THE COST OF QUANTUM FEATURES IN REACHER CONTROL: A RESOURCE-ACCOUNTED SIMULATOR STUDY

Anonymous authors
Paper under double-blind review

ABSTRACT

Small quantum circuits have few parameters, but a robot controller may need to run one every time it chooses an action. We study the resulting cost on Gymnasium MuJoCo Reacher-v5. A two-qubit feature map with finite-shot sampling (Q1) and a compact classical control share the same observations, action bounds, PPO schedule, and 50,000 training interactions. We evaluate each of 20 training seeds on the same 100 cases. Q1 finishes $0.00569\,\mathrm{m}$ closer to the target on average. Whether it meets the predeclared $0.010\,\mathrm{m}$ non-inferiority margin depends on how uncertainty is estimated: the one-sided bootstrap bound is $+0.00946\,\mathrm{m}$, while the Student-t sensitivity bound is $+0.01074\,\mathrm{m}$.

Success remained low: 9.4% for the compact control, 10.3% for Q1, and 11.3% for SB3. Q1 used 57.6 million shots per seed and took $0.915\,\mathrm{ms}$ per action at the median, about four times the compact policy's latency. A later 500k-step comparison retained the full state alongside the features. There, Q1 finished $0.02721\,\mathrm{m}$ farther from the target over 18 completed pairs (95% interval $[+0.01840, +0.03647]\,\mathrm{m}$), and two Q1 runs timed out. The follow-up failed its frozen zero-failure rule and remains exploratory. Q1 success was 16.2% on completed seeds, versus 26.8% for sine. A separate 20-seed SB3 screen improved return on every seed, yet two missed the 10% success floor. The evidence therefore concerns the cost of fixed representations in controllers with limited task success. It does not establish practical robot competence or predict performance on a quantum processor.

1 INTRODUCTION

A robot controller has to choose its next action on time. If that choice depends on a quantum circuit, the cost of reading the circuit matters at every step, even when the model has few parameters. Variational circuits have already appeared in reinforcement-learning policies and value functions, including simulated robot navigation and arm control (Chen et al., 2020; Lockwood & Si, 2020; Jerbi et al., 2021; Skolik et al., 2022; Hohenfeld et al., 2024; Acuto et al., 2022; Drăgan et al., 2025). We examine this repeated cost on Reacher-v5, where a simulated two-joint arm tries to bring its fingertip to a target.

A circuit feature map can have a classical counterpart with similar input and output sizes (Schuld et al., 2021; Landman et al., 2023; Schuld & Killoran, 2022; Bowles et al., 2024). This gives us a direct comparison: replace the feature map and measure both the target distance and the work needed to choose an action. The implementation needs care. Our PPO objective conditions on the sampled feature randomness, so its update must replay that randomness when revisiting an action. Drawing a new feature changes the likelihood comparison. Seeds and evaluation cases matter as well, as do tuning and other implementation choices (Henderson et al., 2018; Colas et al., 2018; Agarwal et al., 2021; Andrychowicz et al., 2021; Patterson et al., 2024; Nagarajan et al., 2018).

We ask whether a small quantum feature map can preserve task quality relative to a compact classical representation at an acceptable measured cost. Our primary Reacher cohort pairs 20 training seeds and evaluates every final checkpoint on the same 100 cases. We set the allowable distance gap before collecting the results. The compact policies share an encoder, action and value heads, PPO settings, and interaction budget, with the fixed feature map as the experimental factor. Their compute costs are measured separately. Parameter count, expressivity, and optimizer implementation are not all equal across the comparisons.

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.

This case study builds on existing QRL benchmarking work. Meyer et al. (Meyer et al., 2025) propose a statistical framework and test it on a configurable 6G task. Our contribution is the Reacher comparison and its supporting checks: replaying sampled features during PPO updates, verifying the simulator model, and accounting for circuit executions within the control loop. The v9 run and its recovered qualification agree on the full MuJoCo model hash. Because the run configuration omitted the qualification-file identifier, that agreement verifies model identity after the run rather than proving which qualification was used at dispatch. Earlier runs with only a proxy asset hash remain historical.

The result gives little reason to choose the tested quantum features on cost grounds. Q1 trains about four times slower than the compact classical map and consumes 57.6 million shots per seed in CPU simulation. Its median action latency is also roughly four times higher. The quality comparison is less decisive: the predeclared interval rule passes narrowly, a sensitivity check misses, and both policies rarely reach the target. Later experiments help explain the limits of that finding. We retain those failed screens because a small gap between two weak controllers is insufficient evidence for useful robot control.

2 Related work

Quantum reinforcement learning. Quantum machine learning includes several computational models (Biamonte et al., 2017; Cerezo et al., 2022). Early work represented policies or value functions with quantum states (Dong et al., 2008); later studies placed variational circuits inside agents for discrete and continuous control (Chen et al., 2020; Lockwood & Si, 2020; Jerbi et al., 2021; Skolik et al., 2022; Wu et al., 2020; Lan, 2021). Quantum RL has also been tested in simulated robotics: Hohenfeld et al. study wheeled-robot navigation with a hybrid circuit and DDQN; Acuto et al. study a virtual robotic arm using simulated quantum circuits; and Drăgan et al. evaluate a hybrid DDPG controller for continuous-action, LiDAR-based maze navigation (Hohenfeld et al., 2024; Acuto et al., 2022; Drăgan et al., 2025). These works establish that quantum RL in robotics, including continuous-action navigation, predates our study. Our narrower question is what a finite-shot feature map costs inside PPO on Reacher-v5 when circuit calls, shots, and per-action latency are measured alongside task quality.

Why these classical controls. Encoding data into a circuit can produce combinations of sine and cosine functions (Schuld et al., 2021; Pérez-Salinas et al., 2020; Mitarai et al., 2018). This makes simple trigonometric features a relevant comparison, rather than an arbitrary weak baseline. Work on classical random features and quantum feature spaces further motivates checking the representation itself (Rahimi & Recht, 2007; Landman et al., 2023; Havlíček et al., 2019; Schuld & Killoran, 2019). We therefore give the compact controls the same input bottleneck, feature count, and action readout. This removes some alternative explanations for a difference, although it does not make the functions they can represent identical.

Benchmarking and reproducibility. Meyer et al. (Meyer et al., 2025) define a sample-complexity estimator and statistical outperformance rule, test them on BeamManagement6G, and report CPU simulation runtimes. Their experiments average across 100 runs. Robotics studies also compare training time, stability, parameter counts, and test-time success; Hohenfeld et al. report training-resource measurements for simulated DDQN navigation, while Drăgan et al. compare repeated continuous-action maze-navigation runs and similarly sized policies (Hohenfeld et al., 2024; Drăgan et al., 2025). We do not claim these dimensions as new. Our study does not introduce a new estimator or benchmark. It uses 20 paired training seeds on continuous-control Reacher-v5, with a fixed 100-case evaluation cohort. The narrower implementation checks are that PPO replays the same stochastic feature in its likelihood ratio, the environment record pins the full MJCF model, and each policy call records circuit executions, shots, and end-to-end action latency. These checks complement, rather than replace, the statistical questions Meyer et al. address or the robotics evaluations in prior work.

The external SB3 reference answers a different question: how well does the compact family control the arm compared with an ordinary neural policy? It has more parameters and different PPO settings, so its performance gap cannot be attributed to the feature map alone. Implementation choices can substantially affect PPO results (Engstrom et al., 2020; Andrychowicz et al., 2021). We consequently

use the compact classical arm to study a representation change and SB3 to expose the family's task-quality limits. Neither comparison replaces the other.

Repeated seeds address another source of uncertainty: learning can produce different policies even when the settings are unchanged (Henderson et al., 2018; Agarwal et al., 2021). A fixed evaluation cohort makes the comparisons use the same cases, but it does not remove variation between trained policies. We therefore report uncertainty over training seeds and keep tuning separate from the held-out cohort (Eimer et al., 2023). Simulator provenance and execution conditions are recorded as well (Lynnerup et al., 2020; Fan et al., 2018; Mayor et al., 2025). Appendix G connects these choices to the broader literature.

Shot noise and software cost. Finite-shot readout makes the policy features noisy, and the noise depends on the state. Parameter-space and network noise offer classical comparisons (Plappert et al., 2018; Fortunato et al., 2018). Shot-frugal optimizers make the sample count itself an explicit cost (Sweke et al., 2020; Kübler et al., 2020). The gradient estimator and backend also affect how a differentiable quantum model is defined (Bergholm et al., 2018; Javadi-Abhari et al., 2024; Preskill, 2021). Section 5.6 tests the readout-noise explanation with a classical map; that control matches marginal variance, not covariance across qubits.

3 Setting and fairness rules

What we compare. We measure how far the fingertip is from the target at the end of an episode. For each trained policy, we average this distance over the same 100 starting cases. We then compare policies across training seeds, rather than treating every episode as an independent experiment. Writing this average as $d_s(\pi)$, the primary difference is

$$\Delta_s = d_s(\pi_{Q1}) - d_s(\pi_{\mathrm{cl}}), \qquad \bar{\Delta} = \frac{1}{20}\sum_{s=1}^{20} \Delta_s. \tag{1}$$

Negative differences favor Q1. We call Q1 non-inferior only if the one-sided 95% upper confidence bound is below $0.01\,\mathrm{m}$: the test asks whether the data rule out an average loss of one centimetre or more. This allowance is one fifth of our success radius. We chose it before evaluation; it is a project threshold rather than a validated engineering tolerance. The paired policies share a seed value but train in separate processes.

For v9, each evaluation case starts from a different reset seed. The shared manifest contains 100 such cases, one episode apiece. Compact policies use the same training seeds and interaction budget; evaluation uses only the final checkpoint. Earlier schedules differ. In particular, v8 repeated 10 reset seeds across 10 episode indices, so we do not describe it as 100 distinct initial conditions.

The quantum policy receives no input withheld from its control. Failed and timed-out cells stay in the artifact set and enter the predeclared sensitivity analysis. We report parameter storage separately from execution cost: reducing stored parameters does not by itself reduce the work needed to choose an action.

Quantum execution tiers. Q0 computes exact expectations and uses no shots. Q1 samples full bitstrings from the ideal circuit distribution, then estimates both Pauli-Z values from each sample. This preserves their joint sampling structure. Q2 adds a fixed synthetic noise profile: depolarizing survival $(1-p_1)^{n_1}(1-p_2)^{n_2}$ over logical gate counts, with $p_1 = 10^{-3}$, $p_2 = 10^{-2}$, and symmetric readout error $r = 2 \times 10^{-2}$. Q2 cannot run analytically. All three are software settings; Q2 is a stress test, not a model of a specific device.

4 Methods

4.1 One policy family, interchangeable feature maps

The controller first compresses the robot's state to two angles. A fixed classical or quantum map converts them to two features, which the action head uses to choose joint torques (Figure 1). In the implementation, we clip the ten observation entries to $[-10, 10]$ and pad them to 32 values before

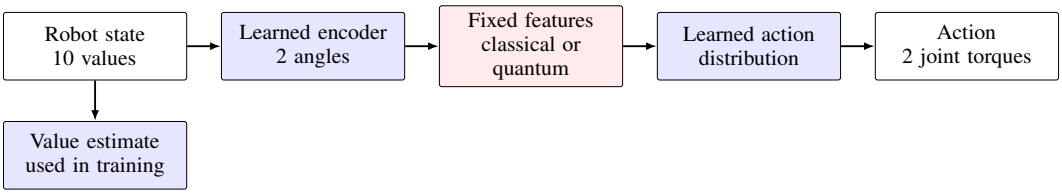

Figure 1: How the compact controller chooses an action. We change the fixed feature map in the highlighted box. The encoder and action distribution learn during training; a separate value estimate helps the training update. Both feature policies have 107 trainable values. Including fixed map parameters gives 109 stored values for sine and 115 for Q1. SB3 is a separate neural reference.

applying a learned linear encoder. The features lie in $[-1, 1]$. A linear readout sets the mean of a Gaussian distribution, with learned, state-independent $\log \sigma$; sampled actions pass through $\tanh$ and are rescaled to the environment's bounds.

The value head takes the padded observation directly. It does not see the encoded angles or feature-map output, so the compact policies share the same critic.

The encoder and policy/value heads learn during v9; feature-map parameters stay fixed (`train_encoder = true`, `train_feature_parameters = false`). There are 107 shared trainable values: 66 in the encoder and bias, 6 in the action readout and bias, 33 in the value head, and 2 for log standard deviation. Stored totals are 109 for the sine map and 115 for Q1.

We do not optimize the circuit parameters. Doing so would require a declared gradient estimator under shot noise, which this comparison does not test. Earlier protocols also froze the encoder; we retain them only as historical or secondary evidence.

Feature maps. The sine map is $\varphi_i = \sin(a_i + \theta_i)$ with $\theta \sim \mathcal{N}(0, 0.05^2)$. The Fourier map averages four harmonics with independent phases, giving eight parameters to match the circuit. The random-MLP map is $\tanh(\tanh(a^\top W_f + c) \odot s)$ with eight parameters. The quantum map returns $\langle Z_0 \rangle, \langle Z_1 \rangle$ of the circuit below.

Circuit. The circuit, `ry_rz_cx_v1`, uses two qubits and two layers. It first encodes each input angle with an R_Y gate. Every layer then applies R_Y and R_Z rotations to both qubits, followed by a linear chain of CNOT gates; the output is the Pauli-Z expectation on each qubit. The circuit has 10 single-qubit gates, 2 two-qubit gates, 8 parameters, and logical depth 7. No transpiler is used, so the recorded compiled depth equals the logical depth. We hash the circuit structure once; the hash is the same across tiers.

A classical control with readout noise. Finite shots add randomness to the quantum features. To check whether that randomness explains a result, we also add binomial noise to classical features. For $\varphi_i \in [-1, 1]$, we draw $k_i \sim \text{Binomial}(S, (1 + \varphi_i)/2)$ and use $\hat{\varphi}_i = 2k_i/S - 1$. Each feature then has the same sampling distribution as a Pauli-Z estimate with that mean. The classical draws are independent, so they do not reproduce correlations between the two quantum outputs. This control runs without a circuit; its emulated shot count is recorded separately.

4.2 Optimizer

The compact policies use PPO's clipped objective (Schulman et al., 2017) and generalized advantage estimation with $\gamma = 0.99$ and $\lambda = 0.95$ (Schulman et al., 2016). Each 50-step rollout gets two update epochs. We fix the clip ratio at 0.2, the learning rate at 3×10^{-3}, the entropy coefficient at 0.01, and the value coefficient at 0.5; advantages are normalized.

Both compact arms update the encoder with simultaneous perturbation stochastic approximation (SPSA): they estimate a direction by perturbing its parameters, using common random numbers for the paired evaluations (Spall, 1992). The encoder learning rate is 0.01; the action and value heads use exact gradients. Thus the circuit weights remain fixed while its inputs can change through

learning. For our stochastic-feature objective, the agent saves the rollout seed and reuses it when calculating the PPO likelihood again. With unchanged parameters, the likelihood ratio must be one (Appendix D). Each update is divided by $|\mathcal{B}|$; the L2 step norm is capped at 10, parameters are clipped to $[-100, 100]$, and training stops if an intermediate value is non-finite. V9 uses 50,000 interactions and 1,000 optimizer updates per compact arm. The preceding three-seed screen checked closeness to SB3 without requiring a minimum success rate.

4.3 Strong baseline

Stable-Baselines3 PPO (SB3) (Raffin et al., 2021) provides the stronger neural reference. Its policy and separate value network use two 64-unit tanh layers, for 9,925 parameters on Reacher-v5. SB3 receives the same 50,000 environment interactions, with $n_{\text{steps}} = 256$, batch size 64, 10 epochs, learning rate 3×10^{-4}, and one PyTorch CPU thread. We do not match its parameter count to the compact policies. The compact classical control isolates the feature map; SB3 shows how those policies compare with ordinary neural control.

4.4 Benchmarks

We use maintained Gymnasium MuJoCo tasks (Brockman et al., 2016; Towers et al., 2024; Todorov et al., 2012; Tassa et al., 2018). Reacher-v5 supplies the primary experiment: 10 observation values, two torque actions, and a 50-step episode at 50 Hz. We derive final fingertip distance and a success indicator at $0.05\,\mathrm{m}$ from the target vector.

Swimmer-v5 has eight observation values, two actuators, and a 1,000-step horizon; its endpoint is forward displacement. Pusher-v5 uses 23 observation values and seven actuators over 100 steps, with object-to-goal distance as its endpoint. Each adapter records simulator version, frame skip, timestep, horizon, action scaling, observation schema, and asset hash. Evaluation stops on a fingerprint mismatch. We also require a classical learning gate before treating any task as a confirmatory quantum benchmark.

4.5 Pre-specified statistics

Each method-seed cell contributes one value: its mean endpoint over 100 evaluation cases. We pair methods by training seed, then use a percentile bootstrap with 10,000 resamples and analysis seed zero. For Reacher, the predeclared non-inferiority decision uses the one-sided 95% upper bound. A Student-t bound serves as a sensitivity check. Secondary Reacher contrasts use paired tests with Holm correction within the declared family (Holm, 1979).

The v9 artifact, replay, resource, and cohort checks pass. Its bootstrap bound falls below the margin; the Student-t bound does not. We also repeat the primary test with failed cells assigned the worst observed seed-level mean. No v9 cells failed, so the complete-case and failure-inclusive results coincide.

Swimmer's frozen protocol named forward displacement, but the first analysis script tested return. A separate script now recomputes the declared endpoint from the immutable seed rows and applies the original standardized margin. It does not pass. We report Swimmer as a negative secondary result, not as a successful replication.

4.6 Checking the experiment implementation

Every run records process wall time, CPU time, peak resident memory, environment and policy calls, optimizer steps, gradients, deadline misses, circuit executions, shots, compilation, model size, qubits, depth, checkpoint hash, worker status, and the raw-ledger hash. The append-only ledger flushes each event. Totals for additive quantities must equal the event sum; capacity fields must equal the maximum.

The SB3 adapter counts optimizer minibatches directly rather than inferring them from its epoch counter. That accounting change came after the released v9 audit (§5.2). New confirmatory MuJoCo runs reject qualifications that include only an asset hash; they require the full model XML hash. Old records remain available for historical analysis. Artifacts refuse overwrites, and evaluation loads

checkpoints from disk in a separate process. In an interruption test, a resumed run reproduced 20 continuation episodes bit for bit.

For 128 random circuits, the simulator agrees with an independent tensor oracle and Qiskit Aer. Maximum analytic error was 2.96×10^{-8}; at 4,096 shots, the standardized residual mean was -9.4×10^{-5} and its RMS was 0.970 (Javadi-Abhari et al., 2024). The repository has 213 collected tests.

5 RESULTS

Appendix X indexes the retained runs, including preparation experiments and excluded attempts, and links each study to its detailed results.

5.1 THE EARLIER SCREEN DID NOT ESTABLISH CONTROLLER COMPETENCE

Before comparing feature maps, we ran the frozen `reacher_v8_controller_requalification` screen. It used three training seeds, 50,000 interactions per seed, and 100 evaluation episodes. Those episodes came from 10 reset seeds repeated across 10 episode indices, not 100 independent starts. All six compact-classical and SB3 cells completed.

The compact policy's return rose from early to late training on every seed (mean change $+7.615$), and its maximum action saturation was 0.01. Its mean paired distance excess over SB3 was $0.00802\,\mathrm{m}$, below the screen's $0.01\,\mathrm{m}$ tolerance. So v8 passed the rule it was given. That rule asked only whether the average distance stayed close to SB3; it set no minimum success rate. Calling this a competence pass would say more than the data do.

We later gave the compact policy four times as many interactions and set explicit success thresholds. It still failed: across three seeds, its mean final distance was $0.1312\,\mathrm{m}$ and success was 8.7%, compared with $0.0812\,\mathrm{m}$ and 34.3% for SB3. No Q1 arm was run at this budget. This later screen does not rewrite v9's frozen statistical test, but it changes how far we should generalize from it. Appendix N reports the gate metrics and hashes.

The retained update logs help locate a likely limitation in this controller. The compact policy's explained variance was negative in every training quartile and fell from -0.305 to -0.880 on average across seeds. SB3's rose from 0.309 to 0.989. Compact-policy returns still improved on all three seeds, with zero action saturation and small approximate KL. This pattern points to a poor value fit rather than a wholesale PPO collapse, but it does not show that the critic caused the task-quality gap: the actor, feature map, and update implementation differ as well.

We tested the critic explanation directly in v11. Replacing the compact policy's linear value head with a 32-unit MLP changed mean final distance by only -0.00189 m; derived success fell from 4.0% to 2.7%. The MLP-critic policy remained 0.03761 m behind SB3 on average and failed the predeclared success and distance gates. This three-seed screen cannot rule out other critic designs, but a wider value head alone did not close the gap (Appendix O).

We then ran a frozen fresh-seed gate for the compact classical feature controller on seeds 111–113, with 200,000 interactions per seed and 100 distinct evaluation starts. It failed the task-quality criteria: mean distance was $0.13553\,\mathrm{m}$ and success 4.7%, compared with $0.09223\,\mathrm{m}$ and 28.0% for full-state SB3. Learning returns increased on all three seeds and action saturation stayed at 0.01, but neither result met the endpoint-quality thresholds. We therefore ran no quantum arm. A later v15 screen also found that increasing this custom controller's feature dimension from 2 to 10 did not close the gap to SB3: its mean distance was $0.17559\,\mathrm{m}$ and success 5.0%, versus $0.09549\,\mathrm{m}$ and 23.7% for SB3. Because the change also increases trainable parameters, v15 is a diagnostic architecture screen, not a causal test of the bottleneck alone (Appendix S). These screens reinforce the competence limitation; they are not evidence about a quantum controller.

5.2 THE BOOTSTRAP RESULT DEPENDS ON THE INTERVAL METHOD

The primary cohort, `reacher_v9_full_identity_reacher`, paired 20 training seeds (44–63) across Q1, the compact classical feature policy, and SB3. Each arm received 50,000 interactions and used the same 100 evaluation cases. All 60 cells completed; no training seed or evaluation case

Table 1: v9 Reacher primary cohort. Values are means and standard deviations across 20 training seeds; each seed was evaluated on the same 100 cases. Q1 meets the predeclared bootstrap non-inferiority threshold, but the Student-t sensitivity bound exceeds the margin. Resource measures are from the Modal CPU simulation, not quantum hardware.

Method	Stored params	Distance (m)	Success	Train wall (s)	p50 (ms)	Train shots
Classical feature	109	0.14187 ± 0.03113	0.0935	107.2	0.231	0
Q1, 128 shots	115	0.13618 ± 0.02856	0.1025	416.7	0.915	5.76×10^7
SB3 PPO MLP	9,925	0.12937 ± 0.02336	0.1130	50.1	0.309	0

failed. We loaded each final checkpoint in a fresh process for evaluation. The recovered model records agree, although the run did not retain the qualification-file identifier used at dispatch (Appendix B).

Mean final fingertip distance was $0.13618\,\mathrm{m}$ for Q1 and $0.14187\,\mathrm{m}$ for the compact classical policy. The paired estimate, Q1 minus classical, was $\Delta = -0.00569\,\mathrm{m}$ (95% percentile-bootstrap interval $[-0.02386, +0.01241]\,\mathrm{m}$).

The one-sided bootstrap upper bound was $+0.00946\,\mathrm{m}$, just below the $0.010\,\mathrm{m}$ margin. A Student-t sensitivity bound of $+0.01074\,\mathrm{m}$ exceeds it. We retain the predeclared bootstrap decision, but the sensitivity check shows that a small change in uncertainty estimation changes the conclusion. Neither result establishes superiority: the two-sided interval includes both improvements and losses. Non-inferiority concerns an allowed loss, which is a different question from whether the candidate is better (Piaggio et al., 2012; Wellek, 2010). No cell failed, so the failure-inclusive and complete-case results are identical.

Success provides a second check on that interpretation. Q1 and the compact control reached the target in only about one episode in ten. Their small average difference therefore describes two policies with limited success. The predeclared test can answer its relative-distance question even though the tested controller family is inadequate for reliable reaching.

5.3 Corrected Q0/Q1/Q2 feasibility pilot

The corrected quantum pilot ran nine cells after the v7 direction checks. Q0, Q1, and Q2 each used the same policy path, 50,000 interactions, three seeds, and a two-case evaluation cohort. Mean final distance was $0.1605\,\mathrm{m}$ for Q0, $0.1587\,\mathrm{m}$ for Q1, and $0.1856\,\mathrm{m}$ for Q2; the compact classical policy averaged $0.1592\,\mathrm{m}$ and SB3 $0.1557\,\mathrm{m}$. One Q2 seed reached $0.2407\,\mathrm{m}$. None of the methods recorded a derived success. This is a feasibility pilot, not another primary estimate.

Each quantum seed executed 450,000 circuit calls. Q0 used analytic expectations and no shots; Q1 and Q2 each used 57.6 million shots. Mean evaluation p50 latency was about $1.07\,\mathrm{ms}$ for Q0, $1.22\,\mathrm{ms}$ for Q1, and $1.00\,\mathrm{ms}$ for Q2. With three seeds and high Q2 variance, the pilot cannot support a quantum-advantage claim.

5.4 Superseded Reacher-v5 non-inferiority study

The earlier `reacher_v5` study completed 100 cells across 20 seeds and five methods. An audit then found that PPO resampled finite-shot features during updates. Even with unchanged parameters, the likelihood ratio could differ from one. We withdrew its non-inferiority decision. The corrected agent saves and reuses each rollout feature seed, but that fix cannot repair already trained checkpoints. Appendix A retains the historical results and their exclusions. Later evaluations of those checkpoints describe their behavior only; they do not validate the corrected training method.

5.5 The resource frontier is unfavorable to Q1

Timing claims use the sequential v9 cohort and apply to the recorded Modal CPU-simulator setup. The compact sine policy had a mean per-seed median action latency of $0.231\,\mathrm{ms}$; Q1 took $0.915\,\mathrm{ms}$, and SB3 took $0.309\,\mathrm{ms}$. Training showed the same pattern. Q1 averaged 416.7 s per seed, versus 107.2 s for the compact control and 50.1 s for SB3. It also used 450,000 circuits, 57.6 million shots, and 3.33 GB peak training RSS per seed.

Table 2: Shot-frontier replication on Reacher-v5: Q1 against the sine map with binomial readout noise at the same shot count. Twenty paired seeds per cell; distance in metres; 95% bootstrap intervals; the upper bound is one-sided 95%. Train shots are per seed for Q1; the control executes no circuits.

Shots	Q1 distance	Control distance	Difference	95% CI	Upper bound	Q1 train shots
16	0.1265	0.1222	$+0.0044$	$[-0.0006, +0.0126]$	0.0116	2.4×10^6
256	0.1245	0.1216	$+0.0028$	$[-0.0008, +0.0084]$	0.0075	3.84×10^7
1,024	0.1237	0.1223	$+0.0014$	$[-0.0011, +0.0050]$	0.0044	1.536×10^8

Mean p50 and p99 latency stayed under the 20 ms reference deadline, though a few individual cells missed it. Q1 used more training time and had higher action latency than the compact control. The timings do not predict QPU latency, energy, or price. We exclude concurrent worker wall time because scheduling contention would change the quantity being measured. The resource frontier and raw event ledgers accompany the study artifacts.

5.6 Shot frontier and the marginally noise-matched control

The primary classical control has no readout noise, which leaves circuit structure and finite-shot randomness entangled. A separate protocol, `shots_v8`, added independent binomial noise to the sine features at the same marginal variance as Q1. It covered 20 paired seeds at 16, 256, and 1,024 shots; all 120 cells completed.

Q1's point estimate was worse at every budget: $+0.0044$, $+0.0028$, and $+0.0014\,\mathrm{m}$. None of the paired intervals excluded zero. At 256 and 1,024 shots, the reference upper bound fell below $0.01\,\mathrm{m}$, but this secondary study made no formal non-inferiority claim (Table 2; Figure 4).

Mean return also favored the classical control. The paired bootstrap differences were -0.375 (CI $[-0.669, -0.125]$), -0.294 ($[-0.559, -0.075]$), and -0.260 ($[-0.514, -0.050]$). The protocol defined no hypothesis family for these secondary outcomes, so we treat them as descriptive. Matching marginal readout noise did not reveal a quality gain for Q1.

These estimates are not pooled with v9; `shots_v8` has its own policy settings and artifacts. The noisy classical control has lower distance point estimates and uses no circuits, but the intervals do not establish a distance advantage. Its independent feature draws match only marginal shot variance, not cross-qubit covariance, so we cannot attribute the return gap to circuit structure. Five-seed ablations of width, depth, map family, and Q2 noise intensity (Appendix E) remain exploratory. They do not rank feature maps in general.

5.7 Secondary checks do not reverse the result

We also re-evaluated archived v5 checkpoints under 16 stress conditions: observation and action noise, action delay, and dynamics changes of up to $\pm 20\%$. The suite contains 960 cells and includes held-out target sectors. These policies predate the corrected Q1 training path. In the graded conditions, paired Q1-classical distance differences ranged from $+0.0001$ to $+0.0023\,\mathrm{m}$. Each pointwise upper bound was below $0.01\,\mathrm{m}$, but the bounds were not simultaneous. Held-out targets hurt every method; Q1's upper bound rose to $0.012\,\mathrm{m}$. Condition-level results appear in Figure 5 and Appendix E.

Repeated evaluation found no detectable latency drift. The check measures deterministic replay and finite-shot variation, not physical aging or calibration drift.

With stochastic actions, Q1 minus classical was $+0.0014\,\mathrm{m}$ and the upper bound was $0.0056\,\mathrm{m}$. This secondary evaluation does not repair the primary gate.

5.8 Secondary Swimmer assessment: no cross-task replication

Swimmer-v5 gives us a second recognized simulator, but not a positive replication. The frozen protocol specified forward displacement; the first analysis script tested return. We disclose that error and reanalyze the retained per-seed displacement rows for all 20 seeds and 100 cases.

Mean displacement was $1.0182\,\mathrm{m}$ for Q1 and $1.0199\,\mathrm{m}$ for the sine control. Q1 minus control was $-0.00168\,\mathrm{m}$ (95% paired seed-bootstrap interval $[-0.01669, +0.01168]\,\mathrm{m}$). Here greater distance is better, so the test uses the lower confidence bound. That bound, $-0.01419\,\mathrm{m}$, falls below the allowed loss of $-0.00700\,\mathrm{m}$ (0.25 control standard deviations). Non-inferiority is not established. Pusher also failed its classical learning gate, so it remains negative feasibility evidence. Neither task supports a cross-task superiority claim.

6 Discussion

Choosing a feature map for this controller involves more than its distance score. In v9, the non-inferiority decision changes with the interval method, while the extra simulation cost is directly measured. The later v20 comparison is less favorable to Q1. Across 18 completed pairs, it increased final distance by $0.02721\,\mathrm{m}$ (95% interval $[0.01840, 0.03647]\,\mathrm{m}$). Success was 16.2% for Q1 and 26.8% for sine, and Q1 took longer to produce actions. Two Q1 runs timed out. Those failures violate the frozen zero-failure rule, so the completed-pair estimate must remain exploratory.

The controller itself also limits what we can learn about the feature map. V9 learned the shared encoder and action/value heads while keeping circuit parameters fixed. V20 instead used SB3 with the full state and fixed additional features. This removes the requirement that all action information pass through two features, but also changes the optimizer and training budget. The contrast between v9 and v20 therefore cannot isolate the cause of their different results. Within v20, both arms share those choices, making its completed-pair comparison informative about the two fixed maps under that specific configuration.

The separate v21 SB3 screen makes another distinction clearer: all 20 seeds improved their training return, but two still missed the required success floor at evaluation. Return includes behavior throughout an episode, whereas success asks whether the final fingertip is close enough to the target. Improvement in one need not imply reliable performance on the other. Together with the failed compact-controller screens, this is why we report absolute success alongside relative distance instead of interpreting every near tie as useful control.

For this reason, we interpret the study as evidence about fixed representations and their measured simulation cost. It leaves the behavior of trained quantum circuits and physical quantum processors unresolved. The practical lesson from these runs is narrower: count the work needed for each action, and check absolute task success before treating closeness to a classical baseline as useful control.

7 Limitations

All quantum results use CPU simulation; Q2 noise is synthetic, and no QPU was used. The v20 state-augmented comparison had two Q1 training timeouts and fails its frozen zero-failure gate; its complete-case statistics are exploratory. Success remained modest even for the sine comparison, so this is not evidence of reliable robot control. The $0.01\,\mathrm{m}$ margin is project-defined. Swimmer misses its corrected margin; stress tests use archived v5 checkpoints, and v15 changes width and parameterization together.

8 Conclusion

The tested quantum features added simulation cost without establishing a reliable quality benefit. Reacher v9 narrowly met one non-inferiority rule and missed another; the longer v20 comparison favored sine on the 18 completed pairs, with two Q1 timeouts limiting the inference. Even full-state SB3 missed the frozen per-seed success requirement in v21. Swimmer and Pusher provide no successful cross-task replication. These results support a limited conclusion about the policy families we tested: small circuits alone did not make control cheaper, and similar average distance did not imply reliable reaching.

Reproducibility statement

We retain trajectories, checkpoints, and resource events for reanalysis. The local v9 archive contains 602 manifest-listed files. Later archives contain 94 files for v15, 445 for v16, 29 for v17, 27 for v18, 32 for v19, 294 for v20, and 142 for v21. The v20 archive includes both timeout records and the exploratory analysis. The code-release manifest binds the v9 archive, and separate manifests verify later runs. These files are staged locally; reviewers do not currently have access to the raw archives. This limits independent reproduction of the trajectory-level analysis from the submission materials.

Ethics statement

This study uses only physics simulators and no personal data. We report failed gates, protocol errors, and resource costs to avoid overstating simulator evidence.

AI use and contributions. The authors chose the research question, designed the experiments, and developed the initial implementation. OpenAI Codex assisted with debugging and optimizing the experiment software, including fixes to errors in the authors' initial code. It also assisted with test design, literature search, analysis-code review, preparation of figures and tables, and manuscript editing. The authors made the research decisions and remain responsible for verifying the code, results, references, and final claims. Both authors contributed to the study design, implementation, experiments, analysis, and writing.

REFERENCES

Alberto Acuto, Paola Barillà, Ludovico Bozzolo, Matteo Conterno, Mattia Pavese, and Antonio Policicchio. Variational quantum soft actor-critic for robotic arm control. *arXiv preprint arXiv:2212.11681*, 2022.

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc Bellemare. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pp. 29304–29320, 2021.

Marcin Andrychowicz, Anton Raichuk, Piotr Stańczyk, Manu Orsini, Sertan Girgin, Raphaël Marinier, Léonard Hussenot, Matthieu Geist, Olivier Pietquin, Marcin Michalski, Sylvain Gelly, and Olivier Bachem. What matters for on-policy deep actor-critic methods? A large-scale study. In *International Conference on Learning Representations*, 2021.

Ville Bergholm, Josh Izaac, Maria Schuld, Christian Gogolin, Shahnawaz Ahmed, Vishnu Ajith, M. Sohaib Alam, Guillermo Alonso-Linaje, B. AkashNarayanan, Ali Asadi, et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*, 2018.

Jacob Biamonte, Peter Wittek, Nathan Pancetti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195–202, 2017. doi: 10.1038/nature23474.

Xavier Bouthillier, Cécile Laurent, and Pascal Vincent. Unreproducible research is reproducible. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 725–734. PMLR, 2019. URL https://proceedings.mlr.press/v97/bouthillier19a.html.

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.

Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, 2021.

Marco Cerezo, Guillaume Verdon, Hsin-Yuan Huang, Lukasz Cincio, and Patrick J. Coles. Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9):567–576, 2022. doi: 10.1038/s43588-022-00311-3.

Samuel Yen-Chi Chen, Chao-Han Huck Yang, Jun Qi, Pin-Yu Chen, Xiaoli Ma, and Hsi-Sheng Goan. Variational quantum circuits for deep reinforcement learning. *IEEE Access*, 8:141007–141024, 2020.

Karl Cobbe, Oleg Klimov, Chris Hesse, Taehoon Kim, and John Schulman. Quantifying generalization in reinforcement learning. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 1282–1289. PMLR, 2019. URL https://proceedings.mlr.press/v97/cobbe19a.html.

Cédric Colas, Olivier Sigaud, and Pierre-Yves Oudeyer. How many random seeds? Statistical power analysis in deep reinforcement learning experiments. *arXiv preprint arXiv:1806.08295*, 2018.

Daoyi Dong, Chunlin Chen, Hanxiong Li, and Tzyh-Jong Tarn. Quantum reinforcement learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 38(5):1207–1220, 2008.

Theodora Augustina Drăgan, Alexander Künzner, Robert Wille, and Jeanette Miriam Lorenz. Continuous quantum reinforcement learning for robot navigation. In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence*, volume 1, pp. 807–814, 2025. doi: 10.5220/0013371800003890.

Yan Duan, Xi Chen, Rein Houthooft, John Schulman, and Pieter Abbeel. Benchmarking deep reinforcement learning for continuous control. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pp. 1329–1338. PMLR, 2016. URL https://proceedings.mlr.press/v48/duan16.html.

Vedran Dunjko and Hans J. Briegel. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, 81(7):074001, 2018.

Bradley Efron and Robert J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

Theresa Eimer, Marius Lindauer, and Roberta Raileanu. Hyperparameters in reinforcement learning and how to tune them. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 9104–9149. PMLR, 2023. URL https://proceedings.mlr.press/v202/eimer23a.html.

Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Madry. Implementation matters in deep policy gradients: A case study on ppo and trpo. In *International Conference on Learning Representations*, 2020.

Linxi Fan, Yuke Zhu, Jiren Zhu, Zihua Liu, Orien Zeng, Anchit Gupta, Joan Creus-Costa, Silvio Savarese, and Li Fei-Fei. Surreal: Open-source reinforcement learning framework and robot manipulation benchmark. In *Proceedings of the 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pp. 767–782. PMLR, 2018. URL https://proceedings.mlr.press/v87/fan18a.html.

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration. In *International Conference on Learning Representations*, 2018.

Keren Gili, Marta Mauri, and Alejandro Perdomo-Ortiz. Generalization metrics for practical quantum advantage in generative models. *arXiv preprint arXiv:2201.08770*, 2023.

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

Peter Henderson, Riashat Islam, Philip Bachman, Joelle Pineau, Doina Precup, and David Meger. Deep reinforcement learning that matters. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

Mohamed Hibat-Allah, Marta Mauri, Juan Carrasquilla, and Alejandro Perdomo-Ortiz. A framework for demonstrating practical quantum advantage: comparing quantum against classical generative models. *Communications Physics*, 7(1):68, 2024. doi: 10.1038/s42005-024-01552-6.

Hans Hohenfeld, Dirk Heimann, Felix Wiebe, and Frank Kirchner. Quantum deep reinforcement learning for robot navigation tasks. *IEEE Access*, 12:87217–87236, 2024. doi: 10.1109/ACCESS.2024.3417808.

Sture Holm. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics*, 6(2):65–70, 1979.

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.

Riashat Islam, Peter Henderson, Maziar Gomrokchi, and Doina Precup. Reproducibility of benchmarked deep reinforcement learning tasks for continuous control. *arXiv preprint arXiv:1708.04133*, 2017.

Ali Javadi-Abhari, Matthew Treinish, Kevin Krsulich, Christopher J. Wood, Jake Lishman, Julien Gacon, Simon Martiel, Paul D. Nation, Lev S. Bishop, Andrew W. Cross, Blake R. Johnson, and Jay M. Gambetta. Quantum computing with Qiskit. *arXiv preprint arXiv:2405.08810*, 2024.

Sofiene Jerbi, Casper Gyurik, Simon Marshall, Hans Briegel, and Vedran Dunjko. Parametrized quantum policies for reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 34, pp. 28362–28375, 2021.

Scott Jordan, Yash Chandak, Daniel Cohen, Mengxue Zhang, and Philip Thomas. Evaluating the performance of reinforcement learning algorithms. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 4962–4973. PMLR, 2020. URL https://proceedings.mlr.press/v119/jordan20a.html.

Jonas M. Kübler, Andrew Arrasmith, Lukasz Cincio, and Patrick J. Coles. An adaptive optimizer for measurement-frugal variational algorithms. *Quantum*, 4:263, 2020.

Daniël Lakens. Equivalence tests: A practical primer for t tests, correlations, and meta-analyses. *Social Psychological and Personality Science*, 8(4):355–362, 2017.

Qingfeng Lan. Variational quantum soft actor-critic. *arXiv preprint arXiv:2112.11921*, 2021.

Jonas Landman, Slimane Thabet, Constantin Dalyac, Hela Mhiri, and Elham Kashefi. Classically approximating variational quantum machine learning with random Fourier features. In *International Conference on Learning Representations*, 2023.

Yunchao Liu, Srinivasan Arunachalam, and Kristan Temme. A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, 17:1013–1017, 2021. doi: 10.1038/s41567-021-01287-z.

Owen Lockwood and Mei Si. Reinforcement learning with quantum variational circuits. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 16, pp. 245–251, 2020.

Nicolai A. Lynnerup, Laura Nolling, Rasmus Hasle, and John Hallam. A survey on reproducibility by evaluating deep reinforcement learning algorithms on real-world robots. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 466–489. PMLR, 2020. URL https://proceedings.mlr.press/v100/lynnerup20a.html.

Walter Mayor, Johan Obando-Ceron, Aaron Courville, and Pablo Samuel Castro. The impact of on-policy parallelized data collection on deep reinforcement learning networks. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 43331–43352. PMLR, 2025. URL https://proceedings.mlr.press/v267/mayor25a.html.

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.

Nico Meyer, Christian Ufrecht, Maniraman Periyasamy, Daniel D. Scherer, Axel Plinge, and Christopher Mutschler. A survey on quantum reinforcement learning. *arXiv preprint arXiv:2211.03464*, 2022.

Nico Meyer, Christian Ufrecht, George Yammine, Georgios Kontes, Christopher Mutschler, and Daniel Scherer. Benchmarking quantum reinforcement learning. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 43934–43964. PMLR, 2025. URL https://proceedings.mlr.press/v267/meyer25b.html.

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

Prabhat Nagarajan, Garrett Warnell, and Peter Stone. Deterministic implementations for reproducibility in deep reinforcement learning. In *2nd Reproducibility in Machine Learning Workshop at ICML*, 2018.

Brian A. Nosek, Charles R. Ebersole, Alexander C. DeHaven, and David T. Mellor. The preregistration revolution. *Proceedings of the National Academy of Sciences*, 115(11):2600–2606, 2018.

Andrew Patterson, Samuel Neumann, Martha White, and Adam White. Empirical design in reinforcement learning. *Journal of Machine Learning Research*, 25(318):1–63, 2024. URL https://jmlr.org/papers/v25/23-0183.html.

Adrián Pérez-Salinas, Alba Cervera-Lierta, Elies Gil-Fuster, and José I. Latorre. Data re-uploading for a universal quantum classifier. *Quantum*, 4:226, 2020.

Gilda Piaggio, Diana R. Elbourne, Stuart J. Pocock, Stephen J. W. Evans, and Douglas G. Altman. Reporting of noninferiority and equivalence randomized trials: extension of the CONSORT 2010 statement. *JAMA*, 308(24):2594–2604, 2012.

Joelle Pineau, Philippe Vincent-Lamarre, Koustuv Sinha, Vincent Larivière, Alina Beygelzimer, Florence d'Alché Buc, Emily Fox, and Hugo Larochelle. Improving reproducibility in machine learning research (a report from the NeurIPS 2019 reproducibility program). *Journal of Machine Learning Research*, 22(164):1–20, 2021.

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. In *International Conference on Learning Representations*, 2018.

John Preskill. Quantum computing in the NISQ era and beyond. *Quantum*, 2:79, 2018.

John Preskill. Quantum computing in the nisq era and beyond. *SIAM Journal on Computing*, 50(5): S8–S11, 2021.

Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Information Processing Systems*, volume 20, 2007.

Donald J. Schuirmann. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics*, 15(6):657–680, 1987.

Maria Schuld. Supervised quantum machine learning models are kernel methods. *arXiv preprint arXiv:2101.11020*, 2021.

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

Maria Schuld and Nathan Killoran. Is quantum advantage the right goal for quantum machine learning? *PRX Quantum*, 3(3):030101, 2022.

Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3):032430, 2021.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations*, 2016.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Sukin Sim, Peter D. Johnson, and Alán Aspuru-Guzik. Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, 2(12):1900070, 2019.

Andrea Skolik, Sofiene Jerbi, and Vedran Dunjko. Quantum agents in the Gym: a variational quantum algorithm for deep Q-learning. *Quantum*, 6:720, 2022.

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

Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K. Fährmann, Barthélémy Meynard-Piganeau, and Jens Eisert. Stochastic gradient descent for hybrid quantum-classical optimization. *Quantum*, 4:314, 2020.

Yuval Tassa, Yotam Doron, Alistair Muldal, Tom Erez, Yazhe Li, Diego de Las Casas, David Budden, Abbas Abdolmaleki, Josh Merel, Andrew Lefrancq, Timothy Lillicrap, and Martin Riedmiller. Deepmind control suite. *arXiv preprint arXiv:1801.00690*, 2018.

Emanuel Todorov, Tom Erez, and Yuval Tassa. MuJoCo: A physics engine for model-based control. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5026–5033, 2012.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Stefan Wellek. *Testing Statistical Hypotheses of Equivalence and Noninferiority*. Chapman & Hall/CRC, 2nd edition, 2010.

Shaojun Wu, Shan Jin, Dingding Wen, and Xiaoting Wang. Quantum reinforcement learning in continuous action space. *arXiv preprint arXiv:2012.10711*, 2020.

A Protocol history, superseded runs, and exclusions

First confirmatory attempt (`reacher_v1`, laptop run, superseded). The same protocol file was later rerun on Modal with corrected agents; this paragraph covers only the earlier laptop run with the legacy implementation. It used 20 seeds, six methods (linear PPO, classical feature PPO, SB3, Q0, Q1, and Q2), and 25,000 interactions per cell. Training ran sequentially on an Apple-silicon laptop, with 128 training and 1,024 evaluation shots. All 120 cells eventually completed.

The Q1-minus-classical difference was $-0.0001\,\mathrm{m}$ (CI $[-0.0165, +0.0161]$), and its one-sided upper bound was $0.0136\,\mathrm{m}$, so the gate failed. In secondary contrasts, Q0 and Q2 were significantly worse than the classical feature map ($+0.021$ and $+0.022\,\mathrm{m}$, Holm $p = 0.022$); Q1 was also worse than linear PPO and SB3. Figure 2a shows the seed-level values. The hybrid agents had 101 trained classical parameters and 8 fixed circuit parameters. Q1 used 9.6×10^6 training shots per seed; its latency was seven times the linear baseline, and its training wall time was four times as long.

The first attempt stopped after 106 cells when a memory watchdog killed it. A second run filled the disk after 101 cells. We resumed only missing cells, moved the partial cell aside, and excluded it from analysis. Only the completed run was analyzed.

Diagnosis. Every NumPy arm's learning curve declined from the first to the last training quartile; late-minus-early return ranged from -27.8 to -39.7. A one-parameter finite-difference check exposed the gradient error. The legacy agent reported -1.587, while the numerical gradient was -0.561. Its update treated the transformed policy mean as a raw linear logit. The hybrid version had the same defect and no gradient through the circuit expectations. We froze those agents for reproducibility and replaced them with the feature-map agent in Section 4.1.

Stability gate (`shots_v5`). The sine and Q1 arms each used three paired seeds and 50,000 interactions. Both missed the required non-negative late-minus-early return, at -22.3 and -31.9; one Q1 seed saturated its control and lost 85 return units. We traced the collapse to an update step applied once per transition without dividing by rollout length, causing a 50-fold overshoot on Reacher. The repaired update divides by $|\mathcal{B}|$. We then logged KL, clip fraction, saturation, gradient norm, and explained variance for every update.

Classical-first competence gate (`reacher_v7`). A repaired-agent pilot had shown improving returns for both the compact policy and SB3. The later v7 screen, however, failed the declared competence test. At 50,000 steps, compact PPO averaged $0.1603\,\mathrm{m}$ final distance versus $0.1374\,\mathrm{m}$ for SB3, and all three compact seeds had negative late-training explained variance. We reopened quantum testing only as an exploratory audit, not as a qualified non-inferiority study. The three-seed matched-feature pilot (`reacher_v4`) and frozen `reacher_v5` protocol remain in the record under their original scopes.

Exclusions. We excluded the first Q2 noise pilot because its single-qubit noise term was inert, and the first Pusher smoke because its asset hash depended on non-portable native memory. Robustness analyses v1 and v2 also stay out: one had a reporting-schema error, and the other mixed environment seeds in its longevity summary. The stochastic-deployment attempts used action RNG that was not fixed independently per cell. We exclude timing fields from the parallel confirmatory run because scheduling contaminated them; its task metrics remain valid. Pusher's 500k and 1M qualifications remain negative evidence only. Finally, one preempted Q0 seed in the Pusher 200k extension and one preempted SB3 seed in the Swimmer gate are retained as incomplete attempts beside their completed replacements.

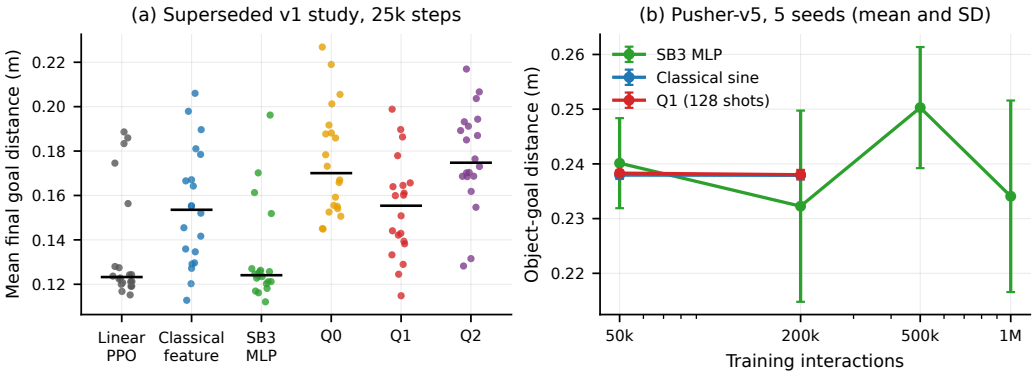

Figure 2: (a) The superseded first Reacher study with the legacy agents at 25,000 interactions; its gate failed and its agents were later found to have a gradient defect. (b) Pusher-v5 object-goal distance against interaction budget; no method, including SB3 at one million interactions, learns to move the object toward the goal.

Table 3: Superseded Reacher-v5 study, 20 seeds per method and 100 shared evaluation cases. These values are retained for provenance only; the finite-shot training interpretation is withdrawn because rollout feature realizations were not replayed during PPO updates.

Method	Params	Final distance (m)	Success	Return	Latency (ms)	Train shots	Circuits
Classical feature	109	0.1223 ± 0.0030	0.045	-9.81	0.299	0	0
SB3 PPO MLP	9,925	0.1171 ± 0.0060	0.085	-9.10	0.420	0	0
Q0 analytic	115	0.1237 ± 0.0067	0.055	-10.07	–	0	150,000
Q1 128 shots	115	0.1234 ± 0.0055	0.054	-10.06	1.119	1.92×10^7	150,000
Q2 128 shots	115	0.1259 ± 0.0165	0.045	-10.09	–	1.92×10^7	150,000

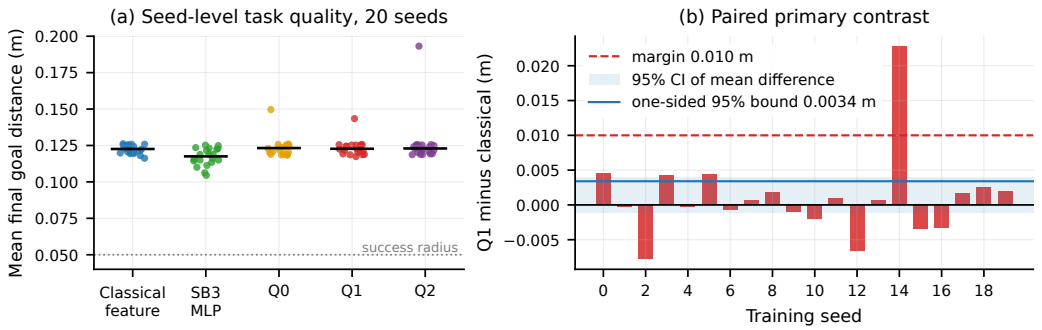

Figure 3: Superseded Reacher-v5 result. (a) Seed-level mean final distance per method. (b) Paired Q1 minus classical difference per seed; the original upper bound is shown for provenance, not as a current non-inferiority claim.

B HARNESS INTEGRITY EVIDENCE

Independent oracle. The reference simulator in `tests/` represents the state as a $(2,)^n$ amplitude tensor and applies each gate with explicit slice arithmetic. It shares no helper code with the production backend. Tests cover analytic expectations for one to four qubits and one to three layers (tolerance 10^{-6}), plus finite-shot means and variances at 512 shots over 200 seeds against $(1 - \langle Z \rangle^2)/S$. They also check fixed-seed determinism, the Q2 noise profile and its refusal of analytic execution, gate counts, logical and compiled depth, circuit hashes, and nine invalid-input paths. The review found incorrect depth and gate counts for general n-qubit circuits; the two-qubit policy outputs did not change.

Qiskit Aer cross-validation. We evaluated 128 random two-qubit, two-layer circuits with both the NumPy backend and Qiskit Aer 0.17.2 (Qiskit 2.5.2). Sampling seeds differed between implementations, so finite-shot agreement was assessed statistically. The largest analytic error was 2.96×10^{-8}. At 4,096 shots, standardized residuals had mean -9.4×10^{-5} and RMS 0.970. Both predeclared checks passed; the report is in the repository.

Interruption and resume. We stopped a Q1 worker at episode 20 of a 40-episode run, immediately after a checkpoint boundary. It left an `interrupted` artifact and a partial ledger of 3,081 events. A fresh process resumed from that checkpoint and was compared with an uninterrupted run.

The 20 continuation episodes matched exactly in actions (maximum absolute difference 0), returns, and metrics. Policy, value, bias, $\log \sigma$, and circuit parameters also matched. Python, NumPy, and PyTorch RNG states and seed streams were identical, as were environment steps, policy calls, gradient evaluations, 6,000 circuit executions, and 768,000 shots. The resumed process recorded one extra compilation call because it started fresh; the report notes that difference.

v9 release audit. The audit covers 60 Reacher method-seed cells, with 20 cells per method. All use one evaluation-manifest hash and have complete resource records; no checkpoint hashes mismatch. Dispatch was sequential and isolated. The v9-specific qualification `qualifications/reacher_v9_full_identity_01/mujoco_reacher_v5.json` and the evaluation manifest both record MuJoCo 3.12.0 and full MJCF SHA-256 `a10215aae27636504b1d41eb98c09ad50dbcb9f4ad888ed7593ff652053168db`. The run configuration did not retain the qualification ID/hash, so these matching records verify model identity but not which qualification file was consumed at dispatch. The root-level asset-hash-only qualification is not used for this cohort. The bootstrap rule passes, while the Student-t sensitivity bound does not. These checks validate the simulator run, not hardware performance or broad controller competence.

C READOUT NOISE AS STATE-DEPENDENT EXPLORATION

Let $\varphi_i = \langle Z_i \rangle$ and let $\hat{\varphi}_i = S^{-1} \sum_{s=1}^{S} z_{i,s}$ with $z_{i,s} \in \{-1, +1\}$ and $\Pr(z = +1) = (1 + \varphi_i)/2$. Then $\mathbb{E}[\hat{\varphi}_i] = \varphi_i$, $\mathrm{Var}[\hat{\varphi}_i] = (1 - \varphi_i^2)/S$, and $\hat{\varphi}_i = 2k_i/S - 1$ with $k_i \sim \mathrm{Binomial}(S, (1 + \varphi_i)/2)$. Across qubits the $z_{i,s}$ share a shot, so $\hat{\varphi}_i$ and $\hat{\varphi}_j$ are correlated through $\langle Z_i Z_j \rangle - \langle Z_i \rangle \langle Z_j \rangle$; the marginals hold regardless. With a linear readout the latent mean becomes $\hat{\mu} = \mu + W^\top \epsilon$ where $\epsilon = \hat{\varphi} - \varphi$ is zero-mean and bounded, so the effective exploration noise on the tanh-Gaussian latent is $W^\top \epsilon + \sigma\xi$ with covariance $W^\top \mathrm{diag}((1 - \varphi^2)/S)\, W + \sigma^2 I$ plus the cross-qubit correction. The added variance scales as $1/S$, vanishes where $|\varphi_i| \to 1$, and is largest where $\varphi_i \approx 0$. The latent linear readout is unbiased before the tanh transformation. The executed action need not be unbiased because tanh and action-bound transformations are nonlinear. Replacing φ by a classical map preserves the marginal moment statements but not the circuit's joint cross-qubit distribution. The binomial wrapper is therefore a marginally shot-noise-matched control, not an exact action-level surrogate. Let $\varphi_i = \langle Z_i \rangle$. A single-shot measurement gives $z_{i,s} \in \{-1, +1\}$, with $\Pr(z_{i,s} = +1) = (1 + \varphi_i)/2$. Averaging S shots yields $\hat{\varphi}_i = S^{-1} \sum_s z_{i,s}$. Its mean is φ_i and its variance is $(1 - \varphi_i^2)/S$; equivalently, $\hat{\varphi}_i = 2k_i/S - 1$ for $k_i \sim \mathrm{Binomial}(S, (1 + \varphi_i)/2)$.

The estimates for different qubits need not be independent. Each bitstring supplies all $z_{i,s}$ from the same shot, so

$$\mathrm{Cov}(\hat{\varphi}_i, \hat{\varphi}_j) = \frac{\langle Z_i Z_j \rangle - \langle Z_i \rangle \langle Z_j \rangle}{S}. \tag{2}$$

This covariance does not change the marginal means or variances above.

Now let $\epsilon = \hat{\varphi} - \varphi$. A linear readout changes the latent mean to $\hat{\mu} = \mu + W^\top \epsilon$. The extra noise is zero-mean and bounded. Before the $\tanh$ and action-bound transforms, the policy latent is $\mu + W^\top \epsilon + \sigma\xi$, with covariance $W^\top \Sigma_\epsilon W + \sigma^2 I$, where Σ_ϵ includes the cross-qubit terms. Shot noise falls as $1/S$, vanishes as $|\varphi_i|$ approaches one, and is largest near zero.

The linear readout is unbiased before the nonlinear action transforms; the executed action need not be. A classical binomial wrapper preserves the marginal mean and variance, but loses the circuit's joint cross-qubit distribution. It is a marginally shot-noise-matched control, not an exact action-level substitute.

D WHY STOCHASTIC FEATURE SEEDS MUST BE REPLAYED

Let z denote the random seed used to obtain a finite-shot feature vector $\varphi(x, z)$ for observation x. PPO stores an action u, its rollout log-likelihood $\ell_{\mathrm{old}} = \log \pi_{\theta_{\mathrm{old}}}(u \mid x, z)$, and the feature seed z. The clipped surrogate therefore uses

$$r(\theta) = \exp\left(\log \pi_\theta(u \mid x, z) - \ell_{\mathrm{old}}\right). \tag{3}$$

At unchanged parameters, $r(\theta_{\mathrm{old}}) = 1$ up to floating-point roundoff. If the update recomputes the feature with an independent seed z', it instead evaluates

$$\widetilde{r}(\theta_{\mathrm{old}}) = \exp\left(\log \pi_{\theta_{\mathrm{old}}}(u \mid x, z') - \log \pi_{\theta_{\mathrm{old}}}(u \mid x, z)\right), \tag{4}$$

which is not one in general. The discrepancy is present before any parameter update and can activate PPO clipping or change the policy gradient. Storing and replaying z is therefore part of the likelihood definition for this stochastic-input policy, rather than an optional variance-reduction choice.

E ADDITIONAL TABLES AND FIGURES

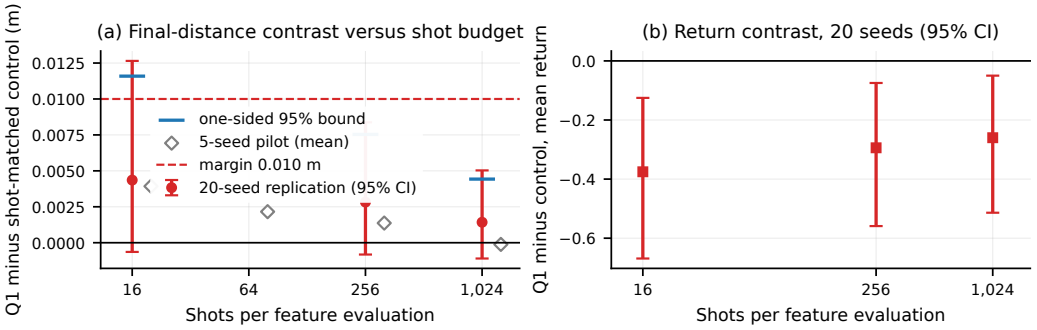

Figure 4: Shot frontier. (a) Q1 minus shot-matched classical control on final distance; filled points are the 20-seed replication, ticks its one-sided upper bounds, open diamonds the five-seed pilot means, and the dashed line the $0.01\,\mathrm{m}$ margin. (b) The same contrast on mean return; every interval excludes zero.

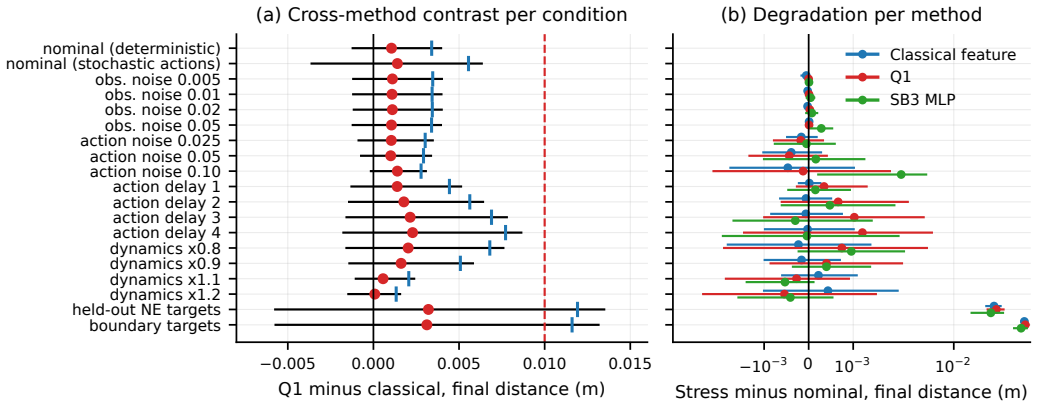

Figure 5: Evaluation-time stress tests of archived v5 checkpoints, 20 seeds per cell. (a) Paired historical Q1 minus classical final distance per condition: 95% bootstrap intervals (black), one-sided upper bounds (blue ticks), and the $0.01\,\mathrm{m}$ reference line (dashed). (b) Stress minus nominal per method on a symmetric log scale; only the target-shift conditions degrade any method after Holm correction. These evaluations do not establish corrected-Q1 training claims.

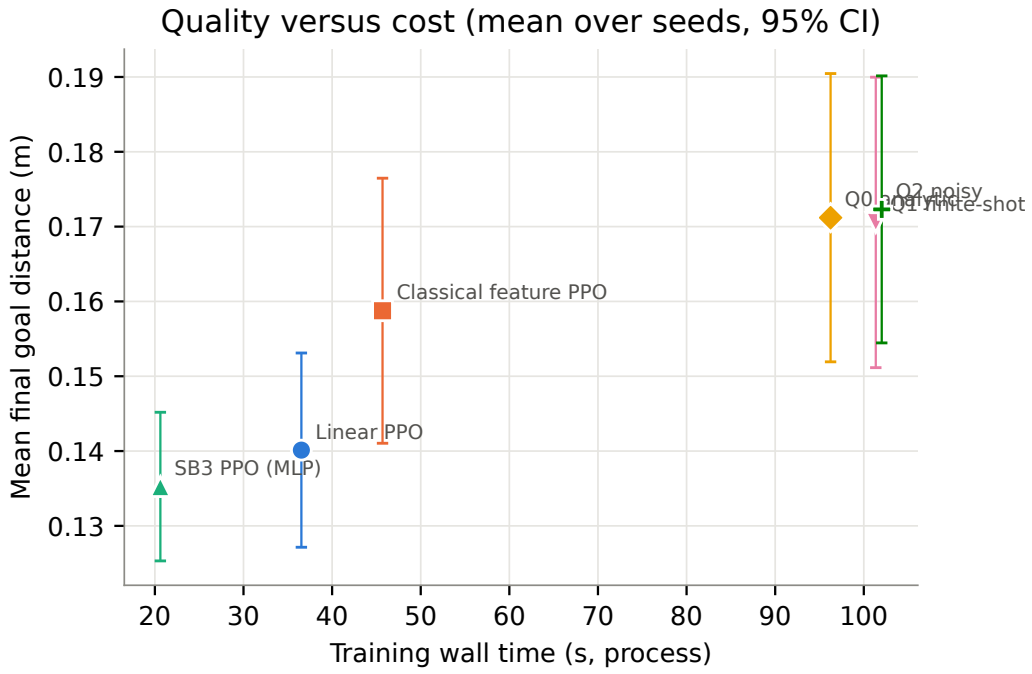

Figure 6: Historical six-method sequential Reacher audit (v1), retained for provenance. It is separate from the three-method v9 cohort in Table 1. This run lacks the full model-identity record required for current claims; its plotted costs and quality must not be substituted for v9 measurements.

Width, depth, feature-map, and noise ablations. The five-seed exploratory protocol `shots_v6` varied circuit settings and classical maps at 50,000 interactions. All 85 cells completed (Table 5; Figure 8). Four qubits, with 193 stored parameters, were $0.0145\,\mathrm{m}$ worse than two; the interval was wide ($[-0.0008, +0.0414]$), and latency more than doubled. One versus two layers differed by $0.0002\,\mathrm{m}$. Fourier and sine maps differed by $0.0003\,\mathrm{m}$, while the random-MLP map was $0.018\,\mathrm{m}$ worse, driven by one seed.

At the default noise scale, Q2 and Q0 differed by $0.00003\,\mathrm{m}$. A corrected 15-cell Q2 sweep (`shots_v7`) used 0.5, 1, and 2 times the default error rates; mean distances were 0.1229, 0.1222, and $0.1211\,\mathrm{m}$. The apparent improvement with noise was $-0.0018\,\mathrm{m}$ (CI $[-0.0043, +0.0004]$), which does not exclude zero. We retired the earlier noise profile after oracle review showed its single-qubit term had no observable effect, and excluded that pilot.

Pusher-v5. The five-seed, seven-method pilot completed all 35 cells at 50,000 interactions. Compact policies left object-to-goal distance at $0.238\,\mathrm{m}$ (between-seed SD below $0.001\,\mathrm{m}$); SB3 improved return but not distance, which was $0.240\,\mathrm{m}$. At 200,000 interactions, Q1 minus sine was $+0.0001\,\mathrm{m}$ and SB3 reached $0.232\,\mathrm{m}$ (50k-to-200k improvement CI $[-0.017, +0.002]$). Longer SB3-only runs reached $0.250\,\mathrm{m}$ at 500,000 interactions and $0.234\,\mathrm{m}$ at one million, with no derived successes (Figure 2b).

The longer SB3 runs did not establish successful pushing. We therefore did not expand quantum training to the 500k or one-million-step budgets. The earlier 50k and 200k quantum comparisons remain exploratory. Table 8 summarizes the methods at each budget.

Table 4: Swimmer-v5 matched comparison at 500,000 interactions. Displacement in metres, higher is better; the pilot rows use five seeds, the replication rows twenty. Latency is the per-seed median from the same run and is descriptive only, since these cells ran in parallel.

Study	Method	Params	Displacement (m)	Return	Latency (ms)	Train shots
Pilot (5)	SB3 PPO MLP	9,669	1.3652 ± 0.5499	33.99	0.31	0
Pilot (5)	Classical sine	109	1.0225 ± 0.0302	25.54	0.25	0
Pilot (5)	Q0 analytic	115	1.0135 ± 0.0231	25.30	0.93	0
Pilot (5)	Q1 128 shots	115	1.0137 ± 0.0233	25.31	1.03	1.92×10^8
Replication (20)	Classical sine	109	1.0199 ± 0.0280	25.48	0.21	0
Replication (20)	Q1 128 shots	115	1.0182 ± 0.0143	25.42	0.94	1.92×10^8

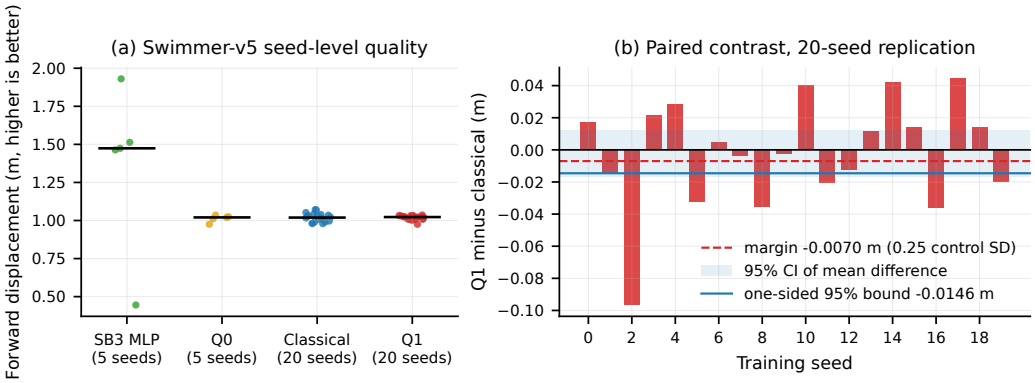

Figure 7: Swimmer-v5. (a) Seed-level forward displacement; the five-seed pilot supplies the SB3 and Q0 columns and the 20-seed replication the classical and Q1 columns. (b) Paired Q1 minus classical difference per seed with the standardized margin and the one-sided lower bound; the lower bound falls below the negative margin, so non-inferiority is not established.

Endpoint-corrected descriptive reanalysis. The protocol named `forward_distance_m`, but the original hypothesis code tested return. We exclude that result. The separate script `scripts/reanalyze_swimmer_endpoint.py` instead averages the declared displacement over 100 cases within each training seed, then bootstraps the 20 paired seeds. Q1 minus sine is $-0.0017\,\mathrm{m}$ (95% percentile interval $[-0.0167, +0.0117]\,\mathrm{m}$). The source rows and script are checksum-linked in the anonymous package. This correction remains descriptive; it does not restore confirmatory status.

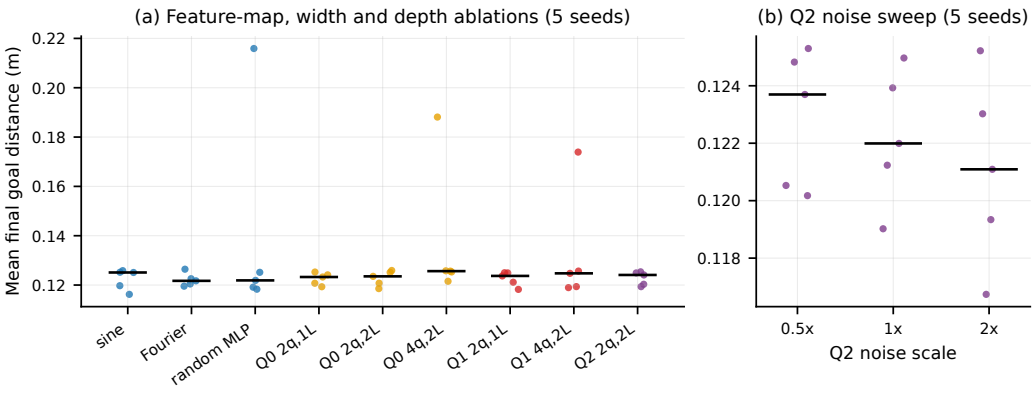

Figure 8: Exploratory five-seed ablations on Reacher-v5 (bars mark medians). (a) Classical maps, circuit width and depth, and the Q2 tier at 128 shots. (b) The corrected gate-accumulated Q2 noise profile at three intensities.

Table 5: Exploratory five-seed ablations on Reacher-v5 at 50,000 interactions (`shots_v6`) and the corrected Q2 noise sweep (`shots_v7`). Latency is the per-seed median from parallel cells and is descriptive only.

Condition	Shots	Params	Distance (m)	Return	Latency (ms)	Train shots
sine	–	109	0.1224 ± 0.0042	-9.98	0.25	0
Fourier	–	115	0.1221 ± 0.0027	-10.03	0.24	0
random MLP	–	115	0.1401 ± 0.0425	-10.36	0.20	0
sine + readout noise	16	109	0.1212 ± 0.0038	-9.83	0.37	0
sine + readout noise	64	109	0.1205 ± 0.0044	-9.86	0.36	0
sine + readout noise	256	109	0.1209 ± 0.0062	-9.91	0.33	0
sine + readout noise	1,024	109	0.1226 ± 0.0040	-9.99	0.40	0
Q0 2 qubits, 1 layer	–	111	0.1225 ± 0.0025	-9.96	0.79	0
Q0 2 qubits, 2 layers	–	115	0.1228 ± 0.0031	-9.89	1.02	0
Q0 4 qubits, 2 layers	–	193	0.1373 ± 0.0285	-10.42	2.29	0
Q1 2 qubits, 2 layers	16	115	0.1251 ± 0.0044	-10.17	0.99	2.4×10^6
Q1 2 qubits, 2 layers	64	115	0.1226 ± 0.0028	-9.93	0.88	9.6×10^6
Q1 2 qubits, 2 layers	256	115	0.1223 ± 0.0033	-9.93	0.98	3.84×10^7
Q1 2 qubits, 2 layers	1,024	115	0.1225 ± 0.0033	-9.91	0.97	1.536×10^8
Q1 2 qubits, 1 layer	128	111	0.1226 ± 0.0029	-9.90	0.77	1.92×10^7
Q1 4 qubits, 2 layers	128	193	0.1325 ± 0.0233	-10.38	2.43	1.92×10^7
Q2 2 qubits, 2 layers	128	115	0.1228 ± 0.0028	-9.89	1.01	1.92×10^7
Q2 noise $0.5\times$	128	115	0.1229 ± 0.0024	-9.92	0.83	1.92×10^7
Q2 noise $1\times$	128	115	0.1222 ± 0.0023	-10.00	0.98	1.92×10^7
Q2 noise $2\times$	128	115	0.1211 ± 0.0033	-9.87	1.08	1.92×10^7

Table 6: Selected paired contrasts from the five-seed ablations (`shots_v6`), in metres, with 95% bootstrap intervals. Positive favors the second-named arm. These are post hoc and exploratory.

Contrast	Difference	95% CI
Q1 16 shots minus control 16 shots	$+0.0039$	$[+0.0005, +0.0080]$
Q1 64 shots minus control 64 shots	$+0.0022$	$[+0.0005, +0.0041]$
Q1 256 shots minus control 256 shots	$+0.0014$	$[-0.0031, +0.0059]$
Q1 1,024 shots minus control 1,024 shots	-0.0001	$[-0.0035, +0.0026]$
Q0 4 qubits minus Q0 2 qubits	$+0.0145$	$[-0.0008, +0.0414]$
Q0 1 layer minus Q0 2 layers	-0.0002	$[-0.0028, +0.0016]$
Fourier minus sine	-0.0003	$[-0.0034, +0.0029]$
random MLP minus sine	$+0.0177$	$[-0.0020, +0.0545]$
Q0 minus sine	$+0.0004$	$[-0.0031, +0.0033]$
Q2 minus Q0	$+0.00003$	$[-0.0004, +0.0005]$
Q1 256 shots minus Q0	-0.0005	$[-0.0009, -0.0002]$
Q1 1,024 shots minus Q0	-0.0003	$[-0.0006, -0.0001]$
sine + noise 64 shots minus sine	-0.0019	$[-0.0033, -0.0006]$
sine + noise 256 shots minus sine	-0.0015	$[-0.0036, -0.0002]$
Q2 noise $2\times$ minus $0.5\times$	-0.0018	$[-0.0043, +0.0004]$

Table 7: Graded robustness suite (960 cells, 20 seeds per cell). Mean final distance per method and the paired Q1 minus classical contrast with its one-sided 95% upper bound; no contrast is significant after Holm correction. The last two rows come from the nine-condition suite that included target shifts.

Condition	Classical	Q1	SB3	Q1 − cl.	95% CI	Upper
nominal	0.1223	0.1234	0.1171	$+0.0011$	$[-0.0012, +0.0040]$	0.0034
observation noise 0.005	0.1223	0.1234	0.1171	$+0.0011$	$[-0.0012, +0.0040]$	0.0035
observation noise 0.01	0.1223	0.1234	0.1172	$+0.0011$	$[-0.0012, +0.0040]$	0.0034
observation noise 0.02	0.1223	0.1234	0.1172	$+0.0011$	$[-0.0011, +0.0040]$	0.0034
observation noise 0.05	0.1223	0.1234	0.1174	$+0.0011$	$[-0.0012, +0.0039]$	0.0034
action noise 0.025	0.1222	0.1232	0.1171	$+0.0010$	$[-0.0009, +0.0035]$	0.0030
action noise 0.05	0.1219	0.1229	0.1173	$+0.0010$	$[-0.0007, +0.0034]$	0.0029
action noise 0.10	0.1219	0.1233	0.1193	$+0.0014$	$[-0.0001, +0.0031]$	0.0028
action delay 1	0.1223	0.1237	0.1173	$+0.0014$	$[-0.0013, +0.0051]$	0.0044
action delay 2	0.1223	0.1240	0.1176	$+0.0018$	$[-0.0014, +0.0064]$	0.0056
action delay 3	0.1223	0.1244	0.1168	$+0.0021$	$[-0.0016, +0.0078]$	0.0069
action delay 4	0.1223	0.1246	0.1171	$+0.0023$	$[-0.0018, +0.0086]$	0.0077
dynamics $\times 0.8$	0.1221	0.1241	0.1181	$+0.0020$	$[-0.0016, +0.0076]$	0.0068
dynamics $\times 0.9$	0.1222	0.1238	0.1175	$+0.0016$	$[-0.0014, +0.0058]$	0.0051
dynamics $\times 1.1$	0.1225	0.1231	0.1166	$+0.0006$	$[-0.0010, +0.0024]$	0.0021
dynamics $\times 1.2$	0.1228	0.1228	0.1167	$+0.0001$	$[-0.0015, +0.0016]$	0.0013
held-out NE targets	0.1541	0.1573	0.1461	$+0.0032$	$[-0.0057, +0.0135]$	0.0119
boundary targets	0.1982	0.2014	0.1863	$+0.0031$	$[-0.0057, +0.0131]$	0.0116

Table 8: Pusher-v5 exploratory results, five seeds each, object-to-goal distance in metres. No cell recorded a derived success at any budget.

Budget	Method	Params	Distance (m)	Return	Train shots
50k	SB3 PPO MLP	11,919	0.2401 ± 0.0082	-40.05	0
50k	sine	129	0.2380 ± 0.0007	-48.11	0
50k	Fourier	135	0.2382 ± 0.0005	-48.73	0
50k	random MLP	135	0.2380 ± 0.0006	-47.71	0
50k	Q0	135	0.2383 ± 0.0004	-50.31	0
50k	Q1 128 shots	135	0.2383 ± 0.0005	-50.34	1.92×10^7
50k	Q2 128 shots	135	0.2383 ± 0.0005	-50.10	1.92×10^7
200k	SB3 PPO MLP	11,919	0.2323 ± 0.0175	-35.89	0
200k	sine	129	0.2379 ± 0.0003	-46.61	0
200k	Q1 128 shots	135	0.2380 ± 0.0009	-46.75	7.68×10^7
500k	SB3 PPO MLP	11,919	0.2503 ± 0.0111	-36.61	0
1M	SB3 PPO MLP	11,919	0.2341 ± 0.0175	-37.96	0

Table 9: Resource columns of the primary run that the release audit excludes from evidence because the seeds ran as parallel cloud workers. Means over 20 seeds. Peak RSS is process-level. Reported for completeness only.

Method	Train wall (s)	Train CPU (s)	Peak RSS (MB)	Eval wall (s)	Checkpoint (kB)
Classical feature	95.3	72.1	3,324	2.2	14.9
SB3 PPO MLP	42.7	41.5	3,515	2.7	1,216
Q0	195.3	165.5	3,326	5.7	15.1
Q1	205.0	174.9	3,325	5.8	15.1
Q2	205.4	175.9	3,326	5.9	15.1

F HYPERPARAMETERS AND PROTOCOL CONSTANTS

Table 10: v9 settings and protocol-specific departures; older cohorts are described by their frozen protocol files.

Setting	Value
Feature size (padded observation)	32
Qubits / layers / feature dimension	2 / 2 / 2
Encoder initialization	$\mathcal{N}(0, 0.1^2)$, bias 0; trained in v8/v9, frozen in earlier protocols
Feature-map parameter initialization	$\mathcal{N}(0, 0.05^2)$, frozen
Readout initialization	$\mathcal{N}(0, 0.1^2)$, bias 0
Initial $\log \sigma$	-0.5, clipped to $[-5, 2]$
Learning rate / epochs per episode	3×10^{-3} / 2
Encoder SPSA learning rate / perturbation	0.01 / 0.05; common random numbers
Clip ratio / γ / λ	0.2 / 0.99 / 0.95
Entropy / value coefficient	0.01 / 0.5
Step norm cap / param. clip / value-error clip	10 / 100 / 100
Interactions (Reacher v9 / Pusher / Swimmer)	50k / protocol-specific / 500k
Training and evaluation shots (Q1, Q2)	128 / 128 (v5); frontier 16, 256, 1,024
Historical recorded-field Reacher audit	25k interactions; 128 / 1,024 shots
Q2 error rates (p_1, p_2, r)	10^{-3}, 10^{-2}, 2×10^{-2}; sweep $0.5\times$ to $2\times$
Evaluation cohort	v9: 100 distinct reset seeds, 1 episode each; historical cohorts vary
Inference deadline	20 ms (Reacher, Pusher), 40 ms (Swimmer)
Non-inferiority margin	0.01 m (Reacher); 0.25 control SD (frontier, Swimmer)
Bootstrap	percentile, seed 0, 10,000 resamples
Maximum failed seeds per method	1 (confirmatory), 0 (pilots)
SB3 PPO policy	MlpPolicy, two 64-unit tanh layers, separate value network
SB3 PPO optimizer	n_{steps} 256, batch 64, 10 epochs, lr 3×10^{-4}, 1 thread

G LITERATURE-TO-CLAIM MAP

The references below provide background for specific parts of the study. They are not evidence that this circuit works on a physical robot.

Quantum reinforcement learning. Early proposals and later variational policy or value models show that quantum reinforcement learning predates this study (Dong et al., 2008; Chen et al., 2020; Lockwood & Si, 2020; Jerbi et al., 2021; Skolik et al., 2022). Robotics examples include simulated wheeled-robot navigation, a simulated virtual arm, and continuous-action LiDAR maze navigation (Hohenfeld et al., 2024; Acuto et al., 2022; Drăgan et al., 2025). These are prior demonstrations of quantum RL in simulated robotics, not comparisons on our Reacher-v5 setup. Surveys of quantum machine learning and variational algorithms help distinguish a proposed method from a demonstrated hardware advantage (Dunjko & Briegel, 2018; Meyer et al., 2022; Cerezo et al., 2021; Preskill, 2018).

Feature maps and classical controls. Work on data encoding and quantum feature spaces informs our two-dimensional expectation-value map (Mitarai et al., 2018; Pérez-Salinas et al., 2020; Schuld et al., 2021; Havlíček et al., 2019; Schuld & Killoran, 2019). We use the random-feature literature to motivate the Fourier and sine controls (Rahimi & Recht, 2007; Landman et al., 2023). Parameter count alone is not enough to compare these representations. Prior analyses call for reporting the encoding, classical substitute, and cost of evaluation together (Huang et al., 2021; Schuld & Killoran, 2022; Bowles et al., 2024).

Trainability and expressivity. Circuit behavior can change with depth, initialization, and the measured output (Sim et al., 2019; McClean et al., 2018; Pineau et al., 2021). We keep all eight circuit parameters fixed. Training still updates the shared observation encoder, action readout, value head,

and log standard deviation. Both compact policies use the same simultaneous perturbation stochastic approximation (SPSA) update for the encoder, with common random numbers. The experiment tests a fixed quantum feature map inside a trainable policy, not a fully trainable quantum controller.

RL evaluation and tuning. PPO and generalized advantage estimation define the learning setup (Schulman et al., 2017; 2016); continuous-control benchmarks and MuJoCo/Gymnasium papers establish the simulator context (Duan et al., 2016; Todorov et al., 2012; Towers et al., 2024). Seed sensitivity, evaluation variance, and implementation choices have all affected published RL results (Henderson et al., 2018; Colas et al., 2018; Jordan et al., 2020; Agarwal et al., 2021; Andrychowicz et al., 2021). That is why we retain a fixed evaluation cohort and report seed-level intervals (Cobbe et al., 2019; Bouthillier et al., 2019). Robot-learning work also points to simulator versions and assets as part of reproducibility (Lynnerup et al., 2020; Fan et al., 2018). We separate tuning from final evaluation, following Eimer et al. (2023).

What differs from Meyer et al. Meyer et al. (Meyer et al., 2025) define an empirical sample-complexity estimator and a statistical outperformance criterion for QRL, then study those questions in BeamManagement6G. They report performance averaged over 100 training runs and provide simulator runtime measurements. This paper uses a different task and endpoint: paired final distance on Reacher-v5, with 20 training seeds and a shared set of 100 evaluation cases. We add a PPO likelihood-replay audit, a full MuJoCo model hash check, and control-loop latency and shot/circuit counts. This is a case study that complements their statistical framework; it is not a new QRL benchmark or a replacement for their sample-complexity analysis.

Statistics and decision rules. We distinguish non-inferiority from equivalence testing (Schuirmann, 1987; Wellek, 2010; Lakens, 2017), and report a one-sided interval for the predeclared endpoint (Piaggio et al., 2012). Secondary contrasts use Holm's step-down correction (Holm, 1979); seed-level uncertainty uses bootstrap intervals (Efron & Tibshirani, 1993). None of these references validates our $0.01\,\mathrm{m}$ margin. We chose that threshold for this study; no external engineering standard supports it.

Noise and resource accounting. Parameter- and action-space noise provide context for our shot-noise control (Plappert et al., 2018; Fortunato et al., 2018). Work on shot-frugal and stochastic optimization motivates recording circuit calls, gradients, and shots rather than treating each circuit execution as free (Sweke et al., 2020; Kübler et al., 2020; Spall, 1992). Q2 is only a stated synthetic stress model. We do not interpret it as a hardware result. Parallel on-policy collection can change optimization and timing (Mayor et al., 2025); accordingly, concurrent worker wall time is not used for sequential latency claims.

Reading the map. The citations establish prior methods, known evaluation problems, and the statistical tools used here. They do not show that this circuit is useful on a physical robot. In the main text, each result points to a versioned artifact and carries a status: confirmatory, exploratory, superseded, or failed.

Related settings and study design. Some cited quantum-advantage work concerns supervised learning (Liu et al., 2021; Schuld, 2021) or generative models (Hibat-Allah et al., 2024; Gili et al., 2023). Those settings help locate the broader question, but their results cannot establish an advantage for our sequential-control task. Continuous-control reproducibility is addressed directly by Islam et al. (2017). We also distinguish decisions made before evaluation from explanations developed after seeing outcomes, as motivated by preregistration (Nosek et al., 2018). A frozen rule makes a decision traceable; it does not establish that a chosen margin or controller is practically useful.

H Scope of the contribution

For this fixed circuit and policy family, replay is a correctness requirement: PPO must reuse the finite-shot feature that produced the rollout likelihood. Our resource logs give a separate warning. A small parameter count can hide repeated circuit calls, shots, latency, and memory. The v9 result adds one more limit: it passes the predeclared bootstrap rule but fails the Student-t sensitivity check, while the artifact record keeps failed and superseded runs visible.

We did not test a physical robot or QPU, and these results say nothing about whether a two-qubit circuit can compete with larger classical models on other tasks. The circuit itself is fixed; learning its parameters would require a separate gradient estimator and study budget. What we test is narrower: one fixed representation inside a policy with a shared, trainable encoder and action head.

Table 11: Comparison roles and matching conditions. SB3 is an external quality reference, not a parameter-matched arm.

Property	Compact feature-map arms	SB3 reference
Raw state input	Reacher state, 10 dimensions	Same
Action space and bounds	Two-dimensional continuous action	Same task action space
Environment interactions	50,000 per training seed	50,000 per training seed
Trained parameters	107 shared values: encoder (66), action readout (6), value head (33), and log standard deviation (2)	SB3 MLP policy/value networks
Feature representation	Fixed linear, sine, Fourier, random-MLP, or quantum map	Learned two-layer MLP
Stored policy parameters	109 for sine; 115 for Fourier, random-MLP, and quantum maps; fixed map weights are stored but not trained	9,925 trained values
PPO implementation	Project implementation with replayed feature seeds	Stable-Baselines3 PPO
Total compute	Not matched; wall time, CPU time, shots, and calls are measured	Not matched; wall/CPU time and memory are measured
Role in inference	Primary compact comparison	External control-quality reference

I CLAIM LEDGER

This ledger distinguishes what the experiment supports from results that are secondary or not established. "Supported" means the implementation and released summaries answer the stated question. "Secondary" marks evidence outside the frozen primary test; "Not established" means a required experiment or validity condition is missing.

Table 12: Claim ledger for the released evidence.

Claim	Evidence	Allowed interpretation
Simulator semantics	Independent tensor oracle and Qiskit Aer comparison	Supported for the stated CPU simulator and circuit family; not evidence about a QPU.
Resource accounting	Append-only event ledgers, conservation checks, and fresh-process resume	Supported for recorded runs; parallel scheduling is excluded from latency claims.
Reacher Q1 non-inferiority	v9: 20 paired training seeds, 100 shared evaluation cases, full MJCF identity; Q1–classical bootstrap upper bound $+0.00946\,\mathrm{m}$ against margin $+0.010\,\mathrm{m}$; Student-t sensitivity upper bound $+0.01074\,\mathrm{m}$	Protocol rule passes narrowly; conclusion is sensitive to interval method, and no quality superiority is established.
Shot frontier	Twenty paired seeds at 16, 256, and 1,024 shots with marginally noise-matched controls	Secondary descriptive evidence; Q1 does not show a quality benefit over the control.
Archived stress suite	Evaluation-only perturbations of historical v5 checkpoints	Secondary evaluation evidence; it does not validate corrected-Q1 training.
Swimmer endpoint reanalysis	The 20-seed protocol declared forward displacement; a hash-linked script recomputes that endpoint from all paired seed rows and applies its standardized margin	Corrected negative secondary result; Q1 non-inferiority is not established. The original analysis-code mismatch is disclosed.
Compact-controller competence	v10–v13 fixed-seed exploratory gates; v13 compares full-state SB3 with a learned 2-D bottleneck	The tested compact controllers fail the task-quality gates. No quantum run was admitted; this does not isolate quantum features.
Quantum advantage or hardware performance	No physical QPU execution and no corrected primary gate	Not measured and not claimed.

Confirmatory eligibility predicate. We treat a result as eligible for a primary claim only when every gate is true. For a method–cohort analysis, define

$$\mathcal{G} = I_{\mathrm{env}} \wedge I_{\mathrm{manifest}} \wedge I_{\mathrm{replay}} \wedge I_{\mathrm{resource}} \wedge I_{\mathrm{screen}} \wedge I_{\mathrm{statistical}}, \tag{5}$$

where the six indicators require, respectively, a complete simulator/model fingerprint, identical evaluation cases, valid stochastic-input replay, conserved resource events, a passed pre-run controller screen, and a passed predeclared statistical endpoint. The v8 screen was relative: it tested a 0.01 m average distance tolerance against SB3, not an absolute success target. The later v10 screen added success thresholds and failed them. We therefore use the screen indicator to describe v9's formal protocol eligibility, not practical competence. The superseded v1 Modal audit lacks the full-model hash and failed the Q1 non-inferiority criterion, so it has $I_{\mathrm{env}} = 0$ and $I_{\mathrm{statistical}} = 0$ under this definition. The v9 cohort records the full model hash, uses the shared manifest and replay checks, conserves its resource ledgers, follows the passed v8 relative screen, and passes the predeclared bootstrap rule; thus v9 meets its frozen protocol's formal statistical gate. The Student-t sensitivity bound misses the margin. The low success rates, and v10's later failure of stricter success thresholds, leave the practical-control question unresolved. This predicate records the frozen protocol result; it does not certify that the policy is a capable robot controller.

J Independent seed-level reanalysis

We provide a second analysis path for the shot-frontier result. The script `scripts/reproduce_shot_frontier.py` reads only the released seed-level CSV, checks that the two conditions have the same unique seeds at each shot count, and recomputes paired differences and percentile intervals. It does not read the original analysis JSON or any checkpoint. With 10,000 resamples and analysis seed 0, it reproduces the reported point estimates and the one-sided upper bounds to the displayed precision (Table 13). The resulting JSON is stored beside

the input CSV as `independent_reanalysis.json`; its presence is checked by the release manifest. This is a consistency check on the reported secondary result, not a new hypothesis test.

Table 13: Independent reanalysis of the shot-frontier CSV. Differences are Q1 minus the shot-matched classical control; positive distance is worse for Q1. Intervals are percentile bootstrap intervals over the 20 paired training seeds.

Shots	Seeds	Distance difference	95% CI	One-sided upper	Return difference
16	20	$+0.0044$	$[-0.0006, +0.0126]$	$+0.0116$	-0.3752
256	20	$+0.0028$	$[-0.0008, +0.0084]$	$+0.0075$	-0.2941
1,024	20	$+0.0014$	$[-0.0011, +0.0050]$	$+0.0044$	-0.2599

K METRIC DEFINITIONS AND RESOURCE CONSERVATION

The endpoint is aggregated in two stages. For training seed s and evaluation case e, let $d_{s,e}$ be the final fingertip-to-target distance in metres. The seed-level endpoint is $D_s = |\mathcal{E}|^{-1} \sum_{e \in \mathcal{E}} d_{s,e}$, where $|\mathcal{E}| = 100$. Success is the derived indicator $\mathbb{1}[d_{s,e} < 0.05\,\mathrm{m}]$; it is reported separately from the native environment reward. Return is summed over the episode. Control effort is $|\mathcal{E}|^{-1} \sum_t \|u_t\|_2^2$, action variation is the mean $\|u_t - u_{t-1}\|_1$, and saturation is the fraction of action coordinates within 10^{-6} of an action bound. We compute these metrics for each episode before averaging within seed. A truncation emits the same terminal metric record as a termination, with its cause retained in the artifact.

For a policy call, latency is measured from the observation entering the policy to the action leaving it. We report p50, p95, and p99 over calls, plus the fraction above the 20 ms Reacher deadline. Environment stepping, preprocessing, classical policy time, quantum execution time, gradient time, optimizer time, and serialization time are recorded as separate spans. Training and evaluation totals are never combined in the primary tables.

Let e index events in a resource ledger. Additive quantities use conservation identities:

$$C_{\mathrm{circuits}} = \sum_e c_e, \qquad S_{\mathrm{shots}} = \sum_e s_e, \qquad G_{\mathrm{grad}} = \sum_e g_e, \qquad T_{\mathrm{compile}} = \sum_e t_e. \tag{6}$$

Peak RSS and maximum latency are not additive; the release audit checks them as maxima over their raw events. Every event carries its phase, method, seed, worker status, and checkpoint ID. A killed worker may leave an incomplete ledger, but the finalizer cannot mark that cell complete until the ledger hash, checkpoint hash, and expected event totals agree. This is why a timeout remains a failure record instead of disappearing from the denominator.

L EXECUTION MATRIX AND EVIDENCE STATUS

Table 14 lists the studies that appear in the paper. The status is assigned by the protocol, artifact, and analysis gates, not by the numerical result. In particular, a completed run can remain unqualified. Appendix X extends this summary to all top-level run and analysis directories in the Modal volume inspected on 26 September 2026, including preparation runs and interrupted attempts.

Table 14: Study inventory. Cell counts are method–seed cells; the evaluation count is per cell when applicable.

Protocol	Task	Cells	Endpoint and cohort	Evidence status
`reacher_v7`	Reacher-v5	6	50k interactions; 100-case controller screen	competence gate failed
`reacher_v8`	Reacher-v5	6	50k interactions; 3 seeds per method; 100 shared cases	relative distance screen passed; no absolute success requirement
`reacher_v9`	Reacher-v5	60	3 methods, 20 seeds each; 50k interactions; 100 shared cases	qualified primary cohort; bootstrap NI passes, Student-t sensitivity fails
`reacher_v1` (laptop, legacy agents)	Reacher-v5	120	25k interactions; 20 seeds; 100 paired cases	superseded; gradient defect
`reacher_v1` (Modal, corrected agents)	Reacher-v5	120	25k interactions; 20 seeds; 100 paired cases	historical resource audit; identity incomplete; NI failed
`shots_v8`	Reacher-v5	120	20 seeds at 16, 256, 1,024 shots	valid secondary replication
`reacher_v5_robustness_v1`	Reacher-v5	540	archived checkpoints; 9 conditions	evaluation-only secondary
Graded robustness v2	Reacher-v5	960	archived checkpoints; 16 stress conditions	evaluation-only secondary
`swimmer_v3`	Swimmer-v5	40	20 seeds; 100 cases per cell; displacement endpoint reanalyzed	negative corrected secondary result; margin not met
Pusher gates	Pusher-v5	35+	50k to 1M interactions; SB3 qualification included	negative baseline evidence

The inventory separates the failed v7 screen from the relative v8 screen that gated v9. V8 used three seeds and no minimum success rate; its pass is not evidence that the compact policy solved Reacher. V10 later added explicit success thresholds and failed them. The Swimmer protocol specified displacement before training; the first analysis code tested return by mistake. The independent reanalysis uses the originally specified endpoint, keeps its failed margin, and reports the analysis correction rather than hiding it. No exploratory or superseded result is pooled into the v9 primary contrast.

M COMPUTE

Resource-vector interpretation. For a method m, let $\rho(m) = (P, Q, D, S, C, T, M, F)$ denote stored parameters, qubits, circuit depth, shots, circuit executions, latency, peak memory, and failure indicators. We call a method resource-dominated only when another method is no worse on the reported quality and resource coordinates and strictly better on one. We do not collapse these coordinates into a scalar cost without declared workload-specific weights.

Reacher v9 ran sequentially on 4-core, 16 GiB Linux CPU workers; each method-seed resource record reports wall time, CPU time, and process peak RSS. The cohort contains 60 cells and used a shared evaluation manifest. Swimmer v3 ran on 2-core, 8 GiB workers and has its own immutable evaluation schedule. These are provider CPU-simulator measurements. The archived v1 and v5 resource figures remain historical and are not used for v9 timing claims.

N EXPLORATORY 200K CONTROLLER-COMPETENCE SCREEN

The frozen protocol `reacher_v10_controller_competence_200k` compared the compact classical feature policy with SB3 PPO at 200,000 interactions for seeds 71–73. Each checkpoint was evaluated on the same 100 distinct reset seeds (manifest hash `0105aae36f3f147bf64c5e97f4106adc7fd984925e92c4e7902316480f7445b0`). All six cells completed without evaluation failures. The compact policy's return improved on every seed and its largest action-saturation fraction was 0.01, but the final-distance and success criteria failed. The mean compact distance was $0.13117\,\mathrm{m}$ versus $0.08123\,\mathrm{m}$ for SB3; the mean paired gap was $0.04994\,\mathrm{m}$, compared with the $0.010\,\mathrm{m}$ gate tolerance. Mean derived success was 8.67% for the compact policy and 34.33% for SB3. Compact seed success rates were 10%, 9%, and 7%; none exceeded the predeclared per-seed threshold of 10% except the first, which only met it. The gate also required pooled success of at least 25%.

Table 15: Seed-level results from the exploratory v10 competence screen. Each entry averages 100 evaluation cases; this three-seed screen is a gate, not a confirmatory comparison.

Seed	Method	Final distance (m)	Success rate	Late–early return
71	Compact PPO	0.13007	0.10	+15.15
71	SB3 PPO	0.08826	0.22	+25.37
72	Compact PPO	0.13133	0.09	+14.77
72	SB3 PPO	0.06575	0.57	+26.22
73	Compact PPO	0.13211	0.07	+15.28
73	SB3 PPO	0.08967	0.24	+26.82

The protocol-level decision is `fail`: the compact policy passes the improvement and saturation criteria but fails the pooled success, per-seed success, and distance-to-SB3 criteria. No quantum method was run after this gate failed. The report, resolved run configuration, and immutable evaluation manifest are included under `paper_artifacts/modal_2026-09-23/`; the screen is exploratory and does not alter the v9 estimate or support a 200k quantum claim.

Update diagnostics. We partitioned each saved update log into four contiguous, equal-sized groups and averaged the logged metrics within each group. The compact policy has a linear value head over the padded observation; its explained variance is negative throughout and declines over training. SB3's explained variance rises above 0.95 after the first quartile. Compact-policy returns nevertheless improve in all three seeds (+15.15, +14.77, and +15.28 between the early and late episode quartiles), without action saturation. The signs therefore do not support a simple policy-collapse story. They do motivate a controlled follow-up with a nonlinear value function, while holding the actor, PPO schedule, interaction budget, and evaluation cohort fixed. This diagnosis is exploratory: update counts differ between methods, and explained variance alone cannot identify the cause of the performance gap.

Table 16: Mean update metrics by contiguous training quartile in the v10 screen, averaged over three seeds. These are descriptive diagnostics; the update algorithms and numbers of updates differ between methods.

Method	Quartile	Explained variance	Value loss	Approx. KL	Clip fraction
Compact PPO	1	−0.305	27.953	0.0002	0.0001
Compact PPO	2	−0.476	6.987	0.0005	0.0008
Compact PPO	3	−0.653	1.899	0.0009	0.0020
Compact PPO	4	−0.880	0.764	0.0023	0.0108
SB3 PPO	1	0.309	35.329	0.0092	0.0820
SB3 PPO	2	0.957	0.126	0.0142	0.1520
SB3 PPO	3	0.984	0.047	0.0196	0.1984
SB3 PPO	4	0.989	0.035	0.0288	0.2457

The aggregation is regenerated by `scripts/analyze_reacher_v10_updates.py`. The six source summaries now sit in the adjacent `source_summaries/` directory. The machine-readable diagnostic records each file's SHA-256 hash.

O EXPLORATORY CRITIC-CAPACITY FOLLOW-UP

To test the value-fit hypothesis more directly, v11 kept the compact feature policy and training schedule fixed while replacing its linear value head with a 32-unit tanh MLP. We ran both compact variants and SB3 for 200,000 interactions on seeds 81–83. All three used the same 100-case evaluation manifest (SHA-256 `12095d27cd5eb98a0e03fd8de917cfc8ba4f0f3c344fde51f0f45f20e485fc86`); all nine method-seed evaluations contain 100 records and no failures. The protocol was exploratory and explicitly disallowed confirmatory claims.

Table 17: Exploratory v11 mean final distance and derived success by training seed. The linear and MLP-critic variants share the compact actor and are the controlled critic-capacity contrast. SB3 is a competence reference, not an otherwise matched ablation.

Seed	Method	Final distance (m)	Success rate
81	Compact, linear critic	0.14353	0.04
81	Compact, MLP critic	0.14434	0.02
81	SB3 PPO	0.10370	0.19
82	Compact, linear critic	0.14676	0.04
82	Compact, MLP critic	0.14177	0.04
82	SB3 PPO	0.08456	0.35
83	Compact, linear critic	0.14534	0.04
83	Compact, MLP critic	0.14387	0.02
83	SB3 PPO	0.12890	0.03

Across the three paired seeds, the MLP critic changed mean final distance by $-0.00189\,\mathrm{m}$ relative to the linear critic. Its derived success was 2.7%, versus 4.0% for the linear critic. These three seeds are too few for a reliable treatment-effect estimate, but the observed change is small and does not recover the SB3 gap. The predeclared competence gate compared the MLP-critic policy with SB3 and failed: the mean paired distance excess was $0.03761\,\mathrm{m}$ (tolerance $0.010\,\mathrm{m}$), pooled success was 2.7% (threshold 25%), and no seed reached the 10% per-seed success threshold. Improvement from early to late training was positive on all three seeds, and maximum action saturation was 0.01, so this was not a saturation collapse.

The follow-up weakens the claim that critic capacity alone explains the compact controller's poor task performance, at least for this setup. It does not establish that critic capacity is irrelevant: the comparison has only three seeds. No quantum arm was run after the competence gate failed. The v11 run records no usable source commit, and its architecture metadata labels the trainable encoder update as analytic even though the implementation uses SPSA. We therefore use v11 only for the task-quality diagnostic above, not for gradient-cost or code-version claims. Raw evaluations, checkpoints, and the gate report are retained under `paper_artifacts/modal_2026-09-25/reacher_v11_value_critic_200k_01/`.

P SB3 LOW-DIMENSIONAL BOTTLENECK COMPETENCE GATE

The next test isolated whether the poor compact-policy results could be explained by the project PPO implementation. We used Stable-Baselines3 PPO in both arms, with the same 10-dimensional Reacher state, two-dimensional action, training schedule, 200,000 interaction budget, and 100 fixed evaluation cases per seed. The reference used the full state. The candidate applied a fixed two-dimensional sine bottleneck before the otherwise standard policy network. This was a three-seed exploratory competence screen, not a quantum comparison and not a confirmatory test. Its frozen thresholds required at least 25% pooled derived success, at least 10% in every seed, no more than 5% action saturation, and no more than 0.01 m mean distance excess over the reference. The absolute saturation

ceiling was not calibrated to the full-state arm; we report its nominal failure but do not interpret it as a collapse signal.

Table 18: v12 results by training seed. Success is the project's derived Reacher metric; evaluation had 100 fixed cases in each cell and no failures.

Seed	Method	Final distance (m)	Success	Max saturation
91	Full-state SB3	0.06695	36%	0.387
91	Sine bottleneck	0.13432	11%	0.361
92	Full-state SB3	0.09025	19%	0.350
92	Sine bottleneck	0.14829	4%	0.357
93	Full-state SB3	0.08613	30%	0.373
93	Sine bottleneck	0.13506	9%	0.369

Across seeds, the bottleneck's mean final distance exceeded the reference by 0.05811 m; pooled success was 8.0%, versus 28.3% for full-state SB3. The candidate improved late-versus-early return in all three seeds, but that did not translate into competent endpoint performance. It failed the distance, pooled-success, and per-seed-success checks while passing the learning-trend check. The frozen protocol also flagged its 0.369 maximum action saturation against an absolute 0.05 ceiling. That ceiling was not calibrated to Reacher: the full-state reference reached 0.387, 0.350, and 0.373 on the same seeds. The candidate's paired saturation excesses were -0.026, +0.007, and -0.004, so saturation was not evidence of collapse. We correct that interpretation here; the task-quality gate still fails. This indicates that the tested fixed bottleneck is a poor representation under this setup; it does not establish that quantum features are intrinsically ineffective.

The run used seeds 91–93 and the shared evaluation-manifest hash `4b9a93b1d8f88cea1d158db3d7b9f1a5beaa9e76bf321561681a113781f87b1b`. It completed all six method-seed cells and 600 evaluations without evaluation failures. The local 48-file result bundle is 129,520,611 bytes; its per-file checksums are recorded in the run's raw-artifact manifest. This exploratory bundle is distinct from the released v9 primary-study archive, and its source commit identifier is omitted here to preserve double-blind review.

Q Learned bottleneck competence screen

After the fixed sine projection failed, we tested whether allowing the compact representation to learn would close the gap. This exploratory screen compared full-state SB3 PPO with the same PPO configuration using a trainable 10-to-2-dimensional bottleneck. Both arms received 200,000 interactions for each of three training seeds. Every checkpoint was evaluated on the same 100 fixed reset cases; this was a controller-development gate, not a quantum comparison or confirmatory study.

Table 19: v13 learned-bottleneck screen. Success is a derived Reacher metric; each row summarizes 100 fixed evaluation cases with no failures.

Seed	Method	Final distance (m)	Success	Max saturation
101	Full-state SB3	0.10694	12%	0.373
101	Learned bottleneck	0.12088	15%	0.357
102	Full-state SB3	0.11465	13%	0.400
102	Learned bottleneck	0.12708	4%	0.344
103	Full-state SB3	0.11608	16%	0.404
103	Learned bottleneck	0.12805	6%	0.369

The learned representation's mean distance was 0.12534 m, compared with 0.11256 m for full-state SB3. Its paired excess was +0.01278 m, above the protocol's 0.01 m tolerance; success was 8.3% versus 13.7%. Late-minus-early return was positive for all three candidate seeds (mean +22.44), and the candidate did not exceed the reference's action saturation by the allowed 0.05. Those checks do not offset the failed task-quality thresholds: pooled success had to reach 25%, and every seed had to reach 10%. The frozen gate therefore failed and no quantum arm was run. With three seeds, this is a

development result about the tested bottleneck and budget, not a population estimate or a claim about quantum policies.

The shared evaluation manifest and its checksum are recorded with the raw bundle. It contains all six training records and 600 evaluation records. The first dispatch, which failed because one method was routed to the wrong runner before candidate training, is separately preserved and excluded. The v13 report in the accompanying documentation gives the protocol, resource, provenance, and verification details.

R FRESH-SEED FEATURE-CONTROLLER COMPETENCE GATE

After the learned bottleneck also missed its task-quality thresholds, we retested the compact classical feature controller on new training seeds. The frozen v14 protocol used seeds 111–113, 200,000 environment interactions per method and seed, and one evaluation episode from each of 100 distinct reset seeds. The full-state SB3 reference and candidate shared the same immutable evaluation manifest (SHA-256 `b59e9be4dda1e04c5157cd2aaadab207bb1273e3f60971d1e27d23e36878e7ac`). All six cells and 600 evaluations completed without failures. The evaluation manifest and cell records identify Gymnasium Reacher-v5, MuJoCo 3.12.0, and the same full MJCF hash as v9 (`a10215...3168db`).

Table 20: Fresh-seed v14 competence gate. The feature controller is classical; no quantum method was run. Each entry summarizes 100 fixed evaluation cases.

Seed	Classical feature distance / success	Full-state SB3 distance / success	Paired distance gap (m)
111	0.13471 / 6%	0.07422 / 39%	+0.06049
112	0.13603 / 4%	0.10876 / 18%	+0.02727
113	0.13585 / 4%	0.09373 / 27%	+0.04213
Mean	0.13553 / 4.7%	0.09223 / 28.0%	+0.04330

The frozen gate required a mean candidate-to-reference distance gap no greater than $0.01\,\mathrm{m}$, pooled candidate success of at least 25%, and at least 10% success on every seed. The candidate missed all three: its mean gap was $0.04330\,\mathrm{m}$, pooled success 4.7%, and per-seed success 4%–6%. It did pass the learning-trend and saturation checks: late-minus-early return was positive on all seeds (mean +14.15), and maximum action saturation was 0.01. All cells shared the same complete evaluation manifest. Because the candidate failed the endpoint-quality criteria, the protocol did not admit a quantum comparison. This is a three-seed controller-development screen, not a confirmatory estimate and not a conclusion about quantum policies. The raw cell artifacts and independently recomputed gate decision are retained with the experiment records; the source snapshot is recorded, while the Modal cell manifests report the Git commit as unavailable.

S FRESH-SEED FEATURE-WIDTH SCREEN

The exploratory v15 protocol asked whether a 10-feature version of the custom classical controller would pass the same competence gate against full-state SB3. It also retained the 2-feature controller as a reference arm. The run used 200,000 interactions for each of seeds 211–213 and one episode on each of 100 shared evaluation cases. All nine method–seed cells and all 900 evaluation episodes completed without failures. The shared manifest hash is `f4e362ff6725e4633e07d2c87fddf4e06e4bd1318e9df6b4965124a181c4cfcd`.

Table 21: Exploratory v15 feature-width screen. Distance and success are averages across the three training seeds; the standard deviation is across seed-level means. Success uses the derived 0.05 m threshold. No quantum method was run.

Method	Final distance (m)	Success	Trainable parameters
Custom PPO, 2-feature map	0.13151 ± 0.00087	7.3%	109
Custom PPO, 10-feature map	0.17559 ± 0.05927	5.0%	397
Full-state SB3 PPO	0.09549 ± 0.02087	23.7%	9,925

The predeclared candidate was the 10-feature custom PPO. Its mean paired distance excess over SB3 was $+0.08011\,\mathrm{m}$ (three seeds; one-sided 95% percentile-bootstrap upper bound $+0.10597\,\mathrm{m}$), far above the $0.01\,\mathrm{m}$ gate. Mean candidate success was 5.0%, and its lowest seed was 1%; the gate required 25% pooled and 10% per seed. It passed the learning-trend and saturation criteria but failed all task-quality criteria. The protocol therefore stopped before any quantum follow-up.

This is not a one-factor test of bottleneck width. The encoder and readout dimensions change with the feature map, and the parameter count rises from 109 to 397. We report v15 as an exploratory architecture/parameterization screen; it does not establish that width alone caused the weaker result. Training-time comparisons are omitted because seed workers ran concurrently. The recovered 94-file raw archive and the analysis outputs are checksum-bound in the accompanying release materials. The local audit matched the recorded source snapshot to a clean source tree; the Modal manifest itself reported the Git commit as unavailable. We omit the repository revision from this double-blind manuscript.

T STATE-PRESERVING Q1 FOLLOW-UP

The exploratory v16 run tested whether the earlier feature-map comparison was being obscured by a two-dimensional bottleneck. It used Reacher-v5 with all ten state values retained. The two augmented policies received either fixed sine features or Q1 finite-shot features; the third arm was full-state SB3. All arms used SB3 PPO, targeted 200,000 interactions per training seed, and executed 200,192 because SB3 collects complete 256-step rollouts. They used the same 100-case evaluation manifest. The Q1 map used two qubits, two layers, and 128 shots. Circuit parameters remained fixed. The run used 20 seeds (601–620), and all 60 method-seed cells and 6,000 evaluation episodes completed without training or evaluation failures. Four workers were preempted and resumed; their resume manifests and one incomplete attempt directory remain in the raw record. According to the Modal coordinator log, cancellation was recorded after all scheduled cells had committed; the immutable run files independently support the completed-cell accounting but do not establish that event order.

Table 22: Exploratory state-preserving Reacher follow-up, v16. Means and standard deviations are across 20 training seeds; each checkpoint was evaluated on the same 100 cases. These results do not replace the v9 primary cohort and do not authorize a confirmatory claim.

Method	Distance (m)	Success	Train wall (s)	Mean per-seed p50 (ms)	Train shots
Full-state SB3	0.11187 ± 0.01534	11.4%	164.4	0.252	0
State + sine	0.11811 ± 0.01519	11.2%	169.9	0.279	0
State + Q1	0.11887 ± 0.01151	10.6%	314.5	0.948	2.61×10^7

The primary exploratory contrast was Q1 augmentation minus sine augmentation: $+0.00076\,\mathrm{m}$, with a paired 95% percentile-bootstrap interval $[-0.00556, +0.00716]\,\mathrm{m}$ and one-sided upper bound $+0.00620\,\mathrm{m}$. The corresponding Student-t upper bound was $+0.00657\,\mathrm{m}$. Although both bounds lie below the protocol's numerical $0.01\,\mathrm{m}$ margin, the protocol explicitly disallows confirmatory claims. The per-comparison `claim_eligible` flag covers the numerical endpoint calculation; the experiment-level gate fails because this analysis is exploratory and the protocol disallows claims. We report the estimates as descriptive only. The full-state SB3 arm also reached only 11.4% success on this cohort, so the run does not establish competent control. Q1's training wall time was 1.85 times that of the sine arm, and its mean per-seed p50 action latency was 3.40 times higher. It executed

204,196 circuits and used 26,137,088 shots per seed during training, followed by 5,000 evaluation circuits and 640,000 evaluation shots. No method missed the 20 ms action deadline.

The full-state-versus-augmented comparisons are descriptive; the analysis did not produce paired confidence intervals or multiplicity-adjusted tests for those secondary contrasts. The training logs offer a limited diagnostic. Averaged over seeds, explained variance in the final training quartile was 0.988 for full-state SB3, 0.989 for sine augmentation, and 0.988 for Q1 augmentation. Action saturation was zero in that quartile for all three. Those two diagnostics do not establish overall PPO stability or explain the low success rate.

The analysis used 10,000 paired-seed bootstrap resamples with seed zero. The shared evaluation-manifest hash is `28e5fa69fd5d63f515634bd86ad8c0d8c7243090c728000d29b629a5a766bec4`. The protocol was `reacher_v16_state_augmented_q1`; its resolved configuration records Gymnasium 1.3.0, MuJoCo 3.12.0, SB3 2.9.0, and the full Reacher model XML hash. The local dispatch record identifies a committed source snapshot and a clean experiment-source tree at launch, but the remote run configuration omits the commit identifier. This link therefore comes from the local dispatch record, not the run manifest. The complete raw run and generated analysis remain local; no reviewer-accessible archive has been published.

U 500K-STEP FULL-STATE SB3 COMPETENCE SCREEN

The exploratory v17 screen asked whether a longer run with the standard full-state SB3 controller would reach a stronger task-performance level than the earlier compact-policy screens. It trained SB3 PPO for 500,000 target transitions on each of three fresh training seeds (701–703), then evaluated each checkpoint on the same immutable set of 100 Reacher-v5 cases. SB3 collected complete 256-step rollouts, resulting in 500,224 transitions per seed. No quantum or augmented-feature policy was run.

The frozen v17 gate passed its return-improvement criteria (3/3 seeds), pooled success threshold, and per-seed success threshold. Mean final distance was $0.08343\,\mathrm{m}$ and mean derived success was 29.3%. But the gate also required the maximum action saturation across all training updates to be at most 5%. That criterion failed: the seed-level maxima were 36.5–36.7%, all on the initial, untrained policy update. Saturation was zero throughout the final update quartile in all three runs. This later diagnostic suggests an initialization transient rather than a late-training collapse, but it does not change the preregistered v17 decision. The v17 screen therefore failed, and it cannot qualify the reference for a quantum comparison.

The v18 protocol prospectively moved the saturation check to the final training quartile and used fresh seeds and a disjoint evaluation cohort. This exploratory screen still failed: pooled success was 24.33%, below its frozen 25% criterion, despite positive return changes on all seeds and zero late-quartile saturation. Its threshold was not relaxed. A fresh v19 SB3-only replication uses the same gate on new seeds and evaluation cases; neither v18 nor the three-seed v19 screen is confirmatory evidence. No quantum comparison was admitted before the exploratory classical gate passed. V19 then passed that gate on three fresh seeds, but no matched quantum follow-up was run; the screen should not be read as a confirmatory qualification.

V18's per-seed final distances were $0.11537\,\mathrm{m}$, $0.08143\,\mathrm{m}$, and $0.08587\,\mathrm{m}$; derived success was 10%, 30%, and 33%, respectively. Mean distance was $0.09422\,\mathrm{m}$ and mean early-to-late return change was +14.823. The complete screen and its limitations are documented in `docs/reacher_v18_tail_stability_report.md`. V19's mean final distance was $0.08947\,\mathrm{m}$, pooled success was 32%, and mean return change was +19.149; individual seed success rates were 45%, 35%, and 16%. It passed the declared exploratory gate. The 100-case manifest hash is `3572ea4205cfda058ee6985200a9760b883a10ffc53e8b786f072279f8ae4336`. This new screen is described in `docs/reacher_v19_tail_stability_report.md`; the 32-file local manifest also retains incomplete canceled attempts.

The v17 evaluation-manifest hash is `d087858b773c7fd473ea8dd99f93ccaa262de2bf634e70ac484b00cebe517c13`. The 29 recovered artifact files pass the local raw-artifact manifest check. The run used the full Reacher MJCF hash `a10215aae27636504b1d41eb98c09ad50dbcb9f4ad888ed7593ff652053168db` and the source revision is pinned in its

provenance record; its identifier is redacted in this double-blind manuscript. The records remain local; no reviewer-accessible archive has been published.

V 500K-STEP STATE-AUGMENTED FEATURE COMPARISON (V20)

V20 compared fixed Q1 and sine feature maps inside the same state-augmented SB3 PPO policy. Both methods used all ten Reacher state values, a two-feature augmentation, identical action heads and PPO settings, a 500,000-step target, and the same 100-case evaluation manifest. The frozen cohort had 20 training seeds (801–820). Q1 used two qubits and 128 shots; neither the feature encoder nor circuit parameters were trained. The evaluation-manifest hash is `81846893cab01f88080dd8d286291aa0671383dbbbadb485a407db0f40b59f88`.

The paired analysis was regenerated from the immutable evaluation artifacts with 10,000 bootstrap resamples and seed zero. Eighteen Q1 seeds completed fresh-process evaluation; Q1 seeds 812 and 813 timed out during training at 499,712 and 489,216 interactions, respectively. The sine arm completed all 20 seeds. The protocol allowed zero failed confirmatory seeds, so its confirmatory gate fails. We report the complete-case comparison as exploratory and retain the failures rather than treating these rows as task outcomes.

Table 23: Exploratory v20 results. Values are mean ± standard deviation across completed training seeds; the Q1 denominator is 18/20 and the sine denominator 20/20. Training wall times are descriptive because seed workers ran concurrently.

Method	Final distance (m)	Success	Train wall (s)	p50 latency (ms)	Train shots
State + sine	0.09273 ± 0.02065	$26.8\% \pm 13.2\%$	585.5 ± 70.1	0.391 ± 0.057	0
State + Q1	0.11783 ± 0.02229	$16.2\% \pm 9.1\%$	1066.8 ± 120.6	1.220 ± 0.139	65.31×10^6

For the 18 paired completed seeds, Q1 minus sine final distance was $+0.02721\,\mathrm{m}$ (95% percentile-bootstrap interval $[+0.01840, +0.03647]\,\mathrm{m}$; one-sided upper bound $+0.03495\,\mathrm{m}$), above the $0.01\,\mathrm{m}$ non-inferiority margin. Q1 had higher distance in 17 of the 18 complete pairs. A failure-inclusive analysis using the protocol's worst-observed-seed rule had an upper bound of $+0.03667\,\mathrm{m}$; it also failed the margin. These estimates point against Q1 in this policy family, but the timeouts and failed protocol gate mean the study is not confirmatory.

The completed Q1 seeds averaged 510,229 training circuit executions and 65,309,312 shots, plus 5,000 evaluation circuits and 640,000 evaluation shots. Their per-seed median action latency averaged $1.220\,\mathrm{ms}$; the sine control averaged $0.391\,\mathrm{ms}$. Training CPU time averaged 1,000.7 s for Q1 and 543.2 s for sine. These wall and CPU measurements come from concurrent Modal seed workers and should not be interpreted as a controlled serial speed comparison. No method missed the 20 ms action deadline.

The run configuration records Gymnasium 1.3.0, MuJoCo 3.12.0, SB3 2.9.0, the exact qualification identifier and file digest, and the full model hash `a10215aae27636504b1d41eb98c09ad50dbcb9f4ad888ed7593ff652053168db`. The source revision is recorded in the local run configuration and redacted from this double-blind manuscript. The 294-file recovered run passes its local checksum manifest. Its raw artifacts and analysis remain local and have not been uploaded to a reviewer-accessible anonymous host.

W TWENTY-SEED FULL-STATE SB3 COMPETENCE SCREEN (V21)

V21 tested the full-state SB3 PPO reference on 20 fresh training seeds (821–840), with a 500,000-interaction target and 100 distinct, shared evaluation starts (75100–75199). All 20 seeds completed; each used 500,224 environment steps due to PPO rollout-boundary accounting. All evaluation artifacts contain exactly the declared 100 cases and use manifest hash `7b452cc9eab1133daaf0e8c1ebaa184d958f27a24974837077e27820b314eab0`. No evaluation case failed.

Table 24: Exploratory v21 full-state SB3 screen. Statistics are across 20 training seeds; success is derived from final distance below 0.05 m.

Metric	Mean ± SD	Median	Range
Final distance (m)	0.08088 ± 0.01832	0.07556	0.05763–0.13000
Success rate	$33.45\% \pm 13.95\%$	35.5%	7–58%
Late-minus-early return	$+15.10 \pm 5.86$	+12.96	+11.86–+36.18

Every seed improved its late-training return. The maximum action saturation over all updates was 60.2%; in the predeclared final-quarter window the maximum was zero. The frozen exploratory gate passed the 20/20 completion, manifest, return-improvement, positive-mean-return, final-quarter saturation, and pooled-success criteria. It failed the per-seed success criterion because seeds 823 and 833 achieved 9% and 7%, below the 10% floor. We do not relax the threshold or call this a pass. Return improved on all seeds and final-quarter saturation was zero, yet two seeds fell below the success floor. The return gains therefore do not establish absolute task competence or qualify a quantum comparison.

The run's 142 manifest-listed artifacts pass the checksum verifier. The recovered raw tree and a commit-redacted local staging archive are kept separately; neither has been uploaded to an anonymous host. The gate output and full run details are in `docs/reacher_v21_sb3_competence_20seed_report.md`.

X Complete run catalog and supporting experiments

The study grew through controller repairs, failed screens, and changes to the evaluation procedure. This catalog keeps that history visible. It covers the top-level directories returned by the Modal artifact-volume listing on 26 September 2026 and points to the results reported elsewhere in the paper. Names below are artifact identifiers, not public download links. The raw archives remain staged locally.

A directory establishes that an attempt or analysis was retained. It does not establish that every scheduled cell completed. Where we have only a historical directory or no reconciled result, we make no numerical claim. Analysis revisions and recovery shards do not count as additional independent studies. This distinction matters for RL comparisons, where implementation changes and repeated tuning can otherwise be mistaken for replication (Henderson et al., 2018; Andrychowicz et al., 2021; Patterson et al., 2024).

X.1 Preparation, repairs, and the primary Reacher cohort

Early feature-map preparation. `shots_v3_reacher_smoke_01`, `shots_v3_reacher_pilot_01`, and `shots_v4_reacher_pilot_01` are retained preparation runs. Their presence in the volume is verified, but they supply no primary estimate in this paper. The same restriction applies to the early exploratory `reacher_v2_exploratory_pilot_01`.

First Reacher run family. The volume contains `reacher_v1_pilot_01`, `reacher_v1_confirmatory_01`, and `reacher_v1_confirmatory_02`. These names distinguish dispatch attempts; they must not be counted as independent confirmatory replications. The original laptop experiment is described separately in Appendix A. Later retained directories are `reacher_v1_corrected_primary_classical_01`, `reacher_v1_corrected_primary_quantum_01`, `reacher_v1_identity_matched_01`, and `reacher_v1_sequential_timing_01`. They document the transition to corrected agents and sequential timing. Their historical identity records do not establish the full-model provenance of the v9 cohort, and none contributes observations to its primary estimate.

Stability work before v5. `shots_v5_reacher_stability_01` is the failed stability experiment discussed in Appendix A. Its four analysis directories, `shots_v5_reacher_stability_01-analysis-v1`, `shots_v5_reacher_stability_01-analysis-v2`, `shots_v5_reacher_stability_01-analysis-v3`, and `shots_v5_reacher_`

stability_01-analysis-v4, are successive analyses of that run. The subsequent classical preparation attempts are reacher_v3_classical_ppo_stability_pilot_01 and reacher_v3_classical_ppo_stability_pilot_02; reacher_v4_matched_quantum_features_pilot_01 is the matched-feature pilot. These exploratory runs precede the frozen reacher_v5_confirmatory_quantum_features_01 cohort, whose finite-shot training interpretation was later withdrawn after the replay audit (Section 5.4).

Corrected evaluation and controller screens. reacher_v6_corrected_distinct_eval_01 retained corrected exploratory evaluations, including eight recovered SB3 seed artifacts for seeds 20–27 on one 100-case manifest. Those recovered cells do not constitute a new 20-seed comparison. The volume also contains reacher_v7_modal_stability_gate_01, reacher_v7_stability_02, and reacher_v7_modal_quantum_feature_pilot_01. The failed competence screen in Appendix A and the nine-cell Q0/Q1/Q2 feasibility pilot in Section 5.3 have different evaluation scopes. Sharing a version prefix does not make their cohorts interchangeable.

Primary cohort. reacher_v8_controller_gate_01 is the relative-distance screen in Section 5.1. reacher_v9_full_identity_01 contains the 60-cell primary cohort in Section 5.2. Its 20 paired training seeds, common 100-case schedule, and interval-sensitive decision are the basis of the main Reacher claim. Earlier pilots are not pooled with it.

X.2 Ablations, readout noise, and repeated evaluation

The five-seed ablation run shots_v6_reacher_ablations_pilot_01 completed 85 cells; its analysis is shots-v6-reacher-ablations-pilot-analysis-v1. It varies shot count, qubit count, circuit layers, and the classical map family (Tables 5 and 6). shots_v6_reacher_ablations_smoke_01 and smoke-analysis-v1 are preparation artifacts for that study. The pilot motivated the separate 120-cell shot replication, shots_v8_reacher_shot_frontier_confirmatory_01, analyzed in shots-v8-reacher-shot-frontier-confirmatory-analysis-v1 (Section 5.6). Its three shot budgets are comparisons within one declared study, rather than three independent confirmations of v9.

The Q2 noise sweep has a separate correction history. shots_v7_q2_noise_smoke_01 and shots_v7_q2_noise_pilot_01 used the obsolete noise profile with an inert single-qubit term. They remain excluded. The corrected preparation run is shots_v7_q2_noise_smoke_02; the completed study is shots_v7_q2_noise_pilot_02, with analysis shots-v7-q2-noise-pilot-analysis-v2. Across its 15 cells, mean final distance was $0.122904\,\mathrm{m}$, $0.122228\,\mathrm{m}$, and $0.121083\,\mathrm{m}$ at $0.5\times$, $1\times$, and $2\times$ synthetic noise. Every paired interval between noise levels included zero. The ordering alone gives no evidence that stronger noise helps control.

reacher_v5_robustness_v1 contains the original nine-condition suite: 540 cells and 54,000 episodes. Its canonical analysis is v3. Analysis v1 had a schema error, while v2 mixed environment seeds in the longevity summary. Repeated deployment in the corrected analysis compares early and late episodes within each fixed environment seed. Classical and SB3 distance drift was zero; Q1 drift was $+0.000147\,\mathrm{m}$, with an interval spanning zero. No method had detectable latency drift. This check concerns repeated software execution and sampling variation.

The graded extension is reacher_v5_graded_robustness_v2. Its 960 cells cover 16 conditions (Table 7). The retained directories reacher_v5_graded_robustness_v2_dynamics_090, reacher_v5_graded_robustness_v2_dynamics_110, and reacher_v5_graded_robustness_v2_dynamics_120 are recovery shards tied to that study's manifest. They do not add independent evidence beyond the reconciled condition records. Held-out and boundary target evaluations from the original suite appear alongside the graded conditions in the table.

Stochastic-action deployment has three retained versions: reacher_v5_stochastic_deployment_v1, reacher_v5_stochastic_deployment_v2, and reacher_v5_stochastic_deployment_v3. Only v3 fixes action RNG independently per cell and supplies the reported result. Its Q1-minus-classical distance difference is $+0.001399\,\mathrm{m}$ (95% interval $[-0.003610, +0.006327]\,\mathrm{m}$), with one-sided upper bound $0.005553\,\mathrm{m}$. All these deployment and

stress evaluations use archived v5 checkpoints. They describe those policies; they cannot establish the performance of the corrected Q1 training path.

X.3 THE SECOND-TASK EXPERIMENTS

Pusher preparation is retained as `pusher_v1_exploratory_smoke_01` and `pusher_v1_exploratory_smoke_02`, with `pusher-v1-smoke-analysis-v1` and `pusher-v1-smoke-analysis-v2`. The first smoke's asset hash depended on native memory, so that attempt is excluded. The numerical studies are `pusher_v1_exploratory_pilot_01`, `pusher_v2_learning_extension_pilot_01`, `pusher_v3_sb3_qualification_500k_01`, and `pusher_v4_sb3_qualification_1m_01`. Their analysis directories are, respectively, `pusher-v1-exploratory-pilot-analysis-v1`, `pusher-v2-learning-extension-analysis-v1`, `pusher-v3-sb3-qualification-500k-analysis-v1`, and `pusher-v4-sb3-qualification-1m-analysis-v1`. Table 8 reports the progression from 50k to one million interactions. No budget produced derived successes. Small differences between feature maps therefore do not establish useful manipulation.

Swimmer began with `swimmer_v1_sb3_learning_gate_01` and its analysis `swimmer-v1-sb3-learning-gate-analysis-v1`. SB3 passed the five-seed learning screen at 500k interactions. The follow-up, `swimmer_v2_matched_quantum_pilot_01`, completed 20 cells and is analyzed in `swimmer-v2-matched-quantum-pilot-analysis-v1`. Mean forward displacement was $1.3652\,\mathrm{m}$ for SB3, $1.0225\,\mathrm{m}$ for the classical feature policy, $1.0135\,\mathrm{m}$ for Q0, and $1.0137\,\mathrm{m}$ for Q1. The Q1-minus-classical difference was $-0.008744\,\mathrm{m}$ (95% interval $[-0.055124, +0.023419]\,\mathrm{m}$). This pilot exposes the gap to the larger classical reference even when the compact policies have similar means.

`swimmer_v3_matched_quantum_confirmatory_01` is the subsequent 40-cell replication. The volume retains `swimmer-v3-matched-quantum-confirmatory-analysis-v1` and `swimmer-v3-matched-quantum-confirmatory-analysis-v2`. Those analysis versions must be read with the later displacement-endpoint correction, described in Section 5.8 and Appendix J. The corrected test misses its margin. Reanalyzing the saved endpoints does not create a new cohort.

X.4 CONTROLLER FOLLOW-UPS THROUGH V21

The later experiments ask why the policies remain weak and whether a stronger classical controller can qualify a new comparison. Their exact run names and paper locations are collected below. They are separate exploratory protocols; their repeated attempts are not pooled into the primary test.

Table 25: Later Reacher studies. Each row points to the full account, including the gate decision and available provenance.

Retained run directory	Paper account
`reacher_v10_controller_competence_200k_01`	Appendix N: quality gate failed
`reacher_v11_value_critic_200k_01`	Appendix O: wider critic did not close gap
`reacher_v12_sb3_bottleneck_gate_01`	Appendix P: fixed bottleneck failed
`reacher_v13_learned_bottleneck_gate_01`	Routing failure; excluded attempt
`reacher_v13_learned_bottleneck_gate_02`	Appendix Q: learned bottleneck failed
`reacher_v14_featuremap_v3_gate_01`	Appendix R: fresh-seed gate failed
`reacher_v15_width_isolation_01`	Appendix S: wider features failed
`reacher_v16_state_augmented_20260925`	Appendix T: full state retained; low success
`reacher_v17_sb3_absolute_competence_01`	Appendix U: initial saturation fails gate
`reacher_v18_sb3_tail_stability_01`	Appendix U: pooled success below 25%
`reacher_v19_sb3_tail_stability_replication_01`	Appendix U: exploratory three-seed pass
`reacher_v20_state_augmented_500k_01`	Appendix V: 18 pairs complete; two Q1 timeouts
`reacher_v21_sb3_competence_20seed_01`	Appendix W: 20 complete; two miss success floor

V17's early dispatch and timeout records remain part of its history, even though a later recovery produced three evaluated seeds. V19 likewise retains canceled partial attempts. Completion and gate success are separate: v17 failed its original maximum-saturation rule, v18 failed pooled success, and v19 passed a revised exploratory gate on fresh seeds. In the larger v21 screen, two of 20 seeds missed the per-seed floor. Reading these screens together shows variation across seeds that a single three-seed pass would hide. It does not justify changing the frozen thresholds after evaluation.

X.5 Simulator validation and recovery checks

Appendix B reports the independent tensor oracle, Qiskit Aer comparison, and interrupted-checkpoint continuation check. The locally retained Qiskit reports include `independent-qiskit-validation-v1.json` and `independent-qiskit-validation-v2.json`. The later report retains 128 circuit cases and the same summary values: maximum analytic error 2.9627×10^{-8} and finite-shot standardized-residual RMS 0.9703. These report versions are validation records, not two independent robot studies. Circuit agreement checks the simulator calculation (Javadi-Abhari et al., 2024); it does not establish that the policy learns the robotics task. The interruption check answers a different question: whether saved state reproduces the same continuation in a fresh process.

X.6 Reading the result tables

Final distance and success answer different questions. Distance measures how close the arm finishes, including episodes that miss the target. Success counts only episodes ending within the project's five-centimetre radius. Two methods can have similar mean distance while both succeed rarely. This is why the v9 non-inferiority test cannot establish useful control on its own. Its one-centimetre margin limits an average loss relative to the chosen control; it does not impose an absolute accuracy requirement.

The pairing also matters when cells fail. In v20, the sine column averages 20 completed seeds and the Q1 column averages 18. Subtracting those column means compares different seed sets. The paired

estimate of $+0.02721\,\mathrm{m}$ instead uses the same 18 seeds on both sides. The two Q1 timeouts remain failures in the gate and the sensitivity analysis. They have no observed final-distance score and cannot be assigned zero distance, which would falsely count them as perfect reaching.

Latency describes how long an action takes to compute; training wall time includes the work needed to learn the policy. These costs need separate interpretations. In v9, Q1's mean per-seed median action latency was $0.915\,\mathrm{ms}$, versus $0.231\,\mathrm{ms}$ for sine. Both are below the 20 ms reference deadline at that summary level, although individual deadline misses are recorded separately. The ratio therefore shows extra CPU-simulator work, not that every quantum-feature action was too late. Nor does it estimate the energy or latency of running the circuit on a physical quantum device.

Finally, the independent sample for the primary uncertainty calculation is a trained policy seed. Evaluating 100 cases per seed gives a better estimate of each policy's performance. It does not turn 20 training seeds into 2,000 independent training runs. This is the reason for averaging episodes before bootstrapping paired seeds, following the concerns about uncertainty in RL evaluation discussed by Agarwal et al. (2021).

Together, this catalog and the numerical appendices account for the retained experimental work without assigning the same evidential role to every run. The primary interval, the exploratory controller results, and failures each remain traceable to their own cohort.