies outside the scope of this proposition altogether. Proposition 2 is therefore not a general theorem about why this literature omits accelerated optimizers; it is the narrower statement that a state-free autoregressive rollout cannot express stateful optimizer dynamics, which covers one of the two families we audit and not the other. That the comparison sets consist of surrogates over plain gradient descent is consistent with it, but we do not claim it is explained by it.

5 SETUP

Resource accounting. We report one quantity throughout: *circuit evaluations* on the device. Every method is wrapped so that all of its evaluations are counted, including those made inside acceptance tests and inside any inner optimization loop, and no other resource axis is used for empirical method comparisons. Where a method performs exactly one gradient per iteration, its iteration count and its circuit-evaluation count differ by the constant factor below, so ratios are the same in either; we nonetheless quote circuit evaluations everywhere so that methods mixing the two channels are on the same footing. We use "query" as a synonym for one circuit evaluation and never for an iteration.

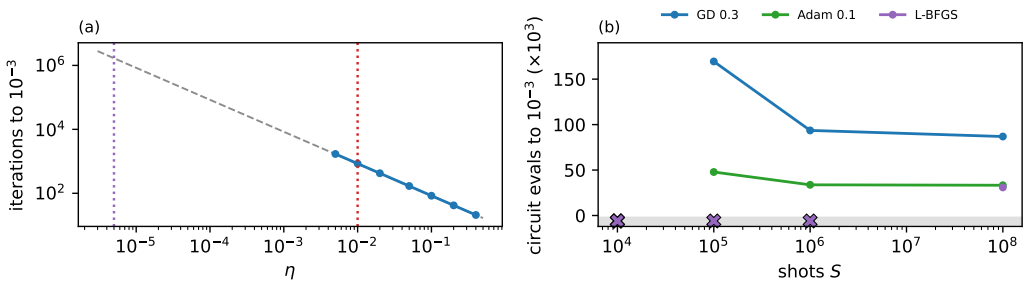


Figure 2: *(a)* Proposition 1 on a 20-dimensional quadratic; grey dashed line is $\Theta(1/\eta)$ extrapolated, dotted verticals are the audited reference steps 10^{-2} (red) and 5×10^{-6} (purple), where the baseline needs $\sim 10^6$ iterations. *(b)* Finite sampling, $n = 20$ TFIM; crosses on the grey band mark configurations never reaching the target.

Two-channel accounting. Charging an energy and a gradient alike, as a single "call", would overcharge methods that lean on energy-only evaluations. One energy costs G circuit evaluations, G being the number of commuting groups of H ($G = 2$ for TFIM). Each parameter-shift partial derivative needs two energies, so a full gradient over p parameters costs $2pG$. We therefore count the two channels separately and report circuit evaluations,

$$\text{units} = G\, n_{\text{energy}} + 2pG\, n_{\text{grad}}.$$

Under finite sampling each circuit evaluation uses S shots. We match S across methods within every comparison, so the total shot count is S times the reported circuit-evaluation count and all within-setting ratios are unchanged; we report the coarser unit because it is the one the audited papers use. This matters for the Koopman surrogate, which scans 1000 predicted points with energy-only evaluations; it does not change the physics-informed comparison, whose cost is dominated by the gradient calls of its inner loop (18 energy versus 84–192 gradient calls per run across every configuration we measured). We report it because it is the accounting choice most likely to have favoured our conclusion, and it does not.

Metric. We fix a target energy $E^\star$—the value the reference optimizer of the audited method converges to—and count queries to reach $E \le E^\star + 10^{-3}$, the accuracy used in that literature. The criterion is one-sided by design; Appendix F explains why, and states how the baseline step sizes were selected and frozen.

Systems. Transverse-field Ising (TFIM) and Heisenberg Hamiltonians with the hardware-efficient ansatz (Kandala et al., 2017) of the released implementation at depth 3, at $n = 8, 20, 40$ qubits ($p = 48, 120, 240$ parameters), contracted exactly as tensor networks (Gray, 2018). Initial parameters are drawn uniformly from $[0, 1]$, as in both codebases we audit. Spreads are over five seeds unless stated; the Koopman experiment at its reduced size uses seven.

6 Results

6.1 The headline speedup is available from the step size alone

Table 1 reports the circuit evaluations needed to reach $E^\star$, expressed as a speedup over the reference optimizer used in this literature (gradient descent, $\eta = 0.01$). Raising η to 0.3—changing nothing else—already matches the $30\times$ figure reported for learned surrogates. Adam and L-BFGS go substantially further. Crucially, the faster optimizers do not trade accuracy for speed: their final energies are *lower* than $E^\star$ in every configuration we ran. The rows for $n = 40$ and for the Heisenberg Hamiltonian are discussed in Appendix B, where we also address the regime in which the ordering could most plausibly reverse.

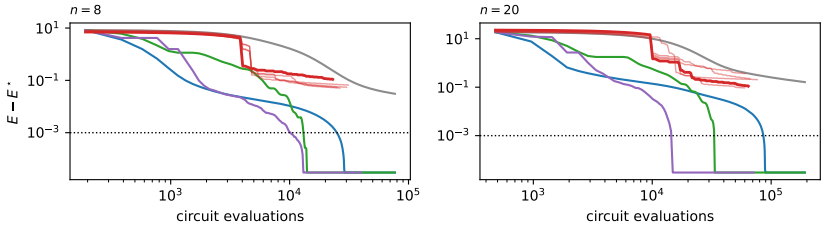


Figure 3: Accuracy against circuit evaluations, all seeds, both axes logarithmic. The surrogate (red) outruns the optimizer it is derived from (grey, $\eta = 10^{-2}$) early, then plateaus and is overtaken; its curve ends because the released protocol runs a fixed number of outer iterations. Values are floored at 3×10^{-5}, so reaching the floor means passing the target; the dotted line is the 10^{-3} accuracy defining the speedup ratio.

Table 1: Circuit evaluations to reach $E^\star$, as a speedup over gradient descent with $\eta = 0.01$ (the reference optimizer of the surrogate literature). Per-seed speedups are averaged; entries are mean $\pm$ s.d. over five seeds. Column "base" is the absolute circuit-evaluation count of that reference optimizer, in units of 10^3.

System	base	GD η=0.3	momentum	Adam	L-BFGS
TFIM, n=20	2756	35.3±4.8	49.4±15.7	74.1±14.6	178.2±25.2
TFIM, n=40	5646	31.0±3.6	42.3±3.7	77.4±13.2	168.5±9.9
Heisenberg, n=20	2846	32.1±4.4	26.6±8.2	61.8±29.4	224.9±49.7

6.2 Head-to-head at a matched query budget

Figure 3 and Table 2 run the released surrogate implementation unmodified, at its own default configuration, through the same counter. The surrogate is substantially faster than the reference gradient descent it is built on, over its initial budget.

The sharpest comparison is at a matched budget. We take the number of circuit evaluations the surrogate itself consumed and ask where each optimizer stands after the same number. Adam and L-BFGS are already *past* the target ($E - E^\star < 0$) at that point, while the surrogate is still 0.07, 0.20 and 1.28 above it at $n = 8, 20, 40$; even tuned-step gradient descent is one to nearly two orders of magnitude closer. Counting instead the evaluations needed to reach the target, Adam uses under four fifths of the surrogate's budget at every size and L-BFGS between a quarter and two fifths, and both reach an accuracy the surrogate does not attain in any run.

Tuned-step gradient descent is the exception and we state it plainly: under this accounting it needs 1.43 to 1.70 times what the surrogate spends, at every size. It does converge to the target, which the surrogate does not, but the claim that standard optimizers are cheaper rests on Adam and L-BFGS, not on a tuned step alone. This is also a place where the two-channel accounting works against us: charging the surrogate's acceptance tests as energies rather than gradients makes it cheaper than a single-rate count would, and we adopted that convention because it is the correct one, not because it helps.

Separately, the released implementation places on its iteration axis only about a fifth of the device evaluations it actually makes (5.5 ± 1.6 across 15 runs). We report this as a reproducible property of the released code, not as a statement about the numbers in the original paper.

6.3 The same pattern in a second surrogate family

The audit is not about one model. We repeat it on the Koopman/dynamic-mode-decomposition surrogate (Luo et al., 2023), importing the authors' own released DMD routine and keeping their protocol constants (four base steps per piece, 1000 predicted points scanned by energy evaluation,

Table 2: Circuit evaluations to reach $E^\star$ on TFIM, in units of 10^3. For the surrogate we report what it consumed and the accuracy gap it ended at, because it does not reach $E^\star + 10^{-3}$ in any run. "Plotted ratio" is the surrogate's actual device evaluations divided by the number of points its released code places on the iteration axis. Mean ± s.d. over five seeds.

	circuit evaluations to reach $E^\star$ ($\times 10^3$)					gap at the surrogate's own budget				
n	GD η=0.01	GD η=0.3	Adam	L-BFGS	surrogate	GD η=0.3	Adam	L-BFGS	surrogate	ratio
8	967±39	34±9	15±2	8±2	20±3	0.007±0.003	-0.002±0.001	-0.005±0.002	0.071±0.023	5.0±0.9
20	2757±52	79±9	38±7	16±2	48±7	0.020±0.008	-0.010±0.007	-0.027±0.010	0.200±0.107	4.8±0.5
40	5647±34	184±20	75±12	34±2	129±35	0.024±0.016	-0.046±0.020	-0.067±0.018	1.277±1.247	6.6±2.4

Table 3: Koopman surrogate in its own over-parameterised regime, in thousands of circuit evaluations; five seeds at the published configuration and seven at the reduced one. "Succeeded" counts seeds reaching $E^\star + 10^{-3}$; the speedup column averages over successful seeds only, which flatters the surrogate.

method	to reach $E^\star$	spent	speedup	succeeded
published config (5q, depth 250, p=2500)				
GD $\eta=5\times10^{-6}$ (their reference)	250000	250000	1×	5/5
QuACK	883±316 (3/5)	1310	283×	3/5
GD $\eta=10^{-3}$	1234±21	—	203×	5/5
Adam $\eta=10^{-2}$	414±56	—	604×	5/5
reduced config (3q, depth 30, p=180)				
GD $\eta=5\times10^{-6}$ (their reference)	18000	18000	1×	7/7
QuACK	29±20 (3/7)	83	625×	3/7
GD $\eta=10^{-3}$	91±0	—	198×	7/7
Adam $\eta=10^{-2}$	7±3	—	2734×	7/7

$\arg\min$ taken as the next iterate, $5n$ pieces) and their reference step $\eta = 5 \times 10^{-6}$. We run this at their published configuration, 5 qubits at depth 250 ($p = 2500$), whose compilation alone takes over three hours, and additionally at a smaller size in the same over-parameterised regime (Larocca et al., 2023) ($p \gg 2 \cdot 4^n$), 3 qubits at depth 30, where more seeds are affordable.

Table 3 shows the same structure as before, at both sizes. At the published configuration the surrogate reaches the target in three seeds of five and accelerates its own reference by $283\times$ when it does; Adam reaches it in five of five, at $604\times$, using about a third of the queries the surrogate spends. At the smaller size the pattern repeats with more seeds: three successes in seven against seven in seven, and an order of magnitude in cost. The surrogate's difficulty here is not its speedup when it works, which is real and large, but that whether it arrives at all is unreliable—and that a model-free optimizer arrives every time, sooner and at a fraction of the cost.

The scope of this result needs stating precisely, because the Koopman paper does not rest only on the configuration we audit. It also reports the surrogate accelerating Adam—$4.63\times$ on LiH and $3.24\times$ on Ising—so the surrogate is not presented as tied to small-step gradient descent. What our measurement bears on is the headline $> 200\times$ configuration, whose base optimizer is gradient descent at $\eta = 5 \times 10^{-6}$: there a tuned model-free optimizer is both cheaper and more reliable than the surrogate. Whether the surrogate adds anything on top of a *tuned* optimizer is a different question, and answering it requires comparing tuned Adam against the surrogate driving that same tuned Adam. We did not run that comparison, and we therefore make no claim about it.

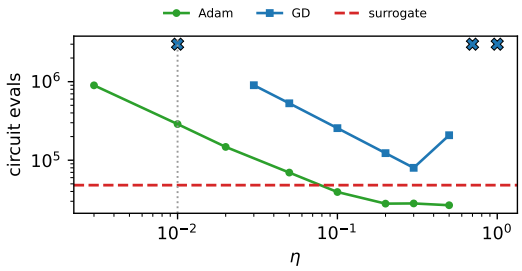
We stress that Proposition 2 does *not* apply here: the sliding-window construction carries trajectory history and is not state-free. Proposition 1 does apply, and applies more sharply, since $\eta = 5 \times 10^{-6}$ is three orders of magnitude below the step used in the physics-informed model.

6.4 Under finite sampling

Everything above is exact simulation. Since the surrogate fits a smooth model of the trajectory, it might be expected to average out per-query noise and come into its own once sampling is finite—the

Table 4: Accuracy gap at the surrogate's own budget under finite sampling; lower is better, negative means the target was passed. Five seeds per row. σ_E is the noise actually injected; S is the shot count it corresponds to for the Hamiltonian of Appendix F. The expected reversal does not occur: at $n = 20$, $\sigma_E = 0.18$ the surrogate is more than twice as far from the target as Adam.

n	shots S	σ_E	GD η=0.3	Adam	surrogate
8	7.5×10^3	0.035	+0.009±0.004	+0.002±0.008	+0.079±0.024
8	7.5×10^2	0.110	+0.020±0.013	+0.020±0.009	+0.079±0.026
20	7.4×10^3	0.057	+0.027±0.014	+0.011±0.014	+0.198±0.092
20	7.4×10^2	0.180	+0.124±0.017	+0.099±0.019	+0.222±0.109



optimizer	converges for $\eta \in$	beats the surrogate for $\eta \in$	width
GD	[0.03, 0.5]	—	—
Adam	[0.003, 0.5]	[0.1, 0.5]	5×

Figure 4: Our conclusion across the optimizer's own hyperparameter, $n = 20$, three seeds. Crosses at the top mark step sizes at which the optimizer never converges; the dotted vertical is the audited method's reference step $\eta = 10^{-2}$; the dashed horizontal is what the surrogate spends, 48k, as in Table 2. Adam beats it over a 5× range of η but *not* at $\eta = 10^{-2}$ or below, where it needs 288k. Tuned gradient descent never gets below the surrogate's budget.

regime a VQA actually runs in. We repeat the head-to-head with a sampled oracle: each expectation value is estimated from S shots, $\hat{E} = E + \mathcal{N}(0, \|c\|_2^2/S)$, gradients being parameter-shift differences of two such estimates.

Table 4 does not show that reversal. Every method degrades as S falls, but the ordering is unchanged and the gap does not close: at $n = 20$ with $\sigma_E = 0.18$ Adam sits 0.099 ± 0.019 above the target and the surrogate 0.222 ± 0.109. The one casualty is L-BFGS, our strongest result in exact simulation, which needs accurate curvature and fails to reach the target below $S = 10^8$ (Figure 2b); we do not carry it forward as a practical recommendation. SPSA (Spall, 1992), whose two evaluations per step are independent of p and which is the natural choice here, did not converge to the 10^{-3} target at any of $n = 8, 20, 40$—which we read as that target lying below what a stochastic-approximation method is built to deliver rather than as a verdict on it.

6.5 Is our own conclusion sensitive to the optimizer we chose?

The paper's argument is that a reported speedup can depend on a hyperparameter of the baseline. It would be incoherent not to ask the same of ourselves. Figure 4 sweeps the step size of both added optimizers over two decades, far wider than the three-point grid used to fix the reported settings.

Two things follow. First, the ordering we report is not an artifact of one lucky step size, but the band is narrower than a single comparison suggests: Adam is cheaper than the surrogate for every $\eta \in [0.1, 0.5]$, a fivefold range, and plain gradient descent for no η at all. The second half of that is the same conclusion Table 2 reaches at every size, and it is why our claim rests on Adam and L-BFGS rather than on a tuned step. Second, the ordering does reverse at $\eta \leq 10^{-2}$—exactly the reference value used by the audited method—and it reverses by a factor of six, not a hair. That is Proposition 1 seen from the other side: at a small enough step the baseline is slow enough that a

shots S	model σ_E	observed s.d.	ratio
1024	0.2253	0.2311	1.03
4096	0.1127	0.1208	1.07
16384	0.0563	0.0555	0.99

ansatz depth	transpiled depth	mean error	s.d. over points	range
1	7	$+0.022$	0.188	$[-0.29, +0.22]$
2	13	-0.044	0.081	$[-0.13, +0.09]$
4	25	-0.150	0.292	$[-0.60, +0.25]$

Table 5: **Left:** the Gaussian sampling model against `ibm_kawasaki`; one parameter point, ten repetitions per shot count, counts interleaved so a calibration change cannot masquerade as a dependence on S. **Right:** device error against exact simulation, six random parameter points at each of three depths, 8192 shots (sampling s.d. 0.080).

Table 6: Optimizers run on `ibm_kawasaki`, 4 qubits, transpiled depth 15, 4096 shots, 30 iterations, matched device budget, four seeds. Optimisation uses the device; the resulting parameters are then scored exactly.

optimizer	device circuit evaluations	energy of the parameters found	seeds
Adam η=0.1	1980	-12.21 ± 0.05	4
GD η=0.01 (their reference step)	1980	-11.86 ± 0.06	4

surrogate over it looks good, and our own baseline would be no different if we chose its step as carelessly.

6.6 Does the sampling model hold on hardware?

Section 6.4 assumes $\hat{E} = E + \mathcal{N}(0, \|c\|_2^2/S)$. We tested that on a device: one energy of a 4-qubit, depth-2 circuit, estimated repeatedly on `ibm_kawasaki` at three interleaved shot counts.

The variance model survives (Table 5): observed spread matches prediction to within 7% over a $16\times$ range of S. The device error it does not model is larger and is no fixed offset—over eighteen configurations its mean is -0.057 with spread 0.207 across parameter points, two and a half times the sampling noise at the same shot count, growing with depth from 0.081 to 0.292 and reaching -0.60 at individual points. (An earlier version of this section reported $+0.254$ from a single parameter point and called it a systematic offset; the sweep shows that was one draw from a distribution centred near zero.) That is two orders of magnitude above the 10^{-3} at which speedup ratios are defined, so nothing can be certified to have reached that target here. We varied the parameter point and the depth, not the device, the day, the Hamiltonian, the transpilation or the mitigation; drift we cannot address at all, having recorded submission rather than execution times ($t = -1.8$, not significant).

Running the optimizers on the device tells a simpler story. Each optimises *using* `ibm_kawasaki` at 4096 shots, and its final parameters are then scored in exact simulation—separating search quality from device error and sidestepping a target that error makes unreachable. At a matched 1980 device circuit evaluations Adam finds parameters worth -12.21 ± 0.05 against -11.86 ± 0.06 for the reference step over four seeds (Table 6): a small gap, but consistent across seeds. At 4 qubits and depth 2 the instance is far too easy for Proposition 1 to bite, so this is not hardware confirmation of the simulation result, only evidence that the optimizers behave on the device as they do in simulation, at roughly two hours of device time per arm.

7 Conclusion

Learned trajectory surrogates do accelerate their reference dynamics. In the two headline configurations we audit, however, the reported gains are normalised against gradient descent with a small fixed step, and that is a poor reference point for statements about the quantum measurement cost of VQAs: changing the step size alone recovers the reported speedup, and standard optimizers exceed it on the same device-cost axis. Echoing broader concerns about benchmarking practice in quantum machine learning (Bowles et al., 2024), we suggest that claims of quantum resource reduction be reported against a tuned classical optimizer on an explicit query-count axis, and we release the instrumented harness to make that comparison routine.

REPRODUCIBILITY STATEMENT

Every number in this paper is generated from raw result files by the scripts released with it rather than transcribed by hand, and a checker re-reads each figure quoted in the text against the table it comes from. Section 5 states the metric, the resource unit and how the baseline step sizes were selected and then frozen; Appendix F gives the Hamiltonians, ansatz, parameter counts, seeds, surrogate configurations, sampling model and software versions, together with the measurement pitfalls we hit and the results we discarded. Both audited implementations are run unmodified at their released defaults; our harness, the raw outputs and the hardware job records are included with the submission.

REFERENCES

Amira Abbas, Robbie King, Hsin-Yuan Huang, William J Huggins, Ramis Movassagh, Dar Gilboa, and Jarrod McClean. On quantum backpropagation, information reuse, and cheating measurement collapse. *Advances in Neural Information Processing Systems*, 2023.

Andrew Arrasmith, Lukasz Cincio, Rolando D Somma, and Patrick J Coles. Operator sampling for shot-frugal optimization in variational algorithms. *arXiv preprint arXiv:2004.06252*, 2020.

Xavier Bonet-Monroig, Hao Wang, Diederick Vermetten, Bruno Senjean, Charles Moussa, Thomas Bäck, Vedran Dunjko, and Thomas E O'Brien. Performance comparison of optimization methods on variational quantum algorithms. *Physical Review A*, 107(3):032407, 2023.

Joseph Bowles, Shahnawaz Ahmed, and Maria Schuld. Better than classical? the subtle art of benchmarking quantum machine learning models. *arXiv preprint arXiv:2403.07059*, 2024.

M Cerezo, Andrew Arrasmith, Ryan Babbush, et al. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021a.

M Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12(1):1791, 2021b.

Tianlong Chen, Xiaohan Chen, Wuyang Chen, Howard Heaton, Jialin Liu, Zhangyang Wang, and Wotao Yin. Learning to optimize: A primer and a benchmark. *Journal of Machine Learning Research*, 23(189):1–59, 2022.

Johnnie Gray. quimb: A python package for quantum information and many-body calculations. *Journal of Open Source Software*, 3(29):819, 2018.

Erik J Gustafson, Juha Tiihonen, Diana Chamaki, Farshud Sorourifar, J Wayne Mullinax, Andy C Y Li, Filip B Maciejewski, Nicolas P D Sawaya, Jaron T Krogel, David E Bernal Neira, et al. Surrogate optimization of variational quantum circuits. *Proceedings of the National Academy of Sciences*, 122(36):e2408530122, 2025.

Yiming Huang, Yajie Hao, Yuxuan Du, Jing Zhou, Xiao Yuan, and Xiaoting Wang. PALQO: Physics-informed model for accelerating large-scale quantum optimization. *Advances in Neural Information Processing Systems*, 38:65527–65549, 2025.

Benjamin D M Jones, Lana Mineh, and Ashley Montanaro. Benchmarking a wide range of optimisers for solving the Fermi–Hubbard model using the variational quantum eigensolver. *Quantum Science and Technology*, 10(4):045032, 2025.

Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M Chow, and Jay M Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, 2017.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

Martin Larocca, Nathan Ju, Diego García-Martín, Patrick J Coles, and Marco Cerezo. Theory of overparametrization in quantum neural networks. *Nature Computational Science*, 3(6):542–551, 2023.

Dong C Liu and Jorge Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical Programming*, 45(1):503–528, 1989.

Di Luo, Jiayu Shen, Rumen Dangovski, and Marin Soljacic. QuACK: Accelerating gradient-based quantum optimization with koopman operator learning. *Advances in Neural Information Processing Systems*, 36, 2023.

Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1):4812, 2018.

Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, 2018.

Charles Moussa, Max Hunter Gordon, Michal Baczyk, M Cerezo, Lukasz Cincio, and Patrick J Coles. Resource frugal optimizer for quantum machine learning. *Quantum Science and Technology*, 8(4):045019, 2023. doi: 10.1088/2058-9565/acef55.

Ken M Nakanishi, Keisuke Fujii, and Synge Todo. Sequential minimal optimization for quantum-classical hybrid algorithms. *Physical Review Research*, 2(4):043158, 2020.

Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J Love, Alán Aspuru-Guzik, and Jeremy L O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5:4213, 2014.

Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational Physics*, 378:686–707, 2019.

Peter J Schmid. Dynamic mode decomposition of numerical and experimental data. *Journal of Fluid Mechanics*, 656:5–28, 2010.

Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99(3):032331, 2019.

James C Spall. Multivariate stochastic approximation using a simultaneous perturbation gradient approximation. *IEEE Transactions on Automatic Control*, 37(3):332–341, 1992.

James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4:269, 2020. arXiv:1909.02108.

Naoki Yamamoto. On the natural gradient for variational quantum eigensolver. *arXiv preprint arXiv:1909.05074*, 2019.

A AI USE STATEMENT

Parts of this paper's text, code and analysis were produced with the assistance of generative AI. The authors take full responsibility for the final content of this paper, including any text produced with the assistance of generative AI.

B SCALE AND HAMILTONIAN DEPENDENCE

The most plausible way for this picture to reverse is at scale: in the barren-plateau regime (McClean et al., 2018; Cerezo et al., 2021b), large steps and quasi-Newton directions could fail where a learned surrogate still helps. We do not observe a reversal up to $n = 40$, the largest size reported in the work we audit; if anything the gap widens, with the surrogate ending 1.28 ± 1.25 above target at a budget where L-BFGS is 0.069 below it—an error bar larger than the mean, so the surrogate is not merely worse at $n = 40$ but erratic. Nor is the effect specific to the Ising Hamiltonian: swapping in a Heisenberg Hamiltonian while holding the ansatz fixed leaves the ordering unchanged. All rows are five seeds; at $n = 40$ the surrogate uses a 294M-parameter network and costs hours of classical compute per run.

C Discussion

Proposition 1 accounts for the part of the measurements that concerns the reference step; Proposition 2 is a statement about representable transition laws and does not by itself explain any particular surrogate's failure. The physics-informed residual $\theta^{(t+1)} - \theta^{(t)} + \eta\,\partial_\theta \hat{E}$ fits a gradient-descent step, so the gradient-flow path is the model's predictive target; Proposition 2 further shows that a rule carrying only t and θ cannot reproduce the general transition law of a stateful optimizer. The restriction to plain gradient descent is inherited from the derivation rather than chosen in the experimental design, which is also why the competing methods in these comparisons share the same base optimizer: they are surrogates for the same flow. By Proposition 1, the ratio reported against that flow then says as much about its step size as about the surrogate.

We read this as a scope statement rather than an error. Within its scope the machinery works, and works well. The difficulty is that the scope coincides with the regime in which a model-free alternative is available.

D Limitations

Our audit covers the hardware-efficient ansatz on two Hamiltonian families up to 40 qubits, in exact simulation without shot noise or device noise; we have now measured the finite-sampling regime (Section 6.4) but only at $n = 8$ and $n = 20$, at two shot counts and five seeds each, and with an idealised Gaussian sampling model rather than device noise. Our hardware measurements (Section 6.6) validate the sampling model and characterise the device error across parameter points and three depths, but on one device on one day, without error mitigation, and on one Hamiltonian; how that error behaves across devices, over time, or under mitigation we did not measure. Our hardware optimizer comparison is a single 4-qubit setting with four seeds and compares only the two classical optimizers; we did not run either surrogate on the device, so the hardware evidence bears on the sampling model and on optimizer behaviour, not on the surrogates. We do not cover the chemistry ansatz (UCCSD) used in part of the original evaluation. The head-to-head comparison uses the released default configuration; the original numbers may have been produced under settings we did not recover. We report classical wall-clock separately because the surrogate consumes substantial classical compute (a 73.5M-parameter network trained per outer iteration at $n = 20$) to save quantum queries, a trade that is favourable under some cost models and not others.

E Proofs

Proposition 1. Gradient descent on a quadratic gives $\theta^{(t)} = (I - \eta A)^t \theta^{(0)}$. Decomposing $\theta^{(0)}$ in the eigenbasis of A, the component along the μ-eigenspace contracts as $(1 - \eta\mu)^t$ and every other component contracts at least as fast. If that component is non-zero—the worst case, and the generic one—reaching relative accuracy ε requires $t \geq \log(1/\varepsilon)/\big(2 \log \frac{1}{1-\eta\mu}\big)$, which is $\Theta(\log(1/\varepsilon)/(\eta\mu))$ whenever $\eta\mu \ll 1$; the matching upper bound follows from the same expression. If instead $\theta^{(0)}$ has no component in that eigenspace the rate is governed by a larger eigenvalue and the count is smaller, which is why the statement is worst-case. Multiplying by the per-iteration cost $2pG$, which does not depend on η, gives $C_{\mathrm{GD}}(\eta)$, and the ratio statement is then immediate for any circuit-evaluation cost C_M not depending on η.

Proposition 2. Immediate from the state form. Fix θ and two velocities $v_1 \neq v_2$. Both states (θ, v_1) and (θ, v_2) have the same parameter vector, and their next increments are $\Delta_i = \beta v_i - \eta \nabla E(\theta)$, so $\Delta_1 - \Delta_2 = \beta(v_1 - v_2) \neq 0$ whenever $\beta > 0$. A map g with $\Delta = g(t, \theta)$ would have to return both values at the same argument, since t is common to the two states.

An earlier version of this paper argued the point differently, from the fact that an underdamped heavy-ball orbit returns arbitrarily close to earlier points in θ-space while its velocity has rotated, and claimed a separation of increments bounded below by a constant independent of the proximity δ. That claim is false for a convergent trajectory: as $\delta \to 0$ the near-revisits occur at ever later iterates, where both $\theta^{(t)}$ and $\Delta^{(t)}$ tend to zero, so the increment separation vanishes with δ. Numerically, the largest increment gap among pairs closer than δ falls from 4.4×10^{-1} at $\delta = 10^{-1}$ to 2.3×10^{-6}

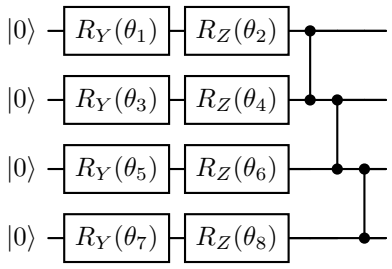
at $\delta = 10^{-6}$, tracking δ rather than approaching a constant. The state-space argument above is the correct one, and is the only one the paper relies on.

F REPRODUCTION DETAILS

Systems and ansatz. Transverse-field Ising and Heisenberg, built through `quimb`'s Hamiltonian constructors at the coefficients hard-coded in the released physics-informed implementation. Those constructors are written in spin operators $S^\alpha = \sigma^\alpha/2$ rather than Pauli matrices, so in Pauli terms the two open chains are

$$H_{\text{TFIM}} = -\sum_i Z_i Z_{i+1} - \tfrac{1}{2}\sum_i X_i, \qquad H_{\text{Heis}} = \tfrac{1}{4}\sum_i \big(X_i X_{i+1} + Y_i Y_{i+1} + Z_i Z_{i+1}\big),$$

with DMRG ground energies -20.400 and -8.682 at $n = 20$. An earlier version of this appendix quoted the constructor arguments ($J = -4$, $h = 1$) as if they were Pauli coefficients, which contradicts those ground energies by about a factor of four. Both systems use the same hardware-efficient ansatz used by that implementation: per layer, R_Y and R_Z on every qubit followed by a chain of CZ gates, depth 3, one layer of which is



Parameter counts are 48, 120 and 240 at $n = 8, 20, 40$. Initial parameters are drawn uniformly from $[0, 1]$, matching both released codebases. Circuits are contracted exactly as tensor networks with `quimb` (Gray, 2018) and JAX autodiff; no shot noise unless stated.

Resource unit. units $= G\, n_{\text{energy}} + 2pG\, n_{\text{grad}}$ with $G = 2$ commuting groups: an energy costs G circuit evaluations and each of the $2p$ parameter-shift energies behind one gradient costs G as well. An earlier version of these experiments charged gradients at $2p$, which is inconsistent with charging energies at G and undercharges gradient-heavy methods relative to energy-heavy ones by a factor of G. All numbers reported here use the corrected unit. For the physics-informed model we charge its acceptance test as an energy evaluation even though the released code also computes the gradient there, since the protocol never uses it. We charge it the two trajectory points it consumes from the warm-up, its 18 acceptance evaluations and the gradient calls of its inner loop—18 energy and 84–192 gradient calls per run—and not the remaining warm-up steps, which the released code runs as the comparison arm rather than for the surrogate. Charging the full 20-step warm-up instead would raise its cost by 9.6k circuit evaluations at $n = 20$, against it.

Surrogate configurations. The physics-informed model is run unmodified at its released defaults: warm-up 20 VQE iterations, 2 trajectory samples, 2000 predicted steps, network width $50{\cdot}2np$, 3000 training epochs per outer iteration, Adam at 10^{-4}. Each run is given its own working directory: the released code writes the trajectory to `Ising_data.npy` in the current directory, so parallel runs overwrite one another and silently return identical results across seeds.

For the Koopman surrogate we import the authors' `sliding_window_dmd_nonsq` and keep their constants: 4 base steps per piece, window size 1, 1000 predicted points each evaluated by an energy measurement, $\arg\min$ taken as the next iterate, $5n$ pieces, base step 5×10^{-6}. Their published configuration is 5 qubits at depth 250; that circuit costs about two hours of XLA compilation, so we preserve the over-parameterised regime their claim concerns ($p \gg 2 \cdot 4^n$) at 3 qubits and depth 30 (180 parameters). Pieces whose extrapolation diverges are truncated rather than aborting the run.

Finite sampling. $\hat{E} = E + \mathcal{N}(0, \|c\|_2^2/S)$ per expectation value, with gradients formed from parameter-shift differences of two such estimates. The noise we injected was computed from $\|c\|_2 = 18$ at $n = 20$, which follows from reading the constructor arguments as Pauli coefficients; for the Hamiltonian above $\|c\|_2 = \sqrt{24} \approx 4.90$, so the injected variance is $13.5\times$ the nominal value, at every size. Scaling $\|c\|_2$ by k is exactly equivalent to scaling S by $1/k^2$ in both the energy and the gradient channel, so these runs are exact draws from the stated model at a smaller shot count, and it is that effective count we report: what the generating code labels $S = 10^5$ and 10^4 is roughly 7.5×10^3 and 7.5×10^2 shots. The error is conservative—more sampling noise, not less—and is identical across methods within each row, so the comparison is unaffected. The accuracy criterion is evaluated on the *true* energy at the current parameters: it measures where the optimizer is, not what an experimenter could certify, which at these shot counts would be nothing near 10^{-3}.

Compute. All experiments run on a single laptop CPU (plus Metal for the surrogate's network). The complete set of runs reported here is a few hours, dominated by the physics-informed model, which trains a 73.5M-parameter network per outer iteration at $n = 20$ and 294M at $n = 40$.

Metric and target. Following the surrogate literature we fix a target energy $E^\star$ and count the queries needed to reach it, at the accuracy $\Delta E = |\hat{E} - E^\star| \le 10^{-3}$ used there. We take $E^\star$ to be the value the *reference optimizer of the method under audit* converges to after 6000 iterations, i.e. its own limit. This is not the exact ground state, which a depth-3 ansatz does not reach: at $n = 20$ the DMRG ground energy is -20.400 while $E^\star = -20.362$. Using the reference optimizer's own limit is the reading most favourable to the surrogate, since that limit is precisely what the surrogate is trained to predict.

The criterion is one-sided, $E \le E^\star + 10^{-3}$: overshooting to a *lower* energy counts as success. This is not cosmetic. A two-sided window $|E - E^\star| \le 10^{-3}$ is crossed in a single step by fast optimizers, which then never register as having arrived; in an earlier version of these experiments that silently recorded L-BFGS as a failure on one seed of five.

How the baselines were chosen. This is the point on which an audit of this kind can most easily cheat, so we state the protocol before the results. The two reference steps ($\eta = 10^{-2}$ and 5×10^{-6}) are the ones the audited papers use. For the added baselines we selected a step on a single system, $n = 8$ TFIM, from a three-point grid—$\{0.03, 0.1, 0.3\}$ for gradient descent and $\{0.01, 0.05, 0.1\}$ for Adam—and then held that choice fixed everywhere else: at $n = 20$ and $n = 40$, on the Heisenberg Hamiltonian, and under finite sampling. In the Koopman regime the two added steps ($\eta = 10^{-3}$ for gradient descent and 10^{-2} for Adam) were not selected by a grid at all. They were fixed in advance, three and four orders of magnitude above that method's reference 5×10^{-6}, and are the only two we ran there. L-BFGS-B was run at library defaults and tuned not at all. The surrogates, by contrast, were run unmodified at their own released default configurations, including network width, training epochs and prediction horizon. A per-system search on our side would only widen the gaps we report; we did not perform one.

F.1 Experimental configuration

F.2 What each experiment measures

- **Table 1** sweeps the added optimizers against the reference step and reports circuit evaluations to $E^\star$ as a ratio. Five seeds per cell; the ratio is formed seed-by-seed and averaged, and it is a ratio of a heavy-tailed variable, so the standard deviations are indicative rather than confidence intervals.
- **Table 2** runs the released physics-informed implementation and reads off both the queries it consumes and where each optimizer stands at that same budget. The second block is the comparison we consider decisive, because it is invariant to how one chooses the accuracy threshold.
- **Table 3** repeats the exercise for the Koopman surrogate in the regime its claim concerns. Five seeds at the published configuration and seven at the reduced one; the surrogate reaches the target in three of each, so its speedup column is conditioned on success.

Table 7: Fixed experimental settings. Everything not listed was left at the released or library default; we tuned no hyperparameter of any surrogate; the baseline-selection protocol is in Section 5.

quantity	value
qubits n	$8,\ 20,\ 40$ (TFIM, Heisenberg); 3 (Koopman regime)
ansatz depth	3; 30 in the over-parameterised regime
parameters p	$48,\ 120,\ 240$; 180
initial parameters	$\mathcal{U}[0,1]^p$, as in both released codebases
seeds	$2025,\ 7,\ 42,\ 11,\ 99$ (and $123,\ 7777$ for the Koopman runs)
reference horizon	6000 gradient steps to define $E^\star$
success criterion	$E \le E^\star + 10^{-3}$, one-sided, on the true energy
ground-state check	DMRG2, bond dimensions $16/32/64/128$, tol 10^{-10}
shot counts S	$10^4, 10^5, 10^6, 10^8$; $\|c\|_2 = 18$ at $n = 20$
resource unit	$G\, n_{\text{energy}} + 2pG\, n_{\text{grad}}$, $G = 2$

Table 8: Optimizer settings. The reference steps are the ones the two surrogate papers use. For the added baselines we picked a step from a three-point grid on one system at $n = 8$ and then held it fixed everywhere, including at $n = 20$ and $n = 40$ and on the Heisenberg Hamiltonian; L-BFGS-B was not tuned at all. A per-system search would only widen the gap we report.

optimizer	setting	how chosen
GD (reference, physics-informed)	$\eta = 10^{-2}$	as released
GD (reference, Koopman)	$\eta = 5 \times 10^{-6}$	as released
GD (tuned)	$\eta \in \{0.1,\ 0.3\}$	grid over $\{0.03, 0.1, 0.3\}$
Heavy-ball	$\eta = 0.03$, $\beta = 0.9$	textbook defaults
Adam	$\eta = 0.1$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$	η from $\{0.01, 0.05, 0.1\}$
L-BFGS-B	`scipy` defaults, `maxiter`=`maxfun`=3000	no tuning

- **Figure 3** is the same data as Table 2 without a threshold: accuracy against queries, all seeds shown.
- **Figure 2(a)** is a synthetic check of Proposition 1 on a quadratic, not a VQA experiment. **(b)** adds sampling noise to the $n = 20$ oracle.

F.3 Negative and discarded results

We report these because each would mislead a reader who repeated the experiment naively.

- Three ports of the Koopman surrogate preceded the one reported. The first accumulated the trajectory across pieces rather than refitting a fresh window per piece; the second aborted the run when an extrapolation diverged instead of truncating it. Both produced energies far worse than the method's, and both were our errors, not the method's.
- Our first head-to-head used a two-sided accuracy window and recorded L-BFGS as failing on one seed of five, when in fact it had crossed the window in a single step.
- Our first multi-seed batch of the physics-informed surrogate returned bit-identical results across seeds. The released code writes the trajectory to a fixed filename in the working directory, so concurrent runs overwrite one another.
- The step-size sweep of Figure 4 was first reported from a result file generated before the $2p \to 2pG$ correction to the resource unit, and its companion table was written by hand rather than generated. Every cost in it was therefore exactly half the correct value, and the surrogate's budget on the figure was an unsourced constant, 116k against the 47k of Table 2. Re-running the sweep under the current unit narrows the range over which Adam beats the surrogate from $25\times$ to $5\times$, removes gradient descent from that range entirely, and

Table 9: Software. All results were produced on one laptop; no GPU is required except for the physics-informed surrogate's network, which we ran on Metal.

package	version
Python	3.9.6
numpy	1.26.4
scipy	1.13.1
quimb	1.11.2
jax	0.4.30
torch	2.8.0
openfermion	1.6.1
matplotlib	3.9.4

deepens the reversal at $\eta = 10^{-2}$ from $1.2\times$ to $6\times$. Both the figure and its table are now generated from the same result file as Table 2, and a check ties them together.

- The Hamiltonians were quoted in the constructor's spin-operator arguments as though they were Pauli coefficients, which put them at odds with the ground energies reported alongside them, and the sampling noise was computed from the same misreading. See Appendix F: the injected noise is $13.5\times$ the nominal variance, so the sampled runs are exact draws from the stated model at a shot count $13.5\times$ smaller than labelled, which is what Table 4 now reports.
- We checked whether charging energy and gradient evaluations at a single flat rate had favoured our conclusion. It had not: the physics-informed surrogate spends 18 energy calls against 84–192 gradient calls, so its cost is dominated by the latter under either accounting.