COMPOSITIONAL QUANTUM SURROGATES: CIRCUIT-GUIDED REPRESENTATIONS FOR QUANTUM SURROGATE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Quantum expectation surrogates amortize measurements across related circuit controls, but downstream tasks require different forms of reuse. Existing predictive-surrogate methods learn mean-value functions; how the fitted representation serves ordered predictions, observable reuse, and surrogate-based optimization remains open. We introduce Compositional Quantum Surrogates (CQS) through two circuit-guided architectures: CQS-R, a recurrent model for ordered layers or drive periods, and CQS-P, a masked Fourier product model for observable-dependent interactions. With matched masks and frequency coverage, product composition reduces moment prediction error by 10.5% relative to complete Fourier regression. A query-wise recurrent control reaches similar accuracy on 12-period trajectories but requires 78 updates, compared with 12 shared updates for CQS-R. In VQE, linear moment readouts enable Hamiltonian reuse, and Fourier structure enables analytic continuation; on two-layer grids, continuation mainly improves attainment near a feasible reference. The results favor dependency-aware products for moments, shared recurrence for trajectories, and continuation-stabilized search for VQE pre-training.

1 INTRODUCTION

Many quantum-computing workloads repeatedly query related parameterized circuits. A variational quantum eigensolver (VQE) estimates energies while searching over control parameters (Peruzzo et al., 2014; Kandala et al., 2017; Cerezo et al., 2021a). Floquet experiments measure local observables across successive drive periods (Zhang et al., 2022). Predictive surrogates turn previously acquired outcomes into functions of the circuit controls, allowing new controls, periods, observables, or Hamiltonian coefficients to be evaluated without a fresh measurement of each queried circuit. The central design issue is not only prediction accuracy, but also whether the fitted representation supports the computation that the downstream task repeats.

Classical shadows estimate multiple properties from measurements of a state (Huang et al., 2020; Aaronson, 2018; Struchalin et al., 2021), while learning across state families extends prediction to unmeasured instances (Huang et al., 2022). For parameterized processors, predictive mean-value surrogates model observables across circuit controls, with applications spanning variational pre-training, Floquet response prediction, and phase identification (Liao et al., 2026). Trigonometric predictors exploit gate periodicity to estimate these functions without reconstructing a full quantum state or measuring the test circuit (Liao et al., 2026; Schuld et al., 2021). How the fitted model shares computation across related outputs remains a separate design choice.

Dense Fourier interactions grow rapidly with the number of active controls, even when an observable depends on only a subset of gates. Fitting each requested period independently repeats work across an ordered trajectory. These settings call for separating the circuit dependencies supplied to the model, the representation of their interactions, and the computation shared across outputs. Optimization adds a further test: predictors with similar held-out error can guide different search trajectories.

We introduce Compositional Quantum Surrogates (CQS), a representation framework for expectation prediction that composes circuit controls either sequentially or multiplicatively. CQS-R follows circuit execution order with a shared gated recurrent unit (GRU) (Cho et al., 2014; Chung et al.,

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.

2014). CQS-P learns Fourier factors under observable-dependent control masks and combines them multiplicatively. Both models are fitted from task labels and receive only controls at inference. Linear moment readouts allow a fitted moment model to be recombined across Hamiltonians, while Fourier structure allows analytic Gaussian continuation of the fitted objective during optimization.

The experiments compare prediction quality and downstream use. Under matched scalar supervision on transverse-field Ising model (TFIM) circuits, performance depends on the label regime. A representation-by-mask control attributes the larger moment-prediction gain to observable dependencies, with a further 10.5% within-mask MSE reduction from products over complete Fourier regression. In Floquet symmetry-protected topological (FSPT) prediction, a query-wise GRU reaches similar accuracy to CQS-R but requires 78 recurrent updates for a 12-period trajectory rather than 12. In VQE, product moments with continuation improve attainment near a feasible-circuit reference on two-layer grids; few starts improve appreciably beyond that reference.

Our main contributions can be summarized as follows:

- We propose a circuit-guided surrogate framework with two complementary instantiations: shared recurrent updates for ordered prediction queries and observable-dependent Fourier products for structured moment prediction.
- We design controlled comparisons that separate dependency information, low-rank product fitting, and per-output recurrent depth. These controls show an additional product fitting benefit within matched masks and identify shared recurrence as computation reuse for full-trajectory prediction.
- We evaluate surrogate-assisted VQE with paired endpoint metrics, matched continuation controls, and feasible-energy references, showing where continuation stabilizes search and where the simple feasible circuit bounds endpoint improvements.

2 Related Works

2.1 Learning quantum properties

Shadow tomography and classical shadows estimate many properties from copies of a state (Aaronson, 2018; Huang et al., 2020), with experimental demonstrations of measurement-efficient property estimation (Struchalin et al., 2021). Learning across structured state families instead predicts properties of new instances; local ground-state properties of gapped Hamiltonians admit such prediction under specified assumptions (Huang et al., 2022). Our task uses parameterized circuits and periodic drives as the instance family. Measured properties supply supervision, while controls alone specify a new prediction query. Measurement reuse within a state and generalization across controls are distinct; this work focuses on the latter and on downstream reuse.

2.2 Predictive quantum processor surrogates

The 2026 Nature Communications predictive-surrogate method of Liao et al. (2026) first formulated the processor mean-value surrogate view used by our reference comparisons, using learned mean-value functions for variational pre-training and Floquet phase identification. We implement the public support-two TFIM and sampled-feature FSPT fitting rules on the circuit families studied here. These reference predictors already exploit trigonometric structure; the FSPT sampler also permits interactions beyond support two. Our comparison isolates the form of structure: fixed trigonometric feature maps with fitted output coefficients versus learned sequential or factorized representations. Section 3 distinguishes the reference fitting algorithms from their measurement labels and task assumptions.

2.3 Periodic features and factorized prediction

Quantum feature maps connect circuits to kernels (Schuld & Killoran, 2019; Havlíček et al., 2019), and Fourier analysis relates expressivity to parameter encoding (Schuld et al., 2021; Caro et al., 2021). Feature approximations and kernel regression provide classical fitting tools (Rahimi & Recht, 2007; Rasmussen & Williams, 2006). Tensor-product regression uses local features and low-rank

coefficient tensors (Kolda & Bader, 2009; Stoudenmire & Schwab, 2016). Saiapin & Batselier (2025) combine canonical polyadic (CP) Fourier models with learned feature mixtures and alternating least squares. CQS-P adapts this low-rank function family through circuit masks, supervised factor fitting, and harmonic orders determined by repeated controls.

2.4 Recurrent representations and variational objectives

GRUs learn state representations through shared sequence updates (Cho et al., 2014; Chung et al., 2014). Related physical modeling approaches learn coordinates with linear Koopman evolution (Lusch et al., 2018) or represent non-Markovian quantum processes using learned Lindblad operators (Banchi et al., 2018). CQS-R uses recurrent computation for expectation prediction; its hidden state follows circuit order without being constrained to identify a physical state. For VQE, the relevant outcome is the true energy reached after optimizing circuit parameters (Peruzzo et al., 2014; Kandala et al., 2017; Cerezo et al., 2021a), and trainability studies emphasize the role of the optimization landscape (McClean et al., 2018). We therefore assess held-out prediction and optimization endpoints separately, and compare continuation on both product and reference representations.

3 Background and Problem Setup

Controls, expectations, and labels For controls $\mathbf{x}$, processor program $U_{\mathbf{x},k}$, layer or drive period k, and observable O_j, define

$$y_{k,j}(\mathbf{x}) = \mathrm{Tr}[O_j \mathcal{P}(U_{\mathbf{x},k}, \rho_0)]. \tag{1}$$

Here $\mathcal{P}$ applies the program to ρ_0 through an ideal evolution or a realized noisy processor channel. A training example contains a control vector and target estimates $\tilde{y}_{k,j}$ for a specified set of periods and observables. Exact labels satisfy $\tilde{y}_{k,j} = y_{k,j}$. Simulation and finite measurements define different supervision regimes for the same prediction interface. The observable set fixes the output coordinates, and inference uses controls rather than measurements from the test circuit. The predictors therefore estimate the specified expectations, not the full quantum state.

Prediction and pre-training objectives Prediction evaluates $\hat{y}_{k,j}$ on held-out controls. Final-layer energy prediction is the scalar case at $k = L$; trajectory prediction supervises several k. VQE instead uses $H_J = \sum_o c_o(J) O_o$ and the learned energy $\hat{E}_J(\boldsymbol{\theta}) = \sum_o c_o(J) \hat{m}_o(\boldsymbol{\theta})$. Starting from matched initial controls, an optimizer returns $\boldsymbol{\theta}_{\mathrm{end}}^{(s,r)}$ for seed s and restart r. Its task metric is

$$\bar{E}_{\mathrm{end}}^{(s)} = \frac{1}{R_{\mathrm{start}}} \sum_{r=1}^{R_{\mathrm{start}}} E_J(\boldsymbol{\theta}_{\mathrm{end}}^{(s,r)}). \tag{2}$$

We use $R_{\mathrm{start}} = 20$ and score every endpoint with the exact simulator. This measures surrogate-assisted pre-training, rather than a subsequent quantum refinement stage or the best selected restart.

Measurement-matched reference predictors The reference comparators follow the Nature Communications predictive-surrogate methods of Liao et al. (2026). TFIM comparators use support-two trigonometric features; FSPT additionally uses the public sampled-feature h_{qs} algorithm in expanded gate coordinates. One-dimensional VQE uses h_{cs}-style shadow records; two-dimensional VQE uses h_{qs}-style grouped finite-shot measurements. Both use exact test energies. Final-layer prediction supplies the same scalar h_{cs}-style estimates from local marginals to every learner, including the h_{qs}-style fit. Algorithm and label source are thus distinct. FSPT uses exact labels with separate coefficient vectors for each period and observable. Hardware additionally compares a step-conditioned static model under the recurrent model's early-prefix validation rule.

Generalization and data reuse A trajectory is the splitting unit: all targets of a sampled control vector belong to one partition. Final-layer TFIM and FSPT repeats reuse one control pool per size with paired partitions and neural initializations. VQE generates a control and measurement dataset per seed. The main FSPT evaluation supervises every selected period and tests new controls, rather than forecasting beyond the training horizon.

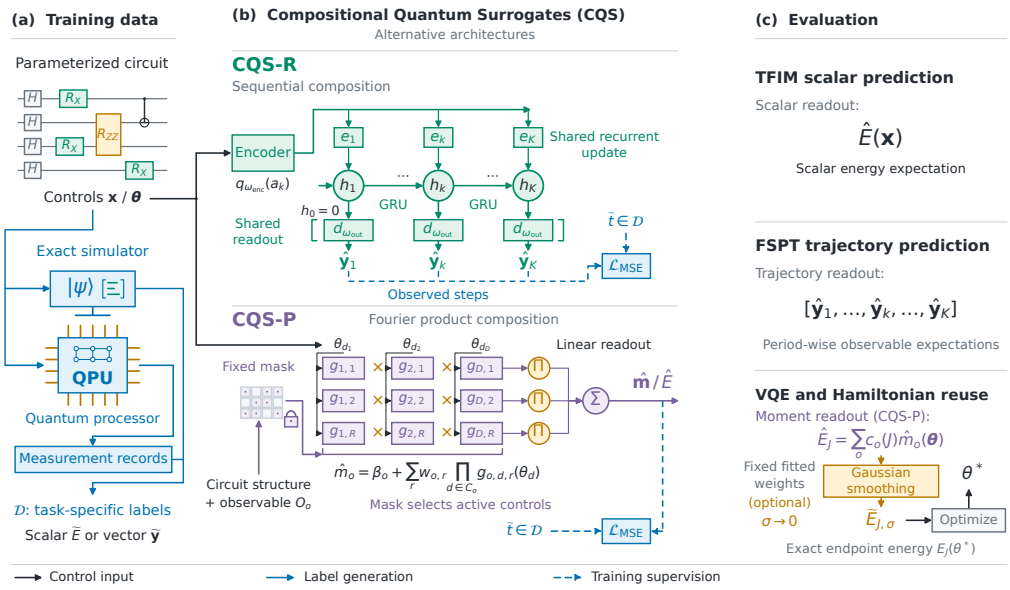


Figure 1: **Compositional Quantum Surrogates.** CQS-R composes ordered circuit inputs with a shared recurrent update and reads out requested expectations from the resulting state. CQS-P composes observable-specific Fourier factors over the controls selected by circuit-structure masks. Both instantiate the same expectation-prediction interface while using different composition operators. Moment outputs can be linearly recombined into Hamiltonian energies, and the product form admits analytic smoothing for optional VQE continuation.

4 Methodology

CQS places a circuit-guided composition operator before a common expectation readout. For a requested output, write

$$\hat{\mathbf{y}}_k(\mathbf{x}) = d_{\omega_{\text{out}}}\left(\Phi_{\omega_{\text{rep}}}(\mathbf{x}; k)\right). \tag{3}$$

Here ω_{rep} and ω_{out} denote learned representation and readout parameters. We use $\boldsymbol{\theta}$ for gate-angle controls. The two instantiations differ in the composition operator $\Phi_{\omega_{\text{rep}}}$: CQS-R composes ordered controls through shared recurrent updates, whereas CQS-P composes observable-dependent Fourier factors over the controls that can affect each output. For CQS-P, $\Phi_{\omega_{\text{rep}}}$ collects the output-specific product features, and $d_{\omega_{\text{out}}}$ applies their corresponding linear readouts.

4.1 A Common Expectation-Prediction Interface

The readout predicts either a vector of moments or a scalar energy, depending on the supervised target. Moment supervision minimizes $\|\hat{\mathbf{m}} - \tilde{\mathbf{m}}\|_2^2$, where $\mathbf{m}$ collects the chosen observable expectations. Scalar supervision minimizes the squared error of a direct energy output or of $\sum_o c_o \hat{m}_o$. In the scalar case, the individual heads are latent contributions and need not be identifiable physical moments. This distinction is architectural: moment supervision exposes reusable observable estimates, while scalar supervision fits only the requested linear functional. Experimental protocols, label sources, and evaluation metrics are given in Appendix B.

4.2 CQS-R: Sequential Composition

Encoder and shared update Let a_k be the input at layer or period k. The encoder is a two-layer multilayer perceptron (MLP) with SiLU activations,

$$e_k = q_{\omega_{\text{enc}}}(a_k). \tag{4}$$

The deterministic recurrent representation initializes its hidden state to zero and applies the same GRU parameters at every step (Cho et al., 2014; Chung et al., 2014):

$$h_0 = 0, \qquad h_k = \mathrm{GRU}_{\omega_{\mathrm{rec}}}(e_k, h_{k-1}). \tag{5}$$

The hidden state stores computation for expectation prediction rather than a constrained physical state. Period-indexed inputs can make the update time-dependent even though the parameters are shared, allowing the representation to follow circuit order without imposing an autonomous dynamics model.

Expectation readout The readout maps the composed state to the requested observable vector:

$$\hat{\mathbf{y}}_k(\mathbf{x};\omega) = d_{\omega_{\mathrm{out}}}(h_k), \qquad \hat{y}_{k,j}(\mathbf{x};\omega) = [d_{\omega_{\mathrm{out}}}(h_k)]_j, \tag{6}$$

where $\omega = (\omega_{\mathrm{enc}}, \omega_{\mathrm{rec}}, \omega_{\mathrm{out}})$. Training minimizes supervised loss over observed output coordinates,

$$\mathcal{L}(\omega) = \frac{1}{|\mathcal{D}|} \sum_{(\mathbf{x},k,j,\tilde{y})\in\mathcal{D}} \ell(\hat{y}_{k,j}(\mathbf{x};\omega), \tilde{y}), \tag{7}$$

with mean-squared error used in the experiments. Because one forward recurrence produces all prefixes, CQS-R reuses intermediate computation across requested periods. The method matches tasks in which outputs are indexed by circuit depth, period, or prefix. Appendix B and Appendix C report the batching, checkpoint, hardware-label, and recurrent-control protocols.

4.3 CQS-P: Masked Fourier Product Composition

Observable-dependent masks For each output observable o, reverse circuit propagation constructs a conservative control set C_o without using expectation labels. For Pauli circuits, the construction tracks which controls can influence the propagated measurement operator; for support-based tasks, it propagates support unions rather than enumerating all Pauli branches. These masks define the coordinates that participate in each output's product representation. Mask construction and implementation complexity are given in Appendix N.

Factorized periodic representation For each active control θ_d and channel r, define a learned periodic factor

$$g_{o,d,r}(\theta_d) = a_{o,d,r,0} + \sum_{q=1}^{Q} \big[a_{o,d,r,q}\cos(q\theta_d) + b_{o,d,r,q}\sin(q\theta_d)\big]. \tag{8}$$

Multiplication combines the active factors into one feature per observable o and rank channel r, and a learned linear readout forms the prediction:

$$z_{o,r}(\boldsymbol{\theta}) = \prod_{d\in C_o} g_{o,d,r}(\theta_d), \qquad \hat{m}_o = \beta_o + \sum_{r=1}^{R} w_{o,r} z_{o,r}(\boldsymbol{\theta}). \tag{9}$$

The product form couples Fourier coefficients across active coordinates while keeping the composition operator fixed and commutative. The masks determine the variables in each product, so different observables can share the same factorization principle while depending on different circuit coordinates.

Moment readout and Hamiltonian recombination When the supervised outputs are observable moments, a Hamiltonian energy with coefficients $c_o(J)$ is obtained by the linear readout

$$\hat{E}_J(\boldsymbol{\theta}) = \sum_o c_o(J)\hat{m}_o(\boldsymbol{\theta}). \tag{10}$$

Changing the Hamiltonian coefficients therefore reuses the fitted moment representation without retraining. This reuse follows from the observable-level supervision and the linear energy map; the experiments specify which observable bases and Hamiltonian families use this readout.

Table 1: **Independent control benchmarks.** Mean ± sample SD over five independently generated datasets; lower is better for all metrics. Each row names the CQS variant and its matched reference: local-mask CQS-P vs. complete Fourier, CQS-R vs. Query-GRU, and CQS-P+C vs. Fourier+C. Reported metric values are scaled by 10^{-2}; settings and counts are unscaled.

Task / metric	CQS variant	Matched reference
Masked moments / MSE	CQS-P: $7.735_{\pm 0.485}$	Complete Fourier: $8.643_{\pm 0.230}$
Trajectory / MAE	CQS-R: $19.271_{\pm 0.380}$	Query-GRU: $18.980_{\pm 0.327}$
2×3 VQE / energy	CQS-P+C: $-300.477_{\pm 0.536}$	Fourier+C: $-297.249_{\pm 2.149}$
3×3 VQE / energy	CQS-P+C: $-450.593_{\pm 1.227}$	Fourier+C: $-422.701_{\pm 36.679}$

4.4 Analytic Continuation for Product Objectives

For VQE search, the fitted Fourier objective can be smoothed analytically before optimization returns to the unsmoothed predictor. We optimize

$$\widetilde{E}_{J,\sigma}(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\epsilon} \sim \mathcal{N}(0,\sigma^2 I_D)}[\hat{E}_J(\boldsymbol{\theta} + \boldsymbol{\epsilon})]. \tag{11}$$

Here D is the number of independently perturbed physical control coordinates, and I_D is the $D \times D$ identity. Gaussian convolution of a Fourier representation multiplies a coefficient with frequency vector $\mathbf{w}$ by $\exp(-\sigma^2 \|\mathbf{w}\|_2^2/2)$. Independent control perturbations preserve Eq. (9), so the same product evaluation applies after scaling each factor's frequency-q coefficients by $\exp(-q^2\sigma^2/2)$. Fitted weights remain fixed throughout continuation. The annealing schedule and matched optimization controls are reported in Appendix L.

5 Experiments

The experiments follow the three uses of the proposed representations: expectation prediction from shared scalar labels, trajectory prediction with reusable sequential computation, and surrogate-based VQE search. The TFIM and FSPT sweeps establish the main task-dependent trends. Independent-data controls then isolate dependency information, per-output depth, and the contribution of continuation to optimization endpoints.

5.1 Experimental Setup

All reported simulation metrics use exact held-out targets or exact endpoint energies for evaluation. The TFIM and FSPT sweeps use paired splits on one control pool per size, while the VQE and independent-control studies use paired datasets, labels, starts, and method comparisons. Appendix C gives dataset-level contrasts and per-dataset values.

We implement the public predictive-surrogate algorithms of Liao et al. (2026): support-two trigonometric features for TFIM and sampled expanded-gate features for FSPT. These references are fixed-feature or sampled-feature regressors, so their coverage is determined by the specified support cap and sampled basis rather than learned during fitting. Appendices L and M give objectives, regularization, stopping rules, and label sources. Additional controls include squared-L2 trigonometric ridge fits, state-reset GRUs, period MLPs, complete Fourier regression, and Query-GRU.

Table 1 collects the matched controls used below to examine moment fitting, full-trajectory computation, and VQE endpoints on independently generated datasets.

5.2 Expectation Fitting under Shared Labels

The first experiment fixes the scalar supervision and compares how well each representation predicts an expectation. Two-layer one-dimensional TFIM circuits use $N = 8, 10, 12, 14$ and 2,000/400/400 training/validation/test controls. Every learner receives the same h_{cs}-style energy estimates with $T = 32$ shots per input. Test energies are exact and unnormalized. Both neural models use a 250-epoch cap and validation checkpoint selection. Appendix B.1 specifies the circuit and label generator.

Table 2: **TFIM final-layer energy prediction.** Exact-test MAE, mean ± sample standard deviation over three paired splits of one pool per N. NC denotes the Nature Communications predictive-surrogate reference of Liao et al. (2026). All predictors share $T = 32$ scalar labels. Here h_{cs} denotes the support-two classical-shadow-style NC fitting profile, and h_{qs} denotes the sampled-feature NC fitting profile; both use the same shared scalar labels in this table. NC-tuned selects regularization on validation labels. All predictions are direct, without continuation. Bold marks the lowest row mean. Means and sample SDs: $\times 10^{-1}$, with settings/counts unscaled.

N	CQS-R	CQS-P	NC h_{qs}	NC h_{cs}	NC-tuned
8	$1.625_{\pm 0.158}$	$\mathbf{1.562}_{\pm 0.031}$	$4.021_{\pm 0.647}$	$3.478_{\pm 0.142}$	$2.124_{\pm 0.206}$
10	$\mathbf{2.032}_{\pm 0.131}$	$2.218_{\pm 0.065}$	$5.507_{\pm 0.288}$	$5.025_{\pm 0.401}$	$2.382_{\pm 0.059}$
12	$\mathbf{2.393}_{\pm 0.061}$	$2.518_{\pm 0.025}$	$7.833_{\pm 0.412}$	$6.389_{\pm 0.433}$	$2.648_{\pm 0.053}$
14	$3.069_{\pm 0.288}$	$\mathbf{2.690}_{\pm 0.197}$	$8.788_{\pm 0.410}$	$7.631_{\pm 0.377}$	$3.149_{\pm 0.077}$

Regularization tuning accounts for much of the gap to the reference predictor (Table 2). The learned models retain lower mean MAE than the validation-tuned reference across sizes, but neither dominates the other under noisy supervision; the recurrent margin at the largest size is small relative to split variation. With exact supervision, the product model is lower at every tested size (Table 34), showing that the product advantage depends on label regime as well as circuit family. Appendix M gives the shot and allocation studies.

Moment-structure control. The scalar comparison tests prediction under shared labels; a separate moment-supervised control tests where the product model gains accuracy. Correct masks outperform full and matched-random masks (Appendix L.3), and local masks improve both products and complete Fourier regression. Under the same mask and frequency coverage, products retain a 10.5% MSE reduction (Table 1), while the smaller energy-MAE difference shows that moment accuracy need not translate proportionally to the Hamiltonian-weighted sum.

5.3 Sequential Composition for Trajectory Prediction

The second experiment tests the ordered-output setting for which CQS-R is designed. We predict local-Z expectations of an FSPT chain from $|0\rangle^{\otimes N}$ under the drive

$$U_F(\delta, \mathbf{J}) = \left[\prod_{i=0}^{N-1} e^{iJ_i Z_{i-1} X_i Z_{i+1}}\right] \left[\prod_{i=0}^{N-1} R_{X_i}(\pi - 2\delta)\right], \tag{12}$$

with missing boundary Z factors replaced by the identity, $\delta \sim \mathcal{U}[0, 2]$, and $J_i \sim \mathcal{U}[0, 1]$ (Else & Nayak, 2016; von Keyserlingk & Sondhi, 2016; Potter et al., 2016; Zhang et al., 2022). The shared rotation angle is $\phi = \pi - 2\delta$. We cross $N \in \{4, 6, 8, 10\}$ with $K \in \{6, 12, 24\}$, using paired 2,000/400/400 trajectory splits and exact labels. All K periods are supervised and evaluation uses new controls. The product adapter uses Fourier order $Q_\phi = 1$ for this angle, a compact first-harmonic truncation whose excluded early-period harmonics are derived and tested in Appendix M.5. Complete settings and results appear in Appendix E and Table 11.

Shared recurrence gives the lowest mean MAE among the three matrix predictors across all tested sizes and horizons (Figure 2). Its margin over NC narrows as the supervised horizon grows. Diagnostics on shared early-period targets show a growing fitting error as the horizon extends, beyond the effect of adding harder late-period outputs (Appendix J). Sequential reuse improves prediction in this regime, while longer responses remain difficult to fit.

The independent controls separate prediction depth from full-trajectory computation. Query-GRU restores per-output depth without sharing states and reaches similar observed accuracy to CQS-R (Table 1), while the restricted-budget Query-GRU has larger error. For $K = 12$, CQS-R uses 12 recurrent updates to produce a full trajectory, compared with 78 updates for Query-GRU. The recurrent advantage is therefore computation reuse across period queries, rather than a demonstrated accuracy gap over depth-matched independent queries.

At $N = 8, K = 12$, expanding $Q_\phi = 1$ to 8 lowers first-period MAE from 0.1291 to 0.0506, while periods 2–12 remain at 0.3101 versus 0.3099. Full-trajectory MAE falls by 2.3%. With rank

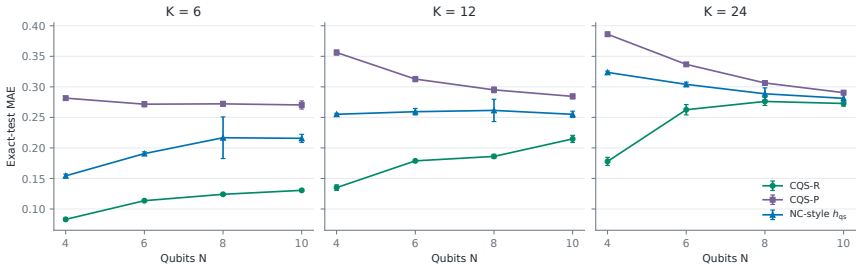


Figure 2: **FSPT qubit-count and supervised-horizon sweep.** Three paired partition/initialization runs per point, with bars showing sample standard deviations. All periods are supervised. The recurrent model uses $k/24$. Products use $Q_\phi = 1$ and distinct period heads, as does NC for period identification. CQS-R has the lowest mean MAE throughout, with a narrower NC gap at larger N and K.

Table 3: **Independent VQE datasets.** True endpoint energy, mean $\pm$ sample SD over three independently generated datasets, each retaining all 20 starts. NC is h_{cs}-style for chains and h_{qs}-style for grids. Lower is better. Reported metric values: $\times 10^0$; settings/counts unscaled.

Setting	CQS-R	CQS-P+C	NC	NC+C
1D $N = 6, L = 1$	$-1.118_{\pm 0.735}$	$\mathbf{-2.631}_{\pm 0.114}$	$-2.547_{\pm 0.045}$	$-2.581_{\pm 0.012}$
1D $N = 6, L = 2$	$1.671_{\pm 1.348}$	$\mathbf{-1.286}_{\pm 0.431}$	$-0.718_{\pm 0.091}$	$-0.739_{\pm 0.076}$
2D $2 \times 3, L = 1$	$-2.000_{\pm 0.100}$	$\mathbf{-2.667}_{\pm 0.153}$	$-2.199_{\pm 0.100}$	$-2.633_{\pm 0.115}$
2D $2 \times 3, L = 2$	$-2.661_{\pm 0.587}$	$\mathbf{-3.009}_{\pm 0.003}$	$-2.684_{\pm 0.291}$	$-2.710_{\pm 0.262}$
2D $3 \times 3, L = 1$	$-2.566_{\pm 0.306}$	$\mathbf{-4.500}_{\pm 0.000}$	$-3.031_{\pm 0.233}$	$-4.497_{\pm 0.003}$
2D $3 \times 3, L = 2$	$-3.713_{\pm 1.004}$	$\mathbf{-4.506}_{\pm 0.014}$	$-4.272_{\pm 0.242}$	$-4.281_{\pm 0.212}$

and training rules matched, the first-period gain identifies a frequency-coverage limit of the tested product adapter. The nearly unchanged later-period error shows that harmonic expansion alone does not resolve the trajectory gap; rank, data, and longer-horizon fitting effects are examined in the appendices.

5.4 Moment Representations for Optimization and Reuse

VQE tests the fitted representations through the controls selected by an optimizer. We use $N = 6$ chains with independent gate angles and 2×3 and 3×3 grids with shared angles per layer. Methods share measurement records, 20 starts per seed, and 100 Adam updates. CQS-R and NC fit scalar energies, whereas CQS-P fits measured moments; exact simulation scores every endpoint. The resulting endpoints compare complete pipelines with different representations and supervision objectives.

The CQS-P+C pipeline attains the lowest mean endpoint energy on the chain and grid settings (Table 3). Matched controls distinguish its sources of reuse and endpoint improvement: moment readouts recombine across Hamiltonian couplings, direct/+C searches isolate continuation, and moment-supervised Fourier fits compare representations under the same search.

Grid complexity is determined by effective controls, not qubit count alone: the plus initialization removes the first shared R_X as a global phase, leaving one effective coordinate at $L = 1$ and three at $L = 2$. The feasible reference $E_{\text{ref}} = -N/2$ and the near-equivalence of single-layer product and complete-Fourier endpoints motivate focusing interpretation on the two-layer grids (Appendices L.5 and L.6).

Both moment models use identical records, starts, and continuation, but differ in parameterization, frequency selection, and regularization. The two-layer endpoints in Table 1 are lower for CQS-P+C. Endpoint distributions in Figure 3 show that products more often reach the feasible-reference neighborhood, while few starts improve appreciably beyond it; the principal gain is search reliability rather than a large improvement in attainable energy. Appendix C reports the full thresholds.

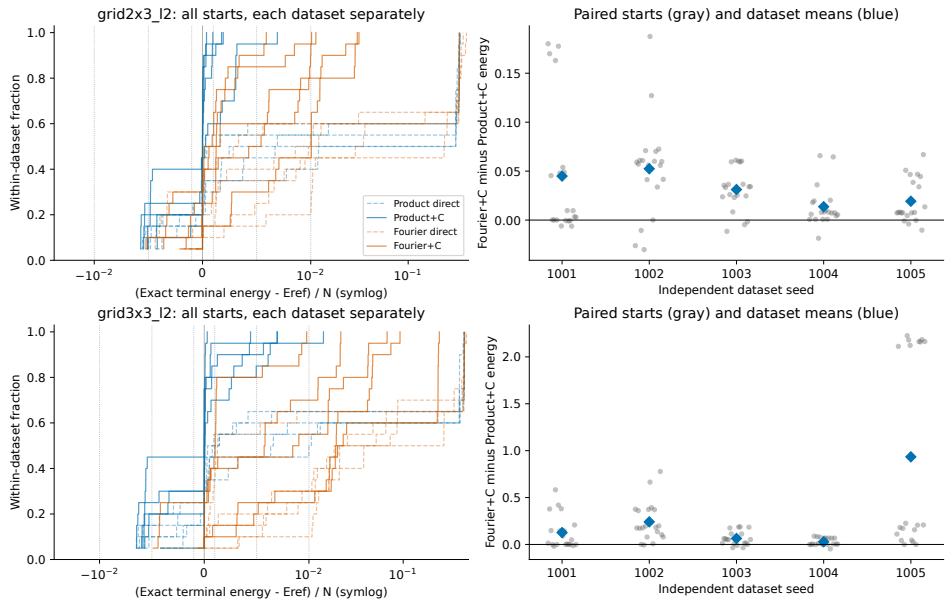


Figure 3: **Two-layer VQE endpoint distributions.** CQS-P+C reaches the feasible-reference neighborhood more consistently than Fourier+C; paired restart differences and dataset means are shown on the right.

Direct optimization of the same fitted products yields much worse grid endpoints, locating the continuation gain in the search objective (Appendix L, Table 26). Appendix D further separates representation, supervision, and continuation; Hamiltonian reuse follows independently from the linear moment readout.

Across tasks, circuit structure supports different forms of reuse. Observable masks drive the largest controlled moment-prediction gain, with an additional product benefit under matched masks and frequency coverage. Shared recurrence gives the lowest mean error in the fully supervised trajectory sweep and reuses updates across period queries; a depth-matched Query-GRU reaches similar accuracy through separate computation. Moment readouts support Hamiltonian recombination, and continuation improves grid-search reliability.

Held-out prediction error and optimization endpoints capture different aspects of surrogate performance. On two-layer grids, product continuation mainly increases the frequency of reaching the feasible-reference neighborhood, with limited improvement beyond it. The useful representation thus depends on prediction accuracy, computational work, and the downstream quantity being estimated or optimized.

6 Conclusions

Compositional Quantum Surrogates match their representation to the outputs a task reuses. CQS-R shares recurrent computation across trajectory queries, while CQS-P combines circuit masks with low-rank Fourier products for observable moments. The experiments show gains in moment fitting, full-trajectory computation reuse, Hamiltonian recombination, and continuation-guided attainment on the evaluated grids. Together, they support choosing surrogate structure according to the downstream query rather than relying on a single representation across tasks.

The evidence is limited to the evaluated circuits and supervision regimes, with persistent later-period FSPT product error and VQE endpoints that depend on the fitted surrogate and continuation schedule (Appendix O). Future work should examine adaptive product frequencies and ranks, longer-horizon recurrent behavior, and broader finite-shot and device-noise settings.

REPRODUCIBILITY STATEMENT

Appendix B specifies task inputs, labels, splits, model selection, and evaluation. Saved predictions and run summaries support the reported aggregates; figure scripts retain source values. Both neural models use PyTorch (Paszke et al., 2019). We provide detailed code for reproducing the experiments in the supplementary material. Appendices L.3, L.4, L.6, and M.2 detail the mask, training-budget, complete-Fourier, and tuned-NC controls, respectively.

ETHICS STATEMENT

This work studies quantum surrogate learning through controlled simulations, public software tools, and documented benchmark-style circuit families. It does not involve human subjects, new dataset collection, personally identifiable information, or deployment in a real-world decision system. The results are simulation-based; their interpretation should not be extended to unvalidated hardware performance or device-independent resource claims. The experimental boundaries, paired controls, and reproducibility details define the scope of the evidence.

AI USE STATEMENT

OpenAI's Codex assisted with manuscript language editing, wording refinement, LaTeX formatting, and consistency checks. It also assisted with experiment planning, code formatting, and drafting experiment scripts. The authors are responsible for the experiments, numerical values, scientific claims, citations, and final submission.

REFERENCES

Scott Aaronson. Shadow tomography of quantum states. In *Proc. ACM SIGACT Symp. Theory Comput.*, pp. 325–338, 2018.

Leonardo Banchi, Edward Grant, Andrea Rocchetto, and Simone Severini. Modelling non-markovian quantum processes with recurrent neural networks. *New J. Phys.*, 20(12):123030, 2018.

Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602–606, 2017.

Matthias C. Caro, Elies Gil-Fuster, Johannes Jakob Meyer, Jens Eisert, and Ryan Sweke. Encoding-dependent generalization bounds for parametrized quantum circuits. *Quantum*, 5:582, 2021.

Matthias C. Caro, Hsin-Yuan Huang, M. Cerezo, Kunal Sharma, Andrew Sornborger, Lukasz Cincio, and Patrick J. Coles. Generalization in quantum machine learning from few training data. *Nat. Commun.*, 13(1):4919, 2022.

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

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

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proc. Conf. Empirical Methods Nat. Lang. Process.*, pp. 1724–1734, 2014.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

Dominic V. Else and Chetan Nayak. Classification of topological phases in periodically driven interacting systems. *Phys. Rev. B*, 93(20):201103, 2016.

Dominic V. Else, Bela Bauer, and Chetan Nayak. Floquet Time Crystals. *Phys. Rev. Lett.*, 117(9): 090402, 2016.

Suguru Endo, Zhenyu Cai, Simon C. Benjamin, and Xiao Yuan. Hybrid Quantum-Classical Algorithms and Quantum Error Mitigation. *J. Phys. Soc. Jpn.*, 90(3):032001, 2021.

Elies Gil-Fuster, Jens Eisert, and Carlos Bravo-Prieto. Understanding quantum machine learning also requires rethinking generalization. *Nat. Commun.*, 15(1):2277, 2024.

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

Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nat. Phys.*, 16(10):1050–1057, 2020.

Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R. McClean. Power of data in quantum machine learning. *Nat. Commun.*, 12 (1):2631, 2021.

Hsin-Yuan Huang, Richard Kueng, Giacomo Torlai, Victor V. Albert, and John Preskill. Provably efficient machine learning for quantum many-body problems. *Science*, 377(6613):eabk3333, 2022.

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

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

Tamara G. Kolda and Brett W. Bader. Tensor Decompositions and Applications. *SIAM Rev.*, 51(3): 455–500, 2009.

Wei-You Liao, Yuxuan Du, Xinbiao Wang, Tian-Ci Tian, Yong Luo, Bo Du, Dacheng Tao, and He-Liang Huang. Demonstration of efficient predictive surrogates for large-scale quantum processors. *Nat. Commun.*, 17(1):4731, 2026.

Bethany Lusch, J. Nathan Kutz, and Steven L. Brunton. Deep learning for universal linear embeddings of nonlinear dynamics. *Nat. Commun.*, 9(1):4950, 2018.

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

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

I. V. Oseledets. Tensor-Train Decomposition. *SIAM J. Sci. Comput.*, 33(5):2295–2317, 2011.

Mateusz Ostaszewski, Edward Grant, and Marcello Benedetti. Structure optimization for parameterized quantum circuits. *Quantum*, 5:391, 2021.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Inf. Process. Syst.*, pp. 8026–8037, 2019.

Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.*, 5(1):4213, 2014.

Andrew C. Potter, Takahiro Morimoto, and Ashvin Vishwanath. Classification of Interacting Topological Floquet Phases in One Dimension. *Phys. Rev. X*, 6(4):041001, 2016.

Ali Rahimi and Benjamin Recht. Random features for large-scale kernel machines. In *Advances in Neural Inf. Process. Syst.*, pp. 1177–1184, 2007.

Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.

Albert Saiapin and Kim Batselier. Tensor Network Based Feature Learning Model. In *Proc. Int. Conf. Artificial Intell. & Stat.*, pp. 3277–3285, 2025.

Maria Schuld and Nathan Killoran. Quantum Machine Learning in Feature Hilbert Spaces. *Phys. Rev. Lett.*, 122(4):040504, 2019.

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

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(3):032430, 2021.

Ryan Shaffer, Lucas Kocia, and Mohan Sarovar. Surrogate-based optimization for variational quantum algorithms. *Phys. Rev. A*, 107(3):032415, 2023.

E. M. Stoudenmire and David J. Schwab. Supervised Learning with Tensor Networks. In *Advances in Neural Inf. Process. Syst.*, pp. 4799–4807, 2016.

G.I. Struchalin, Ya. A. Zagorovskii, E.V. Kovlakov, S.S. Straupe, and S.P. Kulik. Experimental Estimation of Quantum State Properties from Classical Shadows. *PRX Quantum*, 2(1):010307, 2021.

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

Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. Error Mitigation for Short-Depth Quantum Circuits. *Phys. Rev. Lett.*, 119(18):180509, 2017.

Giacomo Torlai, Guglielmo Mazzola, Juan Carrasquilla, Matthias Troyer, Roger Melko, and Giuseppe Carleo. Neural-network quantum state tomography. *Nat. Phys.*, 14(5):447–450, 2018.

C. W. von Keyserlingk and S. L. Sondhi. Phase structure of one-dimensional interacting Floquet systems. I. Abelian symmetry-protected topological phases. *Phys. Rev. B*, 93(24):245145, 2016.

C. W. von Keyserlingk, Vedika Khemani, and S. L. Sondhi. Absolute stability and spatiotemporal long-range order in Floquet systems. *Phys. Rev. B*, 94(8):085112, 2016.

Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nat. Commun.*, 12(1): 6961, 2021.

Xu Zhang, Wenjie Jiang, Jinfeng Deng, Ke Wang, Jiachen Chen, Pengfei Zhang, Wenhui Ren, Hang Dong, Shibo Xu, Yu Gao, Feitong Jin, Xuhao Zhu, Qiujiang Guo, Hekang Li, Chao Song, Alexey V. Gorshkov, Thomas Iadecola, Fangli Liu, Zhe-Xuan Gong, Zhen Wang, Dong-Ling Deng, and H. Wang. Digital quantum simulation of Floquet symmetry-protected topological phases. *Nature*, 607(7919):468–473, 2022.

A Appendix Roadmap

Appendix B defines the tasks, labels, splits, and evaluation metrics used throughout the paper. Appendix C reports the independent-data controls for the representation and computation claims. Appendix D separates representation, supervision, and continuation effects in VQE. Appendix E provides extended task results and diagnostics, Appendix G gives method, implementation, and mask-complexity details, and Appendix O discusses limitations and future work.

B Experimental task definitions

This section defines the learning and evaluation problems independently of model architecture. Implementation hyperparameters and auxiliary controls are reported in the later appendices. In prediction tasks, a control vector or complete control sequence is the splitting unit: all its target coordinates belong to one partition. Training labels determine fitted weights; validation labels determine checkpoints or validation-selected hyperparameters. Test targets are used for scoring, not supplied to the predictor. Simulation tests use exact expectations, while hardware tests use measured expectation estimates.

Notation conventions. N counts qubits, L circuit layers, and K evaluated drive periods; $K_{\max}$ is the fixed period-normalization constant for a specified protocol. Controls are denoted by $\mathbf{x}$, or by $\boldsymbol{\theta}$ when expressed as gate angles, whereas ω denotes learned network parameters. Angles are in radians, with $R_P(\theta) = e^{-i\theta P/2}$ for a Pauli generator P. Observable indices j and o refer to the trajectory and moment-output conventions, respectively, and m counts outputs. R is the product rank, Q a Fourier cutoff, and C_o an observable's retained control set. In product-complexity formulas, D counts allocated factor slots; repeated-control FSPT slots are distinct from independent physical coordinates. The identity dimension in Eq. (11) instead matches the perturbed control vector. h_k denotes a recurrent hidden state, while scalar h denotes the transverse-field coupling.

For test controls $\mathcal{S}$, evaluated periods $\mathcal{K}$, and observable indices $\mathcal{O}$, trajectory MAE is

$$\mathrm{MAE}(\mathcal{S},\mathcal{K},\mathcal{O}) = \frac{\sum_{\mathbf{x}\in\mathcal{S}}\sum_{k\in\mathcal{K}}\sum_{j\in\mathcal{O}}|\hat{y}_{k,j}(\mathbf{x}) - y_{k,j}(\mathbf{x})|}{|\mathcal{S}|\,|\mathcal{K}|\,|\mathcal{O}|}. \tag{13}$$

For the fitted predictor $f(\mathbf{x}) = (\hat{y}_{k,j}(\mathbf{x}))_{k,j}$, replacing the absolute residual by its square gives $\mathcal{R}(f)$, the empirical MSE. Scalar tasks omit the period and observable averages. Unless a task specifies otherwise, $R^2 = 1 - \sum_e(\hat{y}_e - y_e)^2 / \sum_e(y_e - \bar{y})^2$ pools all evaluated entries e. Here $\bar{y}$ is the mean target over evaluated entries. Pearson correlation uses the same entries:

$$r_{\mathrm{P}} = \frac{\sum_e(\hat{y}_e - \bar{\hat{y}})(y_e - \bar{y})}{\sqrt{\sum_e(\hat{y}_e - \bar{\hat{y}})^2 \sum_e(y_e - \bar{y})^2}}, \qquad \bar{\hat{y}} = |\mathcal{E}|^{-1}\sum_{e\in\mathcal{E}}\hat{y}_e,$$

where $\mathcal{E}$ is the evaluated entry set. Correlation is undefined if either variance vanishes; R^2 is undefined if the target variance vanishes. Run-level metrics are summarized by their mean and sample standard deviation. Seeds 42/43/44 change partitions and neural initializations on fixed TFIM/FSPT pools, but generate separate data in VQE. Three GPU timing trials measure execution variability rather than training-seed variability. Unless a caption states otherwise, reported metric values use the scale factor named in the caption, while settings and counts remain unscaled.

B.1 TFIM final-layer energy prediction

Input and physical target. For each $N \in \{8, 10, 12, 14\}$, an input is $\mathbf{x} \in [-1, 1]^{2\times(2N-1)}$, with independently uniform coordinates. Each of two layers applies H on all qubits, then N local R_X rotations and $N-1$ nearest-neighbor R_{ZZ} rotations. The initial state is $|0\rangle^{\otimes N}$. The scalar target is the final expectation of $-J\sum_{i=0}^{N-2} Z_iZ_{i+1} - h\sum_{i=0}^{N-1} X_i$, with $J = -0.1, h = -0.5$. Energy is not divided by N.

Learning and evaluation. A model maps all layer controls to this one final energy, using only final-energy labels for supervision. Each run uses 2,000/400/400 train/validation/test controls. Table 2

supplies identical $T = 32$ scalar shadow-derived labels to every learner. The exact-label task uses exact energies for training and validation instead. Both score exact-test MAE. This task tests generalization to new controls at fixed circuit depth. Appendix K specifies the scalar noise model and Appendix M gives the additional fitting profiles.

B.2 One- and two-dimensional VQE pre-training

Optimization context. Analytic quantum gradients (Schuld et al., 2019) and stochastic-gradient methods (Sweke et al., 2020) provide direct circuit-optimization tools. Sequential coordinate updates exploit the dependence on a selected gate parameter (Nakanishi et al., 2020; Ostaszewski et al., 2021), while surrogate-based optimization uses a fitted response to guide subsequent queries (Shaffer et al., 2023). Our pre-training protocol fits a reusable moment predictor and compares search endpoints on that fitted model. Cost-dependent trainability results (Cerezo et al., 2021b) further motivate separating representation accuracy from optimization success; the continuation controls evaluate that distinction empirically.

Input and learned objective. The input is a variational control vector $\boldsymbol{\theta}$ with coordinates uniform on $[-\pi, \pi]$. Circuits prepare $|+\rangle^{\otimes N}$ once, followed by $L = 1$ or 2 layers of R_X and open-edge R_{ZZ} gates. One-dimensional chains use $N = 6$ and independent gate angles; two-dimensional 2×3 and 3×3 grids share one R_X and one R_{ZZ} angle within each layer. The chain energy is $\langle -J \sum ZZ - h \sum X \rangle$; the grid energy is $\langle J \sum ZZ + h \sum X \rangle$, with $J = -0.1, h = -0.5$ for the primary comparison. Unlike the final-layer prediction task, these circuits do not repeat Hadamards at every layer.

Supervision and output. Chains use 2,000/400/400 controls and ten random-Pauli shots per measured control. Grids use 160/40/200 controls and 10,000 shots in each of the all-X and all-Z groups. Within a run, methods share the acquired records. CQS-R and NC fit scalar energies; CQS-P fits the measured moment vector and combines it with Hamiltonian coefficients. The downstream output is an optimized control vector, not merely an energy prediction. Each fitted surrogate receives the same 20 starting controls and 100 Adam updates. Direct and +C optimization use the objectives specified in Appendix L; exact energy does not guide their updates.

Metric. The exact simulator evaluates all returned controls. Equation (2) averages their 20 true endpoint energies within each seed, then reports the mean and sample standard deviation across seeds. Lower is better, and every restart contributes to the metric. This protocol measures surrogate-assisted pre-training before any subsequent quantum refinement. Exact-energy Adam is reported only as a local-optimization diagnostic. The primary VQE table reports independent seeds 45/46/47. The analytic grid reference and complete Fourier comparison are defined in Appendices L.5 and L.6.

B.3 Hamiltonian reuse

The two-layer, six-qubit chain additionally varies $J \in \{-0.5, -0.3, -0.1, 0.1, 0.3, 0.5\}$ while keeping the circuit family, $h = -0.5$, controls, and measured moment records fixed. A fitted CQS-P moment predictor returns $\hat{\mathbf{m}}(\boldsymbol{\theta})$ and forms a new objective by the linear readout $\hat{E}_J = \sum_o c_o(J)\hat{m}_o$ without refitting its weights. The CQS-R and NC predictors instead fit a scalar energy model for each coupling using the same acquired records. Each coupling then uses the matched VQE starting controls, update budget, and true-endpoint metric above. This task measures reuse across Hamiltonian coefficients within a fixed observable basis. The six chain couplings plus the other five primary settings give eleven distinct conditions; $J = -0.1$ in the two-layer chain is counted once.

B.4 Fully supervised FSPT trajectory prediction

Physical context. Floquet time crystals exhibit subharmonic temporal order (Else et al., 2016; von Keyserlingk et al., 2016). The boundary response considered here is evaluated in the specified symmetry-protected Floquet setting. We report boundary and bulk errors alongside aggregate trajectory prediction error so that local and averaged behavior can be read separately.

The input $\mathbf{x} = (\delta, J_0, \ldots, J_{N-1})$ has independent $\delta \sim \mathrm{Unif}[0, 2]$ and $J_i \sim \mathrm{Unif}[0, 1]$. Repeated application of Eq. (12) to $|0\rangle^{\otimes N}$ defines $y_{k,i}(\mathbf{x}) = \langle Z_i \rangle_k$. A predictor returns the $K \times N$ matrix

$\hat{\mathbf{y}}(\mathbf{x})$ from controls alone. Training and validation supervise every $k \in \{1, \ldots, K\}$, and evaluation uses the same periods at previously unseen controls. Each period is identified either by an input index or by a distinct output head.

The matrix varies $N \in \{4, 6, 8, 10\}$ and $K \in \{6, 12, 24\}$, with 2,000/400/400 controls per paired run. Shorter horizons use prefixes of the same 24-period targets; the GRU/MLP period input is $k/24$. Models are refitted for every (N, K), so varying N or K is a workload sweep, not cross-size or zero-shot horizon transfer. Primary MAE equally weights all test controls, periods, and sites in Eq. (13). Period-wise, boundary, and bulk errors restrict the corresponding index sets before averaging.

B.5 Finite-shot sensitivity and measurement allocation

Both tasks use the $N = 8, L = 2$ final-energy definition in Appendix B.1, a fixed input pool, paired partitions, and exact test energies. Noisy scalar energy labels come from the existing termwise marginal shadow simulation. Identical labels are supplied to both CQS architectures and both NC fitting profiles; no learner receives extra target observations. The noise model does not preserve all correlations of joint multi-qubit hardware outcomes.

The shot-sensitivity task fixes 2,000 training and 400 validation controls and varies $T \in \{8, 16, 32, 64, 128, 256, 512\}$, alongside an exact-label condition. Labels use shared measurement prefixes. The allocation task instead fixes a requested budget cap $B \in \{16000, 32000, 64000, 128000\}$ and varies $M \in \{125, 250, 500, 1000, 2000\}$. With $V = \max(50, \min(400, \lceil M/5 \rceil))$, the implemented allocation is

$$T(B, M) = \max\left(1, \min\left(512, \left\lfloor \frac{B}{M + V} \right\rfloor\right)\right), \qquad B_{\text{actual}} = (M + V)T(B, M). \tag{14}$$

For the tested configurations, $B_{\text{actual}} \leq B$; integer rounding and the 512-shot cap can leave budget unused. Both tasks report exact-test MAE across three paired seeds, using the 80-epoch neural cap. These studies characterize error under the stated noise model and resource allocation. Allocation choices based on test error are retrospective comparisons rather than deployment policies. These fits differ from the main-table $T = 32$ comparison in measurement randomness and neural training cap.

B.6 IBM TFIM measured-prefix prediction

For $N \in \{4, 6, 8, 10\}$, the stored inputs are ten-layer control sequences with $2N - 1$ parameters per layer, sampled in $[-1, 1]$. The circuit prepares $|+\rangle^{\otimes N}$ once, then applies local R_X and nearest-neighbor R_{ZZ} blocks. A prefix k uses only the first k blocks. The output vector contains N local X_i expectations and $N - 1$ nearest-neighbor $Z_i Z_{i+1}$ expectations, rather than one scalar energy. EstimatorV2 supplies target estimates at requested precision 0.08; this is not a fixed shots-per-input specification.

Each size has 24 training, 8 validation, and 16 test controls. Training labels and the GRU validation criterion use $k \leq 4$; primary evaluation uses $k = 5, \ldots, 10$ on test controls. Prediction receives controls, not their measured outcomes. MAE and MSE pool late-prefix observable entries against hardware estimates, not noiseless simulator truth. The $N = 4$ measurement repeats use sampling seeds 20260826, 20260827, and 20260828 in separate acquisition runs. The other sizes each use seed 20260826. Thus seed and acquisition variation are not separated. The $N = 4$ scaling dataset is also the first repeat, giving six unique datasets in total. No new hardware measurements are acquired for the static controls below.

Full-prefix and current-layer controls. Writing $d = 2N - 1$, the history-matched static input is

$$\mathbf{u}_k = [\text{vec}(\mathbf{x}_1, \ldots, \mathbf{x}_k), \mathbf{0}_{(10-k)d}, k/10] \in \mathbb{R}^{10d+1}. \tag{15}$$

Its feature map contains a constant, coordinate-wise sine and cosine, and cosine–cosine and sine–sine products for up to 600 coordinate pairs, sampled once with seed 123. This is the original padded trigonometric representation. We fit ridge with an intercept, no feature standardization, and fixed $\alpha = 1$. The same fixed-regularization procedure also fits the current-layer input $\mathbf{v}_k = [\mathbf{x}_k, k/10] \in \mathbb{R}^{2N}$, which omits preceding controls. Training uses only the 24 training controls at $k \leq 4$, with no validation refit.

Results and remaining depth shift. The fixed full-prefix ridge is less accurate than CQS-R on the stored hardware datasets, while current-layer ridge is also weaker. These comparisons use static controls with fixed regularization; they test whether simple trigonometric features over the available prefix history reproduce the recurrent predictor under the recorded acquisition protocol.

Full-prefix inputs align available control history, but not invariance to depth. At $N = 4, 6, 8, 10$, respectively, 807/1343, 820/1423, 881/1503, and 937/1583 trigonometric features are constant during training and become variable at late test prefixes. Cosines of padded controls equal one, so this includes constant interactions as well as zero features. Their late-coordinate dependence is unidentifiable from early-only training, and $k/10$ also exceeds its training range. The current-layer representation avoids these inactive control positions but sacrifices history. Neither comparison isolates recurrence alone. The original MLP additionally retains its row-level internal validation rule.

Table 4: Hardware late-prefix MAE on all six stored datasets. Static ridge controls use fixed regularization. Only full-prefix ridge receives the complete preceding controls. Lower is better. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

N, seed	CQS-R	Current-layer	Full-prefix $\alpha = 1$
4, 20260826	2.423	2.631	3.217
4, 20260827	2.594	3.109	3.616
4, 20260828	2.948	3.216	3.457
6, 20260826	2.565	3.022	3.069
8, 20260826	2.194	2.747	2.924
10, 20260826	2.198	2.617	3.069

B.7 Mechanism and approximation diagnostics

These experiments reuse the prediction or optimization targets above and change a specified component; they do not define additional quantum applications.

History interventions. Inference-only reset replaces every carried GRU state by zero while keeping the fitted weights. Retrained reset uses that operator during training, validation, and testing with the original schedule. The outcome is prediction error on the same test controls and evaluated periods. The first measures dependence of a fitted solution on history; the second measures performance attainable under the reset architecture. A period MLP supplies a separately fitted no-history comparator.

Horizon, width, and readout probes. Common-prefix comparisons evaluate identical first-six-period targets using models trained on different full horizons; train, validation, and test errors are distinguished. Target-block analyses partition the supervised horizon into periods 1–6, 7–12, and 13–24, then compare prediction errors and target scales. Width/budget interventions change widths 128/256 and epoch caps 300/600 at fixed $K = 24$ and matched splits. Frozen-state probes fit shared or period-specific linear readouts to saved recurrent features using training labels, with validation-only selection and held-out-control scoring. Their target remains local-Z prediction; a change in readout does not establish a jointly retrained architecture's performance. Detailed protocols are in Appendix J and the implementation notes.

Continuation and harmonic diagnostics. The direct-versus-+C VQE comparison keeps fitted weights and initial controls fixed, changes the optimization objective, and scores all true endpoints as in Appendix B.2. The FSPT harmonic diagnostic instead fixes $J_i = 0.37$ and projects the exact first-period function over 4,096 equally spaced $\delta \in [0, 2]$ onto prescribed first- or third-order trigonometric spans. Its metric is projection MSE on that slice, not held-out performance of a newly trained surrogate. It tests an expressivity restriction, independent of the optimizer and number of multiplicative channels. Appendix M provides the exact function and residuals.

C Independent-Data Representation and Computation Controls

These comparisons generate five fresh control pools with data seeds 1001–1005, neural initialization seeds 2001–2005, and measurement seeds 3001–3005. VQE starts use seeds 4001–4005. Methods

share the data, labels, partitions, and starts within each comparison. Models and hyperparameters are selected using training and validation labels, and exact test values or endpoint energies provide evaluation. Each dataset has one initialization, so each complete data-and-fitting run contributes one value. The five dataset means receive equal weight. These datasets remain separate from the five-dataset controls below.

C.1 Representation Crossed with Observable Dependencies

The independent-angle $N = 6, L = 2$ chain uses 2,000/400/400 training/validation/test controls sampled uniformly from $[-\pi, \pi]$. Each training or validation control has ten joint random-Pauli shots, shared by all four arms. Test moment targets are exact. The local mask uses the circuit dependencies of Appendix L.3; the full mask excludes only the initial global-phase R_X controls. Both representations use harmonic order one. Product rank is 32, with Adam learning rate 10^{-3}, weight decay 10^{-4}, batch size 32, cap 300, and patience 30. The two product arms share initial factors and batch order.

Complete Fourier regression uses the exact tensor-product kernel for each output o,

$$K_o(\mathbf{x}, \mathbf{x}') = \prod_{d \in C_o} \left[1 + 2\cos(x_d - x'_d)\right]. \tag{16}$$

This is the inner product of complete tensor features with local basis $[1, \sqrt{2}\cos x_d, \sqrt{2}\sin x_d]$, including all interactions over the same active coordinates as the product model. We center the kernel and targets to leave the intercept unpenalized, normalize the centered kernel by its mean diagonal, and solve with $\bar{K}_o + n\lambda I_n$, where $\bar{K}_o$ is the centered, normalized $n \times n$ training Gram matrix and n is the training-control count. A single λ per arm minimizes mean validation moment MSE over $\{10^{-8}, 10^{-6}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$. Zero centered kernels predict the training mean. The kernel matrix notation is distinct from the trajectory horizon K. Kernel regression uses float64; product training uses float32.

Table 5: **Representation and mask.** Means ± sample SD over five independent datasets. Energy MAE uses $J = -0.1$. P denotes products and F complete Fourier regression. Reported metric values: $\times 10^{-2}$; settings/counts unscaled.

Arm	Moment MSE	Energy MAE
P-local	$7.735_{\pm 0.485}$	$31.606_{\pm 2.756}$
F-local	$8.643_{\pm 0.230}$	$31.874_{\pm 1.820}$
P-full	$19.197_{\pm 0.683}$	$46.426_{\pm 2.302}$
F-full	$18.873_{\pm 0.337}$	$46.364_{\pm 2.324}$

The five F-local minus P-local moment-MSE differences are 2.359×10^{-3}, 1.006×10^{-2}, 8.661×10^{-3}, 1.053×10^{-2}, and 1.379×10^{-2}. Full-minus-local differences average 1.146×10^{-1} for products and 1.023×10^{-1} for Fourier regression. The within-mask comparison changes coefficient factorization, regularization, and fitting together, so it measures the resulting learning procedures under matched dependency information.

C.2 Query-Wise Iteration and Shared Recurrence

The FSPT comparison uses $N = 8, K = 12$, fresh 2,000/400/400 trajectory pools, and exact local-Z labels for all periods. Controls follow Eq. (12). CQS-R, Reset-GRU, and Query-GRU have width 128 and 134,536 parameters; Period-MLP follows the width-256 architecture used in the matrix. All use Adam with learning rate 10^{-3}, weight decay 10^{-4}, batch size 32, cap 300, patience 30, and validation MSE selection. Period inputs retain $k/24$.

For query k, Query-GRU starts at $z_0^{(k)} = 0$ and computes

$$\begin{aligned} z_j^{(k)} &= \mathrm{GRU}_{\omega_{\mathrm{rec}}}\left(q_{\omega_{\mathrm{enc}}}([\mathbf{x}, k/24]), z_{j-1}^{(k)}\right), \quad j = 1, \ldots, k, \\ \hat{\mathbf{y}}_k(\mathbf{x}; \omega) &= d_{\omega_{\mathrm{out}}}(z_k^{(k)}). \end{aligned} \tag{17}$$

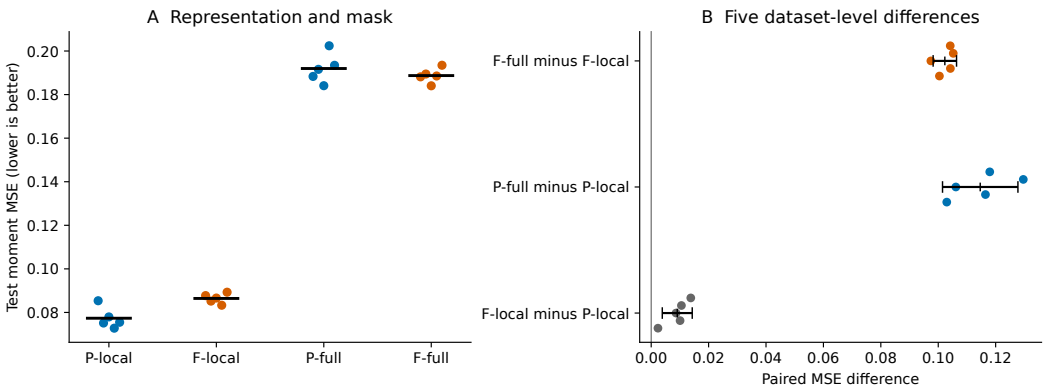


Figure 4: **Structure and representation effects.** Left: five dataset-level errors per arm, with mean bars. Right: paired differences with their means. Correct masks account for the larger improvement; products retain an additional within-mask moment-MSE benefit.

For a fixed query k, the embedding $q_{\omega_{\text{enc}}}([\mathbf{x}, k/24])$ stays constant across the k inner steps while the hidden state changes. Queries share weights but maintain separate states, and active queries are batched without inactive cell updates. CQS-R uses intermediate period encodings, whereas Query-GRU repeats the target-period encoding. The comparison captures both computation sharing and this difference in conditioning.

We also retain the validation-best Query checkpoint within the actual cumulative training-cell budget of CQS-R. Cell counts include the number of samples in each batch; validation work is recorded separately. The prefix is selected from epoch validations and the last whole batch within budget, using the same full Query training trajectory. All CQS-R and full Query runs reached 300 epochs, totaling 7,200,000 and 46,800,000 training-cell updates respectively.

Table 6: **Trajectory MAE on independent datasets.** The budget prefix is a checkpoint of the full Query fit, not an additional fit. MAEs and sample SDs: $\times 10^{-2}$; settings/counts unscaled.

Model	1001	1002	1003	1004	1005	Mean $\pm$ SD
CQS-R	19.430	19.065	18.985	19.869	19.007	$19.271_{\pm 0.380}$
Reset-GRU	28.447	30.097	28.162	30.453	27.949	$29.022_{\pm 1.164}$
Period-MLP	29.879	28.949	29.809	30.119	29.023	$29.556_{\pm 0.533}$
Query full	18.934	19.096	18.915	19.430	18.526	$18.980_{\pm 0.327}$
Query prefix	27.678	27.288	27.032	27.356	26.047	$27.080_{\pm 0.622}$
Training mean	33.334	32.764	33.354	33.142	32.391	$32.997_{\pm 0.413}$

Query-full minus CQS-R differences for seeds 1001–1005 are $(-0.4961, 0.0306, -0.0697, -0.4389, -0.4813) \times 10^{-2}$, using the same scale as Tables 1 and 6. On RTX 4090 GPUs, recorded training-stage time averages 238.4 seconds for CQS-R and 340.3 seconds for Query-GRU; evaluation of 400 test trajectories averages 12.87 and 17.55 milliseconds. Batched execution makes these time ratios smaller than the 6.5-fold cell-count ratio. Cell counts are not FLOPs. Prefix-specific training time was not recorded.

C.3 Two-Layer Optimization and Feasible References

Each grid uses five datasets with 160/40/200 controls. Training and validation labels share joint all-X and all-Z measurement records with 10,000 shots per group and control. Product rank is 32 with harmonic order four; Adam uses learning rate 10^{-3}, weight decay 10^{-4}, batch size 32, cap 3,000, minimum 500 epochs, and patience 100. Complete Fourier regression selects harmonic order $Q \in \{1, 2, 3, 4\}$ and ridge in $\{0, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ on measured validation MSE. Each fit evaluates 36 candidates; all ten datasets select $Q = 2$ and ridge 0.1. The models share supervision and search, while harmonic order, parameterization, and fitting regularization differ. The endpoint contrast therefore compares these fitted procedures as a whole.

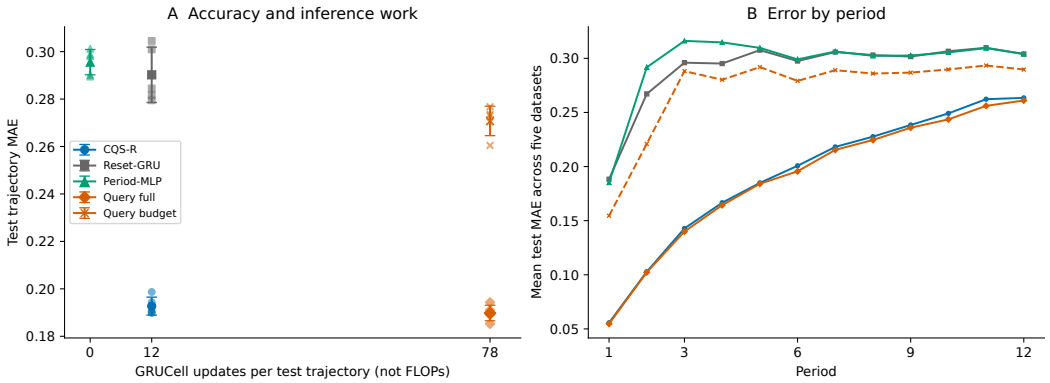


Figure 5: **Accuracy and computation reuse.** Left: dataset-level MAE points, means, and sample SD against recurrent updates per trajectory. Right: per-period MAE averaged over five datasets. Query-GRU approaches shared recurrence after full training; the cell-budget prefix remains less accurate.

Both fitted models support direct and continued optimization from the same 20 starts, with 100 Adam updates, learning rate 0.1, and clipping to $[-\pi, \pi]$. Continuation uses Eq. (11) with $\sigma_t = \max(0, 1 - t/60)$. Search uses CPU float64. Exact energy evaluation follows model selection and all surrogate searches. A second reference selects one training control by its measured energy and evaluates that candidate exactly; it is a single-candidate reference rather than a restart average.

Table 7: **VQE dataset-level contrasts and reference attainment.** Energy contrasts are mean Fourier+C minus CQS-P+C energies. Attainment fractions refer to CQS-P+C and tolerance $\epsilon = 0.001N$. All 20 starts are retained. Reported metric values: $\times 10^{-2}$; settings/counts unscaled.

	2×3, $E_{\mathrm{ref}} = -3$			3×3, $E_{\mathrm{ref}} = -4.5$		
Seed	$E_{\mathrm{F+C}} - E_{\mathrm{P+C}}$	$E \le E_{\mathrm{ref}} + \epsilon$ $\epsilon = 0.001N$	$E < E_{\mathrm{ref}} - \epsilon$ $\epsilon = 0.001N$	$E_{\mathrm{F+C}} - E_{\mathrm{P+C}}$	$E \le E_{\mathrm{ref}} + \epsilon$ $\epsilon = 0.001N$	$E < E_{\mathrm{ref}} - \epsilon$ $\epsilon = 0.001N$
1001	4.492	100.000	15.000	12.658	70.000	5.000
1002	5.240	60.000	20.000	24.090	80.000	20.000
1003	3.119	95.000	25.000	6.423	85.000	30.000
1004	1.367	100.000	40.000	2.871	95.000	30.000
1005	1.920	90.000	20.000	93.417	100.000	45.000

Additional tolerance analysis uses $0.005N$ and $0.01N$, compared with $0.001N$ in the table. At $0.005N$, product/Fourier attainment averages 99%/63% on 2×3 and 94%/43% on 3×3; at $0.01N$ it is 100%/82% and 99%/56%. Product fractions strictly below $E_{\mathrm{ref}} - 0.005N$ average 14% and 22%; none fall below $E_{\mathrm{ref}} - 0.01N$. The larger-grid mean difference is influenced by dataset 1005, where several Fourier searches end well above the reference. The full distributions retain these outcomes.

Each dataset requires 4,000,000 simulated acquisition shots, shared by both models; the ten datasets use 40,000,000 shots. The independent-mask experiment uses a further 120,000 simulated shots, and FSPT uses exact labels. The reported endpoints are exact research evaluations after surrogate search. At the same measurement allocation, validating one deployed candidate would require a further 20,000 shots.

D SEPARATING REPRESENTATION, SUPERVISION, AND CONTINUATION IN VQE

The main VQE pipeline combines a representation, a supervision objective, and a search procedure. We separate these choices in a factorial experiment with two representations (CQS-P and complete Fourier regression), two supervision objectives (scalar energy and Pauli moments), and two searches (direct optimization and Gaussian continuation). The primary comparison fixes the same harmonic cutoff and observable-specific dependency masks. A secondary comparison lets Fourier regression select its cutoff using validation data.

D.1 PAIRED EXPERIMENTAL DESIGN

We use the existing 2×3 and 3×3 spin grids with two shared-control R_X/R_{ZZ} layers, the initial state $|+\rangle^{\otimes N}$, and $H = -0.5 \sum_i X_i - 0.1 \sum_{(i,j)} Z_i Z_j$. Five new data seeds, 1201–1205, each generate 160 training, 40 validation, and 200 test controls uniformly on $[-\pi, \pi]^4$. The first shared R_X angle contributes only global phase, leaving three effective coordinates. Training and validation moments are estimated from joint all-X and all-Z measurement records, with 10,000 simulated shots per group and control. Scalar labels are linear combinations of these same measured moments. Model initialization, measurement sampling, and search starts use separate random streams with seed offsets 1,000, 2,000, and 3,000, respectively.

Both representations use $Q = 4$ and the same per-observable causal masks. CQS-P retains rank 32. Its energy-supervised ablation keeps the same output heads and initial weights as the moment-supervised model, but trains their weighted sum. These heads are latent contributions under energy supervision, rather than individually identified moments. Fourier regression contains every real Fourier mode within the same coordinate-wise cutoff and output mask. Its energy-supervised fit jointly estimates the masked output-head coefficients through their weighted energy sum; its moment-supervised fit estimates each output separately.

CQS-P uses Adam with learning rate 10^{-3}, weight decay 10^{-4}, batch size 32, a cap of 3,000 epochs, a minimum of 500 epochs, and patience 100. The moment objective averages squared errors over controls and outputs. The scalar objective is energy MSE divided by $\|c\|_2^2$, where c is the Hamiltonian coefficient vector. This normalization gives equal expected loss for zero-mean, uncorrelated component errors of equal variance. Both supervision variants select checkpoints by measured validation energy MSE. Fourier regression uses centered ridge fitting with an unpenalized constant and selects its penalty from $\{0, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ by the same validation criterion. Moment fitting regularizes each output separately; scalar fitting regularizes the joint head coefficients for the normalized energy objective. No validation refit is performed.

Each fitted model is frozen before the two searches. All eight conditions share 20 uniformly sampled initial controls per dataset, 100 Adam updates, learning rate 0.1, clipping to $[-\pi, \pi]$, and float64 search arithmetic. Direct search uses $\sigma = 0$ throughout. Continuation decreases σ linearly from 1 to 0 over the first 60 updates, followed by 40 updates on the unsmoothed objective. Exact simulation evaluates test predictions and terminal energies after fitting and search; it does not select models, starts, or endpoints. Thus the primary experiment evaluates 1,600 terminal controls without selecting a best restart by exact energy.

Aggregation and contrasts. We first average the 20 terminal energies within each dataset, then summarize the five dataset means by their mean and sample standard deviation. Difference columns use paired dataset means with shared data and starts. Define the continuation gain as $\Delta_C = E_{\text{direct}} - E_{+C}$, so positive values favor continuation. Representation contrasts are $E_{\text{Fourier}} - E_{\text{CQS-P}}$; supervision contrasts are $E_{\text{energy}} - E_{\text{moments}}$. The representation–continuation interaction is $\Delta_C(\text{CQS-P}) - \Delta_C(\text{Fourier})$, evaluated separately for each supervision objective.

D.2 CONDITIONAL CONTRIBUTIONS TO ENDPOINT QUALITY

Continuation improves both representations. Continuation lowers the mean endpoint energy in all eight within-model comparisons in Table 8. For moment-supervised CQS-P, the direct-minus-continuation gains are 0.8232 on 2×3 and 1.7525 on 3×3. Complete Fourier regression also benefits,

Table 8: **VQE factorial endpoints at matched $Q = 4$.** Exact terminal energy, averaged over all 20 starts within each dataset, then over five datasets. Direct and +C entries are mean $\pm$ sample SD. The last column is the paired continuation gain. Lower energy and positive gain are favorable. Both representations use identical per-output masks and measurement records.

Grid	Model	Labels	Direct	+C	Direct − +C
2×3	CQS-P	Energy	$-1.4341_{\pm 0.3545}$	$-2.7347_{\pm 0.2006}$	1.3006
2×3	CQS-P	Moments	$-2.1376_{\pm 0.2018}$	$-2.9608_{\pm 0.0572}$	0.8232
2×3	Fourier	Energy	$-1.0917_{\pm 0.1569}$	$-2.6943_{\pm 0.2189}$	1.6026
2×3	Fourier	Moments	$-1.1097_{\pm 0.1012}$	$-2.6916_{\pm 0.1392}$	1.5819
3×3	CQS-P	Energy	$-1.6458_{\pm 0.3251}$	$-3.8022_{\pm 0.4341}$	2.1564
3×3	CQS-P	Moments	$-2.7487_{\pm 0.4050}$	$-4.5011_{\pm 0.0073}$	1.7525
3×3	Fourier	Energy	$-1.3654_{\pm 0.3250}$	$-4.0289_{\pm 0.2377}$	2.6635
3×3	Fourier	Moments	$-1.4025_{\pm 0.2581}$	$-3.9900_{\pm 0.1925}$	2.5875

with corresponding gains of 1.5819 and 2.5875. These comparisons change the search objective while holding the fitted weights fixed, identifying continuation as a search effect shared by the fitted product and complete Fourier surrogates under the common optimization budget.

Moment supervision contributes separately. With direct optimization, replacing scalar supervision by moment supervision improves CQS-P by 0.7034 and 1.1029 on the two grids (Table 9). With continuation, the corresponding gains are 0.2260 and 0.6990. On 3×3, the mean endpoint changes from -3.8022 under scalar supervision to -4.5011 under moment supervision despite identical continuation, and exact test energy MSE falls from 1.1028 to 0.1078. Moment supervision therefore improves the fitted objective in this setting after the search procedure is fixed. For Fourier, the corresponding supervision effects remain near zero.

The representation effect depends on the search. At fixed $Q = 4$ and moment supervision, CQS-P improves on Fourier by 1.0278 and 1.3462 under direct search, but by 0.2692 and 0.5112 with continuation. The resulting interaction estimates are -0.7586 and -0.8350, meaning that smoothing reduces the between-representation gap in these runs. Under scalar supervision, both representation contrasts after continuation are small. The full configuration is therefore best interpreted as a fitted pipeline in which representation, supervision, regularization, and search interact. Endpoint quality measures that pipeline behavior; it is not used as a standalone smoothness diagnostic for the product landscape.

Table 9: **Conditional effects at matched $Q = 4$.** Paired mean differences over five datasets. Positive representation and supervision contrasts favor CQS-P and moment supervision, respectively. A positive interaction means that CQS-P benefits more from continuation.

Paired contrast	2×3: mean	3×3: mean
Fourier − CQS-P, energy, direct	0.3424	0.2804
Fourier − CQS-P, energy, +C	0.0404	−0.2267
Fourier − CQS-P, moments, direct	1.0278	1.3462
Fourier − CQS-P, moments, +C	0.2692	0.5112
Energy − moments, CQS-P, direct	0.7034	1.1029
Energy − moments, CQS-P, +C	0.2260	0.6990
Energy − moments, Fourier, direct	0.0180	0.0371
Energy − moments, Fourier, +C	−0.0027	−0.0389
Δ_C(CQS-P) − Δ_C(Fourier), energy	−0.3020	−0.5071
Δ_C(CQS-P) − Δ_C(Fourier), moments	−0.7586	−0.8350

D.3 Validation-selected Fourier reference and interpretation

Fixing $Q = 4$ aligns frequency coverage but can over-parameterize complete Fourier regression for the 160-control training sets. We therefore add a secondary reference that selects $Q \in \{1, 2, 3, 4\}$ and

the same ridge penalties by measured validation energy MSE, while retaining the output masks, data, starts, and searches. This analysis was added after the first two factorial datasets were completed and is reported separately from the fixed-Q design. All 20 fits (two grids, five datasets, two supervision objectives) select $Q = 2$. Table 10 reports the resulting endpoints and paired contrasts against CQS-P with the same supervision and continuation.

For moment supervision, the CQS-P+C advantage decreases to 0.0549 on 2×3 and to 0.2256 on 3×3. The selected Fourier reference therefore narrows the representation gap relative to the fixed-Q comparison, especially on the smaller grid. Both scalar-supervised comparisons after continuation remain near zero. Frequency selection materially changes the representation comparison, so the fixed-Q result and the validation-selected reference together define the supported endpoint claim. These runs also use new datasets, matched output masks, and a common validation-energy criterion, distinct from the complete-Fourier control in Appendix L.6.

Table 10: **Secondary Fourier reference with validation-selected cutoff.** All fits select $Q = 2$; CQS-P remains at $Q = 4$. Endpoint entries are mean $\pm$ sample SD over five dataset means. The final column compares the two representations under identical supervision and continuation; positive differences favor CQS-P.

Grid	Labels	Fourier direct	Fourier+C	Fourier+C − CQS-P+C
2×3	Energy	$-2.1391_{\pm 0.3010}$	$-2.9084_{\pm 0.0831}$	-0.1736
2×3	Moments	$-2.1130_{\pm 0.3005}$	$-2.9059_{\pm 0.1098}$	0.0549
3×3	Energy	$-2.9352_{\pm 0.4542}$	$-4.2734_{\pm 0.1561}$	-0.4713
3×3	Moments	$-2.9393_{\pm 0.4054}$	$-4.2755_{\pm 0.1562}$	0.2256

Attainment and model size. For moment-supervised CQS-P, continuation increases the fraction of starts reaching $E \leq -N/2 + 0.01N$ from 56% to 96% on 2×3 and from 49% to 99% on 3×3. This threshold is relative to a feasible circuit reference, not an exact ground-state energy. The result concerns reliable attainment near that reference. Parameter compression does not explain the improvement in these three-coordinate problems: CQS-P allocates 15,405 and 24,885 trainable parameters, with 9,645 and 15,381 connected through active masks, whereas the complete masked $Q = 4$ Fourier heads contain 4,941 and 7,533 coefficients. For a fixed Hamiltonian, Fourier readouts can additionally collapse to one 729-coefficient energy expansion.

Continuation improves search in this paired design, while moment supervision adds a benefit for CQS-P on the larger grid. Representation differences also change with supervision and frequency selection. The nonzero interactions show that endpoint gains arise from the fitted representation and search procedure together, rather than from independent, additive effects.

E Extended Task Results

This appendix reports the complete FSPT matrix, finite-shot evaluations, hardware measurements, and component interventions underlying the main-text results.

Protocol index. Appendix C gives the five-dataset controls that separate dependency masks, per-output depth, and VQE endpoint stability (Table 1). The broader task comparisons use shared-$T = 32$ TFIM prediction (Table 2; Appendix M.2), the $k/24$ FSPT matrix (Table 11), and independent VQE datasets (Table 3; Appendix L). The exact-label TFIM and FSPT studies are auxiliary fitting profiles reported in Appendix M, separate from the independent-data controls. The hardware study uses stored measured prefixes and a distinct input/validation protocol (Appendix B.6).

E.1 Trajectory Prediction and Floquet Response

Appendix B.4 defines the fully supervised trajectory task.

Matched prediction across sizes and horizons. We vary $N \in \{4, 6, 8, 10\}$ and $K \in \{6, 12, 24\}$. For each N, dataset seed 42 generates 2,800 trajectories of length 24; shorter horizons use their

Table 11: **FSPT prediction across qubit counts and supervised horizons.** Exact-test MAE, mean $\pm$ sample standard deviation over three paired partition/initialization runs. All methods share targets and trajectory splits, and all K periods are supervised. Neural models use a 300-epoch cap. NC uses the public sampled-feature h_{qs} fit. Bold marks the lowest row mean. Reset-GRU and MLP diagnostics appear in the analysis section. Reported metric values: $\times 10^{-1}$, with settings/counts unscaled.

Setting	CQS-R	CQS-P	NC h_{qs}
$N=4, K=6$	$\mathbf{0.830}_{\pm 0.024}$	$2.815_{\pm 0.029}$	$1.543_{\pm 0.030}$
$N=4, K=12$	$\mathbf{1.350}_{\pm 0.045}$	$3.562_{\pm 0.041}$	$2.551_{\pm 0.016}$
$N=4, K=24$	$\mathbf{1.779}_{\pm 0.067}$	$3.861_{\pm 0.034}$	$3.237_{\pm 0.022}$
$N=6, K=6$	$\mathbf{1.136}_{\pm 0.004}$	$2.716_{\pm 0.043}$	$1.907_{\pm 0.029}$
$N=6, K=12$	$\mathbf{1.788}_{\pm 0.008}$	$3.127_{\pm 0.042}$	$2.593_{\pm 0.053}$
$N=6, K=24$	$\mathbf{2.625}_{\pm 0.085}$	$3.368_{\pm 0.027}$	$3.041_{\pm 0.036}$
$N=8, K=6$	$\mathbf{1.240}_{\pm 0.016}$	$2.721_{\pm 0.040}$	$2.167_{\pm 0.341}$
$N=8, K=12$	$\mathbf{1.861}_{\pm 0.033}$	$2.951_{\pm 0.046}$	$2.614_{\pm 0.183}$
$N=8, K=24$	$\mathbf{2.761}_{\pm 0.066}$	$3.063_{\pm 0.026}$	$2.886_{\pm 0.097}$
$N=10, K=6$	$\mathbf{1.305}_{\pm 0.015}$	$2.703_{\pm 0.067}$	$2.157_{\pm 0.066}$
$N=10, K=12$	$\mathbf{2.147}_{\pm 0.058}$	$2.844_{\pm 0.040}$	$2.550_{\pm 0.051}$
$N=10, K=24$	$\mathbf{2.728}_{\pm 0.044}$	$2.905_{\pm 0.035}$	$2.812_{\pm 0.083}$

prefixes. Seeds 42, 43, and 44 define paired 2,000/400/400 splits and network initializations, giving 36 setting–seed combinations per method. Horizons share controls and partitions within each N and seed. GRU and MLP inputs $[\mathbf{x}, k/24]$ keep period scaling fixed; multiplicative and NC output heads identify each period. All K periods are supervised on training controls and evaluated on held-out controls, without observation inputs.

The full and reset GRUs use hidden width 128; the MLP has three width-256 SiLU layers. The GRU and MLP have 134,536 and 136,456 parameters at $N = 8$. Neural models use Adam, batches of 32, a 300-epoch cap, and patience 30. Validation MSE selects checkpoints and the ridge regularizer. The budget is matched in trajectories, not scalar labels: each setting supplies $2{,}000KN$ training targets.

Among the three matrix predictors, CQS-R has the lowest mean MAE throughout the sweep, with a narrower NC gap at longer horizons (Table 11; Figure 2). The multiplicative adapter remains less accurate than both comparators. Its first-period harmonic restriction excludes terms in the exact drive response, providing one concrete mismatch between the representation and the task (Appendix M). The ridge, MLP, and reset-GRU controls in Section F distinguish this frequency issue from the role of accumulated computation.

E.2 Learning from Shared Finite-Shot Labels

Appendix B.5 defines the shot sweep, allocation rule, and actual train-plus-validation measurement cost.

We next hold the input distribution fixed and vary label precision to test how the representations use limited measurements. At $N = 8, L = 2$, both neural models and both NC configurations receive identical scalar training and validation targets on the same partitions. Targets are either exact energies or Pauli-shadow estimates generated from exact termwise marginals with $T \in \{8, 16, 32, 64, 128, 256, 512\}$; all test energies are exact. Neural training has an 80-epoch cap. The measurement randomness and training cap differ from Table 2, so this is a separate paired comparison. Sampling termwise marginals models label noise without preserving the full joint distribution of physical multi-qubit shots.

We also set train-plus-validation budget caps of 16k, 32k, 64k, and 128k shots and vary training controls $M \in \{125, 250, 500, 1000, 2000\}$. Validation uses $\max(50, \min(400, \lceil M/5 \rceil))$ controls. Integer per-input shots, capped at 512, can leave part of a requested budget unused. Appendix M reports every allocation rather than a test-selected allocation policy.

The product model is more accurate with exact labels and at most finite-shot levels, although the ordering between the neural models changes across measurement allocations (Figure 6; Appendix M).

Both outperform the specified NC fitting profiles throughout the shot sweep. Predictive accuracy therefore depends on both representation and label allocation under this noise model; the sweep does not establish a hardware shot-saving factor.

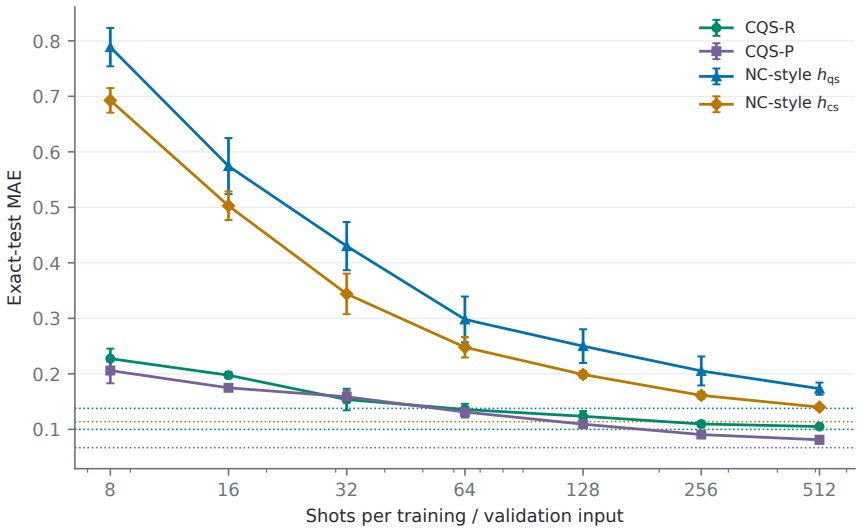


Figure 6: **Finite-shot prediction under matched labels.** All four predictors share each scalar-label realization at $N = 8, L = 2$. Error bars summarize three paired seeds; dotted lines show exact-label performance under the same training cap. The multiplicative version improves over the recurrent version at six of seven shot levels. The complete budget-allocation comparison appears in Appendix M.

E.3 CQS-R ON IBM TFIM HARDWARE

The measured-prefix prediction task and its evaluation horizon are defined in Appendix B.6.

Hardware measurements test prediction with noisy expectation labels. Using Qiskit and EstimatorV2 (Javadi-Abhari et al., 2024), we collect local $\langle X_i \rangle$ and $\langle Z_i Z_{i+1} \rangle$ estimates for TFIM prefixes $k = 1, \ldots, 10$ at target precision 0.08. Each N has 24 training, 8 validation, and 16 test controls. Training and validation use $k = 1, \ldots, 4$; evaluation uses $k = 5, \ldots, 10$ on test controls.

The hardware comparison is a measured-prefix extrapolation diagnostic (Figure 7). CQS-R is compared with fixed static ridge controls that either receive the current layer alone or the padded prefix history. The current-layer ridge performs worse, consistent with the value of preceding controls. Appendix B.6 gives the complete comparisons and describes the depth-dependent feature shift in the full-prefix representation.

F ANALYSIS OF COMPOSITIONAL REPRESENTATIONS

Appendix B.7 specifies the interventions and evaluation targets for each diagnostic.

The representation comparison motivates two different interventions. Within CQS-R, resetting the state removes accumulated computation while preserving the input and update architecture. Within CQS-P, changing direct optimization to continuation preserves the fitted representation and changes the objective used for search. A separate harmonic calculation examines functions excluded by the product drive adapter. These diagnostics test specific components; a comparison between the complete models also changes capacity, supervision in VQE, and training behavior.

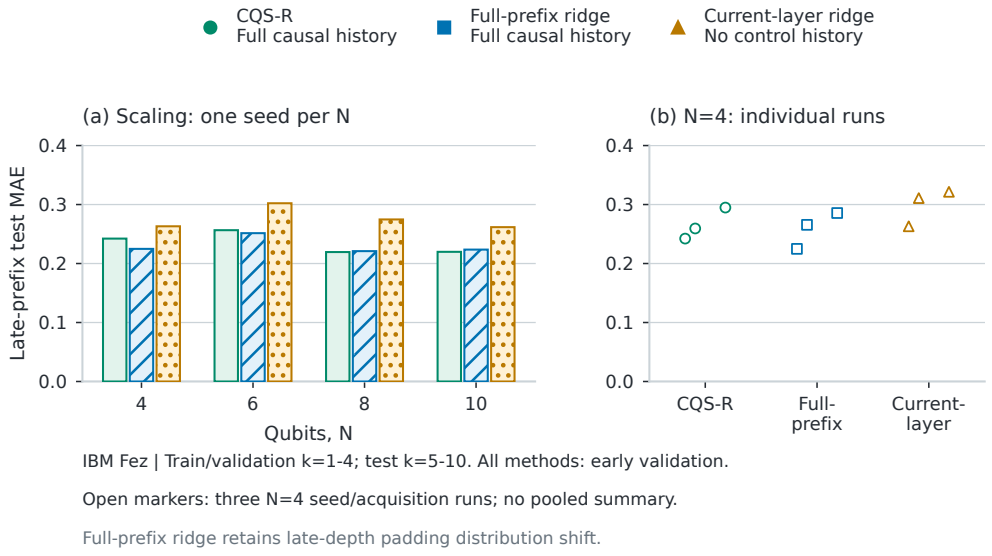


Figure 7: **IBM TFIM measured-prefix prediction.** Left: late-prefix MAE across qubit counts, with one stored dataset per size. Right: individual results for three $N = 4$ acquisition runs. Static ridge controls use fixed regularization; full-prefix ridge receives the preceding controls, while current-layer ridge receives only the active layer. Fits reuse stored measurements, and recurrent predictions are unchanged.

F.1 Diagnostics for Sequential Composition

Does carrying history improve prediction? Removing carried state retains the encoder, GRU parameters, readout, and period information, while reducing each output to one update. Let $G_{\omega_{\text{rec}}}$ denote the GRU update and $d_{\omega_{\text{out}}}$ the readout. The full model and its reset counterpart compute

$$\hat{\mathbf{y}}_k = d_{\omega_{\text{out}}}(G_{\omega_{\text{rec}}}(e_k, h_{k-1})), \qquad \hat{\mathbf{y}}_k^{\text{reset}} = d_{\omega_{\text{out}}}(G_{\omega_{\text{rec}}}(e_k, 0)). \tag{18}$$

An inference-only intervention retains the fitted weights and resets history at evaluation. A retrained reset model uses the second operator throughout training and validation under the same optimization schedule. These tests distinguish a fitted solution's dependence on history from the accuracy of a retrained, single-update predictor. Both retain the period index; neither matches the original model's per-output depth.

Table 12: **History ablation at $N = 8, K = 12$.** All periods are supervised. Test reset changes only the inference operator of the trained CQS-R model; the reset GRU is trained independently. Values summarize three paired runs. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

Method	MAE	MSE	R^2
CQS-R	$1.861_{\pm 0.033}$	$0.690_{\pm 0.028}$	$6.345_{\pm 0.091}$
Reset-GRU (trained)	$2.891_{\pm 0.055}$	$1.501_{\pm 0.042}$	$2.050_{\pm 0.137}$
Period MLP	$2.956_{\pm 0.037}$	$1.554_{\pm 0.033}$	$1.771_{\pm 0.047}$
Trig. ridge	$2.811_{\pm 0.042}$	$1.400_{\pm 0.036}$	$2.586_{\pm 0.072}$
CQS-R (test reset)	$4.477_{\pm 0.073}$	$3.652_{\pm 0.129}$	$-9.337_{\pm 0.433}$

Inference-only resetting sharply degrades prediction. Retraining recovers part of that loss but remains worse than the full recurrence across the matrix (Table 12). The effect therefore extends beyond disrupting a model that was trained to use history, although this comparison still couples state sharing with iterative depth.

What the ablation identifies. The reset GRU retains its nonlinear transformation, while the slightly larger period MLP supplies a second static comparator. Both have higher $N = 8$ errors than the

recurrent model, though resetting also removes iterative depth. Query-GRU restores per-output depth without carrying state between periods and reaches similar observed MAE (Appendix C). Floquet controls and the period index are available to every model, so recurrence changes their processing rather than adding information. Cross-period computation reuse is the recurrent model's supported advantage in this setting.

Error across supervised periods. Prediction error grows over the supervised trajectory, while the recurrent advantage over MLP and ridge narrows (Figure 8). Common-prefix diagnostics also reveal a fitting penalty on training controls when the supervised horizon is extended (Appendix J). This implicates learning across the horizon rather than merely adding more difficult test targets. Because predicted observables are not fed back, the pattern is not evidence of autoregressive observation-error accumulation.

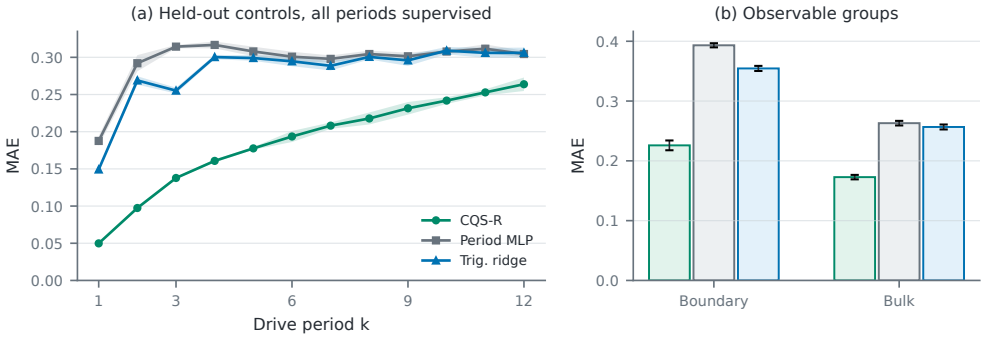


Figure 8: FSPT generalization to held-out controls at $N = 8$. Period-wise MAE, and separate boundary ($j = 0, 7$) and bulk ($j = 1, \ldots, 6$) errors. Bands and error bars show sample standard deviation across the three paired partition/initialization runs. All 12 periods are supervised.

Boundary and bulk observables. We also separate the boundary sites $B = \{0, 7\}$ from the six bulk sites $C = \{1, \ldots, 6\}$. For test controls $\mathcal{S}$ and supervised periods $\mathcal{K}$, a group-specific error is

$$\mathrm{MAE}_A = \frac{\sum_{\mathbf{x}\in\mathcal{S}} \sum_{k\in\mathcal{K}} \sum_{j\in A} |\hat{y}_{k,j}(\mathbf{x}) - y_{k,j}(\mathbf{x})|}{|\mathcal{S}|\,|\mathcal{K}|\,|A|}, \qquad A \in \{B, C\}. \tag{19}$$

The overall metric weights the groups by their sizes: $\mathrm{MAE} = \frac{1}{4}\mathrm{MAE}_B + \frac{3}{4}\mathrm{MAE}_C$. The recurrent error reduction spans both boundary and bulk observables (Figure 8), so the overall advantage is not confined to one spatial group. The larger bulk group contributes more to the aggregate metric, making the separate boundary evaluation important for interpreting local performance.

Boundary observables have larger absolute errors but greater explained variation under CQS-R. Their higher MAE therefore does not by itself imply poorer relative reconstruction: error magnitude and target variation describe different aspects of prediction quality.

Variability and paired comparisons. Partitions are disjoint within runs, but test subsets can overlap across runs from the same pool. Sample standard deviations use denominator $3 - 1$. For baseline b and run s, the relative gain is

$$g_s(b) = 100\left(1 - \frac{\mathrm{MAE}_{s,\mathrm{CQS\text{-}R}}}{\mathrm{MAE}_{s,b}}\right). \tag{20}$$

We summarize paired gains rather than taking the ratio of aggregate means. Three coupled partition/initialization runs neither separate variance sources nor provide population-level significance tests; the paired runs are used as diagnostics of the tested mechanisms.

F.2 Product Expressivity and Optimization

The FSPT adapter tests a boundary of the product parameterization. Its first-period head has one first-harmonic factor per independent control, whereas the exact response contains higher harmonics

of the repeated global control. Appendix M derives this mismatch and the nonzero error of the best first-harmonic projection. Increasing rank alone cannot add the absent frequencies. The restriction belongs to this shared-control factorization, not to Fourier products in general, and helps explain why CQS-P is less accurate throughout the supervised FSPT matrix.

Continuation improves the grid endpoints when applied to fixed product checkpoints, but does not improve every task: direct search remains competitive on the two-layer chain (Table 24). Since the fitted representation is unchanged, this contrast locates the gain in how the surrogate objective is searched. NC+C supplies the corresponding intervention for the reference representation, distinguishing a useful search procedure from an architecture-specific advantage.

The paired final-layer task uses shared scalar labels and a matched neural training schedule; its mixed rankings show no uniform predictive advantage for the product model. VQE combines moment supervision, architecture-specific fitting, and optional continuation, so its endpoints reflect the fitted pipeline. Component comparisons separate direct product optimization from continuation, while five-dataset controls compare complete Fourier models and reference attainment (Appendix C). In FSPT, reset and Query-GRU controls distinguish iterative depth from cross-period reuse. These results favor task-dependent representations evaluated on their downstream metrics.

G Method and Implementation Details for Circuit-Guided Representations

CQS (Compositional Quantum Surrogates) learns expectation-value representations guided by circuit structure. Its two architectures use different composition rules: CQS-R composes learned state updates along ordered circuit layers or drive periods, whereas CQS-P forms products of learned Fourier factors over circuit-dependent controls. Both are evaluated on final-layer TFIM prediction, VQE pre-training, FSPT trajectories, and measurement-budget studies. Hardware and recurrent-state diagnostics focus on CQS-R. Appendix M compares the two architectures with task-specific NC-style regressors. Appendix L specifies the VQE and scalar-label fitting protocols.

Learning settings and generalization. Neural-network wave functions and neural quantum-state tomography learn representations of individual states (Carleo & Troyer, 2017; Torlai et al., 2018). Here, supervision spans circuit controls and the outputs are selected expectations. Data-dependent comparisons of classical and quantum learning (Huang et al., 2021), few-training-data generalization bounds for quantum models (Caro et al., 2022), and memorization-based qualifications of model-class generalization arguments (Gil-Fuster et al., 2024) motivate keeping the data distribution and learning setting explicit. The evidence in this paper is empirical generalization on held-out control splits for the specified circuit families and label regimes.

Relation to low-rank tensor regression. Let $\psi_d(\theta_d)$ collect the constant, sine, and cosine basis functions and write $g_{o,d,r} = \mathbf{a}_{o,d,r}^\top \psi_d$. Expanding Eq. (9) gives the Fourier coefficient tensor

$$A_o = \sum_{r=1}^{R} w_{o,r} \bigotimes_{d \in C_o} \mathbf{a}_{o,d,r} + \beta_o \bigotimes_{d \in C_o} e_0,$$

Here $\mathbf{a}_{o,d,r}$ stacks the constant, cosine, and sine coefficients from Eq. (8), and e_0 selects the constant. This is a CP parameterization with rank at most $R+1$, including the bias. The circuit determines C_o; training learns the factor vectors and readout. Tensor trains provide a different factorization of high-order coefficient tensors (Oseledets, 2011), and tensor-network classifiers apply such factorizations to local feature maps (Stoudenmire & Schwab, 2016). CQS-P adapts the established low-rank function family (Saiapin & Batselier, 2025) through circuit-dependent coordinate restrictions.

Structural information and independent fits. For CQS-P, observable-specific masks are derived from the actual circuit and conservatively retain controls that may affect an output. Excluded coordinates cannot affect that output under the mask construction, while retained coordinates form an overapproximation that can include inactive influences. For CQS-R, sequential composition supplies an ordering and parameter-sharing inductive bias for expectation prediction. The two architectures are fitted independently for each reported task and configuration. Cases that reuse fitted moment predictors across Hamiltonian coefficients or probe frozen checkpoints are stated explicitly below.

Static baselines. Simulation trigonometric features have support size $\|\mathbf{w}\|_0 \leq 2$, where $\mathbf{w}$ is the Fourier frequency vector and ridge regularization. FSPT fits one output coefficient vector per observable and period with shared features. On hardware, static predictors receive a flattened prefix padded to the full horizon plus k/K. The trigonometric map contains constant, sine, cosine, and selected pair terms (at most 600 pairs), with ridge coefficient 1.0. The static MLP uses widths $(96, 64)$, ReLU, and standardization. It enables scikit-learn's internal early stopping on training rows, whereas the GRU uses a separate held-out control split. Thus outer test controls are common but model-selection rules differ. Late prefixes also activate coordinates held at zero during training.

Sequential composition in CQS-R. The deterministic implementation encodes each input step, initializes $h_0 = 0$, updates a GRU cell, and reads directly from its hidden state. Simulation configurations use hidden width 128, Adam with learning rate 10^{-3} and weight decay 10^{-4}, batch size 32, and validation MSE. Hardware GRU models use width 64, full-batch AdamW with learning rate 2×10^{-3} and weight decay 10^{-4}, at most 350 epochs, and patience 60. Hardware validation is restricted to the first four periods. No additional per-observable target normalization or observable embedding appears in these paths.

Input convention. The FSPT matrix networks broadcast controls across periods and append $k/24$ for every (N, K).

Hardware evaluation units. Hardware error entries are flattened over held-out controls, later periods, and observable coordinates. We report each acquisition separately. Data sampling and model initialization use the run seed, so their contributions to variation are not separated.

H HARDWARE METADATA

Relation to noise mitigation. Noise can suppress trainable signal in variational circuits (Wang et al., 2021). Error-mitigation methods instead modify expectation estimation, for example through controlled noise variation (Temme et al., 2017; Endo et al., 2021). These are complementary to fitting a predictor from acquired labels. The hardware comparisons below evaluate learned prediction under the recorded acquisition protocol; they do not measure an additional error-mitigation intervention.

The following table reports the stored hardware runs and fixed static baselines. The full-prefix and current-layer ridge controls are reported in Appendix B.6 and Table 4.

Table 13: IBM TFIM scaling runs. Backend was IBM Fez. Jobs used EstimatorV2 with target precision 0.08, prefix circuits, train periods $k = 1, \ldots, 4$, and test periods $k = 5, \ldots, 10$. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

N	CQS-R MAE	Full-prefix ridge $\alpha = 1$ MAE	Static MLP MAE
4	2.423	3.217	3.400
6	2.565	3.069	4.134
8	2.194	2.924	4.034
10	2.198	3.069	4.173

I FSPT QUBIT-COUNT AND SUPERVISED-HORIZON MATRIX

Matched controls and prefixes. The matrix contains all twelve combinations of $N \in \{4, 6, 8, 10\}$ and $K \in \{6, 12, 24\}$, each evaluated with seeds 42, 43, and 44. A fresh exact-statevector pool is generated for each N using dataset seed 42 and the drive in Equation 12. Each pool contains 2,800 control vectors and 24 periods of local-Z targets. The shorter-K datasets take prefixes of this pool, preserving both control values and target entries. The same split seed therefore identifies the same train/validation/test controls across all three horizons at a fixed N. Each split has 2,000/400/400 trajectories. Networks are trained separately for every setting and seed; no checkpoint is transferred between horizons or sizes.

Information and optimization. All predictors receive controls and the requested output period, either as $k/24$ in the neural inputs or as the period-specific ridge output coordinate. Targets are exact expectations; there are no observation features or observation-assisted denoising. Neural architecture widths and optimizer settings match the main text, including learning rate 10^{-3} and weight decay 10^{-4}. For $N+1$ control coordinates the ridge uses $1+2(N+1)^2$ features and KN outputs, selecting its regularizer from $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 100\}$. Both model selection and training use all selected periods on their respective controls. Equal numbers of trajectories and optimization epochs do not imply equal scalar-label budgets, parameter counts across N, or training FLOPs across K.

Aggregate and individual results. Tables 11 and 14 report mean and sample standard deviation across the three paired runs. Table 15 retains each seed, including the inference-only reset intervention. The artifact also contains R^2, boundary/bulk errors, per-period curves, checkpoints, and validation histories for every setting. The fixed per-N pools are reused across seeds, so these statistics summarize the paired runs rather than independent physical datasets.

Table 14: **FSPT matrix test MSE, $\mathcal{R}(f)$.** Same controls, splits, and uncertainty convention as Table 11. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

N	K	CQS-R	Reset-GRU	Period MLP	Trig. ridge
4	6	$\mathbf{0.149}_{\pm 0.008}$	$0.298_{\pm 0.049}$	$0.516_{\pm 0.033}$	$1.502_{\pm 0.004}$
4	12	$\mathbf{0.363}_{\pm 0.027}$	$1.512_{\pm 0.030}$	$2.013_{\pm 0.077}$	$1.886_{\pm 0.003}$
4	24	$\mathbf{0.583}_{\pm 0.049}$	$2.299_{\pm 0.039}$	$2.305_{\pm 0.044}$	$2.112_{\pm 0.044}$
6	6	$\mathbf{0.261}_{\pm 0.005}$	$0.410_{\pm 0.017}$	$0.744_{\pm 0.033}$	$1.260_{\pm 0.041}$
6	12	$\mathbf{0.597}_{\pm 0.002}$	$1.289_{\pm 0.075}$	$1.748_{\pm 0.043}$	$1.525_{\pm 0.044}$
6	24	$\mathbf{1.229}_{\pm 0.078}$	$1.924_{\pm 0.029}$	$1.901_{\pm 0.029}$	$1.698_{\pm 0.034}$
8	6	$\mathbf{0.316}_{\pm 0.009}$	$0.512_{\pm 0.031}$	$0.846_{\pm 0.025}$	$1.204_{\pm 0.025}$
8	12	$\mathbf{0.690}_{\pm 0.028}$	$1.501_{\pm 0.042}$	$1.554_{\pm 0.033}$	$1.400_{\pm 0.036}$
8	24	$\mathbf{1.382}_{\pm 0.057}$	$1.744_{\pm 0.017}$	$1.710_{\pm 0.013}$	$1.501_{\pm 0.021}$
10	6	$\mathbf{0.352}_{\pm 0.009}$	$0.659_{\pm 0.060}$	$0.894_{\pm 0.020}$	$1.175_{\pm 0.034}$
10	12	$\mathbf{0.931}_{\pm 0.046}$	$1.629_{\pm 0.033}$	$1.510_{\pm 0.035}$	$1.340_{\pm 0.027}$
10	24	$\mathbf{1.393}_{\pm 0.033}$	$1.666_{\pm 0.034}$	$1.637_{\pm 0.031}$	$1.426_{\pm 0.024}$



Figure 9: Detailed comparisons: (a) $N = 8, K = 12$ matrix predictions with $k/24$ inputs; (b) trained and inference-only reset controls for the same runs. Error bars are sample standard deviations across three paired runs.

Table 15: Individual FSPT matrix runs: test MAE, all periods supervised, $k/24$ inputs. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

N	K	Seed	CQS-R	Reset-GRU	Period MLP	Ridge	Test reset
4	6	42	0.843	1.142	1.688	3.032	4.525
4	6	43	0.802	1.172	1.615	3.028	4.583
4	6	44	0.844	1.325	1.647	3.036	4.583

N	K	Seed	CQS-R	Reset-GRU	Period MLP	Ridge	Test reset
4	12	42	1.340	3.020	3.505	3.464	4.914
4	12	43	1.310	2.981	3.611	3.460	5.000
4	12	44	1.399	2.922	3.675	3.465	5.008
4	24	42	1.702	3.895	3.891	3.674	5.063
4	24	43	1.814	3.937	3.942	3.725	5.262
4	24	44	1.822	3.953	3.963	3.746	5.178
6	6	42	1.136	1.462	1.973	2.743	4.226
6	6	43	1.131	1.421	2.000	2.743	4.386
6	6	44	1.140	1.415	1.929	2.660	4.262
6	12	42	1.779	2.626	3.252	3.050	4.636
6	12	43	1.792	2.804	3.291	3.041	4.654
6	12	44	1.793	2.646	3.204	2.957	4.522
6	24	42	2.706	3.469	3.432	3.220	4.671
6	24	43	2.633	3.483	3.476	3.242	4.769
6	24	44	2.537	3.422	3.405	3.174	4.629
8	6	42	1.244	1.602	2.111	2.641	4.186
8	6	43	1.223	1.614	2.125	2.624	4.241
8	6	44	1.254	1.525	2.055	2.571	4.162
8	12	42	1.896	2.952	2.991	2.847	4.463
8	12	43	1.853	2.878	2.960	2.821	4.557
8	12	44	1.833	2.844	2.917	2.766	4.412
8	24	42	2.806	3.183	3.151	2.946	4.552
8	24	43	2.792	3.182	3.148	2.926	4.372
8	24	44	2.685	3.153	3.127	2.902	4.498
10	6	42	1.303	1.890	2.160	2.579	4.160
10	6	43	1.321	1.756	2.140	2.591	4.194
10	6	44	1.291	1.718	2.122	2.512	4.066
10	12	42	2.091	3.019	2.888	2.727	4.423
10	12	43	2.206	3.012	2.871	2.729	4.280
10	12	44	2.146	2.956	2.812	2.670	4.186
10	24	42	2.769	3.051	3.013	2.814	4.088
10	24	43	2.732	3.041	3.016	2.803	4.115
10	24	44	2.682	2.977	2.954	2.760	4.139

J Horizon Effects in Sequential Composition

We investigate the horizon-dependent gap using common-target comparisons, a network-width and training-budget intervention, and frozen-state readout probes. These are post-hoc diagnostics on the matrix's existing control pools. Each comparison uses seeds 42, 43, and 44 with the original paired splits. Checkpoints and regularizers are selected by validation MSE. No test measurement is supplied as an input. The diagnostics concern generalization across controls with all 24 periods supervised.

Common targets reveal a fitting penalty. The full-horizon average mixes different targets as K grows. To hold evaluation targets fixed, we evaluate the separately trained $K = 6, 12, 24$ models on exactly the same first-six periods and test controls. All models use $k/24$, so these prefixes have identical input coordinates. At $N = 8$, first-six MAE rises from 1.240×10^{-1} for training horizon 6 to 2.443×10^{-1} for horizon 24; at $N = 10$ it rises from 1.305×10^{-1} to 2.672×10^{-1} (Table 16). This common-target comparison identifies a fitting penalty from longer supervised horizons, rather than an artifact of averaging additional late-period errors. The forward recurrence leaves an already fitted model's first-six outputs unchanged when future input steps are appended; the observed difference arises from fitting and selecting parameters over the longer supervised horizon.

The same deterioration is present on training controls (Table 17). First-six training MAE rises from 1.025×10^{-1} to 2.392×10^{-1} at $N = 8$, and from 1.070×10^{-1} to 2.634×10^{-1} at $N = 10$. Test errors follow the same direction. This is evidence of a fitting deficit under joint horizon supervision, rather than solely an enlarged generalization gap. It does not distinguish loss weighting, optimization difficulty, or representational limitations by itself.

At $N = 10, K = 24$, CQS-R's first-six test MAE of 2.672×10^{-1} is also slightly higher than ridge's 2.611×10^{-1}. Thus the small full-horizon advantage includes a loss of the early-period advantage, not just dilution by additional late targets.

Table 16: Identical first-six test targets evaluated with models trained to different horizons. Every setting uses $k/24$ inputs and the same split for each seed. Values are mean $\pm$ sample standard deviation. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

N	CQS-R $K = 6$	CQS-R $K = 12$	CQS-R $K = 24$	Ridge $K = 24$
4	$0.830_{\pm 0.024}$	$0.918_{\pm 0.026}$	$1.133_{\pm 0.029}$	$3.063_{\pm 0.010}$
6	$1.136_{\pm 0.004}$	$1.306_{\pm 0.007}$	$1.732_{\pm 0.063}$	$2.743_{\pm 0.019}$
8	$1.240_{\pm 0.016}$	$1.361_{\pm 0.006}$	$2.443_{\pm 0.170}$	$2.653_{\pm 0.031}$
10	$1.305_{\pm 0.015}$	$1.858_{\pm 0.079}$	$2.672_{\pm 0.072}$	$2.611_{\pm 0.074}$

Table 17: First-six CQS-R MAE on training, validation, and test controls. The fitting penalty of longer supervised horizons is already present on the training set. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

N	Training K	Train MAE	Validation MAE	Test MAE
8	6	$1.025_{\pm 0.024}$	$1.220_{\pm 0.020}$	$1.240_{\pm 0.016}$
8	12	$1.214_{\pm 0.006}$	$1.340_{\pm 0.005}$	$1.361_{\pm 0.006}$
8	24	$2.392_{\pm 0.148}$	$2.390_{\pm 0.142}$	$2.443_{\pm 0.170}$
10	6	$1.070_{\pm 0.011}$	$1.290_{\pm 0.021}$	$1.305_{\pm 0.015}$
10	12	$1.740_{\pm 0.087}$	$1.818_{\pm 0.087}$	$1.858_{\pm 0.079}$
10	24	$2.634_{\pm 0.027}$	$2.602_{\pm 0.049}$	$2.672_{\pm 0.072}$

Target scale and block composition. We partition a 24-period trajectory into blocks $B_1 = \{1, \ldots, 6\}$, $B_2 = \{7, \ldots, 12\}$, and $B_3 = \{13, \ldots, 24\}$. For either model, the aggregate MAE decomposes exactly as

$$\mathrm{MAE}_{1:24} = \tfrac{1}{4}\mathrm{MAE}_{B_1} + \tfrac{1}{4}\mathrm{MAE}_{B_2} + \tfrac{1}{2}\mathrm{MAE}_{B_3}. \tag{21}$$

At $N = 10$, target mean absolute magnitude decreases from 3.418×10^{-1} in B_1 to 2.948×10^{-1} in B_3, and variance decreases from 1.985×10^{-1} to 1.601×10^{-1}. These statistics pool test controls, periods within the block, and observables. The late targets therefore have a smaller amplitude scale. A predictor using only the training median for each period and observable reaches late-block MAE 2.928×10^{-1}, compared with 2.869×10^{-1} for ridge and 2.758×10^{-1} for the original CQS-R. At $N = 8$, the corresponding values are 3.122×10^{-1}, 3.029×10^{-1}, and 2.887×10^{-1}. The small late-block gaps coexist with weak recovery of target variation: late-block R^2 is 1.121×10^{-1} for CQS-R and 5.990×10^{-2} for ridge at $N = 10$. These statistics characterize the labels and fitted models; reduced target variance alone does not identify the physical source of prediction difficulty.

Table 18: Target magnitude and block-wise errors for the original $K = 24$ models. The median predictor is fitted on training controls separately for each period and observable. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

N	Periods	Mean $\lvert y \rvert$	CQS-R MAE	Ridge MAE	Median MAE
8	1–6	$3.540_{\pm 0.054}$	$2.443_{\pm 0.170}$	$2.653_{\pm 0.031}$	$3.483_{\pm 0.050}$
8	7–12	$3.128_{\pm 0.034}$	$2.827_{\pm 0.063}$	$2.988_{\pm 0.047}$	$3.106_{\pm 0.033}$
8	13–24	$3.144_{\pm 0.006}$	$2.887_{\pm 0.017}$	$3.029_{\pm 0.006}$	$3.122_{\pm 0.005}$
10	1–6	$3.418_{\pm 0.060}$	$2.672_{\pm 0.072}$	$2.611_{\pm 0.074}$	$3.366_{\pm 0.058}$
10	7–12	$2.912_{\pm 0.047}$	$2.722_{\pm 0.037}$	$2.822_{\pm 0.019}$	$2.893_{\pm 0.042}$
10	13–24	$2.948_{\pm 0.025}$	$2.758_{\pm 0.034}$	$2.869_{\pm 0.028}$	$2.928_{\pm 0.023}$

Network width and nested training budgets. For $N = 8, 10$ at $K = 24$, we train networks with widths 128 and 256 using the original Adam settings, batch size 32, and patience 30. Width changes the encoder, GRU, and readout together, so this tests total network capacity rather than isolated

memory size. Each of the twelve fits runs for at most 600 epochs and saves the validation-best checkpoint within both 300 and 600 epochs. The two budgets therefore share initialization, batch order, and optimizer trajectory. Early stopping is retained; a run stopping before 600 contributes its stopped best checkpoint. Width-128 predictions at the 300-epoch budget reproduce the original matrix predictions. Table 19 reports every configuration, and Table 21 gives actual and selected epochs.

Table 19: Network capacity and nested training budgets at $K = 24$. Width applies to encoder, GRU, and readout; cap is the maximum number of epochs with patience 30. Entries summarize three paired runs. The comparison changes training cost and is a sensitivity analysis. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

N	Width	Cap	All MAE	All MSE	1–6 MAE	13–24 MAE
8	128	300	$2.761_{\pm 0.066}$	$1.382_{\pm 0.057}$	$2.443_{\pm 0.170}$	$2.887_{\pm 0.017}$
8	128	600	$2.602_{\pm 0.034}$	$1.257_{\pm 0.024}$	$2.035_{\pm 0.078}$	$2.844_{\pm 0.014}$
8	256	300	$2.768_{\pm 0.041}$	$1.389_{\pm 0.034}$	$2.464_{\pm 0.069}$	$2.891_{\pm 0.022}$
8	256	600	$2.610_{\pm 0.070}$	$1.260_{\pm 0.047}$	$2.078_{\pm 0.123}$	$2.838_{\pm 0.046}$
10	128	300	$2.728_{\pm 0.044}$	$1.393_{\pm 0.033}$	$2.672_{\pm 0.072}$	$2.758_{\pm 0.034}$
10	128	600	$2.693_{\pm 0.056}$	$1.368_{\pm 0.045}$	$2.566_{\pm 0.126}$	$2.753_{\pm 0.033}$
10	256	300	$2.734_{\pm 0.049}$	$1.400_{\pm 0.041}$	$2.699_{\pm 0.088}$	$2.758_{\pm 0.036}$
10	256	600	$2.708_{\pm 0.060}$	$1.379_{\pm 0.044}$	$2.627_{\pm 0.120}$	$2.753_{\pm 0.040}$

At width 128, extending the cap lowers full-horizon MAE from $2.761 \times 10^{-1}_{\pm 6.600 \times 10^{-3}}$ to $2.602 \times 10^{-1}_{\pm 3.400 \times 10^{-3}}$ at $N = 8$, a paired reduction of $5.75\%_{\pm 1.81\%}$. The gain over unchanged ridge increases from $5.60\%_{\pm 1.63\%}$ to $11.05\%_{\pm 0.67\%}$. At $N = 10$, MAE changes from $2.728 \times 10^{-1}_{\pm 4.400 \times 10^{-3}}$ to $2.693 \times 10^{-1}_{\pm 5.600 \times 10^{-3}}$, a smaller reduction of $1.25\%_{\pm 0.85\%}$; the ridge gain increases from $2.33\%_{\pm 0.65\%}$ to $3.55\%_{\pm 1.07\%}$. First-six errors also improve, from 2.443×10^{-1} to 2.035×10^{-1} and from 2.672×10^{-1} to 2.566×10^{-1}, respectively, but remain far above those of the separately trained $K = 6$ models. All three width-128 $N = 10$ runs stop before epoch 600, after 384–554 epochs. These outcomes show a contribution from the original training budget, with incomplete recovery under the extended schedule.

Increasing width approximately quadruples the parameter count: from 134,536 to 531,208 at $N = 8$ and from 135,050 to 532,234 at $N = 10$. Nevertheless, its paired MAE reductions are $-0.29\%_{\pm 1.35\%}$ and $-0.22\%_{\pm 0.31\%}$ at cap 300, and $-0.33\%_{\pm 1.42\%}$ and $-0.54\%_{\pm 1.63\%}$ at cap 600; negative reductions denote deterioration. Thus simply enlarging this architecture with the unchanged optimizer does not improve mean error. This result does not rule out other capacity changes or width-dependent optimizer tuning. Neither intervention is an equal-compute comparison.

Period-specific decoding of frozen states. We freeze the original width-128 encoder and GRU, extract h_k, and fit linear readouts with an intercept. A shared readout predicts $[1, h_k]W$; a period-specific readout predicts $[1, h_k]W_k$. Both use the same states, controls, targets, and validation trajectories. Ridge selection minimizes validation MSE over the original six-value regularizer grid, using one regularizer for all periods. The objective is row-mean squared loss plus $\lambda \|W\|_F^2$; intercepts are regularized. The shared readout has $129N$ coefficients and the period-specific readout has $24 \cdot 129N$. These probes test linear decodability of a frozen representation and do not retrain the recurrent dynamics.

Period-specific readouts lower MAE relative to shared linear readouts, from 2.980×10^{-1} to 2.842×10^{-1} at $N = 8$ and from 2.857×10^{-1} to 2.764×10^{-1} at $N = 10$. Both remain worse than the original nonlinear readout, whose errors are 2.761×10^{-1} and 2.728×10^{-1} (Table 20). Period identity helps a restricted linear decoder, but this intervention does not show that replacing the shared nonlinear readout with independent heads resolves the gap.

Interpretation and limits of attribution. The evidence supports two complementary observations. Joint long-horizon fitting degrades even identical early targets on training controls, and additional optimization recovers part of that loss. Meanwhile, late targets have smaller magnitude and both learned predictors recover little of their variation beyond a period-wise constant reference, leaving a

Table 20: Readout probes on frozen original $K = 24$ hidden states. Linear heads are fitted with validation-selected ridge regularization. Original denotes the jointly trained nonlinear CQS-R readout. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

N	Readout	MAE	MSE	R^2
8	Original nonlinear	$2.761_{\pm 0.066}$	$1.382_{\pm 0.057}$	$2.298_{\pm 0.235}$
8	Shared linear	$2.980_{\pm 0.015}$	$1.544_{\pm 0.015}$	$1.392_{\pm 0.020}$
8	Per-period linear	$2.842_{\pm 0.051}$	$1.442_{\pm 0.049}$	$1.962_{\pm 0.189}$
10	Original nonlinear	$2.728_{\pm 0.044}$	$1.393_{\pm 0.033}$	$1.779_{\pm 0.111}$
10	Shared linear	$2.857_{\pm 0.028}$	$1.488_{\pm 0.025}$	$1.219_{\pm 0.092}$
10	Per-period linear	$2.764_{\pm 0.034}$	$1.415_{\pm 0.030}$	$1.648_{\pm 0.073}$

narrow absolute error gap. A larger network does not recover the missing advantage under the tested schedule, and independent linear readouts do not outperform the original nonlinear head. These diagnostics support a scoped conclusion: joint long-horizon fitting and weak late-target variation both contribute to the narrowed absolute-error gap. Distinguishing gradient interference, recurrent-state compression, and intrinsic response complexity would require targeted loss-weighting or gradient interventions, richer jointly trained decoders, and independent control pools. All uncertainty here is sample standard deviation over three coupled split/initialization runs on reused pools.

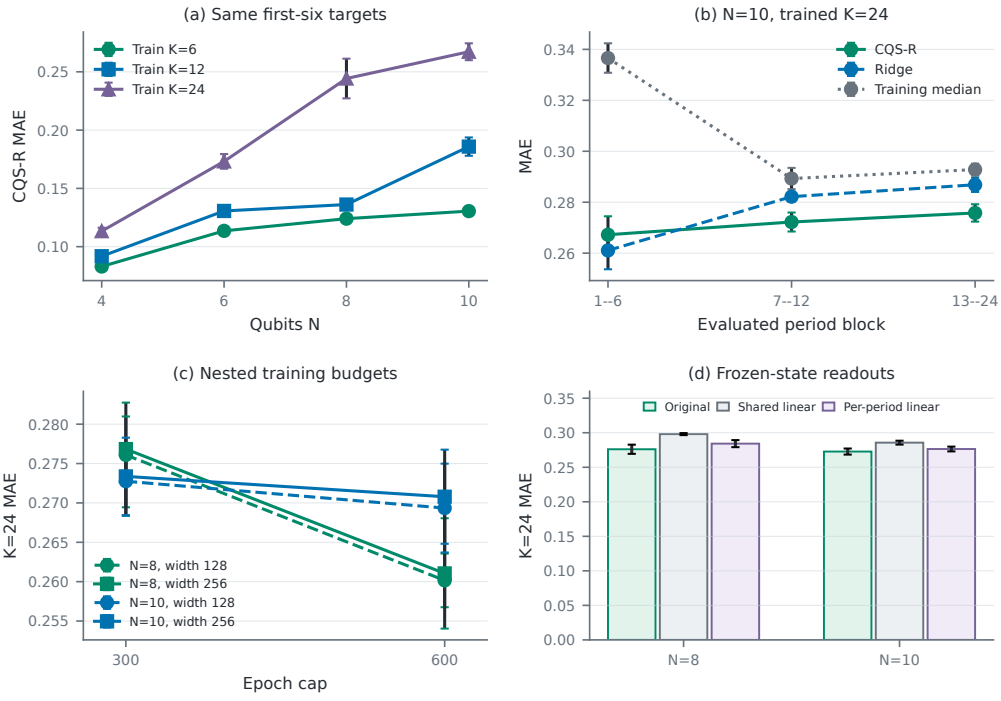


Figure 10: Diagnostics of the narrowing FSPT gap. (a) Fixed first-six targets remove changes in evaluation composition. (b) Block-wise comparisons include a training-only median. (c) Width and budget interventions use nested validation selection. (d) Linear readout probes retain frozen recurrent states. Error bars show sample standard deviation across three paired runs.

K TFIM PAIRED THREE-SEED COMPARISONS

Circuit and scalar target. Each qubit count $N \in \{8, 10, 12, 14\}$ uses its original cached pool of 2,800 independently sampled controls, generated with dataset seed 42. Controls have shape $2 \times (2N - 1)$ with entries uniform on $[-1, 1]$. Starting from $|0\rangle^{\otimes N}$, each layer applies Hadamards to all qubits, local R_X rotations, and nearest-neighbor R_{ZZ} rotations. In particular, the implemented

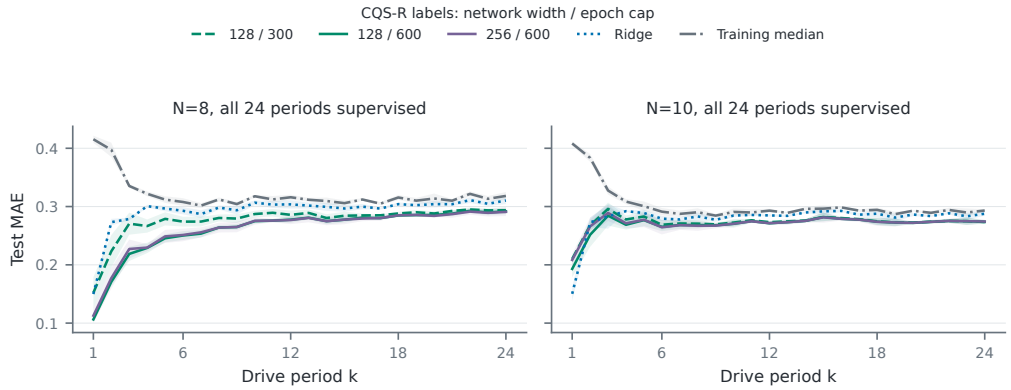


Figure 11: Period-wise test MAE for the diagnostic models. Labels specify network width and epoch cap. All periods are supervised on training controls; shaded bands show sample standard deviation across three paired runs.

Table 21: Every diagnostic fit and nested-budget checkpoint. Ran is the actual number of optimization epochs available within the cap, and Best is the validation-selected epoch. A stopped run may have fewer epochs than the nominal cap. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

N	Width	Seed	Cap	Ran	Best	MAE	MSE
8	128	42	300	300	298	2.806	1.415
8	128	42	600	600	595	2.641	1.283
8	128	43	300	300	300	2.792	1.415
8	128	43	600	600	600	2.583	1.247
8	128	44	300	300	297	2.685	1.317
8	128	44	600	498	468	2.581	1.240
8	256	42	300	300	297	2.812	1.419
8	256	42	600	600	598	2.690	1.314
8	256	43	300	300	300	2.764	1.395
8	256	43	600	600	591	2.559	1.226
8	256	44	300	300	300	2.730	1.352
8	256	44	600	600	593	2.582	1.240
10	128	42	300	300	300	2.769	1.419
10	128	42	600	554	524	2.729	1.392
10	128	43	300	300	300	2.732	1.404
10	128	43	600	384	354	2.723	1.396
10	128	44	300	300	298	2.682	1.355
10	128	44	600	488	458	2.628	1.316
10	256	42	300	300	293	2.775	1.430
10	256	42	600	371	341	2.775	1.427
10	256	43	300	300	282	2.747	1.417
10	256	43	600	600	579	2.687	1.370
10	256	44	300	300	293	2.679	1.353
10	256	44	600	414	384	2.661	1.340

circuit repeats the Hadamards at the start of the second layer; they are not solely an initial-state preparation. The exact scalar label is the unnormalized final energy

$$y = \left\langle -J \sum_{i=0}^{N-2} Z_i Z_{i+1} - h \sum_{i=0}^{N-1} X_i \right\rangle, \qquad J = -0.1, \quad h = -0.5. \tag{22}$$

The recurrent predictor receives the two control blocks in order and is supervised only at its final output. Ridge receives the flattened controls. Neither receives test observations.

Paired repeats and model selection. Seeds 42, 43, and 44 specify the partition, neural initialization, and minibatch order. Seed 42 retains the original contiguous training/validation/test indices $[0, 2000)$,

$[2000, 2400)$, and $[2400, 2800)$; the other seeds permute the same pool before applying these sizes. Each method uses the same partition within a run. These repeats reuse a physical dataset per N and do not separate partition and initialization variability. All twelve neural fits are newly seeded.

The deterministic encoder, GRU, and nonlinear readout have width 128, with dropout and stochastic perturbations disabled. Training retains the original scaling schedule: Adam, learning rate 10^{-3}, weight decay 10^{-4}, batch size 32, cap 250 epochs, and patience 30. The checkpoint minimizes validation MSE. The trigonometric features retain support at most two. For $d = 2(2N - 1)$ flattened controls, it uses $1 + 2d^2$ features.

Finite-shot Table 2 protocol. The main-table prediction rows use shared scalar h_{cs}-style labels rather than exact training labels. For every train and validation input, exact local marginals define a scalar Pauli-shadow energy estimator; a fixed prefix with $T = 32$ shots per input gives the label supplied to both CQS-R and the trigonometric regressor. Held-out test inputs are scored against exact final-layer energies. The trigonometric fit follows the public-code rule audited for the NC-style baseline: non-squared L2 regularization, Adam optimization, $\alpha = 10$, and the same support-two feature basis. Table 22 reports all individual runs.

Table 22: Every finite-shot TFIM run used in Table 2. Both methods train on identical h_{cs}-style scalar labels with $T = 32$ and are evaluated against exact held-out energies. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

N	Seed	CQS-R MAE	h_{cs} MAE	CQS-R $\mathcal{R}(f)$	h_{cs} $\mathcal{R}(f)$	CQS-R R^2	h_{cs} R^2
8	42	1.807	3.642	0.530	2.146	4.300	-13.090
8	43	1.530	3.388	0.383	1.776	6.560	-5.930
8	44	1.538	3.403	0.367	1.759	5.840	-9.940
10	42	1.942	5.488	0.625	4.575	4.770	-28.260
10	43	2.182	4.806	0.749	3.723	3.200	-23.830
10	44	1.972	4.781	0.654	3.490	3.890	-22.630
12	42	2.362	6.574	0.870	7.164	4.110	-38.480
12	43	2.353	6.699	0.904	7.186	3.670	-40.360
12	44	2.462	5.894	0.912	5.543	3.710	-28.240
14	42	2.741	7.403	1.151	8.845	1.890	-52.340
14	43	3.277	7.424	1.677	8.765	-0.470	-44.740
14	44	3.191	8.066	1.581	10.776	-0.470	-61.330

Exact-label auxiliary comparison. We also report an exact-label three-seed comparison as an auxiliary diagnostic. In that protocol, both methods train on exact final-layer energies; the trigonometric ridge coefficient is 1.0 in $\|\Phi W - Y\|_F^2 + \|W\|_F^2$, including intercept regularization. Figure 12 and Table 23 summarize this exact-label diagnostic. Its values are not the source of Table 2. Error bars in Figure 12 are sample standard deviations over the paired runs. The scalar energy is not divided by N, so the raw error scale should be interpreted together with qubit count and the matched baseline.

L Learned Fourier products and paired VQE comparison

Circuit families and energy conventions. VQE circuits prepare $|+\rangle^{\otimes N}$ once, then apply local R_X rotations followed by R_{ZZ} rotations on open-boundary lattice edges in each of $L = 1, 2$ layers. Chains use $N = 6$ and independent gate angles; grids use 2×3 or 3×3 sites with one shared R_X angle and one shared R_{ZZ} angle per layer. Controls are uniform on $[-\pi, \pi]$. We retain the reference implementation's energy conventions: $E_{1\mathrm{D}} = \langle -J \sum ZZ - h \sum X \rangle$ and $E_{2\mathrm{D}} = \langle J \sum ZZ + h \sum X \rangle$, with $h = -0.5$. The main comparison uses $J = -0.1$; the two-layer chain additionally uses $J \in \{-0.5, -0.3, 0.1, 0.3, 0.5\}$. These circuits differ from the final-layer scaling circuits in Appendix K, which apply Hadamards in both layers and use angles in $[-1, 1]$.

Measurements and training data. For each VQE seed, chains use 2,000/400/400 training/validation/test controls and ten independent random-Pauli shots per training or validation control. The stored bitstrings and basis recipes supply X_i and $Z_i Z_j$ shadow estimates. Grids use 160/40/200 controls, with 10,000 shots in each of two commuting measurement groups, all-X and all-Z, giving

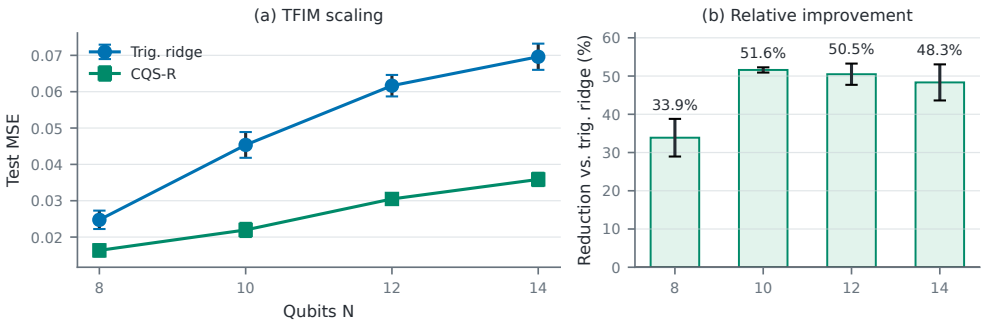


Figure 12: Auxiliary exact-label TFIM three-seed scaling. (a) Mean test MSE for exact final-layer energy. (b) Paired MSE reduction against the fixed-coefficient ridge. Error bars show sample standard deviations across three paired runs.

Table 23: Every auxiliary exact-label TFIM repeat. Ran is the number of optimization epochs; Best is selected by validation MSE. Both methods share exact labels and trajectory splits within each seed. Reported metric values: $\times 10^{-2}$; settings/counts unscaled.

N	Seed	Ran	Best	CQS-R MAE	Ridge MAE	CQS-R MSE	Ridge MSE
8	42	85	55	9.925	12.859	1.717	2.745
8	43	132	102	10.160	12.275	1.748	2.437
8	44	192	162	9.293	11.605	1.437	2.243
10	42	135	105	12.005	18.112	2.393	4.866
10	43	67	37	11.644	16.976	2.193	4.579
10	44	91	61	11.251	16.032	2.003	4.161
12	42	133	103	13.844	19.266	2.978	5.865
12	43	107	77	13.348	20.104	2.991	6.456
12	44	85	55	13.710	19.281	3.177	6.176
14	42	95	65	15.328	21.526	3.685	7.175
14	43	110	80	15.531	20.763	3.700	6.545
14	44	79	49	14.780	20.591	3.375	7.164

20,000 shots per measured control. All models use the same stored estimates within a run. Exact test labels and endpoint energies are used only for evaluation. No additional local labels, gradient labels, or online candidate measurements enter these experiments.

Reference regression. The NC-style fit uses the public experimental rule of Liao et al. (2026): support-two gate-coordinate trigonometric monomials, including repeated shared-control factors, and the objective $\|\Phi w - y\|_2^2 + \alpha\|w\|_2$. The penalty is the unsquared norm. We use $\alpha = 10$ for chains and $\alpha = 0.1$ for grids, float64 Adam with learning rate 0.01, cap 5,000, and the reference exponential-moving-average stopping threshold 10^{-4} with decay 0.9. Initialization follows JAX key 1234. Scalar-energy predictors are fitted separately for each Hamiltonian in both NC and CQS-R. CQS-P instead fits the recorded moment vector once and reuses its readout for all six chain couplings. This changes the supervision objective but not the acquired measurement records.

Architecture and fitting. The CQS-R VQE predictor uses width 128 for its two-layer encoder, GRU, and nonlinear scalar readout, with no dropout or stochastic perturbation. It fits each Hamiltonian's measured scalar energy separately using Adam, learning rate 10^{-3}, weight decay 10^{-4}, batch size 32, a 300-epoch cap, and patience 30. Measured validation MSE selects its checkpoint. Its reported endpoints use direct, unsmoothed Adam on the scalar prediction.

Every observable has 32 product channels. Factors use the features $[1, \sqrt{2}\cos(q\theta), \sqrt{2}\sin(q\theta)]$, with the normalization absorbed into the coefficients of Eq. (8). Independent angles use $q = 1$; shared angles use $q = 1, \ldots, 4$. Both the factor coefficients and final linear readout are learned. The representation is the direct product $z_{o,r}(\boldsymbol{\theta}) = \prod_{d \in C_o} g_{o,d,r}(\theta_d)$ in Eq. (9), where d indexes control coordinates, not recurrent states. Its factors commute; circuit structure enters through their coordinate

assignments and the observable-specific set C_o. The conservative dependency mask propagates possible Pauli support backward through the actual gates, rather than computing expectation values. Factors at excluded coordinates are replaced by one; retaining a coordinate does not establish nonzero physical dependence. The linear moment readout is unbounded.

Training minimizes measured moment MSE using Adam, learning rate 10^{-3}, weight decay 10^{-4}, and batch size 32. Chains use a 300-epoch cap and patience 30. Grids use a 3,000-epoch cap, a minimum of 500 epochs, and patience 100. Checkpoints minimize measured validation MSE. The minimum training period addresses an observed premature-stop failure: with patience 30 and no minimum, the $2 \times 3, L = 2$ seed-44 run stopped at epoch 71 and produced continuation endpoint -1.196. Applying the longer schedule uniformly to all grid tasks and seeds gave -3.001 in that run. Training budgets are thus architecture-specific, not matched wall-clock budgets.

Continuation for both predictors. Each seed uses the same 20 random initializations for all methods. We retain every restart and apply 100 float64 Adam updates with learning rate 0.1, clipping coordinates to $[-\pi, \pi]$. The scale $\sigma_t = \max(0, 1-t/60)$ is identical for CQS-P and NC+C. Gaussian convolution of a trigonometric monomial must account for shared controls. In particular,

$$\mathbb{E}[\cos^2(\theta+\epsilon)] = \tfrac{1}{2}[1 + e^{-2\sigma^2}\cos(2\theta)], \tag{23}$$

$$\mathbb{E}[\sin^2(\theta+\epsilon)] = \tfrac{1}{2}[1 - e^{-2\sigma^2}\cos(2\theta)], \tag{24}$$

$$\mathbb{E}[\sin(\theta+\epsilon)\cos(\theta+\epsilon)] = \tfrac{1}{2}e^{-2\sigma^2}\sin(2\theta). \tag{25}$$

Here $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Independent-coordinate factors multiply. These identities evaluate NC+C without extra training or measurement. Original NC uses its unsmoothed objective throughout. Exact energies never determine an update, checkpoint, restart, or reported subset.

Separating representation, supervision, and optimization. Table 24 compares scalar-supervised CQS-R endpoints with direct Adam and continuation on the same fitted multiplicative model. This within-product comparison isolates the continuation objective. A separate moment-supervised GRU (M-GRU) fits the same measured moments and uses Hamiltonian recombination. Its independent runs use width 128 and a 300-epoch cap. Saved histories show 30 epochs between the selected checkpoint and every early stop; the runner uses patience 30 and no minimum-epoch warmup. The grid product fits instead use cap 3,000, patience 100, and a 500-epoch minimum. Thus the M-GRU/product comparison changes both representation and training budget. The $3 \times 3, L = 2$ exact-energy Adam endpoint -3.016 is a local-optimizer diagnostic, not an ansatz optimum. Gaussian continuation changes the attraction basins while retaining the fitted representation.

Table 24: VQE representation and optimizer comparison. True endpoint energy, mean $\pm$ sample standard deviation of three seed means; each mean retains all 20 restarts. The two multiplicative columns use identical fitted weights. Reported metric values: $\times 10^0$; settings/counts unscaled.

Setting	GRU direct	CQS-P direct	CQS-P+C
1D $N = 6, L = 1$	$-2.117_{\pm 0.488}$	$-2.166_{\pm 0.321}$	$-2.333_{\pm 0.311}$
1D $N = 6, L = 2$	$1.877_{\pm 0.570}$	$-1.461_{\pm 0.231}$	$-1.434_{\pm 0.133}$
2D $2 \times 3, L = 1$	$-2.000_{\pm 0.265}$	$-2.000_{\pm 0.265}$	$-2.633_{\pm 0.153}$
2D $2 \times 3, L = 2$	$-2.263_{\pm 1.014}$	$-1.985_{\pm 0.139}$	$-2.967_{\pm 0.052}$
2D $3 \times 3, L = 1$	$-2.500_{\pm 0.529}$	$-2.500_{\pm 0.529}$	$-4.500_{\pm 0.000}$
2D $3 \times 3, L = 2$	$-4.094_{\pm 0.353}$	$-2.387_{\pm 0.194}$	$-4.503_{\pm 0.013}$

L.1 Independent VQE Component Evaluation

The main VQE table reports independent seeds 45/46/47. These runs use the same circuit families, measurement rules, optimizer settings, and metrics. CQS-P+C improves on NC+C in 16 of 18 paired runs across the six primary settings and 30 of 33 pairs when the chain couplings are included. Couplings reuse measurements and a fitted moment model, so these counts describe paired outcomes rather than independent-dataset significance. The component comparison in Table 26 uses the independent seeds.

Table 25: Hamiltonian reuse in the two-layer six-qubit chain. One fitted CQS-P moment model serves all six couplings; NC fits a scalar-energy regressor for each. CQS-R also fits scalar energies separately for each coupling. All use the same stored measurements. Energies average all 20 initializations within each of three seeds. Reported metric values: $\times 10^0$; settings/counts unscaled.

J	CQS-R	CQS-P+C	NC	NC+C
-0.5	$1.798_{\pm 0.316}$	$-2.174_{\pm 0.280}$	$-0.321_{\pm 0.503}$	$-0.407_{\pm 0.426}$
-0.3	$2.136_{\pm 0.465}$	$-1.773_{\pm 0.346}$	$-0.529_{\pm 0.296}$	$-0.493_{\pm 0.320}$
-0.1	$1.877_{\pm 0.570}$	$-1.434_{\pm 0.133}$	$-0.795_{\pm 0.092}$	$-0.737_{\pm 0.045}$
$+0.1$	$1.918_{\pm 0.774}$	$-1.492_{\pm 0.373}$	$-0.799_{\pm 0.067}$	$-0.838_{\pm 0.044}$
$+0.3$	$2.090_{\pm 0.250}$	$-1.704_{\pm 0.112}$	$-0.781_{\pm 0.064}$	$-0.869_{\pm 0.129}$
$+0.5$	$2.030_{\pm 0.464}$	$-2.186_{\pm 0.135}$	$-0.571_{\pm 0.124}$	$-0.604_{\pm 0.059}$

Table 26: Independent VQE component ablation at $J = -0.1$. M-GRU uses the recurrent architecture with moment supervision and Hamiltonian recombination. CQS-P uses the product representation with direct Adam; CQS-P+C adds continuation on the same fitted product weights. Lower exact endpoint energy is better. Reported metric values: $\times 10^0$; settings/counts unscaled.

Direct optimization: representation and supervision

Setting	CQS-R	M-GRU	CQS-P direct
1D $N = 6, L = 1$	$-1.118_{\pm 0.735}$	$-0.299_{\pm 0.468}$	$-2.499_{\pm 0.100}$
1D $N = 6, L = 2$	$1.671_{\pm 1.348}$	$1.333_{\pm 0.592}$	$-1.159_{\pm 0.148}$
2D $2 \times 3, L = 1$	$-2.000_{\pm 0.100}$	$-2.000_{\pm 0.100}$	$-2.033_{\pm 0.153}$
2D $2 \times 3, L = 2$	$-2.661_{\pm 0.587}$	$-2.737_{\pm 0.229}$	$-2.189_{\pm 0.314}$
2D $3 \times 3, L = 1$	$-2.566_{\pm 0.306}$	$-2.566_{\pm 0.306}$	$-2.567_{\pm 0.306}$
2D $3 \times 3, L = 2$	$-3.713_{\pm 1.004}$	$-4.049_{\pm 0.084}$	$-2.797_{\pm 0.559}$

Continuation and reference fits

Setting	CQS-P+C	NC direct	NC+C
1D $N = 6, L = 1$	$-2.631_{\pm 0.114}$	$-2.547_{\pm 0.045}$	$-2.581_{\pm 0.012}$
1D $N = 6, L = 2$	$-1.286_{\pm 0.431}$	$-0.718_{\pm 0.091}$	$-0.739_{\pm 0.076}$
2D $2 \times 3, L = 1$	$-2.667_{\pm 0.153}$	$-2.199_{\pm 0.100}$	$-2.633_{\pm 0.115}$
2D $2 \times 3, L = 2$	$-3.009_{\pm 0.003}$	$-2.684_{\pm 0.291}$	$-2.710_{\pm 0.262}$
2D $3 \times 3, L = 1$	$-4.500_{\pm 0.000}$	$-3.031_{\pm 0.233}$	$-4.497_{\pm 0.003}$
2D $3 \times 3, L = 2$	$-4.506_{\pm 0.014}$	$-4.272_{\pm 0.242}$	$-4.281_{\pm 0.212}$

Table 27: Development VQE condition–seed means over 20 restarts. Reported metric values: $\times 10^0$; settings/counts unscaled.

Setting	J	Seed	CQS-P+C	NC	NC+C
1D $N = 6, L = 1$	-0.1	42	-2.399	-2.294	-2.384
1D $N = 6, L = 1$	-0.1	43	-1.994	-1.782	-1.783
1D $N = 6, L = 1$	-0.1	44	-2.606	-2.587	-2.685
1D $N = 6, L = 2$	-0.5	42	-1.920	0.259	0.085
1D $N = 6, L = 2$	-0.3	42	-1.419	-0.191	-0.141
1D $N = 6, L = 2$	-0.1	42	-1.281	-0.895	-0.789
1D $N = 6, L = 2$	$+0.1$	42	-1.147	-0.728	-0.811
1D $N = 6, L = 2$	$+0.3$	42	-1.580	-0.854	-0.960
1D $N = 6, L = 2$	$+0.5$	42	-2.042	-0.591	-0.560
1D $N = 6, L = 2$	-0.5	43	-2.475	-0.579	-0.657
1D $N = 6, L = 2$	-0.3	43	-2.110	-0.740	-0.765
1D $N = 6, L = 2$	-0.1	43	-1.517	-0.773	-0.713
1D $N = 6, L = 2$	$+0.1$	43	-1.887	-0.810	-0.888
1D $N = 6, L = 2$	$+0.3$	43	-1.799	-0.749	-0.722
1D $N = 6, L = 2$	$+0.5$	43	-2.308	-0.683	-0.580
1D $N = 6, L = 2$	-0.5	44	-2.128	-0.642	-0.648
1D $N = 6, L = 2$	-0.3	44	-1.790	-0.655	-0.572
1D $N = 6, L = 2$	-0.1	44	-1.504	-0.716	-0.710

Setting	J	Seed	CQS-P+C	NC	NC+C
1D $N = 6, L = 2$	$+0.1$	44	-1.442	-0.860	-0.814
1D $N = 6, L = 2$	$+0.3$	44	-1.733	-0.739	-0.925
1D $N = 6, L = 2$	$+0.5$	44	-2.208	-0.438	-0.671
2D $2 \times 3, L = 1$	-0.1	42	-2.500	-1.900	-2.499
2D $2 \times 3, L = 1$	-0.1	43	-2.600	-1.800	-2.400
2D $2 \times 3, L = 1$	-0.1	44	-2.800	-2.400	-2.700
2D $2 \times 3, L = 2$	-0.1	42	-2.993	-2.708	-2.934
2D $2 \times 3, L = 2$	-0.1	43	-2.906	-2.866	-2.890
2D $2 \times 3, L = 2$	-0.1	44	-3.001	-2.783	-2.779
2D $3 \times 3, L = 1$	-0.1	42	-4.500	-2.498	-4.296
2D $3 \times 3, L = 1$	-0.1	43	-4.500	-2.099	-4.498
2D $3 \times 3, L = 1$	-0.1	44	-4.500	-3.499	-4.298
2D $3 \times 3, L = 2$	-0.1	42	-4.511	-4.382	-4.385
2D $3 \times 3, L = 2$	-0.1	43	-4.509	-4.295	-4.296
2D $3 \times 3, L = 2$	-0.1	44	-4.487	-4.367	-4.310

L.2 Paired compositional architectures under scalar labels

To compare sequential composition with learned Fourier products under common supervision, we fit CQS-R and CQS-P independently using every cached $T = 32$ scalar label and the exact same 2,000/400/400 split at $N = 8, 10, 12, 14$. Both neural models use Adam, learning rate 10^{-3}, weight decay 10^{-4}, batch size 32, a 250-epoch cap, and patience 30. The product model uses rank 32 and one harmonic. Its masks follow the actual repeated-Hadamard circuit: conjugation by each Hadamard exchanges Pauli X and Z support. The sole training target is the scalar weighted energy sum, so no moment labels or physical-moment identification are added. Direct prediction uses $\sigma = 0$, without a VQE search or smoothing. Table 28 reports all paired runs; checkpoint replay verifies the saved test predictions.

Table 28: Paired final-layer prediction with identical scalar $T = 32$ labels. All entries are exact-test MAE. CQS-R and CQS-P are fitted independently under a common epoch cap and checkpoint-selection rule. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

N	Seed	CQS-R (GRU)	CQS-P	NC
8	42	1.807	1.598	3.642
8	43	1.530	1.546	3.388
8	44	1.538	1.541	3.403
10	42	1.942	2.147	5.488
10	43	2.182	2.231	4.806
10	44	1.972	2.275	4.781
12	42	2.362	2.522	6.574
12	43	2.353	2.541	6.699
12	44	2.462	2.491	5.894
14	42	2.741	2.649	7.403
14	43	3.277	2.904	7.424
14	44	3.191	2.517	8.066

L.3 Circuit-mask intervention

We test observable dependencies on the two-layer six-qubit chain with independent angles, using the measured-moment datasets of seeds 45/46/47. Each run supplies 2,000 training, 400 measured-validation, and 400 exact-test controls in $[-\pi, \pi]^{2\times 11}$. All arms remove the first layer's six global-phase R_X controls. Circuit masks retain 67 observable–coordinate entries; a full mask retains all 176 remaining entries across eleven observables. Random masks retain 67 entries and match each observable's counts separately within every layer and gate family (R_X or R_{ZZ}), while changing their positions. Randomization uses seed $100000 + s$ and rejects unchanged rows.

All arms use the same initialized learnable tensors, rank 32, $Q = 1$, minibatch order, and Adam settings: learning rate 10^{-3}, weight decay 10^{-4}, batch size 32, cap 300, patience 30, and no minimum

epoch. Training minimizes measured moment MSE and validation selects the checkpoint. Exact held-out moments and their energy combination at $J = -0.1, h = -0.5$ are scored after fitting. There is no continuation, VQE search, or additional measurement in this intervention. These are additional controls on the existing independent-seed datasets, not a new independent confirmation set.

Table 29: Mask-only intervention on the two-layer $N = 6$ moment task. Mean $\pm$ sample standard deviation over three paired seeds; lower error is better. Random and circuit masks have identical counts within every observable/layer/gate-family block. Reported metric values: $\times 10^{-2}$; settings/counts unscaled.

Mask	Active entries	Moment MSE	Energy MAE
Circuit	67	$7.540_{\pm 0.660}$	$30.870_{\pm 0.880}$
Full	176	$18.660_{\pm 0.250}$	$45.510_{\pm 1.420}$
Random	67	$18.930_{\pm 0.290}$	$45.910_{\pm 1.730}$

Circuit masks improve over both controls on every seed (Table 29). Paired energy-MAE differences are $1.464 \times 10^{-1} \pm 1.640 \times 10^{-2}$ for full minus circuit and $1.503 \times 10^{-1} \pm 1.880 \times 10^{-2}$ for random minus circuit. The random control matches sparsity, so the result supports dependency locations rather than parameter restriction alone in this task. All arms allocate 23,595 parameters and retain dense evaluation. Their active factor/readout slots are 6,795 for circuit/random and 17,259 for full masks; these counts are not identifiable statistical degrees of freedom or sparse-execution speedups. The nine local fits take 79.4 s of training in total.

L.4 Matching the moment-GRU stopping schedule

The original independent M-GRU grid runs use a 300-epoch cap and patience 30, whereas product models use cap 3,000, minimum 500, and patience 100. We rerun both two-layer grids for seeds 45/46/47 with the latter stopping schedule, retaining width 128, measured moment labels, optimizer, initialization, minibatch order, validation rule, starts, and direct VQE search. The extended histories reproduce the original training prefixes. All six runs reach epoch 500 and stop there; four retain their original best checkpoint, while the seed-46 runs select epochs 280 and 366 instead of 56 and 43.

Table 30: Extended M-GRU stopping schedule on independent two-layer grids. All methods use the same measurement records and 20 VQE starts per seed. Entries are true endpoint energy, mean $\pm$ sample standard deviation. Matching cap, minimum, and patience does not match realized epochs or FLOPs. Reported metric values: $\times 10^{0}$; settings/counts unscaled.

Setting	M-GRU original	M-GRU extended	CQS-P+C
2D $2 \times 3, L = 2$	$-2.737_{\pm 0.229}$	$-2.439_{\pm 0.644}$	$-3.009_{\pm 0.003}$
2D $3 \times 3, L = 2$	$-4.049_{\pm 0.084}$	$-3.613_{\pm 0.835}$	$-4.506_{\pm 0.014}$

The extended schedule leaves aggregate endpoints worse in this intervention (Table 30). In each grid, the changed seed-46 checkpoint has slightly lower measured validation MSE but a worse true optimization endpoint. This separates checkpoint prediction quality from search behavior, and rules out the original stopping schedule alone as an explanation of the observed gap under this intervention. M-GRU still uses direct search, so this intervention identifies the stopping-schedule contribution rather than a complete representation-only contrast against product+C. The six extended fits take 55.1 s of local training; timing is reported as local execution context rather than a matched-machine cost comparison.

L.5 Effective dimension and an analytic grid reference

The shared-angle grids have $2L$ raw control coordinates. The initial R_X block acts on $|+\rangle^{\otimes N}$ and changes only global phase, leaving at most $D_{\text{eff}} = 2L - 1$ effective coordinates: one at $L = 1$ and three at $L = 2$. At one effective coordinate, Eq. (9) reduces to a univariate Fourier series after summing rank channels. This setting tests frequency fitting and search, rather than interactions among several effective controls.

For any depth, setting all R_{ZZ} angles to zero leaves a product of X eigenstates with eigenvalue $+1$. For the grid sign convention,

$$E_{\mathrm{ref}} = J \sum_{\langle i,j \rangle} \langle Z_i Z_j \rangle + h \sum_i \langle X_i \rangle = hN = -N/2. \tag{26}$$

At $L = 1$, the final R_{ZZ} block is diagonal and commutes with every $Z_i Z_j$. These expectations remain zero for all controls, while $\langle X_i \rangle \leq 1$ and $h = -0.5$. Hence $E \geq -N/2$, with equality at zero R_{ZZ}: this reference is optimal within the single-layer ansatz. For $L = 2$ it is only a feasible reference, not an ansatz or Hamiltonian ground-state certificate.

Table 31 places the independent-seed endpoints on this scale. The $2 \times 3, L = 1$ average is above the known feasible point; the $3 \times 3, L = 1$ average approaches its ansatz optimum. The two-layer averages improve slightly on the feasible reference. We retain the mean over all starts to measure the stability of the prescribed search; it is distinct from selecting the best restart or initializing at the known point.

Table 31: Shared-grid effective dimension and analytic reference. CQS-P+C values summarize independent seeds 45/46/47. The final column subtracts the fixed feasible reference from the unrounded mean endpoint; negative values are better. The reference is ansatz-optimal only for $L = 1$. Reported metric values: $\times 10^0$; settings/counts unscaled.

Setting	D_{eff}	E_{ref}	CQS-P+C	$\bar{E} - E_{\mathrm{ref}}$
2D $2 \times 3, L = 1$	1	-3.000	$-2.667_{\pm 0.153}$	0.333
2D $2 \times 3, L = 2$	3	-3.000	$-3.009_{\pm 0.003}$	-0.009
2D $3 \times 3, L = 1$	1	-4.500	$-4.500_{\pm 0.000}$	0.000
2D $3 \times 3, L = 2$	3	-4.500	$-4.506_{\pm 0.014}$	-0.006

L.6 Complete Fourier moment control

To test whether low-dimensional Fourier regression accounts for the grid results, we fit the complete frequency cube $|w_d| \leq Q$ over the D_{eff} active shared coordinates. A real basis contains an intercept and $\sqrt{2}\cos(\mathbf{w}^\top \boldsymbol{\theta})$ and $\sqrt{2}\sin(\mathbf{w}^\top \boldsymbol{\theta})$ for one representative of each pair $\{\mathbf{w}, -\mathbf{w}\}$. Its dimension is $(2Q+1)^{D_{\mathrm{eff}}}$. Unlike the low-rank product, this control fits an independent coefficient for each basis function and output moment.

All twelve setting–seed runs reuse the grid data of seeds 45/46/47: 160 training, 40 measured-validation, and 200 exact-test controls. The labels are the stored all-X/all-Z moment estimates, with no new measurements. We jointly select $Q \in \{1, 2, 3, 4\}$ and $\lambda \in \{0, 10^{-8}, 10^{-6}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10\}$ by measured validation-moment MSE. Each candidate minimizes

$$n^{-1} \|\Phi B - \widetilde{M}\|_F^2 + \lambda \|B_{\mathrm{nonconstant}}\|_F^2. \tag{27}$$

Here n is the number of training controls, Φ the real Fourier design matrix, $\widetilde{M}$ the measured-moment matrix, and B the coefficient matrix. The intercept is unpenalized. Centered thin SVD supplies the ridge path, including the minimum-norm solution when $\lambda = 0$. Only training rows fit coefficients; validation rows are not subsequently folded into training. This is a complete-Fourier ridge control, not a relabeling of the original NC unsquared-norm fit.

Direct and continued optimization share all 20 starts, 100 float64 Adam updates, learning rate 0.1, and clipping to $[-\pi, \pi]$. Gaussian convolution multiplies a frequency-vector coefficient by $\exp(-\sigma^2 \|\mathbf{w}\|_2^2/2)$, with $\sigma_t = \max(0, 1 - t/60)$. Coordinates correspond to independent shared controls, so repeated gates do not receive independent perturbations. Exact energies score all endpoints after optimization; they do not select a frequency range, regularizer, or restart.

Measured validation selects $Q = 3$ or 4 on the single-layer grids and $Q = 2$ on the two-layer grids. Fourier+C nearly reproduces both single-layer product endpoints (Table 32). The low-rank product retains lower mean endpoints at two layers, with respective differences of 2.940×10^{-2} and 6.460×10^{-2}. These are comparisons between fitted moment representations under the same continuation, rather than evidence that single-layer optimization requires multiplicative interactions.

Table 32: Complete Fourier moment control on independent grid seeds 45/46/47. Mean ± sample standard deviation of three seed-level true endpoint energies. Both Fourier columns use the same selected coefficients; +C changes only the search objective. Reported metric values: $\times 10^0$; settings/counts unscaled.

Setting	Fourier direct	Fourier+C	CQS-P+C
2D $2 \times 3, L = 1$	$-2.033_{\pm 0.153}$	$-2.667_{\pm 0.153}$	$-2.667_{\pm 0.153}$
2D $2 \times 3, L = 2$	$-2.229_{\pm 0.381}$	$-2.980_{\pm 0.031}$	$-3.009_{\pm 0.003}$
2D $3 \times 3, L = 1$	$-2.567_{\pm 0.306}$	$-4.500_{\pm 0.000}$	$-4.500_{\pm 0.000}$
2D $3 \times 3, L = 2$	$-3.143_{\pm 0.423}$	$-4.442_{\pm 0.041}$	$-4.506_{\pm 0.014}$

The complete twelve-run sweep takes 5.5 s locally with two CPU threads, of which 0.33 s is fitting and validation selection; this is not a matched-machine training-speed claim against the stored neural runs.

M Paired CQS and NC-style simulation comparisons

M.1 Task-specific reference algorithms

We apply the public experimental code of Liao et al. (2026) to the same cached controls, labels, and splits as the neural models. For FSPT, the feature sampler uses 1,000 draws with support cap 25 and key 1234, retaining repeated features and constants. Each period is represented in expanded gate-occurrence coordinates. Products of repeated sine and cosine factors are compressed into exponent counts without changing their values. Every period/site output has independent coefficients and stopping. The objective is $\|\Phi w - y\|_2^2 + \alpha\|w\|_2$, with $\alpha = 1$, Adam step size 0.1, a 10,000-step cap, EMA decay 0.9, and relative stopping threshold 10^{-5}. The public sampler is retained as implemented, rather than replaced by uniform sampling of distinct features.

TFIM uses complete support-two features and the public Fig. 3 h_{qs} fitting profile: $\alpha = 0.1$, step size 0.01, cap 5,000, EMA decay 0.9, and threshold 10^{-4}. Gate parameters are independent in these final-layer tasks. The separately retained h_{cs} fit uses $\alpha = 10$. Both fits can receive the same scalar shadow-derived labels; the subscript identifies the reference fitting profile, not a different label acquisition in this comparison. The NC optimization rule is distinct from the squared L2 ridge objective used in the auxiliary TFIM controls.

The project drive includes N cluster terms with truncated boundary factors, whereas the reference FSPT circuit uses $N - 2$ interior terms. Thus the comparison transfers the public algorithm to our physical task. It does not reproduce every reference circuit or imply its theoretical assumptions for our cached distribution. The first-order CQS-P adapter follows the independent-control convention of our TFIM model, not a complete Fourier representation over expanded gates.

M.2 Validation-tuned NC objective control

Table 2 retains both public-code fitting profiles and adds a stronger validation-selected control. NC-tuned uses exactly their unnormalized support-two features, including a penalized constant, and minimizes

$$F_\alpha(w) = \|Aw - y\|_2^2 + \alpha\|w\|_2. \tag{28}$$

Here $A \in \mathbb{R}^{n \times F}$ is the feature matrix, $w \in \mathbb{R}^F$ the coefficient vector, and $y \in \mathbb{R}^n$ the training-label vector. There is no centering, free intercept, or division by sample count. For a nonzero minimizer, the stationarity condition gives

$$w(t) = (A^\top A + tI_F)^{-1} A^\top y, \qquad 2t\|w(t)\|_2 = \alpha. \tag{29}$$

Here I_F is the $F \times F$ identity and $t > 0$ is an auxiliary scalar, distinct from the continuation step index. We solve this scalar equation using the eigensystem of the smaller Gram matrix. If $\alpha \geq 2\|A^\top y\|_2$, the optimum is zero by the norm subgradient condition. These convex optimality conditions solve the original unsquared-norm objective; using a ridge linear system internally does not replace it with a fixed squared penalty. Each of the 180 candidates passes relative primal–dual gap 10^{-7} and normalized stationarity residual 10^{-5} checks.

We select $\alpha \in \{0.01, 0.03, 0.1, 0.3, 1, 3, 10, 30, 100, 300, 10^3, 3 \cdot 10^3, 10^4, 3 \cdot 10^4, 10^5\}$ by measured validation MSE, without refitting on validation or using exact test targets. The cached 2,000/400/400 splits, termwise marginal-based $T = 32$ scalar labels, and seeds 42/43/44 are unchanged. Thus this control changes optimization and regularization selection, not measurements or features. It is distinct from the original Adam/EMA algorithm profiles.

Table 33: Validation-tuned NC features on the shared-label TFIM task. Mean ± sample standard deviation across the three paired partitions; these are not independent data pools. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

N	Exact-test MAE	Exact-test R^2
8	$2.124_{\pm 0.206}$	$2.943_{\pm 0.597}$
10	$2.382_{\pm 0.059}$	$2.214_{\pm 0.732}$
12	$2.648_{\pm 0.053}$	$2.427_{\pm 0.508}$
14	$3.149_{\pm 0.077}$	$-0.200_{\pm 0.531}$

Selected α values for seeds 42/43/44 are $(100, 300, 100)$ at $N = 8$, $(300, 300, 300)$ at $N = 10, 12$, and $(300, 3000, 3000)$ at $N = 14$. The stronger control reduces the apparent neural advantage. At $N = 14$, its mean R^2 is near zero, as is the recurrent model's $3.170 \times 10^{-2} \pm 1.362 \times 10^{-1}$, while the product model gives $2.385 \times 10^{-1} \pm 7.550 \times 10^{-2}$. Solving at the fixed public α values does not yield the tuned result: for example, at $N = 8$ the convex optima have MAE $1.065 \pm 8.100 \times 10^{-2}$ for $\alpha = 0.1$ and $4.194 \times 10^{-1} \pm 2.070 \times 10^{-2}$ for $\alpha = 10$. Optimization to convergence alone therefore does not explain the improvement; regularization selection matters.

M.3 Training, paired outcomes, and auxiliary comparisons

The paired architecture comparisons use three seeds, 42/43/44, on the existing pools. The CQS-P FSPT representation uses rank 32, harmonic order one, Adam step size 10^{-3}, weight decay 10^{-4}, batch 32 controls, patience 30, and cap 300 for the matrix. TFIM scalar tasks retain caps 250 for Table 2 and exact-label scaling, and 80 for the measurement-budget experiment. Validation MSE selects neural checkpoints. No continuation is used for prediction. Period grouping changes execution only and preserves the full-trajectory objective.

Table 34 gives the exact-label counterpart to the finite-shot main table under the current physically ordered caches and paired three-seed protocol.

Table 34: Exact-label TFIM prediction under the 250-epoch neural cap. Test MAE, three-seed mean ± sample standard deviation. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

Setting	CQS-R	CQS-P	NC h_{qs}
$N = 8$	$0.979_{\pm 0.045}$	$\mathbf{0.537}_{\pm 0.039}$	$1.378_{\pm 0.030}$
$N = 10$	$1.163_{\pm 0.038}$	$\mathbf{0.686}_{\pm 0.026}$	$2.094_{\pm 0.188}$
$N = 12$	$1.363_{\pm 0.026}$	$\mathbf{0.817}_{\pm 0.017}$	$3.114_{\pm 0.126}$
$N = 14$	$1.521_{\pm 0.039}$	$\mathbf{0.939}_{\pm 0.045}$	$4.801_{\pm 0.018}$

M.4 All finite-measurement allocations

Table 35 reports the complete shot sweep. Table 36 and Figure 14 retain every training-count allocation under each requested budget cap. Validation controls count toward the cap. Shots per control are integral and at most 512; for example, $M = 125$ with 50 validation controls can consume at most 89,600 shots even under a 128k cap. The allocation sweep therefore describes capped resource use, rather than equality of actual expenditure at every point. The allocation sweep is a retrospective resource study under simulated termwise-marginal labels. It supports comparisons within this noise model; hardware savings under joint device noise require a separate deployment protocol.

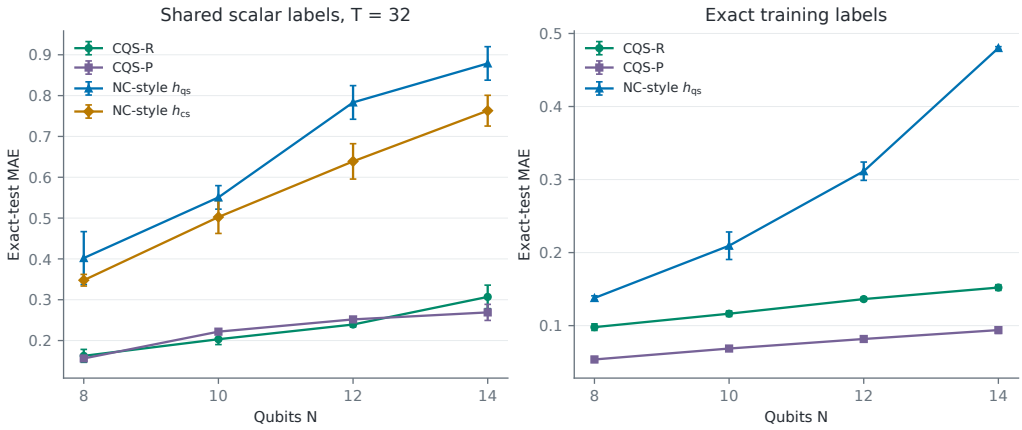


Figure 13: Paired TFIM prediction. Left: shared $T = 32$ labels. Right: exact training labels. NC fitting configurations are distinguished from the common label source. Error bars summarize three seeds.

Table 35: Shot-level TFIM comparison. Exact-test MAE, three-seed mean ± sample standard deviation; neural cap 80. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

Setting	CQS-R	CQS-P	NC h_{qs}	NC h_{cs}
Exact	$1.000_{\pm 0.046}$	$\mathbf{0.670}_{\pm 0.016}$	$1.378_{\pm 0.030}$	$1.138_{\pm 0.046}$
$T = 8$	$2.274_{\pm 0.180}$	$\mathbf{2.060}_{\pm 0.230}$	$7.886_{\pm 0.346}$	$6.927_{\pm 0.221}$
$T = 16$	$1.976_{\pm 0.054}$	$\mathbf{1.748}_{\pm 0.033}$	$5.743_{\pm 0.506}$	$5.028_{\pm 0.257}$
$T = 32$	$\mathbf{1.537}_{\pm 0.195}$	$1.588_{\pm 0.108}$	$4.302_{\pm 0.434}$	$3.441_{\pm 0.365}$
$T = 64$	$1.358_{\pm 0.102}$	$\mathbf{1.312}_{\pm 0.100}$	$2.981_{\pm 0.413}$	$2.479_{\pm 0.184}$
$T = 128$	$1.236_{\pm 0.094}$	$\mathbf{1.093}_{\pm 0.042}$	$2.500_{\pm 0.304}$	$1.989_{\pm 0.061}$
$T = 256$	$1.097_{\pm 0.048}$	$\mathbf{0.906}_{\pm 0.062}$	$2.053_{\pm 0.262}$	$1.612_{\pm 0.060}$
$T = 512$	$1.051_{\pm 0.024}$	$\mathbf{0.812}_{\pm 0.071}$	$1.733_{\pm 0.110}$	$1.401_{\pm 0.044}$

Table 36: Every requested budget cap and training-count allocation. Exact-test MAE, three-seed mean ± sample standard deviation. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

Setting	CQS-R	CQS-P	NC h_{qs}	NC h_{cs}
16k, $M = 125$	$2.516_{\pm 0.147}$	$\mathbf{2.457}_{\pm 0.356}$	$4.284_{\pm 0.487}$	$3.082_{\pm 0.202}$
16k, $M = 250$	$2.479_{\pm 0.037}$	$\mathbf{2.192}_{\pm 0.144}$	$4.457_{\pm 0.413}$	$3.021_{\pm 0.314}$
16k, $M = 500$	$2.631_{\pm 0.111}$	$\mathbf{2.433}_{\pm 0.055}$	$7.292_{\pm 0.245}$	$4.794_{\pm 0.182}$
16k, $M = 1000$	$2.405_{\pm 0.112}$	$\mathbf{2.288}_{\pm 0.075}$	$10.450_{\pm 1.186}$	$7.422_{\pm 0.408}$
16k, $M = 2000$	$2.379_{\pm 0.039}$	$\mathbf{2.098}_{\pm 0.122}$	$8.706_{\pm 0.310}$	$8.145_{\pm 0.243}$
32k, $M = 125$	$2.582_{\pm 0.123}$	$\mathbf{2.344}_{\pm 0.309}$	$3.774_{\pm 0.267}$	$2.868_{\pm 0.159}$
32k, $M = 250$	$2.303_{\pm 0.217}$	$\mathbf{2.207}_{\pm 0.150}$	$4.018_{\pm 0.255}$	$2.660_{\pm 0.207}$
32k, $M = 500$	$2.236_{\pm 0.182}$	$\mathbf{2.082}_{\pm 0.062}$	$5.555_{\pm 0.323}$	$3.466_{\pm 0.160}$
32k, $M = 1000$	$\mathbf{2.140}_{\pm 0.099}$	$2.219_{\pm 0.147}$	$7.908_{\pm 0.467}$	$4.723_{\pm 0.081}$
32k, $M = 2000$	$1.982_{\pm 0.168}$	$\mathbf{1.915}_{\pm 0.047}$	$6.920_{\pm 0.075}$	$5.506_{\pm 0.224}$
64k, $M = 125$	$2.610_{\pm 0.156}$	$\mathbf{2.253}_{\pm 0.194}$	$3.745_{\pm 0.339}$	$2.714_{\pm 0.178}$
64k, $M = 250$	$\mathbf{2.095}_{\pm 0.069}$	$2.187_{\pm 0.079}$	$3.781_{\pm 0.220}$	$2.394_{\pm 0.176}$
64k, $M = 500$	$\mathbf{1.844}_{\pm 0.033}$	$1.908_{\pm 0.080}$	$4.441_{\pm 0.243}$	$2.620_{\pm 0.043}$
64k, $M = 1000$	$\mathbf{1.749}_{\pm 0.193}$	$1.844_{\pm 0.265}$	$5.830_{\pm 0.574}$	$3.402_{\pm 0.115}$
64k, $M = 2000$	$1.568_{\pm 0.104}$	$\mathbf{1.565}_{\pm 0.094}$	$4.659_{\pm 0.610}$	$3.914_{\pm 0.264}$
128k, $M = 125$	$2.545_{\pm 0.142}$	$\mathbf{2.247}_{\pm 0.181}$	$3.612_{\pm 0.497}$	$2.682_{\pm 0.172}$
128k, $M = 250$	$\mathbf{1.806}_{\pm 0.092}$	$1.874_{\pm 0.109}$	$3.686_{\pm 0.258}$	$2.239_{\pm 0.202}$
128k, $M = 500$	$\mathbf{1.615}_{\pm 0.043}$	$1.679_{\pm 0.130}$	$3.674_{\pm 0.105}$	$2.185_{\pm 0.021}$
128k, $M = 1000$	$\mathbf{1.489}_{\pm 0.051}$	$1.531_{\pm 0.121}$	$4.393_{\pm 0.159}$	$2.579_{\pm 0.126}$
128k, $M = 2000$	$1.404_{\pm 0.056}$	$\mathbf{1.384}_{\pm 0.083}$	$3.420_{\pm 0.204}$	$2.758_{\pm 0.032}$



Figure 14: All budget allocations with paired labels. Both axes are logarithmic. Curves summarize three seeds without selecting the lowest test-error allocation.

M.5 A FIRST-PERIOD HARMONIC RESTRICTION

Why increasing rank cannot restore a missing frequency. The current FSPT adapter uses one first-order factor per independent control in each period. Fix the disorder vector $\mathbf{J}$ and consider the first-period head. In each rank channel, all disorder factors are constants with respect to $\phi = \pi - 2\delta$, leaving one factor in $\mathrm{span}\{1, \cos\phi, \sin\phi\}$. Summing channels therefore gives

$$\widehat{y}_{1,i}(\phi, \mathbf{J}) = A_{i,R}(\mathbf{J}) + B_{i,R}(\mathbf{J}) \cos\phi + C_{i,R}(\mathbf{J}) \sin\phi.$$

Increasing R enriches the dependence of these three coefficients on disorder, but leaves the available ϕ frequencies unchanged. Multiplication between different control coordinates does not create additional ϕ factors within this head. Thus a target containing a nonzero second or third harmonic lies outside its hypothesis class at every rank, independently of the optimizer.

The exact response requires higher harmonics. For the drive and initial state used here, the first-period expectation is

$$y_{1,i} = \cos\theta_i \cos\phi - \sin\theta_i \sin\phi \cos^{d_i}\phi, \qquad \theta_i = -2J_i, \tag{30}$$

where $d_i = 1$ at a boundary and $d_i = 2$ in the bulk. Expanding the angle products gives

$$y_{1,i} = \begin{cases} \cos\theta_i \cos\phi - \frac{1}{2}\sin\theta_i \sin(2\phi), & d_i = 1, \\ \cos\theta_i \cos\phi - \frac{1}{4}\sin\theta_i [\sin\phi + \sin(3\phi)], & d_i = 2. \end{cases}$$

For generic disorder with $\sin\theta_i \neq 0$, these responses require second and third harmonics, respectively. At the exceptional values $\sin\theta_i = 0$, this particular obstruction vanishes. The first-period frequency requirement is $Q_\phi \geq 2$ at boundaries and $Q_\phi \geq 3$ in the bulk. Meeting it removes the missing-frequency obstruction; it does not guarantee successful fitting with a fixed rank and training budget.

An irreducible approximation error. For a fixed $\mathbf{J}$ and uniform ϕ over a full 2π period, orthogonality of Fourier modes gives the best first-harmonic approximation error exactly:

$$\inf_{g \in \mathrm{span}\{1, \cos\phi, \sin\phi\}} \mathbb{E}_\phi[(y_{1,i} - g)^2] = \begin{cases} \sin^2\theta_i/8, & d_i = 1, \\ \sin^2\theta_i/32, & d_i = 2. \end{cases}$$

Every first-period $Q_\phi = 1$ head obeys this MSE floor on that slice, even with unlimited rank and perfect optimization. This full-period calculation illustrates the structural error; the experiment samples a different angle interval, for which we compute the projection directly below.

Equation (30) agrees with the cached first-period targets for $N = 4, 6, 8, 10, 20$ to maximum absolute error below 10^{-6}. At fixed $J_i = 0.37$, least-squares projection onto the first-harmonic span over 4,096 equally spaced $\delta \in [0, 2]$ gives MSE 2.285×10^{-2} at the boundary and 1.366×10^{-2} in the bulk (Figure 15). Including harmonics through order three reduces projection residuals to numerical precision. This is an oracle function-space diagnostic, not a trained higher-order model or a retuned baseline. It identifies an expressivity restriction at the first period, not a complete explanation of every long-period error.

Shared gates and later periods. A harmonic cutoff applies to an independent control coordinate, not to the number of distinct parameter names in a gate sequence. Each occurrence of a Pauli rotation contributes at most one Fourier degree in its gate angle. There are N rotations sharing ϕ per period and one occurrence of each cluster angle θ_i. A sufficient per-period frequency envelope is therefore $(N, 1, \ldots, 1)$, giving $(Nk, k, \ldots, k)$ at head k after controls are tied across periods. For $N = 8, K = 12$, this bound is $(96, 12, \ldots, 12)$, not the order-three envelope sufficient for the first-period formula alone. These are conservative accessible-frequency bounds; individual local observables may use lower effective orders. Equation (30) needs only orders two or three, despite N shared rotations. At head k, the adapter multiplies up to k factors evaluated at the same physical ϕ, so its available degree can reach kQ_ϕ. Consequently, the first-period degree-one argument must not be applied unchanged to later heads. Harmonics through order eight in each shared-ϕ factor cover the coordinate-wise envelope for $N = 8$, while rank 32 still restricts the coefficient tensor.

Implication for the observed error. The missing modes contribute structurally to first-period error. Projection MSE on a fixed-disorder slice and trajectory MAE measure different quantities, so the projection does not quantify the full trajectory gap. Retraining with an expanded basis sharply improves the first period while leaving later-period error nearly unchanged (Appendix M.6). Finite-rank approximation, sampling, and fitting effects remain possible sources of the later-period error.

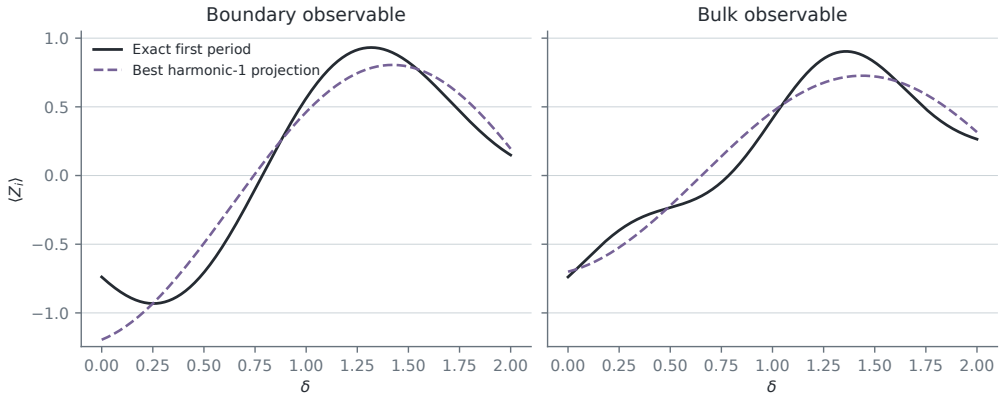


Figure 15: Exact first-period response and its best first-harmonic projection at fixed disorder. The nonzero residual persists regardless of the number of multiplicative channels under the current shared-control first-period parameterization.

M.6 Retraining with expanded harmonics

We retrain the $N = 8, K = 12$ product adapter with shared-control factor orders $Q_\phi = 1$ and 8, retaining first-order cluster-angle factors. Both use rank 32, the same physical-site masks and prefix-specific heads, and the original 2,000/400/400 trajectory partitions for seeds 42/43/44. All twelve periods supply exact training labels; validation MSE selects checkpoints. Adam uses learning rate 10^{-3}, weight decay 10^{-4}, batch size 32, cap 300, and patience 30. Factor coefficients follow the

same inverse-square-root basis-size initialization rule. The expanded model has 821,856 parameters versus 542,304, so rank and fitting rules are matched rather than parameter count.

The first-order fits stop at epochs 179/272/143, selecting epochs 149/242/113. The expanded fits stop at 173/188/300, selecting 143/158/276. All six runs use float32 CUDA computation on RTX 4090 GPUs with TF32 disabled. First-order predictions reproduce the stored matrix predictions. The comparison therefore changes the available harmonics within the fitted product family, without introducing observations or changing the data pool.

Table 37: FSPT harmonic-order control at $N = 8, K = 12$. Test MAE, mean $\pm$ sample standard deviation over three paired partition/initialization runs. Period blocks are averaged within each run before computing the standard deviation. CQS-R and NC retain their matrix predictions. Reported metric values: $\times 10^{-1}$; settings/counts unscaled.

Model	All periods	First period	Periods 2–12
CQS-P, $Q_\phi = 1$	$2.951_{\pm 0.046}$	$1.291_{\pm 0.009}$	$3.101_{\pm 0.051}$
CQS-P, $Q_\phi = 8$	$2.883_{\pm 0.054}$	$0.506_{\pm 0.074}$	$3.099_{\pm 0.052}$
CQS-R	$1.861_{\pm 0.033}$	$0.498_{\pm 0.005}$	$1.984_{\pm 0.036}$
NC h_{qs}	$2.614_{\pm 0.183}$	$0.164_{\pm 0.004}$	$2.837_{\pm 0.199}$

The expanded model improves all three seed MAEs; its paired change is $-6.700 \times 10^{-3} \pm 2.900 \times 10^{-3}$, a 2.3% reduction in the aggregate mean (Table 37). First-period MAE falls from $1.291 \times 10^{-1} \pm 9.000 \times 10^{-4}$ to $5.060 \times 10^{-2} \pm 7.400 \times 10^{-3}$, whereas the mean over periods 2–12 changes from 3.101×10^{-1} to 3.099×10^{-1}. Figure 16 localizes the gain. Frequency expansion addresses the early-period restriction but leaves the full-trajectory gap under this rank and training protocol. The remaining gap motivates separate controls for finite rank, fitting dynamics, and sample size.

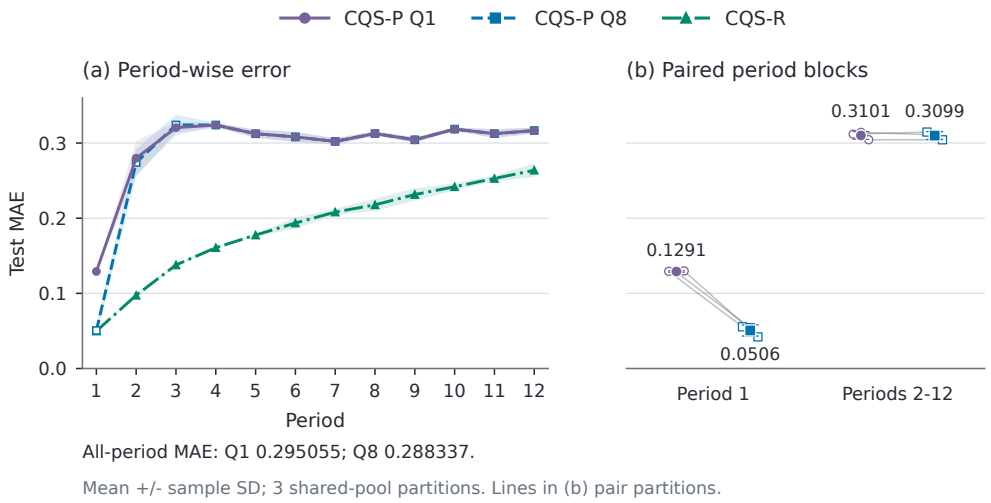


Figure 16: FSPT prediction after expanding shared-control harmonics. Left: period-wise test MAE for the two product variants and the recurrent reference. Right: first-period and later-period errors, computed per run before aggregation; thin lines pair the same splits. Shading and error bars show sample standard deviations over three paired splits of one pool. Most of the product model's improvement occurs at the first period.

Scope of retained controls. The VQE comparison retains its fitted checkpoints and endpoint protocol; the twelve existing grid NC fits reproduce their reference feature predictions. GRU resets, frozen-state probes, and width diagnostics concern CQS-R, not the multiplicative model.

N Mask Construction and Product Complexity

Pauli-branch masks for TFIM and VQE. A Pauli string is represented, up to phase, by binary X/Z supports (p_x, p_z). For a rotation generator (g_x, g_z), anticommutation is determined by

$$b(p, g) = \big[\mathrm{popcount}(p_x \,\&\, g_z) + \mathrm{popcount}(p_z \,\&\, g_x)\big] \bmod 2.$$

The implementation performs the following reverse scan for each output observable:

1. Initialize the string set S to the observable and the control mask to zero.
2. For a known all-site Hadamard block, replace every $(p_x, p_z) \in S$ by (p_z, p_x). Skip the initial R_X block for the plus-initialized VQE circuit, whose action on the initial state is a global phase.
3. For a parameterized rotation, form $B = \{(p_x \oplus g_x, p_z \oplus g_z) : p \in S,\ b(p, g) = 1\}$. If B is nonempty, mark the gate's control coordinate and replace S by $S \cup B$.

Here $\&$ is bitwise intersection and $\oplus$ is exclusive-or. Gates sharing a control mark the same coordinate. The scan stores distinct Pauli strings, but no amplitudes or trigonometric coefficients. Deduplication merges identical strings without evaluating coefficient cancellations, so the resulting dependency set is conservative rather than guaranteed minimal.

Let G be the number of scanned gates, $B_{\max}$ the largest string-set size, and $s = \lceil N/w \rceil$ the number of machine words per support bitset. With expected constant-time hash-table access apart from bitset processing, a coarse time bound is $O(mGB_{\max}s)$; the per-observable working set uses $O(B_{\max}s)$ words, in addition to stored masks. Each branching rotation can at most double the set, so $B_{\max} \le \min(2^b, 4^N)$ after b such rotations. This preprocessing is performed at model construction and can grow exponentially even though subsequent prediction uses no Pauli-string traversal.

Support-union masks for FSPT. The FSPT adapter instead initializes a single support pair (p_x, p_z) for each local-Z observable. It scans the prefix gates in reverse. Whenever $(p_x \,\&\, g_z) \neq 0$ or $(p_z \,\&\, g_x) \neq 0$, it marks the corresponding period/control slot and updates

$$p_x \leftarrow p_x \mid g_x, \qquad p_z \leftarrow p_z \mid g_z,$$

where $\mid$ denotes bitwise union. Otherwise both supports remain unchanged. This rule keeps an overapproximation of X/Z support across possible branches; it does not store individual strings or apply an exact parity test to a branched operator. A zero overlap guarantees commutation for all strings contained in the support bounds. False-positive dependencies can remain.

A prefix with G_k gates costs $O(mG_ks)$ to scan and uses $O(s)$ working support per observable, plus the dense output mask. The current implementation constructs each prefix head separately. Since $G_k = O(kN)$, constructing all K masks costs $O(mNK^2s)$. Both mask procedures use circuit structure and the specified initial-state simplification, without evaluating training or test expectations.

Active factors and allocated factors. For one head with D control slots, m outputs, rank R, and uniform harmonic order $Q \ge 1$, an implementation storing only active factors would allocate

$$P_{\mathrm{active}} = R(2Q + 1) \sum_o |C_o| + mR + m$$

and evaluate them in $O(mR\bar{C}Q + mR)$ operations, where $\bar{C} = m^{-1} \sum_o |C_o|$. The reported implementation instead allocates a dense factor tensor of shape $m \times D \times R \times (2Q + 1)$. It builds the Fourier basis, contracts all factors, and only then replaces inactive factors by one. Thus

$$P_{\mathrm{dense}} = mDR(2Q + 1) + mR + m, \qquad T_{\mathrm{dense}} = O(DQ + mDRQ + mDR) = O(mDRQ).$$

These parameter counts exclude nontrainable masks. Inactive factor coefficients are allocated although their contributions are masked out. For batch size B_{batch}, the factor-value intermediate uses $O(B_{\mathrm{batch}}mDR)$ storage. Dependency masks restrict the represented function but do not by themselves reduce dense contraction work. As depth grows, support expansion can increase $\bar{C}$ toward D, reducing the potential benefit of an active-factor implementation; expansion is circuit-dependent.

Complete FSPT trajectories. Head k has $D_k = k(N+1)$ period/control slots, even though the same physical controls are repeated across periods. With independent heads and uniform Q, the allocated parameter count is

$$P_{\mathrm{traj}} = mR(2Q+1)(N+1)\frac{K(K+1)}{2} + K(mR+m).$$

Summing dense evaluation over heads gives $O(mRQ\sum_k D_k) = O(mRQ(N+1)K^2)$ work. Grouping heads for batched execution changes constants and intermediate storage, not this order. With different orders by control, replace $D_k(2Q+1)$ by the sum of the basis sizes of its slots. These costs are separate from one-time mask construction. They also differ from recurrent cell-update counts: batching and device utilization determine how arithmetic work translates into the measured wall-clock times.

O Limitations and Future Work

The controlled simulations reveal several task-dependent limits. In TFIM, the product and recurrent models change ranking between exact and noisy supervision. In FSPT, the tested $Q_\phi = 1$ product adapter misses first-period frequencies, and increasing Q_ϕ leaves later-period error largely unchanged. VQE endpoints depend on both the fitted surrogate and the continuation schedule; paired summaries appear in Table 1. These differences tie representation choice to circuit family, label regime, and downstream objective.

Held-out expectation error is only one proxy for downstream utility. VQE endpoints are determined by the interaction between fitting error, optimizer path, and feasible ansatz reference. Future evaluations should therefore report the target downstream quantity together with prediction error, especially when the surrogate is used inside an adaptive search.

Adaptive frequency selection and mask-aware rank control are natural extensions for product models. Recurrent studies can distinguish stored computation from horizon extrapolation and examine response features obscured by average MAE. Optimization studies should vary continuation schedules, restart budgets, and Hamiltonian families under paired measurements and validation rules. Broader finite-shot and hardware-facing protocols can then test circuit-guided reuse under device noise, calibration drift, and realistic measurement allocation.