TRAINING HYBRID QUANTUM POLICIES WITH CLASSICAL SURROGATES FOR ACTOR–CRITIC REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reading out a parametrized quantum circuit through finite-shot measurements introduces a discrete stochastic component into hybrid quantum-classical neural networks, making gradient-based training inefficient. In this work, we introduce a surrogate-assisted actor-critic method for training hybrid quantum policies that avoids differentiating the quantum circuit during backpropagation. A classical neural surrogate trained on the circuit's input-output pairs from a replay buffer provides a differentiable substitute for actor updates and critic-target computation, while forward interaction and evaluation retain the quantum circuit's measurements. Our experiments on standard MuJoCo continuous-control tasks demonstrate promising policy learning. Notably, on Humanoid, more runs reach high returns than with learned classical baselines. Moreover, the best average return in our ablation comes from a smaller circuit / surrogate, suggesting that strong performance does not depend on a large network. We also derive conditional upstream gradient-error bounds that separate surrogate fitting from the mean-feature relaxation. Together, the results support surrogate-assisted training as an effective and efficient approach to learning hybrid quantum policies.

1 INTRODUCTION

Parametrized quantum circuits (PQCs) have been used to represent policies and value functions in reinforcement learning (Jerbi et al., 2021; Skolik et al., 2022; Jin et al., 2025). An N-qubit circuit transforms a quantum state in a 2^N-dimensional Hilbert space, with measurements producing N-bit outcomes. Their probabilities depend on how the input is encoded and how the circuit transforms the state, providing a nonlinear feature map (Schuld & Killoran, 2019). In a hybrid policy, classical networks learn to encode observations into circuit controls and map the measured features to actions (Jin et al., 2025).

The functions accessible to a quantum model depend on its input encoding, circuit structure and readout (Schuld et al., 2021). Alongside experiments on standard RL benchmarks, Jerbi et al. (2021) construct environments in which quantum agents have a provable learning advantage over polynomial-time classical learners, assuming the classical hardness of the discrete logarithm problem. This establishes a separation for particular problem families. Meanwhile, the practical value of quantum representations in continuous control still requires empirical study.

The practical challenge is to train the surrounding networks without repeatedly differentiating or querying the circuit on large replay minibatches. The cost can restrict the tasks that are studied: Lokossou et al. (2025) evaluate a hybrid quantum soft actor-critic on Walker2d-v4 and identify the computational demands of Humanoid as a barrier to extending their experiments.

We study a policy whose circuit output is the empirical mode of S measurement shots, that is, the complete bitstring observed most often. This bitstring provides the binary features passed to the action network. Related mode-based readouts have been used in variational quantum optimisation (Zhang & Cheng, 2026). Ordinary backpropagation does not provide gradients through the measurement sampling and discrete mode-selection steps. Parameter-shift rules can differentiate appropriate quantum expectations (Wierichs et al., 2022), but their application here must account for the full shot-aggregation operation.

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.

We fit a classical neural surrogate to circuit control-output pairs stored alongside transitions in the replay buffer. Its predictions supply features for actor updates and critic-target computation without new circuit queries. Gradients train the encoder through the surrogate, whose weights are updated only by response fitting. Action selection and evaluation retain quantum measurements.

We make three contributions. First, we integrate replay-based surrogate fitting into hybrid actor-critic learning, removing circuit queries from minibatch updates. Second, we evaluate learning across four continuous-control tasks and examine training-seed variability, model capacity and deployment features. After training, replacing the circuit with the surrogate's predictions yields a purely classical actor that retains all ten high-return Humanoid seeds. It shares the architecture of the matched fully connected baseline (FCM), allowing the two training procedures to be compared at deployment (Appendix E). The training-time mechanism behind the observed difference remains unidentified. Third, we derive conditional bounds on the encoder's gradient error that separate surrogate-fitting error from the error of replacing sampled features by their mean.

Relation to prior work. Classical surrogates have been used for quantum-model inference (Schreiber et al., 2023; Nair & Ferrie, 2025) and variational optimization (Shaffer et al., 2023; O'Leary et al., 2025). In supervised classification, they provide differentiable proxies for quantum outputs (Batista et al., 2025; Xie, 2025). We extend this approach to off-policy actor-critic learning, fitting shot-aggregated responses from a changing replay distribution for actor updates and bootstrapped critic targets.

The method also relates to gradient estimators for stochastic neurons (Bengio et al., 2013) and surrogate-gradient learning in spiking networks (Neftci et al., 2019). Synthetic gradients predict backpropagated error signals directly (Jaderberg et al., 2017), whereas local generative surrogates approximate stochastic simulators with differentiable models (Shirobokov et al., 2020). Gumbel-Softmax and Concrete provide continuous relaxations of discrete samples (Jang et al., 2017; Maddison et al., 2017), and RELAX combines a learned control variate with correction terms to obtain an unbiased gradient estimator (Grathwohl et al., 2018). In contrast, our surrogate predicts mean features, so differentiating through these predictions generally gives a biased update. We separate response-fitting and mean-feature relaxation errors and compare this update with a sampled-forward straight-through alternative.

2 Surrogate-assisted actor–critic training

2.1 Hybrid quantum-classical policy

The hybrid policy combines an N-qubit circuit with two classical networks: a pre-network Pre_θ and a post-network Post_ψ, parameterized by θ and ψ, respectively. Given a state s, the pre-network produces $(u, c) = (u_\theta(s), c_\theta(s)) = \mathrm{Pre}_\theta(s)$, where $u \in [0,1]^d$ controls the circuit and $c \in [0,1]^{d_c}$ passes directly to the post-network through a bypass connection.

For circuit controls u, each measurement shot follows the Born rule distribution π_u on $\{0,1\}^N$. The aggregation rule A_S selects the most frequent complete bitstring from S independent shots. The post-network then combines this bitstring with the bypass features to produce the action, following the interaction and evaluation path shown in Figure 1:

$$Z^{(1)}, \ldots, Z^{(S)} \overset{\text{iid}}{\sim} \pi_u, \qquad B = A_S(Z^{(1)}, \ldots, Z^{(S)}) \sim P_{u,S}, \qquad a = \mathrm{Post}_\psi(B, c). \tag{1}$$

Note that the aggregated mean $\mu(u) = \mathbb{E}_{P_{u,S}}[B]$ generally differs from the single-shot mean $\mathbb{E}_{\pi_u}[Z]$. We call a tie when several bitstrings share the largest shot count, and our analysis assumes a fixed rule for selecting among them: for a measurable action set D, the deployed policy is the pushforward

$$\Pi_{\theta,\psi}(D \mid s) = \sum_{b \in \{0,1\}^N} P_{u_\theta(s),S}(b)\mathbf{1}\{\mathrm{Post}_\psi(b, c_\theta(s)) \in D\}. \tag{2}$$

The parameters θ and ψ therefore determine the deployed hybrid policy: the pre-network selects the circuit controls, and the post-network maps the measured bitstring and bypass features to an action. The surrogate introduced below is used to compute training updates, it is not evaluated in this hybrid action-selection path.

For a quantum circuit with M layers, the pre-network supplies $2NM$ rotation controls and $2M$ routing controls, giving $d = (2N + 2)M$. After rounding, the routing controls select the qubit pair connected by the entangling gate in each layer. Unless varied, $N = M = d_c = 10$ and $S = 1000$. Appendix C specifies the gate sequence and network architectures.

Interaction and evaluation: retain quantum measurements
$s \longrightarrow \mathrm{Pre}_\theta \longrightarrow (u, c) \longrightarrow (B \sim P_{u,S}, c) \longrightarrow \mathrm{Post}_\psi \longrightarrow a$

Replay fitting: update only the surrogate
Stored (u, B) pairs $\longrightarrow$ BCE fit of $f_\phi(u)$

Actor and critic-target updates: no new circuit queries
$s \longrightarrow \mathrm{Pre}_\theta \longrightarrow (f_\phi(u), c) \longrightarrow \mathrm{Post}_\psi \longrightarrow Q_\omega(s, a)$

Figure 1: Computational schematic of surrogate-assisted training. Top: the hybrid policy selects actions. Middle: stored control-output pairs train the surrogate. Bottom: surrogate predictions supply the features for actor updates and critic-target computation without new circuit queries. Critic targets use the corresponding target networks. Full network diagrams appear in Appendix A.

2.2 Learning the circuit response

A neural surrogate $f_\phi : [0, 1]^d \to (0, 1)^N$ that we call `qtDNN` predicts the aggregated marginal means. We fit it to stored (u, B) pairs with per-bit binary cross-entropy (BCE):

$$\widehat{\mathcal{L}}_{\mathrm{BCE}}(\phi) = -\frac{1}{|\mathcal{M}|N} \sum_{(u,B)\in\mathcal{M}} \sum_{k=1}^{N} [B_k \log f_{\phi,k}(u) + (1 - B_k) \log(1 - f_{\phi,k}(u))]. \tag{3}$$

Each fit uses 32 Adam steps on size-16 resamples from the current 256-transition minibatch $\mathcal{M}$. The surrogate retains its parameters between fits. For the base PQC and classical stochastic channels, ϕ is updated only by BCE fitting: the reinforcement learning loss never updates it. During actor optimization, gradients pass through f_ϕ with respect to its inputs to train the encoder.

2.3 Actor and critic updates

Let $A^\phi_{\theta,\psi}(s) = \mathrm{Post}_\psi(f_\phi(u_\theta(s)), c_\theta(s))$ denote the deterministic actor used during training updates. An overbar (e.g., $\bar{\omega}$) marks target-network parameters. The critic target y and actor loss $\mathcal{L}_A$ are

$$y = r + \gamma(1 - d_{\mathrm{term}}) \min_{j=1,2} Q_{\bar{\omega}_j}\big(s', \mathrm{clip}_{\mathcal{A}}[A^{\bar{\phi}}_{\bar{\theta},\bar{\psi}}(s') + \epsilon]\big), \tag{4}$$

$$\mathcal{L}_A(\theta, \psi) = -\mathbb{E}_{s\sim\rho_{\mathcal{M}}}\Big[Q_{\omega_1}\big(s, A^\phi_{\theta,\psi}(s)\big)\Big]. \tag{5}$$

Here ϵ is clipped target-policy smoothing noise, and d_{term} excludes time-limit truncations. The critics regress to the detached target y using executed actions stored in replay. As in TD3 (Fujimoto et al., 2018), actor and Polyak target updates occur every second critic update. The surrogate target is hard-copied immediately after fitting, so $\bar{\phi} = \phi$ when computing y while only the pre-network, post-network and critics are Polyak-lagged. Section 4 analyzes the fixed-critic actor approximation and discusses the distinct target substitution. Algorithm 1 gives the update order.

3 Experiments

We evaluate policy learning, outcomes across seeds, model capacity, and the update-time substitution. All circuits are simulated. The default setup uses ten qubits, ten layers, 1000 shots and a width-2048 surrogate. Evaluation retains quantum sampling but removes external exploration noise. Periodic scores average ten episodes and completed runs receive a separate terminal evaluation, normally over 30 episodes. The nominal training budget is 3M interactions. Protocols, run-completion status, run-level scores and post-hoc analyses appear in Appendices C-D.

3.1 Policy learning across four environments

Figure 2 shows learning on Humanoid-v4, Hopper-v4, Ant-v5 and HalfCheetah-v5. The Humanoid comparison includes the quantum bottleneck (PQC), a fully connected linear-sigmoid bottleneck

(FC), a fully connected bottleneck matched to the surrogate architecture (FCM), a classical stochastic bottleneck (CSB), and a random-bit generator (RBG). Section 3.2 describes how these configurations are trained. Hopper, Ant and HalfCheetah are summarized at fixed budgets using ten, ten and nine seeds, respectively (Table 1). Environment versions are fixed within each task and are not pooled across versions.

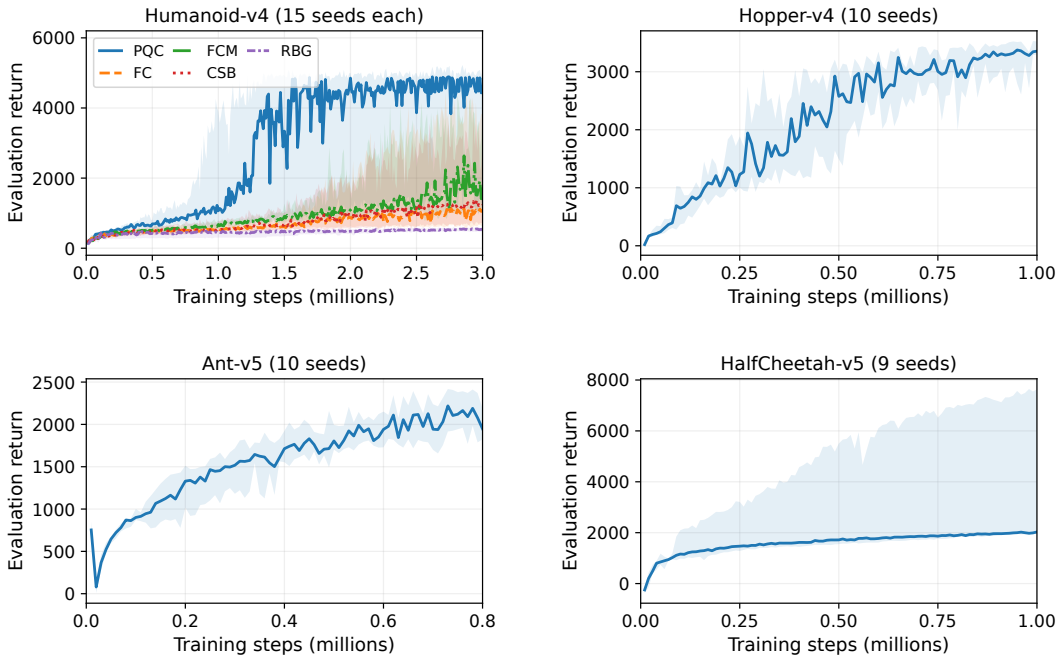


Figure 2: **Surrogate-based updates train policies that retain quantum measurements.** Lines show median ten-episode returns; shading shows the interquartile range across a fixed seed cohort. Humanoid includes all five bottleneck configurations; each has 15 seeds. Appendix D.5 shows individual trajectories for every task and Humanoid configuration.

Environment	Budget	Seeds	Median	IQM [95% CI]
Ant-v5	0.8M	10/10	1,947	1,989 [1,540, 2,335]
Hopper-v4	1.0M	10/10	3,355	3,394 [2,756, 3,498]
HalfCheetah-v5	1.0M	9/10	2,027	3,133 [1,738, 6,759]

Table 1: Additional-task returns at the plotted budgets. Scores use the latest evaluation at or before each budget. Brackets give 95% seed-bootstrap intervals for the trimmed-mean convention in Appendix D.1.

Hopper reaches median return 3355 and Ant 1947. HalfCheetah has a pronounced split: three runs exceed 7600 at 1M interactions while six remain below 2100, giving median 2027. The hybrid policy therefore learns on all four tasks, with consistency varying across seeds and environments. Longer-horizon summaries for nine-seed Ant and Hopper cohorts, and completed-run terminal scores, are reported separately in Appendix D.5.

3.2 High-return outcomes and learning progress

3.2.1 Classical comparisons

FC and FCM train their bottlenecks directly through the actor loss. PQC and CSB use surrogate-assisted updates, while RBG freezes the pre-network, including its bypass. The remaining actor-critic settings are shared (Appendix C.3). Since PQC and CSB have different conditional output entropies, their comparison evaluates the stochastic feature map as a whole (Appendix D.4).

Bottleneck	Mean	Median	IQM [95% CI]	≥ 4000
PQC	3,619	4,588	4,076 [2,513, 4,942]	10/15
FC	2,121	1,138	1,616 [777, 3,064]	4/15
FCM	2,509	2,399	2,387 [1,437, 3,553]	4/15
CSB	1,911	1,246	1,722 [854, 2,766]	1/15
RBG	521	558	537 [481, 577]	0/15

Table 2: Humanoid-v4 terminal returns after nominally 3M interactions, with 15 seeds and 30 terminal episodes per seed for every method. IQM intervals use 20,000 seed-bootstrap resamples. The 4000 threshold is descriptive.

The hybrid policy attains median return 4,588 and reaches 4000 in ten of 15 runs, compared with four for each learned classical baseline (Table 2). Figure 3 shows the complete threshold profile. Late-training median returns also exceed 4000 in ten PQC runs, versus four FC, three FCM, two CSB and no RBG runs. Appendix D.1 gives uncertainty intervals and the full-terminal and threshold tests. The PQC terminal scores in Table 2 are hybrid evaluations. Appendix E shows that the stored surrogate actor can be deployed without the circuit while retaining the same ten high-return Humanoid seeds.

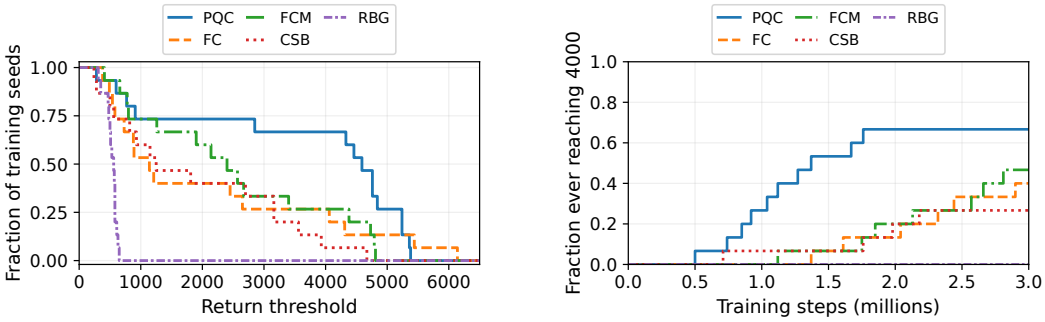


Figure 3: **High-return outcomes and progress toward them.** Left: the fraction of 15 terminal scores meeting each return threshold. Right: the fraction of runs that have reached a periodic return of 4000 by each interaction budget, including transient crossings. The threshold and crossing analysis are post hoc. Pointwise intervals and run-level crossing times appear in Appendices D.2 and D.8.

3.2.2 Progress across training

A complementary post-hoc measure averages each seed's evaluation returns at $0.1, 0.2, \ldots, 3.0$M interactions. Its median is 3144 for PQC, compared with 751, 959, 801 and 470 for FC, FCM, CSB and RBG. Exact Mann-Whitney tests favor PQC against both learned baselines after a four-comparison Holm correction ($p = 0.026$ and 0.033). Ten PQC runs first cross 4000 by 1.76M interactions, while over the full budget, FC, FCM and CSB cross in six, seven and four runs respectively (Figure 3). The primary terminal-score tests remain inconclusive against FC and FCM (Holm-adjusted $p = 0.135$ each), while four other descriptive measures (terminal high-return frequency, late-training returns, grid-averaged learning progress and first-crossing curves) favor PQC. See Appendix D.8 for details.

3.3 Compact configurations and shot budget

We test widths $h \in \{64, 256, 1024, 2048, 4096\}$ at five and ten qubits, with three runs per setting. Five qubits and width 256 give the highest mean terminal return, 5,088, with all three scores above 4000. Its surrogate has 32,261 parameters, 14.7$\times$ fewer than the ten-qubit, width-2048 reference. The compact setting also exceeds 4000 in each seed's late-training median. The three-seed mean ranking is exploratory. We find that the ten qubits with width 1024 configuration has the highest median, suggesting that strong performance does not require the larger configurations tested here.

The shot sweep uses width 2048 and five runs per setting. At five qubits, reducing shots from 1000 to 100 raises mean return from 2537 to 4617, with all five lower-shot runs above 4000. At ten qubits, the corresponding means are 3301 and 4056. Changing shot count changes the aggregated output law as well as cost. A separate CPU characterisation at 200 random ten-qubit controls makes this change explicit: median aggregated entropy rises from 1.92 to 4.62 bits when reducing shots, under smallest-code tie-breaking. Five-qubit controls show the same direction of change (Appendix D.4.1). These measurements show how shot count changes the distribution of bitstrings returned by the circuit. However, they do not establish that higher entropy causes the improvement in policy returns. The reference re-executions share long training prefixes with primary seeds and are never pooled as extra seeds. Their later divergence is consistent with sensitivity to resumed sampling (Appendix D.7).

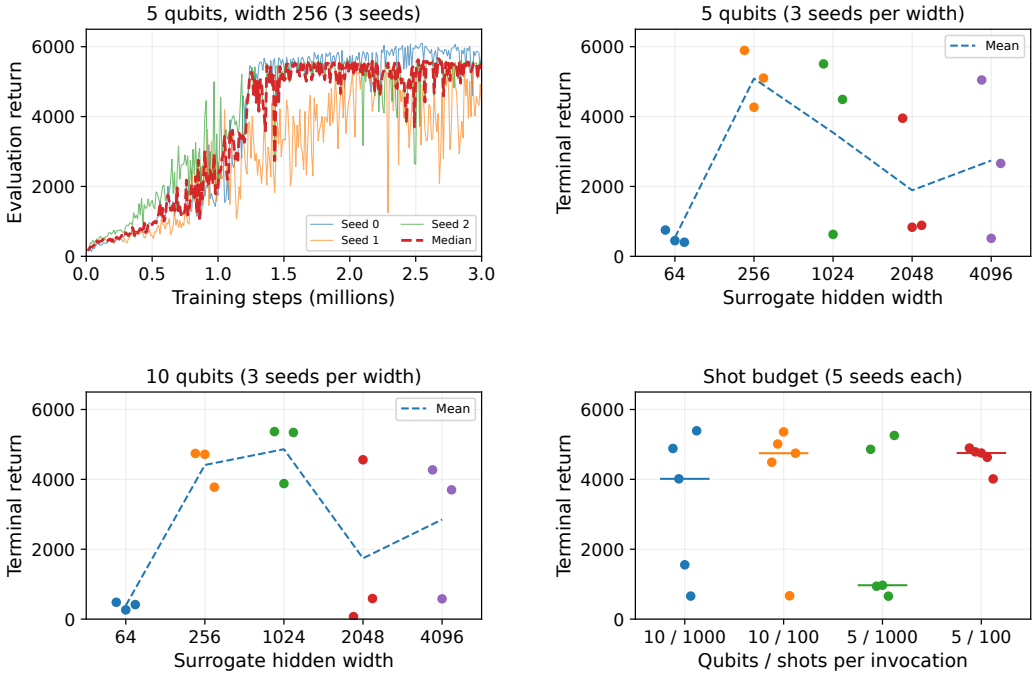


Figure 4: **A compact circuit and surrogate lead the capacity sweep in mean return.** All three compact-setting trajectories and all terminal observations are shown. Dashed lines in the width panels join means, while shot-panel segments are medians. The ten-qubit, width-2048 and ten-qubit, 1000-shot reference cells are same-seed re-executions (Appendix D.7).

3.4 Surrogate features during training updates

We compare three paths for actor updates and critic-target computation. Surrogate-forward uses the continuous features $f_\phi(u)$. Hard-forward straight-through estimation (STE) uses $B + f_\phi(u) - \mathrm{sg}(f_\phi(u))$, where sg is the stop-gradient operator: it returns its argument unchanged but treats it as constant during backpropagation. This retains the sampled bitstring B in the forward pass while using the surrogate's derivative with respect to u in the backward pass. Hard-forward stop-gradient also uses sampled features but freezes the entire pre-network, including its bypass.

All three retain quantum measurements for action selection and evaluation. Comparing surrogate-forward with STE tests whether replacing sampled features with continuous surrogate predictions is an effective training choice. Comparing STE with stop-gradient tests pre-network learning as a whole, including the bypass, rather than only gradients through the surrogate.

One channel invocation evaluates a control vector using S shots and returns their empirical mode. Counts include interaction, evaluation and update calls, and process time also includes surrogate fitting and other classical computation. Calls for optional diagnostics and derivative-target fitting are counted separately.

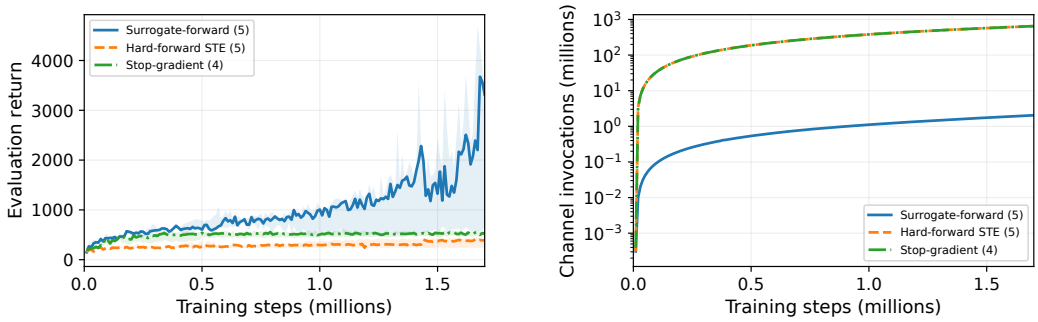


Figure 5: **Surrogate-forward and sampled-forward updates.** Median evaluation returns (left) and cumulative channel invocations (right), with interquartile shading over fixed cohorts: five surrogate-forward, five hard-forward STE and four stop-gradient runs. The surrogate-forward cohort is primary PQC seeds 0-4 throughout the plotted 1.7M interactions.

Update path	n	Return	Hours	Calls
Surrogate-forward	5	3,316	19.1	2.05M
Hard-forward STE	5	395	354.8	650.38M
Hard-forward stop-gradient	4	521	331.6	650.46M

Table 3: Medians at the latest evaluation before 1.7M interactions (1.690–1.691M). Time is cumulative in-process wall time per seed, including evaluation and resumes but excluding queue waiting. This is a matched-interaction, not a time-to-target comparison. The surrogate arm is primary seeds 0–4.

Surrogate-forward training reaches median return 3316, versus 395 for STE and 521 for stop-gradient (Table 3). It uses 318× fewer channel invocations than STE and 18.6× less median logged time. At batch size 256 and actor delay two, both sampled-forward controls add 384 invocations per update on average; surrogate-forward adds none. The small-cohort return contrasts are uncertain: at 1.7M, raw two-sided p-values are 0.056 against STE and 0.190 against stop-gradient. Earlier checkpoints, four-seed sensitivity and all three pairwise tests appear in Appendix D.6. This comparison evaluates a change in the update objective as well as in query cost.

4 Analysis of the Surrogate Approximation

We separate response-fitting error from the substitution of mean features in the actor objective. The analysis applies to PQC and CSB under the stated assumptions. Fix a state s, critic parameters ω, and post-network parameters ψ, and define $G_s(z,c) = Q_{\omega_1}(s, \mathrm{Post}_\psi(z,c))$. With $\mu(u) = \mathbb{E}_{P_{u,S}}[B]$, let

$$F(u,c) = \mathbb{E}_{B\sim P_{u,S}}[G_s(B,c)], \quad F_\mu(u,c) = G_s(\mu(u),c), \quad F_f(u,c) = G_s(f_\phi(u),c). \tag{6}$$

Here F is the sampled fixed-critic objective, F_μ its mean-feature relaxation, and F_f the surrogate update objective. The replay distribution is fixed; these are not environment-return objectives.

4.1 The Response Learned from Replay

The base surrogate is fitted coordinatewise to the *shot-aggregated* bitstring. Its population target is therefore the aggregated marginal mean, not the single-shot Born mean.

Proposition 1 (BCE identifies aggregated marginals)**.** *Let $u \sim \rho$ and $B \mid u \sim P_{u,S}$. For the population per-bit BCE risk $\mathcal{L}$ with natural logarithms and a measurable $f : [0,1]^d \to (0,1)^N$ of*

finite risk,

$$\mathcal{L}(f) - \mathcal{L}(\mu) = \frac{1}{N}\mathbb{E}_{u\sim\rho}\left[\sum_{k=1}^{N} \mathrm{KL}\big(\mathrm{Bern}(\mu_k(u)) \,\|\, \mathrm{Bern}(f_k(u))\big)\right],$$

$$\mathbb{E}_\rho\Big[\|f-\mu\|_2^2\Big] \le \frac{N}{2}[\mathcal{L}(f) - \mathcal{L}(\mu)]. \tag{7}$$

With the usual boundary convention for BCE, μ is the Bayes solution over the closed output cube, up to ρ-null sets. No independence between output bits is required.

The proof is the conditional cross-entropy identity followed by Pinsker's inequality (Appendix B.1). The bound controls prediction error under ρ. A finite replay fit additionally incurs approximation, optimization, and estimation errors; derivative accuracy is a separate property.

4.2 Error propagated to the state encoder

Fix a state s and write $u = u_\theta(s)$ and $c = c_\theta(s)$. Let p_θ denote the number of encoder parameters. We use Jacobians with output coordinates indexing rows and input coordinates indexing columns:

$$J_f := \frac{\partial f_\phi(u)}{\partial u} \in \mathbb{R}^{N\times d}, \qquad J_\mu := \frac{\partial \mu(u)}{\partial u} \in \mathbb{R}^{N\times d},$$

$$J_\theta u := \frac{\partial u_\theta(s)}{\partial \theta} \in \mathbb{R}^{d\times p_\theta}, \qquad J_\theta c := \frac{\partial c_\theta(s)}{\partial \theta} \in \mathbb{R}^{d_c\times p_\theta}.$$

All derivatives are evaluated at the current inputs and parameters. The notation $\nabla_\theta F_\bullet$ denotes the gradient of $F_\bullet(u_\theta(s), c_\theta(s))$ with respect to θ, holding the surrogate, post-network and critic parameters fixed. We use Euclidean norms for vectors and the corresponding induced operator norms for matrices.

Theorem 1 (Conditional upstream-gradient error, including the bypass)**.** *Consider a differentiability point within a fixed routing cell. Write $f = f_\phi(u)$, $\mu = \mu(u)$, and assume $\|f-\mu\|_2 \le \varepsilon_1$, $\|J_f - J_\mu\|_{\mathrm{op}} \le \varepsilon_2$, $\|J_\theta u\|_{\mathrm{op}} \le M_u$, $\|J_\theta c\|_{\mathrm{op}} \le M_c$, and $\|J_\mu\|_{\mathrm{op}} \le M_\mu$. Suppose G_s is continuously differentiable on a neighborhood of $[f,\mu]\times\{c\}$, with $\|\nabla_z G_s(f,c)\|_2 \le L_z$. On the feature segment $[f,\mu]$, let $\nabla_z G_s(\cdot,c)$ and $\nabla_c G_s(\cdot,c)$ be L_{zz}- and L_{cz}-Lipschitz, respectively. Then*

$$\|\nabla_\theta F_f - \nabla_\theta F_\mu\|_2 \le M_u L_z \varepsilon_2 + (M_u M_\mu L_{zz} + M_c L_{cz})\varepsilon_1. \tag{8}$$

The three terms respectively capture Jacobian error, feature error through the circuit path, and feature error through the bypass (Appendix B.2). An accurate BCE fit alone is insufficient. Routing boundaries and activation kinks lie outside this local result.

4.3 Replacing sampled features by their mean

Let $K = \mathrm{conv}(\mathrm{supp}\, P_{u,S})$ and $v(u) = \sum_k \mu_k(u)(1-\mu_k(u))$. Since B is binary, $\mathbb{E}\Big[\|B-\mu\|_2^2\Big] = v(u) \le N/4$, even with correlated bits. If $G_s(\cdot,c)$ is L_z^{supp}-Lipschitz on K, then

$$|F - F_\mu| \le L_z^{\mathrm{supp}}\sqrt{v(u)}. \tag{9}$$

If G_s is also L_z^{fit}-Lipschitz on $[f,\mu]$, then

$$|F - F_f| \le L_z^{\mathrm{supp}}\sqrt{v(u)} + L_z^{\mathrm{fit}}\varepsilon_1. \tag{10}$$

This value bound needs no Jacobian approximation. Live channel diagnostics give mean $\sqrt{v(u)}$ estimates of 1.40–1.47 for PQC and 0.17–0.55 for CSB, compared with the ten-bit maximum 1.58 (Appendix D.4). They measure the variance factor, not the downstream Lipschitz constants. With a continuous Hessian bounded by H on a neighborhood of K, Taylor expansion replaces the first term by $Hv(u)/2$.

For a channel law with fixed finite support and positive differentiable probabilities near u, define $\ell_u(B) = \nabla_u \log P_{u,S}(B)$. Holding c fixed gives the exact identity

$$\nabla_u F - \nabla_u F_\mu = \mathbb{E}\big[\big\{G_s(B,c) - G_s(\mu,c) - \nabla_z G_s(\mu,c)^\top (B-\mu)\big\}\,\ell_u(B)\big]. \tag{11}$$

Under the Hessian condition its norm is at most $\frac{H}{2}\mathbb{E}\left[\|B-\mu\|_2^2 \|\ell_u(B)\|_2\right]$. Extending this comparison to θ also requires a bound on the bypass derivative over the *sampled support hull*: $\nabla_c G_s(\cdot, c)$ must be L_{cz}^{supp}-Lipschitz on K. This adds $M_c L_{cz}^{\text{supp}}\sqrt{v(u)}$ to the relaxation bound. The fitting-segment assumption in Theorem 1 alone does not imply this support-wide condition. Corollary 1 combines the fitting and relaxation contributions and states a sufficient local ascent condition.

For an affine map in the sampled features, the mean-feature relaxation is exact. For nonlinear maps, even a perfect surrogate can reverse the gradient (Appendix B.5). The source-generated tanh-map probes give median mean-feature/reference cosines of 0.994–1.000, while fitted-surrogate directions are mixed. Matched probes with random LeakyReLU and post-network–critic maps also have high median alignment. By contrast, two oscillatory maps with identical values on every binary output give eight-qubit median cosines of 0.752 and 0.294, and opposing directions occur at individual anchors (Appendix B.7.1). Thus the off-vertex geometry matters: neither a benign relaxation nor derivative fitting follows from BCE alone.

4.4 The straight-through direction and critic targets

Let $A = J_\theta u$ and $C = J_\theta c$. Assuming G_s is differentiable at the sampled feature points, the expected upstream STE update is

$$\mathbb{E}[\widehat{g}_\theta^{\text{STE}}] = A^\top J_f^\top \mathbb{E}[\nabla_z G_s(B, c)] + C^\top \mathbb{E}[\nabla_c G_s(B, c)]. \tag{12}$$

It is generally not $\nabla_\theta F$, $\nabla_\theta F_\mu$, or $\nabla_\theta F_f$; replacing only the backward pass is a different approximation (Appendix B.8). The critic is fitted on executed sampled actions, with exploration noise, but its target uses the relaxed actor. A value-gap bound also applies to this target under a Lipschitz assumption on the clipped, smoothed minimum of target critics. It does not establish Bellman consistency, critic accuracy, or convergence of the coupled training process (Appendix B.8).

5 Discussion and conclusion

Surrogate-assisted training produces policies that learn on four continuous-control tasks while using quantum measurements to select actions. On Humanoid, the hybrid evaluations show more high-return outcomes than the learned classical baselines, and the post-hoc learning-progress summaries show a consistent pattern. A compact circuit and surrogate also achieve the highest average return in the capacity sweep. The approximation analysis distinguishes errors in fitting the circuit response from those introduced by mean-feature and straight-through updates.

The trained networks also support deployment without the circuit. Using the stored surrogate to supply the features retains the same ten high-return Humanoid seeds without further fitting (Appendix E). This classical actor has the FCM architecture, so the follow-up compares two training procedures for a shared final architecture; the primary tables continue to report hybrid evaluations. Constant-half and fair-bit features retain nine high-return seeds, while zeros retain six. These substitutions measure dependence on the deployed features, but do not explain why surrogate-assisted hybrid training reaches high returns more often in these runs than end-to-end FCM training.

Limitations. The experiments use simulated circuits and finite seed cohorts. Capacity rankings are exploratory; reference re-executions share primary-run prefixes. PQC and CSB are not entropy-matched, and HalfCheetah shows substantial seed variability. The directional probes use fixed synthetic maps, not trained critics. The relation between estimated channel entropy and outcome is correlational: the controls were collected under the surrogate actor, and no training-time mechanism is identified. The learning results show that surrogate-based updates can train useful hybrid policies without requiring the larger configurations tested here.

Reproducibility statement

Appendix A illustrates the network and training flow; Appendix C documents the implementation, experimental protocols and data coverage; and Appendix D.1 specifies the statistical analysis. The theoretical assumptions and proofs are provided in Appendix B. Appendix B.7 describes the synthetic

probes, for which a complete repeat of all seven configurations agrees in every non-timing numerical field. The channel-characterization and downstream-map protocols are also detailed in the appendices.

We will make the code and analysis package available upon acceptance. The package will include run configurations, source hashes, per-seed scores, raw terminal evaluations, periodic training histories, control-level and probe-level records, and scripts for reproducing the tables and figures. Analysis of the recorded results requires no quantum simulator, and rerunning policy training requires the simulator and environment dependencies specified in Appendix C.

AI USE STATEMENT

The authors used generative AI to revise and polish the manuscript text, to assist with code review and debugging for the experiments and with producing code for selected supplementary studies in the appendix, and to help develop and check the mathematical arguments in the main text and appendices.

The authors have reviewed, verified and improved all AI-assisted work. The authors take full responsibility for the final content of this work, including the mathematical claims, the theoretical and experimental results, their verification and final presentation.

ETHICS STATEMENT

This work does not involve human subjects or personally identifiable data. Potential dual–use considerations arise from quantum computing research in general, but the presented method is limited to reinforcement–learning benchmarks and carries no direct security or safety risks. We note that training with large quantum simulations consumes substantial compute, and future work should continue to assess energy efficiency.

REFERENCES

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron Courville, and Marc G. Bellemare. Deep reinforcement learning at the edge of the statistical precipice, 2022. URL https://arxiv.org/abs/2108.13264.

Nuno Batista, Ana Morgado, Oscar Ferraz, Sagar Silva Pratapsi, Jorge Lobo, and Gabriel Falcao. Medical imaging classification with cold-atom reservoir computing using auto-encoders and surrogate-driven training. In *2025 IEEE International Conference on Quantum Artificial Intelligence (QAI)*, pp. 373–380. IEEE, November 2025. doi: 10.1109/qai63978.2025.00064. URL http://dx.doi.org/10.1109/QAI63978.2025.00064.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation, 2013. URL https://arxiv.org/abs/1308.3432.

Farama Foundation. Humanoid - Gymnasium documentation, 2026. URL https://gymnasium.farama.org/environments/mujoco/humanoid/.

Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1587–1596. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/fujimoto18a.html.

Will Grathwohl, Dami Choi, Yuhuai Wu, Geoffrey Roeder, and David Duvenaud. Backpropagation through the void: Optimizing control variates for black-box gradient estimation, 2018. URL https://arxiv.org/abs/1711.00123.

Max Jaderberg, Wojciech Marian Czarnecki, Simon Osindero, Oriol Vinyals, Alex Graves, David Silver, and Koray Kavukcuoglu. Decoupled neural interfaces using synthetic gradients. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1627–1635. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/jaderberg17a.html.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax, 2017. URL https://arxiv.org/abs/1611.01144.

Sofiene Jerbi, Casper Gyurik, Simon C. Marshall, Hans J. Briegel, and Vedran Dunjko. Parametrized quantum policies for reinforcement learning, 2021. URL https://arxiv.org/abs/2103.05577.

Yu-Xin Jin, Zi-Wei Wang, Hong-Ze Xu, Wei-Feng Zhuang, Meng-Jun Hu, and Dong E. Liu. Ppo-q: Proximal policy optimization with parametrized quantum policies or values, 2025. URL https://arxiv.org/abs/2501.07085.

Romerik Lokossou, Birhanu Shimelis Girma, Ozan K. Tonguz, and Ahmed Biyabani. Quantum deep reinforcement learning for humanoid robot navigation task, 2025. URL https://arxiv.org/abs/2509.11388.

Chris J. Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables, 2017. URL https://arxiv.org/abs/1611.00712.

Sreeraj Rajindran Nair and Christopher Ferrie. Local surrogates for quantum machine learning, 2025. URL https://arxiv.org/abs/2506.09425.

Emre O. Neftci, Hesham Mostafa, and Friedemann Zenke. Surrogate gradient learning in spiking neural networks, 2019. URL https://arxiv.org/abs/1901.09948.

Tom O'Leary, Piotr Czarnik, Elijah Pelofske, Andrew T. Sornborger, Michael McKerns, and Lukasz Cincio. Efficient online quantum circuit learning with no upfront training. *Communications Physics*, 8(1), November 2025. ISSN 2399-3650. doi: 10.1038/s42005-025-02423-4. URL http://dx.doi.org/10.1038/s42005-025-02423-4.

Franz J. Schreiber, Jens Eisert, and Johannes Jakob Meyer. Classical surrogates for quantum learning models. *Phys. Rev. Lett.*, 131:100803, Sep 2023. doi: 10.1103/PhysRevLett.131.100803. URL https://link.aps.org/doi/10.1103/PhysRevLett.131.100803.

Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.*, 122:040504, Feb 2019. doi: 10.1103/PhysRevLett.122.040504. URL https://link.aps.org/doi/10.1103/PhysRevLett.122.040504.

Maria Schuld, Ryan Sweke, and Johannes Jakob Meyer. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Phys. Rev. A*, 103:032430, Mar 2021. doi: 10.1103/PhysRevA.103.032430. URL https://link.aps.org/doi/10.1103/PhysRevA.103.032430.

Ryan Shaffer, Lucas Kocia, and Mohan Sarovar. Surrogate-based optimization for variational quantum algorithms. *Phys. Rev. A*, 107:032415, Mar 2023. doi: 10.1103/PhysRevA.107.032415. URL https://link.aps.org/doi/10.1103/PhysRevA.107.032415.

Sergey Shirobokov, Vladislav Belavin, Michael Kagan, Andrey Ustyuzhanin, and Atılım Güneş Baydin. Black-box optimization with local generative surrogates, 2020. URL https://arxiv.org/abs/2002.04632.

Andrea Skolik, Sofiene Jerbi, and Vedran Dunjko. Quantum agents in the Gym: a variational quantum algorithm for deep Q-learning. *Quantum*, 6:720, May 2022. ISSN 2521-327X. doi: 10.22331/q-2022-05-24-720. URL https://doi.org/10.22331/q-2022-05-24-720.

David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. General parameter-shift rules for quantum gradients. *Quantum*, 6:677, March 2022. ISSN 2521-327X. doi: 10.22331/q-2022-03-30-677. URL https://doi.org/10.22331/q-2022-03-30-677.

Yichen Xie. Quantum surrogate-driven image classifier: A gradient-free approach to avoid barren plateaus, 2025. URL https://arxiv.org/abs/2505.05249.

Siran Zhang and Shuming Cheng. Resource-efficient quantum approximate optimization algorithm via bayesian optimization and maximum-probability evaluation, 2026. URL https://arxiv.org/abs/2603.28413.

A NETWORK ARCHITECTURE AND TRAINING FLOW

The diagrams below show the complete hybrid actor and its surrogate-assisted training procedure. They use $\mathbf{q}_{\mathrm{i}}$ for the circuit controls u, $\mathbf{q}_{\mathrm{o}}$ for the aggregated bitstring B, and $\mathbf{q}_{\mathrm{o,pred}}$ for the surrogate prediction $f_\phi(u)$. The direct connections from PreDNN to PostDNN form the classical bypass c.

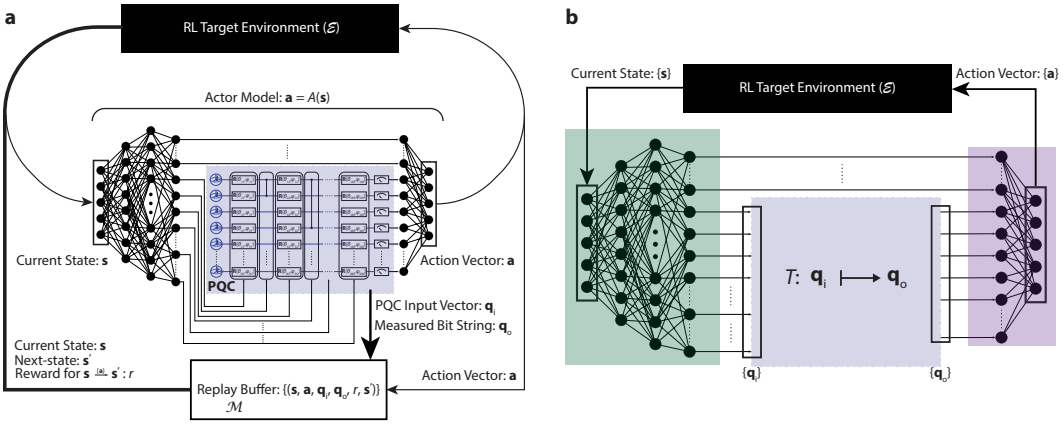


Figure 6: **Hybrid actor and interaction with the environment.** (a) PreDNN encodes the state, the circuit supplies measured features, and PostDNN produces the action. Replay stores the transition together with the circuit controls and aggregated output. (b) The modular view separates the classical networks from the middle module T. For the hybrid policy, T is the quantum channel during interaction and evaluation, and the surrogate during minibatch updates. Circuit connectivity is schematic; Appendix C specifies the implemented gate sequence.

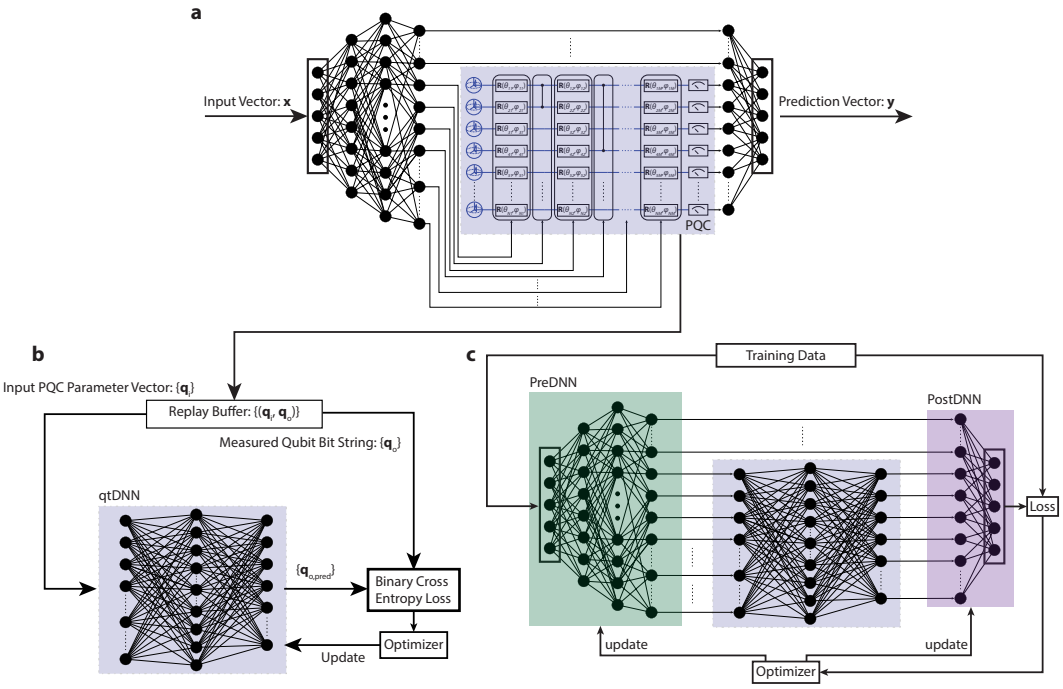


Figure 7: **Surrogate-assisted training within the full network.** (a) The hybrid network evaluates the circuit and records its response. (b) The surrogate learns from stored control–output pairs by binary cross-entropy. (c) The fitted surrogate is held fixed while gradients pass through it to update PreDNN and PostDNN. For the RL actor, the input x is the state s, the output y is the action a, and the outer loss is Eq. (5), evaluated on replay states. The bypass is used in both network paths.

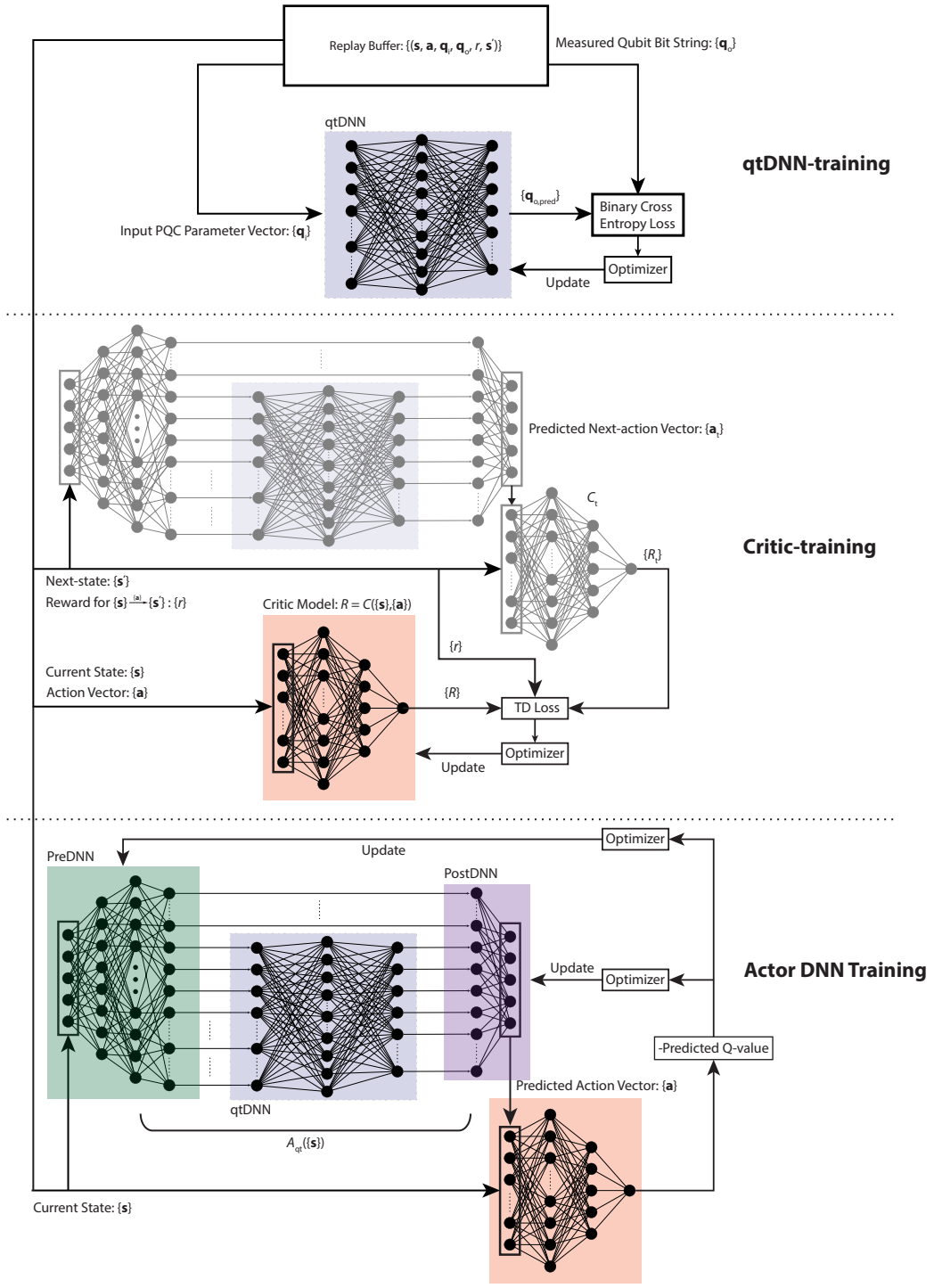


Figure 8: **Replay-based surrogate, critic, and actor updates.** Top: fit QTDNN to recorded circuit responses. Middle: use the target surrogate actor to construct critic targets, then update the critics from replayed transitions. Bottom: hold the surrogate and critic fixed and differentiate the actor objective through them to update PreDNN and PostDNN. The labels R and R_t denote critic estimates, not the reward r. The middle panel represents TD3's twin critics and their targets; the actor update uses Q_{ω_1}. Clipped-double targets, smoothing, terminal masks, and delayed updates follow Eqs. (4)–(5) and Algorithm 1.

B PROOFS AND ADDITIONAL ANALYSIS

We use Euclidean norms for vectors and induced operator norms for Jacobians. The state s, critic parameters ω, and post-network parameters ψ are fixed when differentiating with respect to the pre-network parameters θ, and we suppress s in G_s. Derivative statements apply within a fixed routing cell and away from network kinks. The second-order results additionally assume the stated Hessian bound on the relevant convex hull.

B.1 POPULATION TARGET OF THE BINARY CROSS-ENTROPY FIT

For fixed u and coordinate k, let $p = \mu_k(u)$ and $q = f_k(u)$. Conditioning on u gives

$$\mathbb{E}[-B_k \log q - (1 - B_k) \log(1 - q) \mid u] = -p \log q - (1 - p) \log(1 - q) \tag{13}$$

$$= h(p) + \mathrm{KL}(\mathrm{Bern}(p) \| \mathrm{Bern}(q)), \tag{14}$$

where $h(p) = -p \log p - (1 - p) \log(1 - p)$ uses natural logarithms and $0 \log 0 = 0$. Summing over k, dividing by N, and integrating over ρ proves the identity in Proposition 1. Because BCE is applied coordinatewise, the calculation depends only on the marginals and allows arbitrary correlations between bits.

For Bernoulli laws their total-variation distance is $|\mu_k - f_k|$. Pinsker's inequality therefore gives

$$2(\mu_k(u) - f_k(u))^2 \le \mathrm{KL}(\mathrm{Bern}(\mu_k(u)) \| \mathrm{Bern}(f_k(u))). \tag{15}$$

Summing over coordinates and integrating over ρ gives the norm bound. A finite neural network can also incur approximation and optimization error, while a replay sample introduces estimation error and possible distribution shift. The proposition assumes genuine channel observations. The warm-up placeholders described in Appendix C.5 fall outside that assumption.

Small prediction error does not control derivatives. To see why the gradient bound needs a separate Jacobian assumption, take the constant marginal $\mu(u) = 1/2$ on $[0, 1]$ and

$$f_\varepsilon(u) = \frac{1}{2} + \varepsilon \sin\left(\frac{u - 1/2}{\varepsilon^2}\right), \qquad 0 < \varepsilon \le \frac{1}{4}. \tag{16}$$

The uniform value error is at most ε, and the pointwise excess BCE is $-\frac{1}{2} \log(1 - 4\varepsilon^2 \sin^2((u - 1/2)/\varepsilon^2)) = O(\varepsilon^2)$ uniformly. Nevertheless, $f'_\varepsilon(1/2) = 1/\varepsilon$, while $\mu'(1/2) = 0$. Thus even uniformly vanishing excess BCE cannot replace a Jacobian-error hypothesis.

B.2 PROOF OF THE UPSTREAM-GRADIENT BOUND

Write $A = J_\theta u$ and $C = J_\theta c$. The chain rule, retaining both outputs of the pre-network, yields

$$\nabla_\theta F_f = A^\top J_f^\top \nabla_z G(f, c) + C^\top \nabla_c G(f, c), \tag{17}$$

$$\nabla_\theta F_\mu = A^\top J_\mu^\top \nabla_z G(\mu, c) + C^\top \nabla_c G(\mu, c). \tag{18}$$

Add and subtract $A^\top J_\mu^\top \nabla_z G(f, c)$. Their difference is exactly

$$\begin{aligned} \nabla_\theta F_f - \nabla_\theta F_\mu = {} & A^\top (J_f - J_\mu)^\top \nabla_z G(f, c) \\ & + A^\top J_\mu^\top [\nabla_z G(f, c) - \nabla_z G(\mu, c)] \\ & + C^\top [\nabla_c G(f, c) - \nabla_c G(\mu, c)]. \end{aligned} \tag{19}$$

Apply the triangle inequality and submultiplicativity. The three terms have norms at most $M_u \varepsilon_2 L_z$, $M_u M_\mu L_{zz} \varepsilon_1$, and $M_c L_{cz} \varepsilon_1$, respectively, proving Theorem 1.

For the post-network parameters, let G_ψ denote the same composition with its ψ-dependence explicit. If $\nabla_\psi G_\psi(z, c)$ is $L_{\psi z}$-Lipschitz in z, then, with u, c fixed,

$$\|\nabla_\psi G_\psi(f, c) - \nabla_\psi G_\psi(\mu, c)\|_2 \le L_{\psi z} \varepsilon_1. \tag{20}$$

The post-network bound has no Jacobian-of-f term because u and c are fixed for this parameter block. For a finite replay distribution, differentiating its finite sum and averaging the pointwise bounds is immediate. For a general fixed distribution, the same conclusion holds under a locally integrable dominating envelope for the relevant derivatives, which justifies exchanging differentiation and expectation. A policy-dependent state-occupancy measure would introduce additional derivatives that are outside this fixed-critic analysis.

B.3 VALUE BIAS CAUSED BY THE MEAN-FEATURE RELAXATION

Set $X = B - \mu$. Because $B_k \in \{0, 1\}$,

$$\mathbb{E}\Big[\|X\|_2^2\Big] = \sum_k \mathbb{E}\big[(B_k - \mu_k)^2\big] = \sum_k \mu_k(1 - \mu_k) = v(u) \le N/4. \tag{21}$$

No off-diagonal covariance terms occur in a squared Euclidean norm. If $G(\cdot, c)$ is L_z^{supp}-Lipschitz on $K = \mathrm{conv}(\mathrm{supp}\, P_{u,S})$, then

$$|\mathbb{E}[G(B, c)] - G(\mu, c)| \le L_z^{\mathrm{supp}} \mathbb{E}[\|X\|_2] \le L_z^{\mathrm{supp}} \sqrt{v(u)}. \tag{22}$$

For the second-order result, assume $G(\cdot, c)$ is twice continuously differentiable on an open neighborhood of K and $\sup_{z \in K} \left\|\nabla_z^2 G(z, c)\right\|_{\mathrm{op}} \le H$. Taylor's theorem on every segment from μ to a support point gives

$$G(B, c) = G(\mu, c) + \nabla_z G(\mu, c)^\top X + r_G(B, \mu), \qquad |r_G(B, \mu)| \le \frac{H}{2} \|X\|_2^2 . \tag{23}$$

The linear term has zero expectation. Taking expectations and using Eq. (21) proves $|F - F_\mu| \le Hv(u)/2$.

For Eq. (10), use a separate Lipschitz constant L_z^{fit} on $[f, \mu]$. Then

$$\begin{aligned} |F - F_f| &\le |F - F_\mu| + |G(\mu, c) - G(f, c)| \\ &\le L_z^{\mathrm{supp}} \sqrt{v(u)} + L_z^{\mathrm{fit}} \|f - \mu\|_2 . \end{aligned} \tag{24}$$

The pointwise bound on $\|\nabla_z G(f, c)\|_2$ in Theorem 1 alone does not supply this segment-wide constant. No Jacobian error enters the value comparison.

The actual bias depends on the downstream function and the full output law; the bound alone does not imply that it grows monotonically with N. For example, when $G(z, c)$ is affine in z, the value bias and fixed-c gradient bias both vanish wherever the derivatives exist.

B.4 SCORE IDENTITY AND A SUFFICIENT ASCENT CONDITION

Suppose the channel law has fixed finite support with positive probabilities in an open neighborhood of u, and these probabilities are differentiable. Finite summation justifies differentiation under the expectation. With $\ell_u(b) = \nabla_u \log P_{u,S}(b)$,

$$\mathbb{E}[\ell_u(B)] = 0, \qquad J_\mu = \mathbb{E}[B\ell_u(B)^\top] = \mathbb{E}[(B - \mu)\ell_u(B)^\top], \qquad \nabla_u F = \mathbb{E}[G(B, c)\ell_u(B)]. \tag{25}$$

Subtract the constant $G(\mu, c)\mathbb{E}[\ell_u(B)]$ and the linear contribution $J_\mu^\top \nabla_z G(\mu, c)$ to obtain Eq. (11). Equation (23) then implies

$$\|\nabla_u F - \nabla_u F_\mu\|_2 \le \frac{H}{2} \mathbb{E}\Big[\|B - \mu\|_2^2 \|\ell_u(B)\|_2\Big] \le \frac{H}{2} \sqrt{\mathbb{E}\Big[\|B - \mu\|_2^4\Big]\ \mathbb{E}\Big[\|\ell_u(B)\|_2^2\Big]}. \tag{26}$$

The score moment measures an additional sensitivity of the sampled law. It can be large even when BCE prediction error is small.

The bypass also contributes to the difference between the mean-feature and sampled objectives. Assume, in addition, that G is continuously differentiable in (z, c) on a neighborhood of $K \times \{c\}$ and that $\nabla_c G(\cdot, c)$ is L_{cz}^{supp}-Lipschitz on K. This is a condition on the sampled support hull, not just on the fitting segment $[f, \mu]$. Then

$$\|\mathbb{E}[\nabla_c G(B, c)] - \nabla_c G(\mu, c)\|_2 \le L_{cz}^{\mathrm{supp}} \mathbb{E}[\|B - \mu\|_2] \le L_{cz}^{\mathrm{supp}} \sqrt{v(u)}. \tag{27}$$

Combining this with the pre-network chain rule and Theorem 1 gives the following conditional extension.

Corollary 1 (From surrogate fitting to the sampled critic objective)**.** *Assume Theorem 1; a fixed finite-support channel law with positive differentiable probabilities on an open neighborhood of u; and the Hessian and bypass-Lipschitz conditions on the support hull K stated above. Let*

$$\Delta_{\mathrm{fit}} = M_u L_z \varepsilon_2 + (M_u M_\mu L_{zz} + M_c L_{cz})\varepsilon_1, \tag{28}$$

$$\Delta_{\mathrm{relax}} = \frac{M_u H}{2} \mathbb{E}\Big[\|B - \mu\|_2^2 \|\ell_u(B)\|_2\Big] + M_c L_{cz}^{\mathrm{supp}} \sqrt{v(u)}. \tag{29}$$

Then $\|\nabla_\theta F_f - \nabla_\theta F\|_2 \le \Delta_{\mathrm{fit}} + \Delta_{\mathrm{relax}}$. *If* $g = \nabla_\theta F \neq 0$ *and the right-hand side is strictly smaller than* $\|g\|_2$, *the surrogate gradient is a strict local ascent direction for* F.

Proof. The first assertion follows by inserting $\nabla_\theta F_\mu$ and applying the bounds already proved. If δ is the combined bound, Cauchy–Schwarz gives $\langle g, \nabla_\theta F_f \rangle \ge \|g\|_2^2 - \|g\|_2 \delta > 0$. Differentiability therefore implies improvement along this direction for sufficiently small positive step sizes. □

The strict inner product implies ascent of the fixed-critic objective for a sufficiently small Euclidean step that remains inside the differentiability neighborhood. The fitting and support-wide constants are assumptions of this local result, rather than quantities certified by the experiments. Adam updates, critic evolution, and changes in the state distribution are outside its scope.

B.5 Exact one-qubit counterexample and aggregation-aware differentiation

Example 1 (Zero value error with an opposing gradient)**.** *Prepare one qubit as* $R_y(\vartheta)|0\rangle$, *retain the majority of five shots, and let* $G(z) = z + \frac{1}{4}\sin(2\pi z)$. *At* $\vartheta = \pi/2$ *the exact aggregated marginal is* $\mu = 1/2$. *Hence* $F = F_\mu = 1/2$, *but*

$$F'(\pi/2) = 15/16 > 0, \qquad F'_\mu(\pi/2) = (1 - \pi/2)\, 15/16 < 0. \tag{30}$$

This failure occurs even with a perfect surrogate, $f = \mu$ *and* $J_f = J_\mu$.

In Example 1, the probability of measuring one in a single shot is $p(\vartheta) = \sin^2(\vartheta/2)$. The five-shot majority is one when at least three shots return one, giving

$$\mu(\vartheta) = 10p^3(1-p)^2 + 5p^4(1-p) + p^5 = 10p^3 - 15p^4 + 6p^5, \qquad \frac{\mathrm{d}\mu}{\mathrm{d}p} = 30p^2(1-p)^2. \tag{31}$$

At $\vartheta = \pi/2$, $p = 1/2$, $p' = 1/2$, and $\mu' = 15/16$. Since $G(0) = 0$ and $G(1) = 1$, the exact sampled objective is $F = \mu$. In contrast, $F_\mu = \mu + \frac{1}{4}\sin(2\pi\mu)$ has derivative

$$F'_\mu = \left[1 + \frac{\pi}{2}\cos(2\pi\mu)\right]\mu'. \tag{32}$$

Evaluated at the center, these derivatives are 0.9375 and approximately -0.53512. The scalar gradients have cosine -1, although the two objective values coincide and the surrogate is exact.

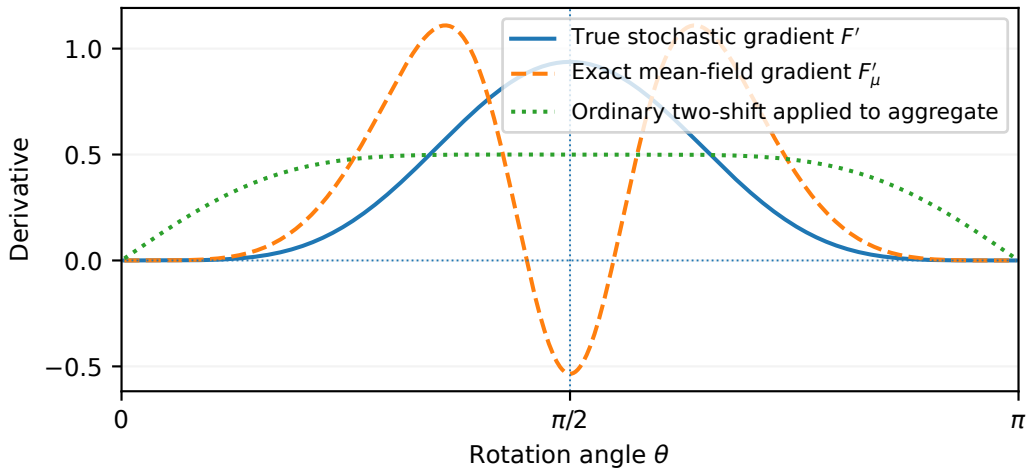


Figure 9: Derivatives in the one-qubit example. The sampled objective and its exact mean-feature relaxation have opposite gradients at $\vartheta = \pi/2$. Applying an ordinary two-shift expression to the aggregated objective gives a third, incorrect derivative. The curves are evaluated from the exact binomial polynomial.

Differentiating the shot-aggregated objective. For this example, $\frac{1}{2}[F(\pi/2+\pi/2) - F(\pi/2 - \pi/2)] = 1/2$, not $15/16$. Nonlinear shot aggregation changes the objective to which a differentiation rule must apply. The ordinary two-shift rule for a single-parameter Born expectation is therefore insufficient here. Generalized shift rules can handle richer Fourier spectra (Wierichs et al., 2022); alternatively, an estimator can be derived directly for the full shot-bag objective.

More generally, enumerate the $K = 2^N$ bitstrings. Under a fixed deterministic tie rule, the probability that mode aggregation returns b is

$$P_{u,S}(b) = \sum_{\substack{n_1+\cdots+n_K=S \\ A_S(n_1,\ldots,n_K)=b}} \frac{S!}{n_1!\cdots n_K!} \prod_{j=1}^{K} \pi_u(j)^{n_j}. \tag{33}$$

Within a fixed routing cell, Eq. (33) is a homogeneous polynomial of degree S in the single-shot probabilities, possibly of lower degree on the probability simplex. For the implemented circuit, each rotation control u_j occurs once, as one angle of $U_3(2\pi u_j, \cdot, 0)$ or $U_3(\cdot, 2\pi u_j, 0)$. Holding other controls fixed, the single-shot probability of an outcome has the form

$$\pi_u(k) = a_k + b_k \cos(2\pi u_j) + c_k \sin(2\pi u_j). \tag{34}$$

For the first angle, the gate is a fixed unitary times $R_y(2\pi u_j)$; conjugating a density matrix produces only constant, sine and cosine terms. For the second, the gate contains a diagonal relative phase with the same frequency. Fixed surrounding gates and parameter-independent noise are linear maps on the density matrix, so they preserve this coordinatewise degree. Consequently $P_{u,S}(b)$ has Fourier degree at most S in each rotation control. For fixed G and c, $F(u, c)$ inherits this degree; $G(\mu(u), c)$ need not. Aggregation can therefore invalidate the ordinary two-shift rule even though it is valid for the single-shot probabilities. Generalized shift rules can exploit the larger spectrum (Wierichs et al., 2022); this is not a lower bound on estimation cost.

The channel score can also be related directly to circuit probabilities. On an interior region where all $\pi_u(k) > 0$, let $w_u(n)$ be the multinomial weight in Eq. (33). Finite differentiation gives

$$\nabla_u P_{u,S}(b) = \sum_{n:A_S(n)=b} w_u(n) \sum_{k=1}^{2^N} n_k \frac{\nabla_u \pi_u(k)}{\pi_u(k)}, \qquad \ell_u(b) = \frac{\nabla_u P_{u,S}(b)}{P_{u,S}(b)}. \tag{35}$$

Each rotation derivative of π_u is available by a suitable shift rule; routing derivatives vanish locally. At zero probabilities, one can differentiate the polynomial directly rather than use the displayed divisions. This is an in-principle expression, not a proposal to enumerate all shot-count vectors during RL training. It identifies aggregation sensitivity absent from the single-shot marginal Jacobian.

We evaluated the closed-form one-qubit expressions at 801 angles. Central finite differences with step 10^{-6} agree with the analytic derivative to below 2.5×10^{-10}. We also checked the finite-support value and gradient identities, including the bypass, and verified that five-shot aggregation can generate frequency five. These numerical checks test the analytic expressions; they do not establish the assumptions along a trained RL policy.

B.6 Computational complexity

The surrogate removes update-time quantum queries but adds classical computation. A single-hidden-layer surrogate with width h, input dimension d, and output dimension N has $(d + N + 1)h + N$ parameters and costs order $h(d + N)$ per dense evaluation. Fitting it for K_{fit} inner steps with sub-batch size b costs order $K_{\text{fit}} bh(d + N)$ per outer update, in addition to the actor and critic computations. These operation counts do not predict end-to-end runtime. The primary configuration uses $h = 2048$ at $N = 10$; a width tied to 2^{N+1} is a possible design choice rather than a requirement. The capacity sweep treats width and qubit count as independent choices.

For an appropriate unaggregated expectation, a naive two-shift estimate with $I = 2NM$ continuous gate angles, batch size b_{RL}, and S shots per circuit would use $2Ib_{\text{RL}}S$ shots. Setting $I = 200$, $b_{\text{RL}} = 256$, and $S = 1000$ gives 102.4 million shots. This illustrates the cost of that particular estimator, not a lower bound or a measured baseline. Its validity for a Born expectation also does not make it valid for mode aggregation. Our measured comparison is with hard-forward STE (Section 3.4).

B.7 SYNTHETIC PROBES OF FITTING AND RELAXATION

This study uses exact CPU simulation of four-, six- and eight-qubit circuits and a fixed random smooth map G, not a trained RL critic. The map is a width-64 tanh network with scalar output scale three. It is fixed within each qubit count. At each synthetic control anchor, we compare the sampled objective $F(u) = \mathbb{E}[G(B)]$, the aggregated-mean relaxation $F_\mu(u) = G(\mu(u))$, and the Born-mean substitute $G(p(u))$, where $p(u) = \mathbb{E}_{\pi_u}[Z]$. Tables 4-5 summarize the seven synthetic configurations described below. We repeated the complete study with the same implementation and settings; all non-timing probe values agree between executions. The software environment and source hashes were recorded for both executions.

Protocols and reference directions. Cold fits use 256 local control–output pairs, with Gaussian control perturbations of standard deviation 0.05, clipped to $[0, 1]^d$, and one shot-aggregated output per control. A model is evaluated after 32 optimization steps and again after 500 cumulative steps. The latter continues the same fit. The product surrogate uses width 2^{N+1} or 64, and the autoregressive surrogate uses width 2^{N+1}. The persistent protocol uses six qubits and width 128, retaining parameters over 12 drifting batches. Each batch adds 64 pairs and performs 32 fitting steps, for 768 pairs and 384 optimization steps in total. It is a growing synthetic replay, not the full RL replay process.

Each anchor uses eight normalized directions in the rotation-control subspace, displacement $h = 0.05$, and 4000 common-random-number (CRN) shot-bag replicates at each endpoint. For direction v_j, the endpoints are $u_j^\pm = \mathrm{clip}_{[0,1]^d}(u \pm h v_j)$. Let $\hat{\mu}_\pm$ and $\hat{F}_\pm$ denote the empirical aggregated means and sampled-objective values at these endpoints. The compared reference vectors have components

$$a_j^F = \frac{\widehat{F}_+ - \widehat{F}_-}{2h}, \qquad a_j^\mu = \frac{G(\widehat{\mu}_+) - G(\widehat{\mu}_-)}{2h}, \tag{36}$$

$$a_j^{\mathrm{Born}} = v_j^\top J_p(u)^\top \nabla_z G(p(u)). \tag{37}$$

The Born Jacobian is computed by an exact ordinary parameter-shift rule on the CPU simulator. The fitted direction uses coordinate finite differences of $G(f(u))$ with step 10^{-3}, with the same endpoint clipping, and is projected onto the eight directions. Both finite-difference calculations retain their nominal denominators after clipping. When clipping is active, these differences compare projected endpoint values rather than ordinary unprojected directional derivatives.

$N/M/S$	Protocol	Defined	F_μ FD	Born PS	Fit 32/384	Fit 500
4/6/1000	Cold	7/10	0.998	0.838	0.092	0.090
6/10/1000	Cold	10/10	0.998	0.196	-0.079	-0.062
6/6/100	Cold	10/10	0.994	0.808	-0.056	-0.086
6/6/1000	Cold	9/10	0.995	0.386	-0.232	-0.159
6/6/1000	Persistent	10/10	0.999	0.642	-0.081	—
6/6/100	Persistent	10/10	0.998	0.886	0.269	—
8/6/1000	Cold	5/5	1.000	0.557	0.224	0.113

Table 4: Median directional cosines against the CRN sampled-objective reference a^F. F_μ FD uses the aggregated-mean reference, and Born PS uses the single-shot mean and its exact shift Jacobian. Fit columns use the product surrogate with width 2^{N+1}. The first fit column is 32 steps for cold fits and 384 for persistent fits. "Defined" counts distinct anchors, not duplicated reference entries across model variants.

Estimated aggregate-mean directions align closely in these fixed-G probes: median cosines range from 0.994 to 1.000 across the seven settings. Ordinary Born-mean directions have lower medians, from 0.196 to 0.886, with negative individual cosines in several settings. The single-shot Jacobian is exact for its own target but does not automatically differentiate the shot-aggregated objective. This is the circuit-specific distinction illustrated analytically in Appendix B.5.

Fitted-surrogate directions are more variable. At four qubits, product width 32 gives medians 0.092 and 0.090 after 32 and 500 steps; product width 64 gives -0.048 and 0.162, and autoregressive width 32 gives -0.032 and 0.155. Six-qubit persistent fits give -0.081 at 1000 shots and 0.269 at 100 shots. These observations distinguish benign relaxation for this particular G from successful

derivative fitting. They neither certify alignment of the trained RL policies nor contradict their measured learning curves.

$N/M/S$	Surrogate	Width	Defined	32/384 steps	500 steps
4/6/1000	ar	32	7/10	-0.032	0.155
4/6/1000	product	32	7/10	0.092	0.090
4/6/1000	product	64	7/10	-0.048	0.162
6/10/1000	ar	128	10/10	0.047	-0.014
6/10/1000	product	64	10/10	0.097	-0.025
6/10/1000	product	128	10/10	-0.079	-0.062
6/6/100	ar	128	10/10	0.056	0.287
6/6/100	product	64	10/10	0.081	0.238
6/6/100	product	128	10/10	-0.056	-0.086
6/6/1000	ar	128	9/10	-0.114	0.092
6/6/1000	product	64	9/10	-0.023	0.131
6/6/1000	product	128	9/10	-0.232	-0.159
6/6/1000	product (persistent)	128	10/10	-0.081	—
6/6/100	product (persistent)	128	10/10	0.269	—
8/6/1000	ar	512	5/5	0.038	0.009
8/6/1000	product	64	5/5	0.027	0.068
8/6/1000	product	512	5/5	0.224	0.113

Table 5: Every fitted surrogate setting in the synthetic study. Entries are median cosines with a^F. The first budget is 32 steps for cold fits or 384 for persistent fits. Three four-qubit reference vectors and one six-qubit, six-layer, 1000-shot cold reference vector vanish, making 18 and six model/budget rows undefined. The coverage denominators include these undefined cases; the medians use only defined cosines. Other configurations have ten defined anchors, except the five-anchor eight-qubit study.

Each cold-protocol control anchor is shared by six model-budget combinations. Reference-direction summaries count each anchor once, so a ten-anchor study contributes ten reference probes rather than 60. Shared CRN noise also couples the estimated mean and sampled directions. High cosine does not imply matching gradient magnitudes or an accurate uniform bound. The counterexample and the probe observations concern different downstream maps: together they show why the relaxation must be assessed for the map actually used.

$N/M/S$	Born PS evaluations	PS seconds / control	32-step fit seconds
4/6/1000	96	0.184	0.004
6/10/1000	240	3.640	0.007
6/6/100	144	1.014	0.006
6/6/1000	144	0.985	0.006
8/6/1000	192	25.900	0.018

Table 6: Recorded CPU microbenchmarks from the cold synthetic studies. The shift computation is an exact single-shot marginal Jacobian for one control vector; the fit uses 256 stored pairs and 32 optimizer steps. They have different outputs and accuracies. These are not RL runtimes or timings of an aggregation-corrected gradient estimator.

The single-shot marginal Jacobian computation uses 240 probability evaluations and takes a median 3.640 seconds per control for six qubits and ten layers; the corresponding values for eight qubits and six layers are 192 evaluations and 25.900 seconds. Corresponding short fits take 0.007 and 0.018 seconds. The timings use single-threaded numerical kernels in a shared four-vCPU environment and include ordinary process contention. They are descriptive measurements, not isolated hardware benchmarks, and are not transferred to the cluster runtime comparisons.

B.7.1 Dependence on the downstream map

For each qubit count, the controls, shot draws, replay data, fitting budgets and projection directions are the same as in the corresponding tanh experiment. The fitted surrogates are also held fixed across downstream maps.

The first alternative has a width-256 LeakyReLU hidden layer, slope 0.01, and a linear scalar head. The second uses the paper's post-network–critic composition: a width-256 LeakyReLU post-network maps the bit features and a fixed bypass to 17 tanh actions in $[-0.4, 0.4]$; a critic combines a width-256 state encoding with the actions, applies another width-256 LeakyReLU layer, and outputs a scalar. Xavier-uniform weights and zero biases match the initialization convention. The 376-dimensional standard-normal state and uniform bypass vector are fixed synthetic draws. This reproduces the architecture, not learned weights or a trained policy's state distribution. The NumPy composition is checked against the actual PyTorch building blocks in `hdqnn/networks.py`.

The remaining alternatives are controlled oscillatory maps,

$$G_{a,k}(z) = \frac{3}{N}\sum_{j=1}^{N}\big[z_j + a\sin(2\pi k z_j)\big], \qquad (a,k) \in \{(1/4,1),(1/2,2)\}. \tag{38}$$

For integer k, both maps agree on every binary feature vector: $G_{a,k}(B) = 3N^{-1}\sum_j B_j$. Their sampled objective is therefore unchanged, whereas the mean-feature objective depends on a and k. The Hessian norm is bounded by $3a(2\pi k)^2/N$. This tests the effect of off-vertex curvature without changing the map's values on deployed features, generalizing the mechanism in Example 1.

N	Downstream map	Defined	Median [IQR]	Minimum	< 0
4	Tanh	7/10	0.998 [0.982, 0.999]	0.913	0
4	LeakyReLU	7/10	0.999 [0.995, 1.000]	0.961	0
4	Post–critic	7/10	0.997 [0.996, 0.998]	0.995	0
4	Sine ($a = 1/4, k = 1$)	7/10	0.972 [0.964, 0.975]	0.950	0
4	Sine ($a = 1/2, k = 2$)	7/10	0.837 [0.620, 0.881]	-0.539	1
6	Tanh	9/10	0.995 [0.992, 1.000]	0.749	0
6	LeakyReLU	9/10	0.996 [0.985, 1.000]	0.811	0
6	Post–critic	9/10	0.993 [0.979, 0.999]	0.949	0
6	Sine ($a = 1/4, k = 1$)	9/10	0.910 [0.906, 0.997]	0.774	0
6	Sine ($a = 1/2, k = 2$)	9/10	0.656 [0.503, 0.788]	-0.984	1
8	Tanh	5/5	1.000 [1.000, 1.000]	0.998	0
8	LeakyReLU	5/5	0.993 [0.989, 0.994]	0.942	0
8	Post–critic	5/5	0.968 [0.965, 0.993]	0.176	0
8	Sine ($a = 1/4, k = 1$)	5/5	0.752 [0.728, 0.860]	-0.775	1
8	Sine ($a = 1/2, k = 2$)	5/5	0.294 [0.169, 0.681]	0.025	0

Table 7: Dependence of the mean-feature relaxation on the downstream map. Each row summarizes distinct matched control anchors, comparing the finite-difference vectors a^{μ} and a^{F}. "< 0" counts opposing estimated directions among defined cosines. The reported coverage includes undefined cases; cosine summaries use only defined directions. These are synthetic fixed-map probes, not measurements of trained RL gradients.

Across the nine random-map/qubit settings, median cosines range from 0.968 to 1.000. The stronger oscillatory map has medians 0.837, 0.656 and 0.294 at four, six and eight qubits, respectively, with negative individual cosines at four and six qubits. The lower-frequency sine map also has an opposing eight-qubit estimate. The random-map probes and the oscillatory probes test different downstream geometries. The latter can produce opposing directions despite using the exact same sampled objective. Because this comparison uses aggregated-marginal references rather than fitted surrogates, its discrepancy concerns the relaxation and reference estimation, not surrogate fitting. Conversely, high alignment for one fixed map does not establish a bound for a learned actor and critic. All reference directions retain the original 4000-bag Monte Carlo and finite-difference errors, including shared CRN noise.

B.8 Straight-through updates and relaxed critic targets

Assume G is differentiable at each sampled feature point and that f, u_θ, and c_θ are differentiable at the inputs under consideration. For a sampled feature B, the automatic-differentiation expression $z = B + f_\phi(u) - \mathrm{sg}(f_\phi(u))$ has forward value B and assigned input derivative J_f. Holding ϕ, ψ, the critic and the replay state fixed gives

$$\widehat{g}_\theta^{\mathrm{STE}} = A^\top J_f^\top \nabla_z G(B,c) + C^\top \nabla_c G(B,c). \tag{39}$$

Taking the finite-support expectation proves Eq. (12). Under the fixed-support and differentiability assumptions of Appendix B.4, the corresponding objective gradients are,

$$\nabla_\theta F = A^\top \mathbb{E}[G(B,c)\ell_u(B)] + C^\top \mathbb{E}[\nabla_c G(B,c)], \tag{40}$$

$$\nabla_\theta F_\mu = A^\top J_\mu^\top \nabla_z G(\mu,c) + C^\top \nabla_c G(\mu,c). \tag{41}$$

Even $J_f = J_\mu$ does not make $J_\mu^\top \mathbb{E}[\nabla_z G(B,c)]$ equal to either fixed-c contribution in general. In Example 1, exact-mean STE gives $(1+\pi/2)15/16$ at $\vartheta = \pi/2$, whereas $F' = 15/16$ and $F'_\mu = (1-\pi/2)15/16$. The implemented stop-gradient control detaches both feature paths and never steps the pre-network optimizer. Its comparison with STE concerns encoder and bypass learning together.

The target substitution. For fixed next state, target networks and smoothing draw, write

$$H_{s',\epsilon}(z,c) = \min_{j=1,2} Q_{\bar{\omega}_j}\big(s', \mathrm{clip}_{\mathcal{A}}[\mathrm{Post}_{\bar{\psi}}(z,c)+\epsilon]\big). \tag{42}$$

Here $u = u_{\bar{\theta}}(s')$, $c = c_{\bar{\theta}}(s')$, $B \sim P_{u,S}$, and $f = f_{\bar{\phi}}$; all target-network parameters are held fixed.

If this function is Lipschitz in z with constants L_H^{supp} and L_H^{fit} on the sampled support hull and fitting segment, respectively, then

$$|\mathbb{E}_B[H_{s',\epsilon}(B,c)] - H_{s',\epsilon}(f(u),c)| \le L_H^{\mathrm{supp}}\sqrt{v(u)} + L_H^{\mathrm{fit}}\,\|f(u)-\mu(u)\|_2\,. \tag{43}$$

Multiplication by $\gamma(1-d_{\mathrm{term}})$ gives the conditional target-value gap. An integrable dominating bound permits averaging over smoothing noise and replay. A finite minimum and action clipping preserve suitable Lipschitz bounds, although they can introduce kinks that preclude a Hessian argument. Critic regressors are trained on executed sampled actions, with exploration noise, whereas the target action is relaxed. This local value statement neither removes that distribution mismatch nor proves that either critic equals the deployed policy's action value.

C Implementation and experimental protocol

This appendix specifies the implementation used for the reported experiments, including its evaluation schedule and checkpoint behavior. The network and training diagrams are in Appendix A.

C.1 Circuit and neural-network architecture

The quantum circuit starts in $|+\rangle^{\otimes N}$. The first $2NM$ sigmoid outputs are multiplied by 2π and used as pairs of angles for $U_3(\alpha,\beta,0)$. At each layer, these gates are applied to all qubits before a single controlled-Z gate. The last $2M$ controls select its two endpoints by $\mathrm{rint}((N-1)u_j)$; NumPy's nearest-even rounding applies at exact half-integers. If the two endpoints coincide, the controlled-Z gate is omitted. Both bit-flip and phase-flip noise with probability 10^{-3} are attached to each U_3 gate. Computational-basis measurement is followed by full-bitstring mode aggregation through CUDA-Q's `counts.most_probable`. The backend's resolution of tied modes is not recorded in the run manifests. The returned string is converted to an N-dimensional vector of zeros and ones.

The pre-network has one width-256 LeakyReLU hidden layer and a sigmoid output of dimension $d+N$. Its last N outputs form the bypass. The post-network concatenates the N channel features with the N bypass features, uses a width-256 LeakyReLU layer, and produces a tanh action mapped to the environment's action bounds. Each of the two critics first transforms the state through a width-256 layer, concatenates the action, and applies another width-256 layer before its scalar output. The hidden LeakyReLU slope is 0.01. Linear layers use Xavier-uniform weight initialization and zero biases.

The default surrogate has a width-2048 LeakyReLU hidden layer and N sigmoid outputs. With $N = M = 10$, $d = 220$ and it has $(220+10+1)2048+10 = 473{,}098$ parameters. In the base hybrid deployment, the surrogate is used only during training; Appendix E evaluates it as an alternative readout. We count its parameters separately from those of the deployed actor, since it contributes to training cost but not to action selection.

Setting	Value or convention
Primary task / nominal budget	Humanoid-v4 / 3,000,000 environment steps
Training seeds	15 per primary bottleneck; 3 per width; 5 per ablation
Quantum default	$N = 10$, $M = 10$, $S = 1000$
Replay / minibatch	Capacity 10^6 / 256, uniform sampling
Surrogate fit	32 Adam steps, sub-batches of 16 with replacement
Learning rates	3×10^{-4} for pre, post, surrogate, and critics
Gradient clipping	Disabled (all configured clipping values are $+\infty$)
Discount / target rate / actor delay	0.99 / 0.005 / 2 critic updates
Warm-up	1001 random actions, due to the inclusive index condition
Exploration noise	Standard deviation 0.05 times the full action range
Target smoothing	Standard deviation 0.2 times first-coordinate half-range
Target noise clipping	$[-0.5, 0.5]$ in absolute action units
Periodic / terminal evaluation	10 / 30 episodes, no external exploration noise
Maximum episode length	1000 environment steps

Table 8: Hyperparameters and implementation conventions for the primary experiments. Environment-step budgets count training interactions only.

C.2 Training order and target construction

Algorithm 1 Replay-trained surrogate-forward TD3 (base quantum configuration)

Initialize pre-network θ, post-network ψ, surrogate ϕ, twin critics ω_1, ω_2, their targets, optimizers, and replay $\mathcal{D}$.
for each environment interaction **do**
  if inside random-action warm-up **then**
    Draw a random action; record zero placeholders for u and B.
  else
    Compute $(u, c) = \mathrm{Pre}_\theta(s)$; query $B \sim P_{u,S}$.
    Act with $\mathrm{clip}_{\mathcal{A}}(\mathrm{Post}_\psi(B, c) + \xi)$.
  end if
  Observe $(r, s', d_{\mathrm{term}})$ and store $(s, u, B, a, r, s', d_{\mathrm{term}})$ in $\mathcal{D}$.
  if warm-up is over and replay has more than 256 entries **then**
    Draw a uniform minibatch $\mathcal{M}$ of 256 transitions.
    Fit ϕ by 32 BCE steps on size-16 resamples of (u, B) in $\mathcal{M}$; set $\bar{\phi} \leftarrow \phi$.
    Construct the surrogate-forward target y in Eq. (4), without channel queries.
    Update both critics by squared-error regression to the detached target.
    if this is every second critic update **then**
      Update θ, ψ using Eq. (5), keeping ϕ fixed but differentiating through its inputs.
      Polyak-update pre, post, and both critic targets with rate τ.
    end if
  end if
  Evaluate at the scheduled crossings using the sampled deployed channel, without exploration noise.
end for

After warm-up, each interaction is followed by an update. The bootstrap multiplier is zero for terminal transitions but remains active at time-limit truncations. We hard-copy the surrogate target after every fit and Polyak-update the pre-network, post-network, and critic targets on the delayed actor schedule. Autograd can accumulate gradients for the surrogate parameters during the actor backward pass; its optimizer is not stepped at that stage, and the buffers are cleared before the next fit.

The exploration noise standard deviation is $0.05(a_{\max} - a_{\min})$ componentwise. Target smoothing instead uses the scalar $0.2(a_{\max,1} - a_{\min,1})/2$ for all action coordinates, then clips the noise in absolute action units. All compared configurations use these same noise conventions.

C.3 CONTROL CONFIGURATIONS

FC and FCM. For FC the bottleneck is a learned linear map followed by a sigmoid. FCM adds the same width-2048 hidden layer as the quantum surrogate. Both use the deterministic module for interaction and updates, training it directly with the actor objective and Polyak-averaging its target parameters. FCM matches the surrogate's architecture and parameter count; its action-selection cost differs because it executes a classical network rather than a sampled circuit. Neither control requires an auxiliary BCE fit.

CSB. The classical stochastic bottleneck uses fixed random matrices in a $d \to 256 \to N$ network, with tanh hidden units and sigmoid output logits scaled by gain 3. The channel-construction seed is fixed at 20260611 across training seeds. Its private shot generator is also initialized with the same constant, 20260612, in every training seed, and its state is not restored from checkpoints. Each shot draws independent Bernoulli coordinates conditional on the controls; the same full-bitstring mode operation is applied to S shots. Mode aggregation can introduce dependencies between output bits even though the individual shots have a product distribution. CSB uses the same surrogate-forward protocol as PQC. Its output law is not entropy-matched: the conditional entropy estimates and variance coefficients in Appendix D.4 are substantially smaller than for PQC. This is a comparison of different stochastic feature maps, not a controlled intervention on quantum correlations alone.

RBG and stop-gradient. RBG returns control-independent random bits and disables upstream learning, including gradients through the bypass. Its fixed random pre-network still provides state-dependent bypass features to the trainable post-network. It is therefore a control for this full frozen-encoder architecture, rather than an isolated test of quantum randomness or entanglement. The hard-forward stop-gradient variant samples the control-dependent quantum channel but detaches both the sampled features and the bypass; the pre-network optimizer is not stepped. Four runs supply the matched-budget comparison in Section 3.4. The surrogate is still fitted from replay, but it supplies no policy-update gradient in this control.

Hard-forward STE. The actor update and critic-target computation both request sampled channel features. The expression $B + f_\phi(u) - \mathrm{sg}(f_\phi(u))$ then supplies the surrogate derivative during backpropagation. Every minibatch element requires a channel call, giving 384 extra calls per update on average at batch size 256 and actor delay two. Switching to surrogate-forward changes the forward objective as well as the query count, so return differences reflect both changes. Batching and parallel execution can change runtime but leave this control-setting count unchanged.

C.4 EVALUATION, TERMINATION, AND CHECKPOINTS

The training loop stops at the first episode boundary after reaching 3M environment steps. The overshoot is at most one episode. A step budget keeps sample counts comparable even when policy quality changes episode length. Evaluation interactions are counted separately.

A ten-episode evaluation occurs on the first completed episode and when the cumulative training count crosses a multiple of $10\times$ the maximum episode length (10,000 steps). A dedicated 30-episode terminal evaluation is scheduled at the end of each completed run. One scalar-sensitivity-regularized run has only ten recorded terminal episodes; Appendix D.1 identifies this exception. Environment reset seeds are generated by $\mathrm{episode_seed}(b, e) = (1000003b + e) \bmod (2^{31} - 1)$; different evaluation episodes use different values. Evaluation disables external exploration noise while retaining the stochastic finite-shot channel.

Elapsed-time counters include simulation, learning, and periodic evaluation, but exclude scheduler waiting. Resumed runs restore the previously logged elapsed time. The last metrics row precedes the separate terminal evaluation, whereas the final channel counter includes it. We therefore use time and call counts from the same periodic checkpoint when comparing update strategies. Matching integer seeds alone does not ensure matched hardware.

C.5 Reproducibility and data availability

Warm-up observations. Initial random actions are stored with $(u, B) = (0, 0)$ placeholders. The surrogate fit does not mask these records, so early BCE updates can be biased by labels that are not genuine channel observations. There are 1001 such records, occupying approximately 1% of replay at 100,000 interactions. With capacity 10^6, they remain until individually overwritten after replay fills. All reported training runs use this unmasked warm-up procedure.

Randomness and restarts. Environment, PyTorch, NumPy, and replay random streams are seeded and checkpointed. CUDA-Q's shot generator is not explicitly seeded or checkpointed, and the private CSB/RBG generator state is absent from model checkpoints. The reported seed labels therefore identify repeated training runs without guaranteeing bitwise reproducibility, especially after a restart.

Hardware and software records. Training uses PyTorch, Gymnasium/MuJoCo, and CUDA-Q. The target-selection code prefers the NVIDIA simulator when a GPU is available; strict mode stops on failure rather than falling back to another backend. The run manifests do not record every run's GPU model and software versions. Runtime comparisons are therefore reported as observed process-time ratios, without device normalization.

Experimental records. Our experimental records contain 159 terminal evaluations. All 75 primary runs have 30 distinct episode returns and complete periodic evaluation histories. Across all experiments, 158 terminal evaluations contain 30 episodes; scalar-sensitivity-regularization seed 1 contains ten. For every run, the mean calculated from the episode returns agrees with the recorded summary. We distinguish completed terminal evaluations, partial histories and same-seed re-executions, and do not count re-executions as additional independent primary runs. Missing evaluations are not assigned zero returns. Appendix D.7 details the shared training histories.

Run completion. The primary Humanoid comparison is complete. Some additional-task and update-path runs were still in progress when we finalized the results for this submission. We report these experiments at the interaction budgets reached by the stated seed cohorts, using only completed evaluations. Each aggregate learning curve uses a fixed cohort, and individual trajectories end at their last recorded evaluation. Runs that have not reached a comparison budget do not contribute to its statistics.

Synthetic-study reproducibility. We executed all seven small-circuit configurations twice with the same implementation and settings. Every non-timing probe value agrees between executions. Reference-direction summaries count each control anchor once, and undefined cosines are excluded from the summaries with their coverage reported explicitly. The channel-characterisation and downstream-map studies have separate protocols and control-level or probe-level results. These CPU studies are distinct from the MuJoCo policy-training experiments. The code and analysis package will be released upon acceptance, including the implementations, configurations, recorded software information and source hashes, both synthetic-study executions, and their numerical comparison.

Environment versions. Run configurations fix Humanoid-v4 and Hopper-v4, and Ant-v5 and HalfCheetah-v5. These IDs specify separate task implementations, not interchangeable labels. In particular, Gymnasium's version history documents changes to observations, reward handling and contact costs between Humanoid-v4 and v5 (Farama Foundation, 2026). All within-task comparisons use the same ID; we neither pool versions nor treat the additional tasks as a matched cross-version benchmark.

D Additional experiments and statistical analysis

D.1 Scores, coverage and uncertainty estimates

For run i with n_i terminal evaluation episodes, the score is $X_i = n_i^{-1} \sum_{j=1}^{n_i} R_{i,j}$. All 75 primary runs have $n_i = 30$ with distinct episode scores. Across the 159 terminal evaluations, 158 contain 30 scores and scalar-sensitivity-regularization seed 1 contains ten. Training runs, not individual evaluation episodes, are the statistical units.

The data release will include per-episode terminal returns, periodic histories, run configurations, and an inventory distinguishing completed runs, partial histories and same-seed re-executions.

For ordered scores $X_{(1)} \leq \cdots \leq X_{(n)}$, let $k = \lfloor n/4 \rfloor$. Our finite-sample interquartile-mean convention is

$$\widehat{\mathrm{IQM}} = \frac{1}{n-2k} \sum_{i=k+1}^{n-k} X_{(i)}. \tag{44}$$

It retains nine central scores when $n = 15$. Pointwise 95% percentile intervals use 20,000 resamples of training runs, with analysis RNG seed 270913 for each statistic. They quantify between-run variation on one task; they are neither simultaneous nor multitask intervals. We follow the emphasis on uncertainty and seed-level distributions in Agarwal et al. (2022).

The probability-of-improvement estimate between configurations A and B is

$$\widehat{p}_{A>B} = \frac{1}{n_A n_B} \sum_{i,j} \left[\mathbf{1}\{X_i^A > X_j^B\} + \frac{1}{2}\mathbf{1}\{X_i^A = X_j^B\} \right]. \tag{45}$$

We bootstrap the two groups separately. Matching seed labels do not establish a paired common-random-number design across different architectures. Terminal-score tests are exact, two-sided Mann–Whitney tests; the data have no ties. Holm adjustment is applied to the four comparisons against PQC. Welch tests and Fisher exact tests at 1500 and 4000 form separate four-comparison families. Both thresholds are descriptive; no pre-registration is claimed. Width, shot and surrogate-family comparisons are exploratory.

PQC vs.	PoI [95% CI]	MW	Welch	Fisher 1500	Fisher 4000
FC	0.70 [0.49, 0.89]	0.135	0.091	0.418	0.131
FCM	0.68 [0.47, 0.87]	0.135	0.101	1.000	0.131
CSB	0.76 [0.56, 0.92]	0.044	0.037	0.527	0.005
RBG	0.92 [0.78, 1.00]	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$	$< 10^{-3}$

Table 9: Primary terminal-score comparisons against PQC. Probability-of-improvement intervals are pointwise seed-bootstrap intervals. Each test column is Holm-adjusted within its own four-comparison family. Fisher columns use the named thresholds. The analysis package to be released upon acceptance will include both raw and adjusted p-values.

Seed	PQC	FC	FCM	CSB	RBG
0	4,453.2	885.4	654.1	2,695.9	564.6
1	4,326.2	722.7	1,891.2	551.1	571.2
2	901.1	6,135.8	2,664.8	275.9	471.5
3	270.3	1,137.6	797.1	926.3	629.6
4	5,375.9	1,201.7	400.8	3,926.4	575.0
5	596.5	4,304.2	2,399.1	816.2	305.9
6	5,355.3	5,437.9	4,379.9	1,802.2	601.7
7	5,237.6	531.5	4,806.9	4,669.3	502.8
8	4,751.5	2,649.4	1,253.5	1,147.1	344.3
9	4,750.4	4,054.6	777.1	236.4	572.1
10	4,834.9	570.6	3,394.4	3,150.3	526.6
11	768.6	2,448.5	4,787.3	3,159.0	462.9
12	5,231.1	480.7	2,563.4	3,558.6	641.9
13	2,845.9	379.3	2,137.9	1,245.7	557.7
14	4,588.0	873.8	4,720.5	498.6	493.0

Table 10: All primary terminal scores, recomputed from raw 30-episode evaluations. Rounded values are shown; all analyses use full precision.

Learning-curve extraction. The metrics log repeats the most recent evaluation between evaluation events. We extract the first completed episode and rows crossing a 10,000-interaction boundary, as specified by the evaluation code. Zero-valued reset markers do not represent evaluations and

are excluded; none occurs among the selected evaluation records. A run must reach the stated interaction budget, and the statistic uses its latest genuine evaluation at or before that budget. This past-only rule prevents future-checkpoint leakage. Main curves use fixed cohorts: 15 for each Humanoid configuration, ten for Ant and Hopper, nine for HalfCheetah, and five/five/four for surrogate-forward/STE/stop-gradient. Lines are medians and bands are interquartile ranges across those runs, not confidence intervals. The compact panel additionally shows individual trajectories. Appendix per-seed lines stop at their own last evaluation, and each plotting axis stops at the last observation in that panel.

D.2 High-return frequencies and late-training behavior

For terminal scores $X_1, \ldots, X_n$, the observed high-return frequency is

$$\widehat{p}_\tau = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1}\{X_i \geq \tau\}. \tag{46}$$

We selected $\tau = 4000$ after inspecting the Humanoid results. It denotes a high-return regime at a finite training budget, not an official success criterion or a mathematical equilibrium. Figure 3 shows the full threshold profile. The exact binomial intervals in Table 11 are conditional on a fixed configuration and threshold; they do not account for exploratory selection. For k successes in n runs, their endpoints are

$$L = \begin{cases} 0, & k = 0, \\ \mathrm{Beta}^{-1}_{k,n-k+1}(0.025), & k > 0, \end{cases} \qquad U = \begin{cases} 1, & k = n, \\ \mathrm{Beta}^{-1}_{k+1,n-k}(0.975), & k < n. \end{cases} \tag{47}$$

The compact setting's three successes in three runs yield $[29.2\%, 100\%]$; this width illustrates the uncertainty of a three-run frequency estimate.

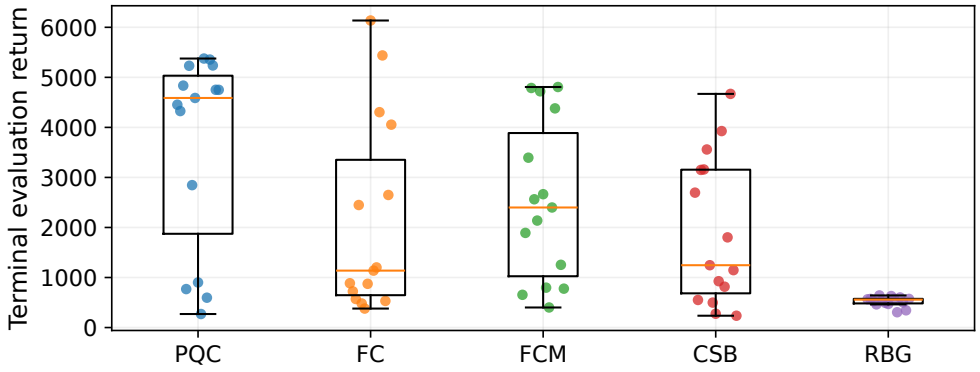


Figure 10: Humanoid terminal distributions. Dots are training-run scores; boxes show medians and interquartile ranges, with 1.5-IQR whiskers. Each primary configuration has 15 raw terminal evaluations.

Late-training summaries use one median per seed from actual evaluations in $[2.5\mathrm{M}, 3\mathrm{M}]$, without smoothing or extrapolation. The counts are 10/15 for PQC, 4/15 for FC, 3/15 for FCM, 2/15 for CSB and 0/15 for RBG. For the compact configuration, the late medians are approximately 5890, 4333 and 5525. Its least stable seed exceeds 4000 at only 62% of these checkpoints, so a high late median does not imply a flat learning curve.

D.3 Capacity, shot and surrogate-family comparisons

The five-qubit, width-256 setting has the largest observed mean and minimum in the sweep; ten qubits with width 1024 have the largest median. Parameter counts are $(d + N + 1)h + N$, with

Configuration	n	Terminal	95% interval	Late median	Ever
PQC	15	10/15	[38.4, 88.2]%	10/15	10/15
FC	15	4/15	[7.8, 55.1]%	4/15	6/15
FCM	15	4/15	[7.8, 55.1]%	3/15	7/15
CSB	15	1/15	[0.2, 31.9]%	2/15	4/15
RBG	15	0/15	[0.0, 21.8]%	0/15	0/15
$N = 5, h = 256, S = 1000$	3	3/3	[29.2, 100.0]%	3/3	3/3
$N = 5, h = 2048, S = 100$	5	5/5	[47.8, 100.0]%	5/5	5/5

Table 11: Terminal and late-training high-return frequencies at threshold 4000. "Late median" counts runs whose median periodic return over 2.5–3M interactions exceeds the threshold. "Ever" counts any periodic evaluation reaching it by 3M. The last two settings are exploratory configurations, not extra primary seeds.

N	Width	Parameters	Mean	Median	Range
5	64	8,069	535	448	[403, 752]
5	256	32,261	5,088	5,103	[4,268, 5,893]
5	1024	129,029	3,542	4,491	[628, 5,508]
5	2048	258,053	1,891	887	[832, 3,953]
5	4096	516,101	2,741	2,660	[515, 5,048]
10	64	14,794	387	416	[264, 481]
10	256	59,146	4,412	4,717	[3,778, 4,742]
10	1024	236,554	4,863	5,342	[3,880, 5,367]
10	2048	473,098	1,741	589	[71, 4,561]
10	4096	946,186	2,851	3,700	[580, 4,272]

Table 12: The full capacity sweep, with three runs per setting. Ranges are minima and maxima, not confidence intervals. The ten-qubit, width-2048 cell contains same-seed re-executions of primary seeds 0-2, not independent replications.

$d = (2N + 2)M$ and $M = 10$. Circuit controls and bypass outputs also change with qubit count, so changing N is not an intervention on surrogate width alone. The comparison concerns complete configurations.

Configuration ($n = 5$)	Mean	Median	Range	≥ 4000
$N = 10, S = 1000$ (re-execution)	3,301	4,015	[660, 5,391]	3/5
$N = 10, S = 100$	4,056	4,748	[668, 5,359]	4/5
$N = 5, S = 1000$	2,537	969	[657, 5,257]	2/5
$N = 5, S = 100$	4,617	4,754	[4,014, 4,897]	5/5
Autoregressive soft rollout	3,092	4,042	[461, 5,187]	3/5
Scalar sensitivity penalty	4,120	4,956	[687, 5,130]	4/5
Directional derivative targets	4,171	4,917	[606, 5,366]	4/5
Forward-path re-execution	3,212	4,675	[242, 5,344]	3/5

Table 13: Five-run exploratory configurations. The ten-qubit, 1000-shot reference and forward-path repeat are same-seed re-executions of primary seeds 0–4. They are displayed separately and never pooled as extra primary seeds. Scalar-sensitivity seed 1 has ten terminal episodes; every other row uses 30 episodes per run.

At ten qubits, reducing shots from 1000 to 100 changes mean return from 3301 to 4056 and median return from 4015 to 4748. At five qubits, the corresponding means are 2537 and 4617. These observations concern different mode-aggregated laws, not only different precision of a fixed expectation estimate. The capacity and shot sweeps test separate configurations.

Autoregressive surrogate. The autoregressive variant is trained with teacher-forced conditional Bernoulli likelihoods and a masked lower-triangular coupling matrix, adding 45 effective coupling parameters at ten qubits. At update time, previous bits are replaced recursively by soft probabilities. This differentiable rollout generally differs from the exact autoregressive marginal means; Proposi-

tion 1 concerns the base coordinatewise BCE model. The variant changes both fitting and downstream relaxation, so its returns do not isolate a benefit from modeling output correlations.

Scalar sensitivity regularization. The source option `jac_reg` differentiates the sum of surrogate outputs and penalizes its projection onto a random unit input direction v. Its penalty obeys

$$\lambda[v^\top J_f(u)^\top \mathbf{1}]^2, \qquad \mathbb{E}_v[vv^\top] = I/d \quad \Longrightarrow \quad \mathbb{E}_v[\text{penalty}] = \frac{\lambda}{d}\left\|J_f(u)^\top \mathbf{1}\right\|_2^2. \tag{48}$$

Output gradients may cancel, so this is neither a Frobenius-Jacobian penalty nor a penalty on $J_f - J_\mu$. The logged fitting loss excludes this extra penalty.

Directional derivative targets. The Sobolev variant adds squared error between surrogate Jacobian–vector products and cached channel finite differences. These labels require additional calls, include finite-shot and finite-difference error, and can cross routing boundaries. The supervised directions are not exact parameter-shift derivatives. These runs are kept distinct from diagnostic-only runs in Table 14.

D.4 Channel entropy and response-fitting diagnostics

The live diagnostic records cover three PQC runs, three CSB runs and five PQC runs with derivative-target fitting. Table 14 reports each run's last diagnostic event, not a uniform bound over the channel. Marginal estimation uses eight replay-control probes with 64 aggregated samples each. Four normalized directions and displacement $\delta = 0.02$ are used for central differences. The table's $\widehat{e}_1$ is a mean marginal discrepancy and $\widehat{D}_\delta$ a mean directional discrepancy. The field `eps2_fro` rescales random-direction measurements; it is not a measured full Jacobian norm or an operator-norm bound.

Channel	Seed	Steps (M)	$\widehat{e}_1$	$\widehat{D}_\delta$	$\widehat{H}(B \mid u)$	$\widehat{\sqrt{v}}$	BCE
CSB	0	2.993	0.213	2.97	0.65	0.40	0.066
CSB	1	2.973	0.051	1.01	0.41	0.17	0.019
CSB	2	2.953	0.186	4.05	1.53	0.55	0.072
PQC	0	2.974	0.689	15.16	6.66	1.47	0.664
PQC	1	2.968	0.809	13.40	5.90	1.40	0.528
PQC	2	2.924	0.693	14.83	7.01	1.44	0.587
PQC + derivatives	0	2.893	0.485	21.13	5.42	1.22	0.533
PQC + derivatives	1	2.970	0.933	18.96	6.79	1.34	0.695
PQC + derivatives	2	2.995	1.024	38.54	5.24	1.09	0.407
PQC + derivatives	3	2.971	1.099	18.96	6.68	1.43	0.615
PQC + derivatives	4	2.781	0.735	16.23	6.40	1.45	0.663

Table 14: All eleven live diagnostic runs, separated by channel and fitting protocol. Entropy is in bits. $\widehat{\sqrt{v}}$ is the mean empirical variance coefficient across the marginal probes. BCE is the median logged episode-end fitting loss over 2.5–3M interactions (nats per bit), not the loss at the single diagnostic event.

PQC diagnostic runs have conditional entropy estimates between 5.90 and 7.01 bits, versus 0.41–1.53 for CSB. Their empirical variance coefficients are 1.40–1.47 and 0.17–0.55, respectively, compared with the binary worst-case bound $\sqrt{10/4} = 1.581$. The mean-feature value bound in Eq. (9) can therefore be substantially looser for the PQC channel. The actual bias still depends on the downstream map and is not determined by this coefficient alone. The BCE and marginal discrepancies likewise measure different channel predictability, not a causal effect of quantum correlations on return.

Entropy estimator limits. Entropy uses 16 replay-control probes, 128 aggregated samples at each, and the plug-in estimate with Miller–Madow correction. If k_{obs} outcomes are observed among R samples, the estimator in bits is

$$\widehat{H} = -\sum_b \widehat{p}_b \log_2 \widehat{p}_b + \frac{k_{\text{obs}} - 1}{2R \ln 2} \le \log_2 R + \frac{R-1}{2R \ln 2}. \tag{49}$$

At $R = 128$ the ceiling is 7.716 bits, below the ten-bit maximum possible true entropy. High estimates near this ceiling do not establish maximal true entropy; rare outcomes can remain unseen. The logged total-correlation estimate subtracts this undersampled joint entropy from summed marginal entropies and can consequently be biased upward. We therefore do not interpret this total-correlation estimate as evidence of entanglement or as an explanation of learning performance.

Directional and Lipschitz proxies. For independent Bernoulli replicate estimates at two unclipped endpoints, the RMS sampling error of the central difference is at most $\sqrt{N/(8R\delta^2)} \simeq 6.99$ for $N = 10, R = 64, \delta = 0.02$. The logger's "noise floor" is this worst-case upper bound, not a lower bound. Actual perturbations include routing coordinates and are clipped; smaller endpoint displacement can increase the bound, and crossing a routing cell invalidates a local smooth-derivative interpretation. Finite-difference bias is separate. The maximum observed downstream gradient and secant norms are empirical maxima, not certified Lipschitz constants on a support hull. The diagnostic cost is $8 \cdot 64(1 + 2 \cdot 4) + 16 \cdot 128 = 6656$ channel invocations per event.

D.4.1 Shot-count and control dependence

We isolate the readout law using the same ten-layer circuit and 10^{-3} bit-flip and phase-flip channels as in training. For each of $N = 5$ and $N = 10$, we evaluate 200 controls drawn uniformly from $[0, 1]^d$. Density-matrix simulation supplies the Born probability vector without sampling error, up to floating-point arithmetic. For each vector and $S \in \{100, 1000\}$, 20,000 independent multinomial shot-count bags estimate the mode-output law. The Born probabilities are exact simulator outputs; the aggregated probabilities, entropy and marginal variance are Monte Carlo estimates.

The main calculation breaks mode ties by the smallest bitstring code, as in the small-circuit study. We also choose uniformly among tied modes using the same shot bags. The latter is a sensitivity analysis, not a claim about CUDA-Q's unrecorded tie convention. Entropy uses the plug-in estimate with Miller–Madow correction; brackets in Table 15 describe variation across controls, not confidence intervals. The 20,000-bag calculation is not subject to the live diagnostic's 128-sample ceiling, but retains finite-sample error. For ten qubits and smallest-code ties, reducing shots raises median

N	S	Tie rule	$H(\pi_u)$	$\widehat{H}(P_{u,S})$ [IQR]	$\widehat{\sqrt{v}}$	Ties (%)
5	1000	Smallest	4.27	0.16 [0.00, 0.79]	0.21	0.4
5	1000	Uniform	4.27	0.17 [0.00, 0.80]	0.20	0.4
5	100	Smallest	4.27	1.28 [0.83, 1.87]	0.65	8.4
5	100	Uniform	4.27	1.25 [0.83, 1.90]	0.64	8.4
10	1000	Smallest	8.35	1.92 [1.14, 2.79]	0.81	9.3
10	1000	Uniform	8.35	1.92 [1.12, 2.82]	0.81	9.3
10	100	Smallest	8.35	4.62 [3.73, 5.63]	1.15	40.8
10	100	Uniform	8.35	4.59 [3.72, 5.78]	1.18	40.8

Table 15: Channel characterisation at 200 random controls per qubit count. Entropies are in bits; entries are medians across controls. Both shot counts and tie rules use the same controls. "Ties" is the percentage of bags with more than one modal bitstring. $\widehat{\sqrt{v}}$ uses the estimated aggregated marginals.

aggregated entropy from 1.92 to 4.62 bits. For five qubits the corresponding medians are 0.16 and 1.28. The alternative tie rule gives the same qualitative random-control comparison. A large Born entropy need not imply a similarly random mode output: repeated shots can concentrate the output on a few modal bitstrings. These are conditional readout measurements, not additional policy-training outcomes or a causal explanation of the shot-sweep returns.

Rotation saturation. A separate sweep holds the routing controls fixed while varying rotation controls as $u^{\mathrm{rot}} = \sigma(gz)$, with z standard Gaussian and $g \in \{0.5, 1, 2, 4, 8, 32, 128\}$. The same 12 Gaussian vectors and routing choices are used at every scale, giving 84 controls per qubit count. This changes rotation saturation without simultaneously changing the routed gate endpoints. Four further controls place all rotation controls exactly at zero or one, with random fixed routing. The full study therefore contains 288 controls per qubit count.

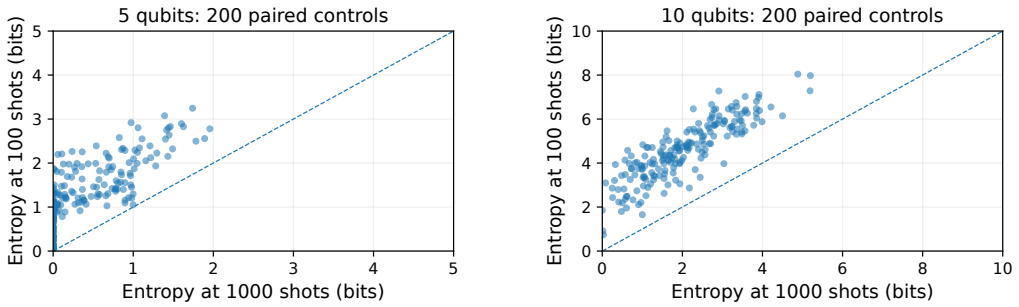


Figure 11: Paired change in aggregated entropy when reducing the shot count. Each point is one random control, using smallest-code ties. The diagonal denotes equal entropy at the two shot counts. Tables 15–16 retain both tie conventions.

N	Logit scale	n	$H(\pi_u)$	$S = 1000$, min	Uniform	$S = 100$, min	Uniform
5	0.5	12	4.16	0.78	0.78	1.65	1.71
5	1	12	4.41	0.91	0.90	1.71	1.66
5	2	12	4.48	0.20	0.20	1.85	1.85
5	4	12	4.00	0.42	0.43	1.09	1.06
5	8	12	4.25	0.23	0.23	1.38	1.40
5	32	12	4.26	1.84	1.84	2.52	2.52
5	128	12	4.94	4.00	4.00	4.19	4.29
5	Endpoints	4	5.00	4.99	5.00	4.95	5.00
10	0.5	12	8.30	1.56	1.58	4.28	4.26
10	1	12	8.60	2.16	2.18	4.84	5.15
10	2	12	8.51	2.02	2.06	4.94	4.99
10	4	12	7.87	2.17	2.13	3.89	3.93
10	8	12	7.78	1.47	1.43	3.79	3.70
10	32	12	8.37	5.02	4.99	6.43	6.49
10	128	12	9.75	8.89	9.02	8.82	9.25
10	Endpoints	4	10.00	9.81	10.00	9.38	10.00

Table 16: Median entropies in the matched saturation sweep, in bits. "Min" breaks ties by the smallest code; "Uniform" samples among tied modes. The endpoint rows use exactly zero-or-one rotation controls, not an extrapolated finite scale. All values for the aggregated law are Monte Carlo estimates; the true entropy is at most N.

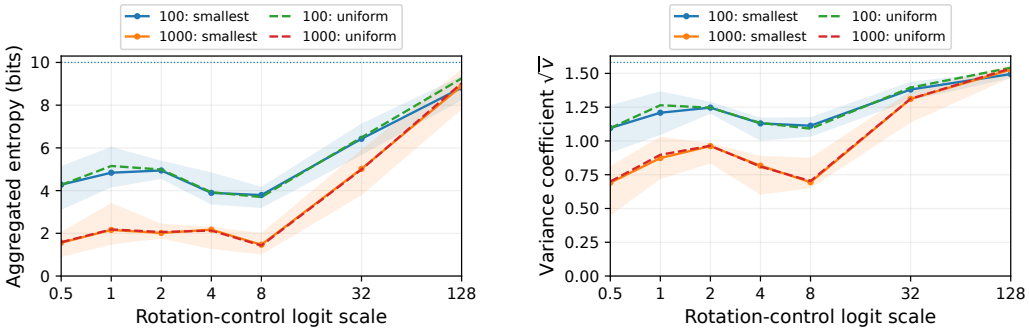


Figure 12: Ten-qubit rotation-saturation sweep. Solid curves and interquartile shading use smallest-code ties; dashed curves use uniform tie-breaking on the same bags. Each point summarizes the same 12 underlying Gaussian controls and fixed routing choices at its logit scale. The horizontal dotted lines are the ten-bit entropy and $\sqrt{10/4}$ variance-coefficient maxima.

At exact rotation endpoints, every U_3 is the identity up to a global phase. The intervening CZ gates can change phases, so the state need not return to $|+\rangle^{\otimes N}$; nevertheless its computational-basis probabilities remain uniform. The Pauli noise also preserves this uniform diagonal. With uniform tie-breaking, symmetry then makes the mode-output law exactly uniform for either shot count. Smallest-code ties break that symmetry and can reduce its entropy, particularly when many modes tie. Thus shot count and tie handling both form part of the readout law; an entropy increase with fewer shots is not a universal monotonicity result.

The random-control and saturation sweeps characterize prescribed circuit inputs. Table 26 and Appendix E.3 instead characterize controls produced by the stored surrogate actors during their own rollouts, not hybrid-policy rollouts. Their seed-level median entropies reach 8.92 bits. The live estimates in Table 14 use 128 samples per control and are constrained by the estimator ceiling. Since control selection, estimation budget and potentially tie handling differ, these studies do not provide a comparison at matched circuit inputs. A higher-replication check on 11 fixed controls compares 20,000 with 80,000 bags across 44 shot/tie cells. The largest observed changes are 0.025 bits in corrected entropy and 0.0028 in $\widehat{\sqrt{v}}$. This is a sensitivity check on selected controls, not a uniform accuracy guarantee.

D.5 Individual histories and additional-task coverage

The main Ant, Hopper and HalfCheetah summaries use ten, ten and nine seeds. Their actual evaluations occur at 790,046–790,726, 990,036–990,911 and exactly 1,000,000 steps, respectively. HalfCheetah seed 3 has two logged episodes through 2000 interactions and one evaluation at 1000; it terminated early and cannot contribute to the 1M cohort. Some additional-task runs were still in progress when we finalized the results, so each summary uses a fixed interaction budget reached by every run in its stated cohort. Table 17 also reports nine-seed cohorts at longer horizons: Ant through 1M interactions and Hopper through 2.1M. Each aggregate learning curve uses a fixed cohort throughout. We plot all recorded periodic evaluations for the five primary Humanoid

Environment	Budget	Seeds	Median	IQM [95% CI]
Ant-v5	0.8M	10/10	1,947	1,989 [1,540, 2,335]
Hopper-v4	1.0M	10/10	3,355	3,394 [2,756, 3,498]
HalfCheetah-v5	1.0M	9/10	2,027	3,133 [1,738, 6,759]
Ant-v5	1.0M	9/10	2,014	2,143 [1,466, 2,585]
Hopper-v4	2.1M	9/10	3,394	3,403 [2,695, 3,507]

Table 17: Additional-task summaries, with coverage shown for every horizon. Longer-horizon subsets are not replacements for the larger short-horizon cohorts.

Environment	Seed	Steps (M)	Episodes	Terminal mean
HalfCheetah-v5	4	3.000	30	2,542
HalfCheetah-v5	5	3.000	30	11,302
HalfCheetah-v5	6	3.000	30	2,540
HalfCheetah-v5	7	3.000	30	2,547
HalfCheetah-v5	8	3.000	30	1,116
HalfCheetah-v5	9	3.000	30	2,416
Hopper-v4	7	3.001	30	3,476
Hopper-v4	8	3.000	30	3,001

Table 18: All completed additional-task runs with separate terminal evaluations. These are a completion-selected subset, not a full-cohort terminal comparison. In particular, the strong HalfCheetah terminal score coexists with several lower-return completed runs.

configurations and the three additional tasks. Each trajectory ends at its last recorded evaluation, without extrapolation or carrying the last value forward. Each horizontal axis ends at the last observation in its panel; Ant therefore ends near 2.14M rather than at 3M.

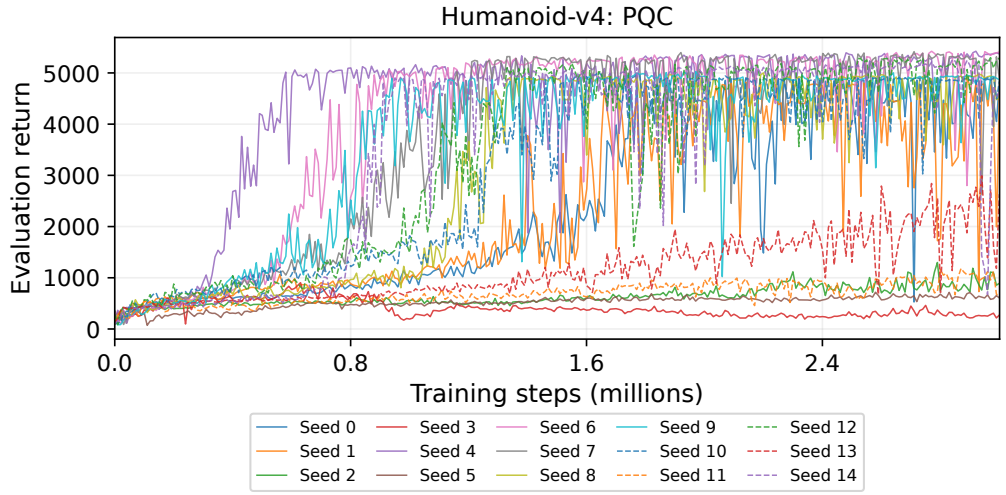


Figure 13: Humanoid-v4, PQC per-seed periodic evaluation histories. Every line uses actual observations through that run's final evaluation.

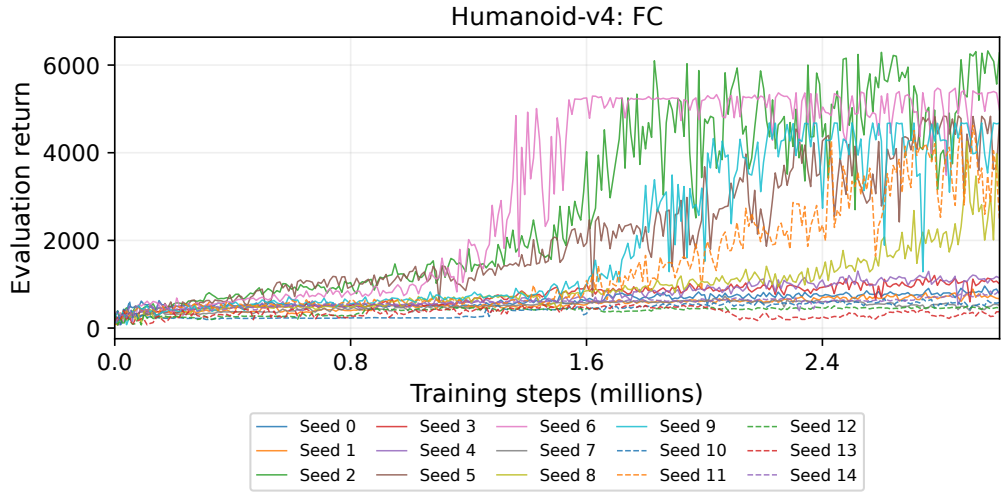


Figure 14: Humanoid-v4, FC per-seed periodic evaluation histories. Every line uses actual observations through that run's final evaluation.

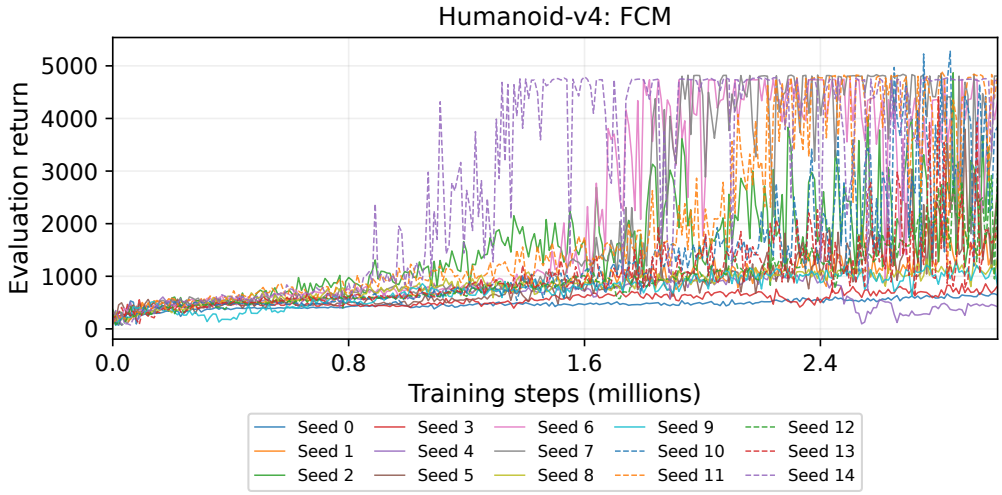


Figure 15: Humanoid-v4, FCM per-seed periodic evaluation histories. Every line uses actual observations through that run's final evaluation.

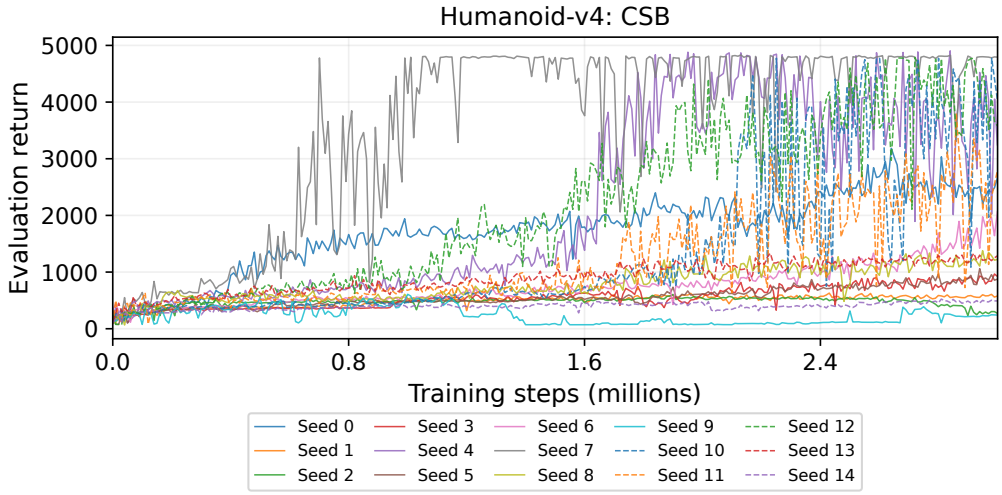


Figure 16: Humanoid-v4, CSB per-seed periodic evaluation histories. Every line uses actual observations through that run's final evaluation.

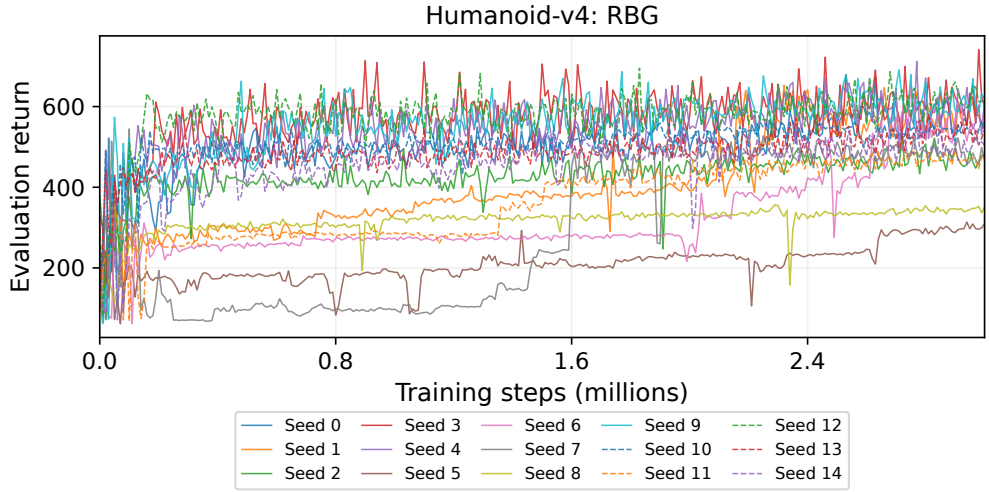


Figure 17: Humanoid-v4, RBG per-seed periodic evaluation histories. Every line uses actual observations through that run's final evaluation.

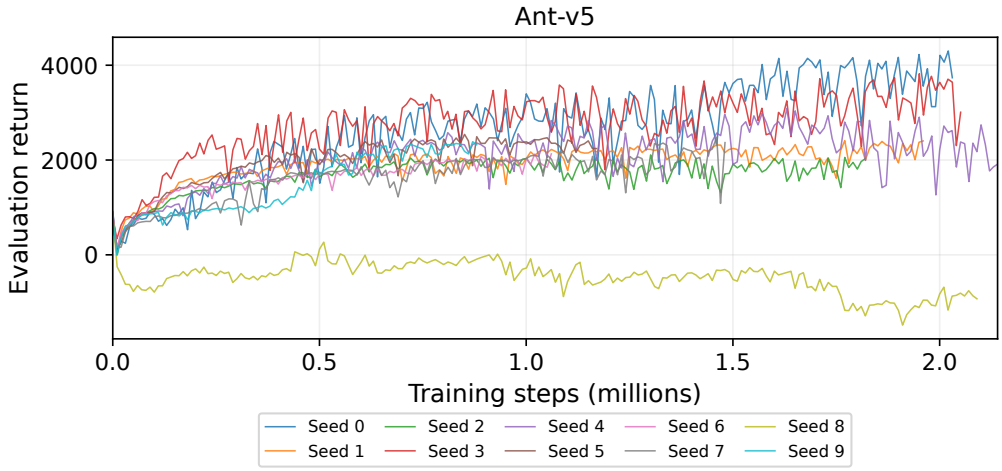


Figure 18: Ant-v5 per-seed periodic evaluation histories. Every line uses actual observations through that run's final evaluation.

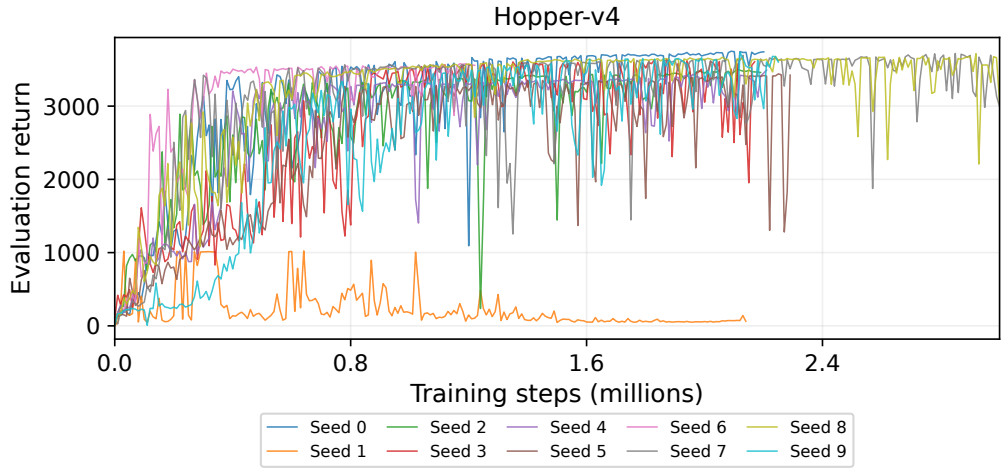


Figure 19: Hopper-v4 per-seed periodic evaluation histories. Every line uses actual observations through that run's final evaluation.

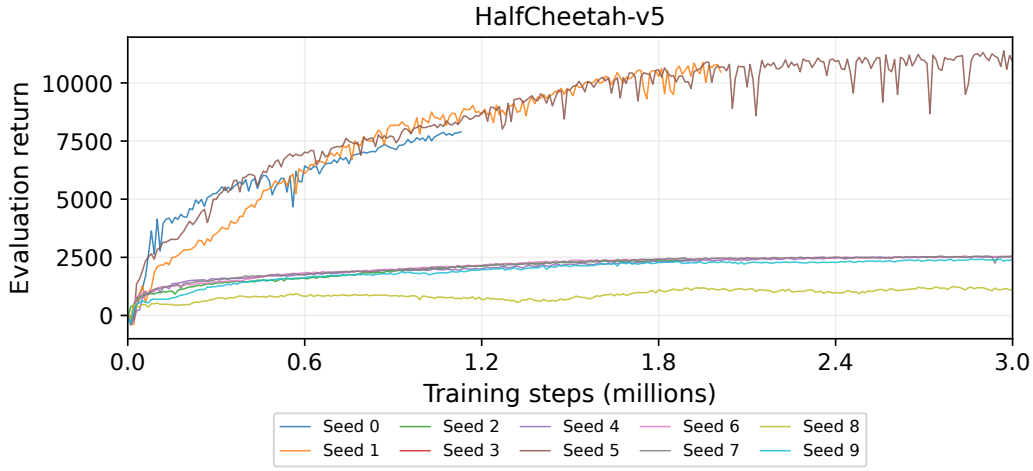


Figure 20: HalfCheetah-v5 per-seed periodic evaluation histories. Every line uses actual observations through that run's final evaluation.

D.6 UPDATE PATHS AT MATCHED INTERACTION BUDGETS

Return, process time and channel count in Table 19 come from the same latest eligible evaluation at each budget. The common 1.7M horizon is the largest 0.1M grid point covered by all five STE and four stop-gradient runs, chosen from step coverage rather than returns. The straight-through and stop-gradient runs were still in progress when we finalized the reported results. The surrogate-forward histories equal primary seeds 0-4 over these budgets. At 1.4M, their median is 1557, whereas the full 15-seed PQC cohort has median 3967. The medians differ because the update-path comparison uses a five-seed subset of the 15-seed primary cohort, not additional independent surrogate-forward runs.

Budget	Update path	n	Return	Hours	Calls
1.0M	Surrogate-forward	5	1,007	10.9	1.11M
1.0M	Hard-forward STE	5	294	205.9	380.83M
1.0M	Hard-forward stop-gradient	4	506	194.0	380.89M
1.4M	Surrogate-forward	5	1,557	15.6	1.61M
1.4M	Hard-forward STE	5	308	290.9	534.86M
1.4M	Hard-forward stop-gradient	4	533	273.1	534.91M
1.7M	Surrogate-forward	5	3,316	19.1	2.05M
1.7M	Hard-forward STE	5	395	354.8	650.38M
1.7M	Hard-forward stop-gradient	4	521	331.6	650.46M

Table 19: Three update paths at common budgets. All horizons reuse the same runs. Times accumulate in-process wall time across resumes, include periodic evaluation, and exclude scheduler waiting. Channel counts also include evaluation.

Budget	Comparison	n_A/n_B	PoI	Raw p	Holm p
1.0M	Surrogate vs. STE	5/5	0.80	0.151	0.302
1.0M	Surrogate vs. Stop-gradient	5/4	0.75	0.286	0.302
1.0M	STE vs. Stop-gradient	5/4	0.10	0.063	0.190
1.4M	Surrogate vs. STE	5/5	0.92	0.032	0.095
1.4M	Surrogate vs. Stop-gradient	5/4	0.75	0.286	0.286
1.4M	STE vs. Stop-gradient	5/4	0.10	0.063	0.127
1.7M	Surrogate vs. STE	5/5	0.88	0.056	0.167
1.7M	Surrogate vs. Stop-gradient	5/4	0.80	0.190	0.222
1.7M	STE vs. Stop-gradient	5/4	0.15	0.111	0.222

Table 20: Exploratory exact two-sided Mann–Whitney comparisons. Each budget forms a three-comparison Holm family; corrections do not make the repeated horizons independent or account for post-hoc budget selection. PoI gives the observed probability of improvement.

The five-versus-four comparison uses all available seeds. Table 21 additionally restricts all paths to labels 0–3, without claiming a paired experimental design. The surrogate-forward median remains higher, but the small-sample tests remain uncertain. The raw $p = 0.032$ surrogate-versus-STE contrast at 1.4M is not a multiplicity-adjusted finding.

Budget	Surrogate	STE	Stop-gradient	Surrogate/stop p
1.0M	779	300	506	0.486
1.4M	1,040	336	533	0.486
1.7M	1,997	326	521	0.343

Table 21: Four-seed subset sensitivity. The last column is an unadjusted exploratory test. These are the same runs as the full-cohort table, not a new experiment.

D.7 SAME-SEED RE-EXECUTIONS AND RESUMED SAMPLING

The forward-path reference (five runs), ten-qubit/1000-shot ablation reference (five), and ten-qubit/width-2048 reference (three) reuse primary seed labels and have long identical histories. We

compare episode indices, interaction counts, training returns, episode lengths, periodic scores, losses and channel counters at tolerance 10^{-6}, excluding elapsed time. Table 22 gives the final identical point and separately evaluated terminal scores. For the forward-path comparison, all five histories remain identical beyond 1.7M, justifying their identification as primary seeds 0–4 within that budget. The first discrepancies occur between approximately 1.58M and 2.15M interactions, around the first

Re-execution	Seed	Last identical (M)	Primary final	Re-execution final
Forward	0	2.098	4,453	4,675
Forward	1	2.154	4,326	4,839
Forward	2	2.101	901	958
Forward	3	1.917	270	242
Forward	4	2.065	5,376	5,344
Shot reference	0	2.098	4,453	4,884
Shot reference	1	2.154	4,326	4,015
Shot reference	2	1.931	901	1,554
Shot reference	3	1.881	270	660
Shot reference	4	2.089	5,376	5,391
Width reference	0	1.806	4,453	71
Width reference	1	2.154	4,326	4,561
Width reference	2	1.580	901	589

Table 22: Thirteen same-seed re-executions with shared training prefixes. Scores after divergence are not extra independent primary seeds. "Last identical" refers to the episode-level log comparison, not only to smoothed or periodic curves.

process-restart regime. The CUDA-Q shot stream is neither explicitly seeded nor checkpointed, so resumed executions can follow different subsequent sample paths. The observed divergence is consistent with restart-sensitive sampling, but the logs do not separate that effect from other execution differences. We report the reference re-executions separately and do not treat them as additional independent primary runs or as a controlled estimate of a restart effect.

D.8 Post-hoc progress, threshold crossings and response fit

To summarize learning across the full interaction budget, define one value per seed:

$$A_i = \frac{1}{30} \sum_{j=1}^{30} \widehat{R}_i(10^5 j), \tag{50}$$

where $\widehat{R}_i(t)$ is the latest genuine ten-episode evaluation at or before t. This is a grid-averaged return, not an exact integral of a continuous learning curve. It uses the same fixed 15-seed cohort per primary configuration and was selected after observing the trajectories. Tables 23–24 report its distribution and four-comparison inference. The correction addresses the four method contrasts, not selection of this statistic among possible learning summaries.

Method	Median grid average	IQM [95% CI]
PQC	3,144	2,638 [1,579, 3,444]
FC	751	913 [560, 1,590]
FCM	959	1,130 [809, 1,648]
CSB	801	924 [598, 1,498]
RBG	470	452 [376, 517]

Table 23: Post-hoc grid-averaged returns across 0.1–3M interactions. Each entry aggregates 15 seed-level values.

First threshold crossing. The first genuine periodic evaluation at or above 4000 is recorded for each seed, with non-crossing runs right-censored at 3M. A transient crossing and a high terminal score are different events: FC crosses in six runs and FCM in seven, but each has four terminal scores above 4000. PQC crosses in ten runs, CSB in four and RBG in none. Figure 3 uses all 15 runs at every plotted step, rather than averaging crossing times over successful runs alone.

PQC versus	PoI [95% CI]	Exact MW p	Holm p
FC	0.78 [0.60, 0.93]	0.009	0.026
FCM	0.73 [0.52, 0.92]	0.033	0.033
CSB	0.76 [0.57, 0.93]	0.013	0.026
RBG	0.94 [0.83, 1.00]	$< 10^{-3}$	$< 10^{-3}$

Table 24: Post-hoc learning-progress comparisons against PQC. Confidence intervals resample training runs; p-values use exact two-sided tests and a four-comparison Holm adjustment.

Seed	PQC	FC	FCM	CSB	RBG
0	1.751	> 3	> 3	> 3	> 3
1	1.660	> 3	> 3	> 3	> 3
2	> 3	1.600	2.561	> 3	> 3
3	> 3	> 3	> 3	> 3	> 3
4	0.491	> 3	> 3	1.750	> 3
5	> 3	2.310	> 3	> 3	> 3
6	0.731	1.360	1.740	> 3	> 3
7	1.031	> 3	1.840	0.701	> 3
8	1.260	2.890	> 3	> 3	> 3
9	0.840	2.030	> 3	> 3	> 3
10	1.361	> 3	2.650	2.170	> 3
11	> 3	2.431	2.120	> 3	> 3
12	1.111	> 3	> 3	1.970	> 3
13	> 3	> 3	2.800	> 3	> 3
14	0.911	> 3	1.111	> 3	> 3

Table 25: First threshold crossing in millions of interactions. > 3 denotes no observed crossing by the budget, not a predicted future crossing.

Behavior associated with high returns. For each primary PQC run, Table 26 summarizes training episodes ending in $[2.5\text{M}, 3\text{M}]$. Among the ten runs with terminal scores above 4000, 69.8–96.7% of these episodes last the full 1000 steps, and median reward per step is 4.82–5.39. Humanoid-v4's reward includes a healthy reward of five, forward reward weight 1.25 and action-cost weight 0.1 (Farama Foundation, 2026). These logs support sustained episode survival; total reward alone does not recover forward speed or separate the reward components. They should not be read as evidence of optimal or fast locomotion.

Response fit and training outcome. Proposition 1 separates response-fitting error from the stochasticity inherent in the labels. The irreducible population risk is

$$\mathcal{L}(\mu) = \frac{1}{N}\mathbb{E}_{u\sim\rho}\left[\sum_{k=1}^{N} h_{\text{nats}}(\mu_k(u))\right], \qquad h_{\text{nats}}(p) = -p\log p - (1-p)\log(1-p). \tag{51}$$

Only the excess risk $\mathcal{L}(f) - \mathcal{L}(\mu)$ measures fitting error. A change in visited controls or shot count can change the irreducible term as well as the fitted model. The channel studies in Appendices D.4.1 and E.3 characterize changes in the output law. Their joint entropies are not the sum of marginal entropies, and neither the random controls nor the surrogate-policy visits estimate this risk on the original replay distribution.

The logged `q_value_loss` is the last surrogate fitting minibatch loss in an update, recorded at episode end; it is not a held-out population BCE. Taking its median over 2.5–3M interactions yields a post-hoc primary-seed Spearman correlation of -0.686 with terminal return ($n = 15$, unadjusted $p = 0.0048$). The median of these losses is 0.252 for the ten high-return runs and 0.653 for the other five (Figure 21). Across 88 completed PQC surrogate-forward Humanoid executions, the corresponding descriptive correlation is -0.637. This pool includes 13 shared-prefix re-executions and different configurations; it is not an 88-independent-seed test. Excluding those re-executions leaves 75 executions with correlation -0.644, still confounded by architecture and fitting choices. For the autoregressive setting the logged loss is a conditional negative log-likelihood rather than the base per-bit BCE. We therefore make no pooled causal or confirmatory significance claim. Across

Seed	Terminal return	Full episodes (%)	Reward / step	Late BCE	$\widehat{H}_{1000}$	$\widehat{\sqrt{v}}_{1000}$
0	4,453	87.1	4.88	0.240	4.54	1.03
1	4,326	76.7	4.82	0.265	4.29	1.01
2	901	0.0	4.89	0.661	8.86	1.49
3	270	0.0	5.35	0.662	8.92	1.49
4	5,376	96.7	5.33	0.296	5.39	1.14
5	596	0.0	4.99	0.653	8.59	1.47
6	5,355	88.0	5.39	0.307	5.43	1.14
7	5,238	95.3	5.32	0.232	3.78	0.95
8	4,751	95.7	4.94	0.183	2.75	0.79
9	4,750	94.4	4.93	0.466	6.52	1.26
10	4,835	87.2	4.90	0.163	2.47	0.74
11	769	0.0	5.00	0.557	7.41	1.35
12	5,231	92.3	5.24	0.228	3.37	0.90
13	2,846	0.1	4.93	0.417	4.11	0.99
14	4,588	69.8	5.16	0.582	8.45	1.45

Table 26: Primary PQC behavior, fitting loss and visited-control estimates. Terminal returns come from the primary hybrid evaluations; episode completion, reward-per-step and BCE summaries concern late training. The last two columns are median estimated conditional entropy (bits) and variance coefficient over 64 controls from surrogate-actor rollouts at $S = 1000$ (Appendix E.3); they do not use hybrid-policy rollouts.

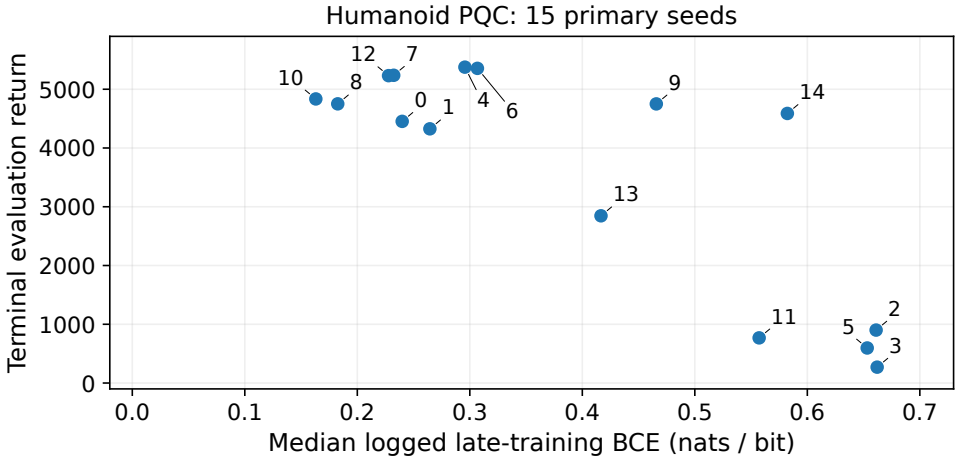


Figure 21: Late fitting loss and terminal return for the 15 primary PQC seeds. Labels identify seeds. This association is post hoc and does not identify causal direction or Jacobian accuracy.

the 15 primary checkpoints, late BCE is positively associated with estimated aggregated entropy (Spearman coefficient $+0.961$, unadjusted $p = 1.30 \times 10^{-8}$; Appendix E.3). At 1000 shots, median entropy across the ten high-return seeds is 4.414 bits, compared with 8.589 for the other five; the corresponding variance coefficients are 1.018 and 1.472. Entropy correlates with recorded terminal return at -0.604 (unadjusted $p = 0.0172$), with an exact two-sided group comparison of $p = 0.0280$. These measurements use controls from the classical surrogate actor, not the hybrid policy or training replay, and do not identify causality or isolate excess BCE. The high-entropy, high-return seed 14 and the lower-entropy, lower-return seed 13 also rule out a deterministic separation by entropy.

D.9 Process time and evaluation-query allocation

In a base PQC run, interaction calls equal the logged training count minus 1001 warm-up actions. The remaining logged calls are periodic evaluations; the difference between the final JSON counter and

Method	Median process hours
PQC	35.1
FC	8.8
FCM	8.2
CSB	41.4
RBG	8.9

Table 27: Median cumulative logged process time at the end of the full 3M primary campaigns, over 15 runs each. These totals precede the separate terminal evaluation and are not normalized across machines.

the final metrics counter is the separate terminal evaluation. Across primary runs, total calls including terminal evaluation range from 3.24M to 5.55M, and evaluation contributes 7.3–46.0%. Longer evaluation episodes naturally require more circuit calls. This allocation explains why an end-to-end invocation ratio is not simply 384, the average additional calls per update in sampled-forward training.

Seed	Interactions (M)	All calls (M)	Evaluation share (%)
0	3.001	4.521	33.7
1	3.001	4.556	34.1
2	3.000	3.385	11.4
3	3.000	3.236	7.3
4	3.001	5.551	46.0
5	3.000	3.312	9.5
6	3.000	5.328	43.7
7	3.000	5.137	41.6
8	3.001	4.941	39.3
9	3.000	5.238	42.7
10	3.000	4.934	39.2
11	3.000	3.431	12.6
12	3.000	5.002	40.0
13	3.000	3.720	19.4
14	3.000	5.169	42.0

Table 28: Primary PQC invocation counts, including periodic and separate terminal evaluation. Environment-step budgets count training interactions only.

E DEPLOYMENT SUBSTITUTIONS AND CHANNEL LAWS AT VISITED CONTROLS

This appendix evaluates the stored policies without further optimization. It separates the role of sampled features at deployment from the training procedure that produced the networks. The hybrid reference is the recorded terminal evaluation used in the primary comparison.

E.1 EVALUATION PROTOCOL AND CLASSICAL CALIBRATION

For each of the 15 primary PQC checkpoints, we retain the learned pre-network, post-network and surrogate, and replace only the features passed to the post-network. The bypass $c_\theta(s)$ is unchanged. Each substitute receives 30 evaluation episodes without external exploration noise, using the same environment-reset seeds as the recorded terminal evaluation (Appendix C). Episode index $e \in \{0, \dots, 29\}$ for training seed i uses

$$r_{i,e} = \big(1{,}000{,}003(i + 777{,}001) + 999{,}999 \cdot 1000 + e\big) \bmod (2^{31} - 1).$$

The logged checkpoint timestep matches the corresponding terminal-evaluation timestep in every run. Reset matching does not reproduce the original quantum shot stream. The hybrid reference is its recorded evaluation, not a fresh quantum evaluation on the local stack.

As a calibration, we re-evaluate the 15 FCM checkpoints with their original deterministic bottleneck. Six local means differ from the recorded means by less than one return point. The median absolute difference is 14.59 points, the mean signed difference is $+27.78$, and the largest absolute difference is 392.44. The same four seeds remain above 4000. Table 29 reports the paired calibration results for all 15 seeds; the primary comparisons use the original FCM evaluations.

For this calibration, let $D_{i,e} = R^{\text{local}}_{i,e} - R^{\text{recorded}}_{i,e}$ and $\overline{D}_i = 30^{-1} \sum_e D_{i,e}$. The standard error is $s_{D,i}/\sqrt{30}$, where $s_{D,i}$ uses the sample standard deviation with denominator 29. All mean differences are within two such *paired-difference* standard errors; the maximum ratio is 1.41. This describes the size of the discrepancies relative to episode-level variation, not equivalence of the software stacks or bitwise replication. Seed 11, for example, has a recorded standard deviation of 6.96 but a local standard deviation of 690.16, so the discrepancy is not confined to policies that were already variable in the original evaluation. We therefore compare substitutes at the training-seed level and do not interpret small local–recorded differences as isolated feature effects.

Seed	Recorded	Local	$\overline{D}$	$\mathrm{SE}(\overline{D})$	$\lvert\overline{D}\rvert/\mathrm{SE}$
0	654.1	654.3	+0.17	0.22	0.77
1	1,891.2	1,959.4	+68.26	98.73	0.69
2	2,664.8	2,692.4	+27.56	54.52	0.51
3	797.1	798.1	+0.99	3.82	0.26
4	400.8	401.4	+0.54	1.48	0.37
5	2,399.1	2,305.2	-93.97	277.61	0.34
6	4,379.9	4,617.7	+237.74	168.39	1.41
7	4,806.9	4,806.9	-0.03	1.13	0.02
8	1,253.5	1,268.1	+14.59	27.01	0.54
9	777.1	776.6	-0.60	3.20	0.19
10	3,394.4	3,786.8	+392.44	292.31	1.34
11	4,787.3	4,661.4	-125.87	125.91	1.00
12	2,563.4	2,445.3	-118.12	298.12	0.40
13	2,137.9	2,151.5	+13.62	247.05	0.06
14	4,720.5	4,719.9	-0.63	0.54	1.16

Table 29: FCM calibration on matched reset seeds, with 30 episodes per checkpoint. Recorded and local columns are episode-mean returns. The last two columns use the standard error of the paired episode differences, not the standard error of the local scores alone. The recorded values are the FCM reference evaluations used in the primary comparison.

E.2 Feature substitutions at fixed network weights

Writing $(u, c) = \mathrm{Pre}_\theta(s)$, every variant acts as $\mathrm{Post}_\psi(z, c)$, followed by clipping to the environment's action bounds. The six choices of z are

Surrogate mean:	$z = f_\phi(u)$,
Rounded surrogate:	$z_k = \mathbf{1}\{f_{\phi,k}(u) \geq 1/2\}$,
Sampled surrogate:	$z_k \sim \mathrm{Bernoulli}(f_{\phi,k}(u))$ independently,
Constant half:	$z = \frac{1}{2}\mathbf{1}$,
Zeros:	$z = 0$,
Fair bits:	$z_k \sim \mathrm{Bernoulli}(1/2)$ independently.

The sampled variants draw fresh features at each action. Their CPU random generators are initialized once per seed and variant, with seed i; these are not the historical quantum draws. Independent Bernoulli coordinates do not reproduce the circuit's joint readout law. Constants and fair bits also leave the learned, state-dependent bypass intact: they are not interventions on the entire state representation.

Table 30 aggregates the 15 checkpoint scores; Table 31 gives each checkpoint. The surrogate mean, rounded surrogate and sampled surrogate all retain the same ten high-return seeds as the recorded hybrid. Constant-half and fair-bit features retain nine, losing seed 1; zeros retain six, losing seeds 1, 4, 6 and 10. No substitute promotes any of the five originally lower-return seeds above 4000.

Features	Seeds	Mean	Median	≥ 4000	Median Δ_i
Recorded hybrid	15	3,619	4,588	10/15	–
Surrogate mean	15	3,684	4,737	10/15	+66
Rounded surrogate	15	3,600	4,613	10/15	+34
Sampled surrogate	15	3,606	4,726	10/15	+0
Constant half	15	3,486	4,535	9/15	-151
Fair bits	15	3,375	4,513	9/15	-4
Zeros	15	2,763	2,978	6/15	-173

Table 30: Deployment substitutions without further training. Each seed score averages 30 episodes; mean and median columns aggregate the 15 seed scores. Δ_i subtracts that seed's recorded hybrid score. The median difference is a median of differences, not a difference of medians. The recorded-hybrid row reports the 450 terminal episodes from the primary comparison; only the feature substitutes are evaluated in this deployment study.

Seed	Hybrid	Mean	Round	Sample	Half	Fair bits	Zeros
0	4,453	4,737	4,629	4,726	4,840	4,513	4,458
1	4,326	4,717	4,613	4,589	2,952	1,950	1,525
2	901	868	1,068	1,022	786	1,000	924
3	270	520	282	277	687	272	292
4	5,376	5,377	5,373	5,376	5,223	5,237	2,978
5	596	664	593	558	674	568	580
6	5,355	5,421	5,390	5,238	5,444	5,405	3,415
7	5,238	5,408	5,401	5,391	5,395	5,356	5,110
8	4,751	4,887	4,898	4,891	4,877	4,750	4,864
9	4,750	4,738	4,489	4,745	4,535	4,846	4,590
10	4,835	4,831	4,713	4,833	4,591	4,829	1,997
11	769	651	892	739	617	634	596
12	5,231	5,265	5,282	5,285	5,033	5,227	4,434
13	2,846	2,486	2,159	2,216	2,205	1,767	1,304
14	4,588	4,684	4,223	4,200	4,426	4,272	4,385

Table 31: All primary-checkpoint deployment means, rounded to integers. "Mean", "Round" and "Sample" refer to the three surrogate features defined above. Each checkpoint is evaluated with all six feature substitutes. Reset seeds match the recorded hybrid evaluation, but the local and historical execution stacks are not demonstrated to be identical.

Among the ten originally high-return seeds, the median surrogate-minus-hybrid change is $+80.85$ points. This is a descriptive summary, not a non-inferiority test. Seeds 4 and 10 differ by $+1.09$ and -4.09 points, with both their local and recorded standard deviations below ten. For seed 10, the small negative difference exceeds two paired-difference standard errors; closeness in return does not mean indistinguishability at that precision. The hybrid mean exceeds every substitute for seeds 10 *and* 13. Its margins over the best substitute are 2.22 and 360.42 points, respectively. These contrasts distinguish minor numerical changes from the larger loss for seed 13.

Classical deployment and training procedures. The stored surrogate mean supplies a deployable classical actor,

$$\widetilde{\pi}(s) = \mathrm{Post}_{\psi}\big(f_{\phi}(u_{\theta}(s)), c_{\theta}(s)\big),$$

without an additional fitting or distillation stage. Its layer shapes, activation functions and bypass match the configured FCM actor. The weights and training objectives differ: FCM learns the bottleneck through the policy loss, whereas the PQC surrogate learns circuit responses by BCE and the circuit supplies behavior during interaction. This follow-up therefore gives a same-architecture deployment comparison between two training procedures, with ten versus four high-return seeds. The primary tables retain their sampled-hybrid evaluations; the deterministic actor is evaluated separately here.

The substitutions support retaining strong policies without executing the circuit at deployment. Success of the constant and random substitutes for many checkpoints also limits any general claim that state-dependent sampled features are necessary for their high returns. Conversely, the losses for some seeds show that not every checkpoint is feature-insensitive. The experiments neither identify a causal benefit of quantum behavior during training nor show that it could not be replaced by another behavior channel; they change deployment features after training, not the training data-generation process.

E.3 Channel characterisation on surrogate-policy rollouts

For each primary PQC checkpoint, 64 control vectors are sampled without replacement from the state visits collected during the 30 surrogate-mean evaluation episodes. The sampling generator is seeded by the training-seed label. These vectors come from the *classical update actor's rollouts*, not the hybrid policy's trajectories, the original training replay distribution, or the separate live-diagnostic runs. Controls within a seed may be dependent; the training seed remains the unit of outcome comparisons.

We characterize these controls using the ten-qubit, ten-layer circuit with bit-flip and phase-flip probabilities of 10^{-3} after each U3 gate, as in Appendix D.4.1. Density-matrix evolution computes the Born law up to floating-point arithmetic. At each control, we estimate the mode-output distribution from 20,000 bags of $S = 100$ or $S = 1000$ shots, using smallest-code tie-breaking. The sampler is reset to seed zero for each checkpoint. This study uses a specified tie convention; the training backend's convention was not recorded. Aggregated entropy is estimated using the plug-in value with the Miller–Madow correction. The Born calculation is therefore exact within the simulator, while aggregated entropy and the variance coefficient remain Monte Carlo estimates.

For seed i, let $u_{i,1}, \ldots, u_{i,64}$ denote the sampled controls. Table 26 reports the seed-level summaries

$$\widehat{H}_{i,1000} = \operatorname*{median}_{\ell=1,\ldots,64} \widehat{H}\big(P_{u_{i,\ell},1000}\big)\,, \qquad \widehat{\sqrt{v}}_{i,1000} = \operatorname*{median}_{\ell=1,\ldots,64} \sqrt{\sum_{k=1}^{10} \widehat{\mu}_k(u_{i,\ell})\big(1-\widehat{\mu}_k(u_{i,\ell})\big)}.$$

These are medians of conditional quantities, not the entropy or variance of the pooled output distribution. Table 32 additionally gives both shot budgets and $\widehat{p}^{\star}_{S}$, the estimated probability that the aggregate output equals the smallest-code Born argmax, summarized by its median over controls.

The per-seed median saturation fraction lies in $[0.945, 0.995]$. This fraction counts all 220 controls below 0.05 or above 0.95, including routing controls, rather than measuring only rotation angles. Median Born entropies range from 7.66 to 9.88 bits. These ranges concern *per-seed medians*; individual controls span a broader range. Saturation therefore coexists with substantially different readout laws across checkpoints, and is not by itself a diagnosis of an almost uniform output.

Seed	Sat.	$H(\pi_u)$	$\widehat{H}_{100}$	$\widehat{H}_{1000}$	$\widehat{\sqrt{v}}_{100}$	$\widehat{\sqrt{v}}_{1000}$	$\widehat{p}^{\star}_{1000}$
0	0.977	8.10	6.26	4.54	1.22	1.03	0.1204
1	0.964	8.02	5.98	4.29	1.21	1.01	0.0993
2	0.991	9.61	8.58	8.86	1.46	1.49	0.0024
3	0.995	9.88	9.03	8.92	1.48	1.49	0.0032
4	0.973	8.14	6.34	5.39	1.23	1.14	0.0411
5	0.991	9.61	8.52	8.59	1.45	1.47	0.0034
6	0.977	8.57	6.98	5.43	1.29	1.14	0.0640
7	0.964	7.96	5.37	3.78	1.13	0.95	0.1517
8	0.964	7.66	4.67	2.75	1.03	0.79	0.2440
9	0.982	8.79	7.23	6.52	1.32	1.26	0.0171
10	0.945	7.66	4.45	2.47	1.01	0.74	0.3848
11	0.982	9.11	7.97	7.41	1.41	1.35	0.0096
12	0.964	7.81	5.37	3.37	1.12	0.90	0.2065
13	0.950	7.96	5.47	4.11	1.14	0.99	0.1011
14	0.986	9.40	8.43	8.45	1.45	1.45	0.0030

Table 32: Visited-control characterisation. Each entry is a median over 64 controls for one primary checkpoint; entropies are in bits. "Sat." is the fraction of all 220 controls outside $[0.05, 0.95]$. $\widehat{H}_S$ and $\widehat{\sqrt{v}}_S$ concern the mode-aggregated law; $H(\pi_u)$ concerns individual shots. Controls are collected under the surrogate-mean policy, not the hybrid policy.

Association with recorded training outcomes. At $S = 1000$, the medians across the ten high-return seeds and five other seeds are respectively 4.414 and 8.589 bits for $\widehat{H}_{i,1000}$, 1.018 and 1.472 for $\widehat{\sqrt{v}}_{i,1000}$, and 0.1099 and 0.0034 for $\widehat{p}^{\star}_{i,1000}$. The entropy–terminal-return Spearman coefficient is -0.604 ($n = 15$, two-sided asymptotic $p = 0.0172$); the exact two-sided Mann–Whitney comparison between the ten and five seed summaries gives $U = 7$ and $p = 0.0280$. The entropy–late-BCE coefficient is $+0.961$ ($p = 1.30 \times 10^{-8}$). All are post-hoc, unadjusted associations; the 960 controls are not treated as 960 independent training outcomes. Entropy and the variance coefficient have the same seed ordering here and are not independent confirmations.

At $S = 100$, the corresponding high-return/other-seed entropy medians are 6.120/8.522 bits and the variance-coefficient medians are 1.211/1.446. The two shot budgets characterize the same controls rather than policies retrained at those budgets. High-return seed 14 has entropy 8.45 bits at 1000 shots, while lower-return seed 13 has 4.11 bits. These exceptions, the non-hybrid control collection, and the absence of a training intervention preclude a claim that low entropy causes good learning. Joint entropy also differs from the marginal-entropy term in Proposition 1; the measurements do not decompose the logged BCE into irreducible uncertainty and fitting error.

Data and reproducibility. We compute deployment scores from individual episode returns, matching checkpoint timesteps and environment-reset seeds to the primary terminal evaluations. We compute the channel summaries from per-control statistics, check the entropy-correction and variance identities, and associate each checkpoint with its late-training BCE using the training-seed label. The FCM calibration is reported separately and is not used to rescale returns.

The records used for this statistical analysis contain episode returns, per-control summaries and channel configurations, but do not include checkpoint weights, saved control vectors, full Born probability vectors or a local software-version manifest. They also lack source hashes for these deployment and visited-control executions. The package to be released upon acceptance will include the protocol scripts, experimental records and statistical-analysis code. These materials allow the reported summaries to be recomputed; the saved records alone do not support independent re-execution of the policy evaluations or density-matrix simulations.