TOWARD FULLY UNITARY QUANTUM BACKPROPAGATION

Anonymous authors
Paper under double-blind review

ABSTRACT

We introduce a fully unitary quantum backpropagation framework that trains deep quantum neural networks without any mid-circuit measurement. The algorithm operates in a hybrid Schrödinger–Heisenberg paradigm: the forward pass alternates weight unitaries with QSVT activation layers (polynomial approximations of non-linearities such as sigmoid or tanh, realised as block-encodings), caching both pre-activation states $|\psi^{(l)}\rangle$ and activation states $|a^{(l)}\rangle$ in ancilla registers; the backward pass propagates the loss operator through both activation and weight layers in the Heisenberg picture and coherently writes gradient states into ancilla registers—entirely without measurement collapse. Three algorithmic enhancements—an operator cache, a barren-plateau detection-and-recovery mechanism, and a quantum Adam optimiser with warmup and cosine decay—are each supported by formal guarantees. Our main theoretical contributions are: (i) an $\mathcal{O}(dL^2 d_a D + Ld_a n)$ per-iteration gate bound (d the gradient QSVT degree, d_a the activation QSVT degree, D the maximum layer depth, L the number of layers) obtained via a quantum *circuit-wrapping* model that propagates the loss operator as a nested block-encoding—exact, with no Pauli-string enumeration or truncation error; (ii) an $\mathcal{O}(1/\varepsilon_{\text{conv}}^4)$ convergence rate matching classical Adam whenever the QSVT accuracy satisfies $\varepsilon_{\text{approx}} \leq \varepsilon_{\text{conv}}$; (iii) a probabilistically sound-and-complete barren-plateau detection guarantee for bounded gradient norm ($G \leq \sqrt{2/3}$); and (iv) a near-matching $\Omega(Ln/\log(1/\varepsilon_{\text{approx}}))$ lower bound, establishing near-optimality when $d{\cdot}D$ is polylogarithmic in n. As an alternative propagation strategy we also develop, in Section L, an approximate *Pauli-propagation mode* with a certified truncation-error bound $2L\,\varepsilon_{\text{tol}}$. Empirically, on the MNIST digit-parity benchmark our circuit-wrapping gradient matches a state-of-the-art Mixture-of-Quantum-Experts baseline at every expert count while incurring $204\times$ less quantum-circuit cost per training step; and because this cost is independent of the parameter count, it affords deeper experts that lift test accuracy to 98.2%—exceeding the baseline's best reported 97.54%—at a $408\times$ per-step cost reduction the baseline cannot match.

1 INTRODUCTION

Backpropagation (Rumelhart et al., 1986) owes its central place in deep learning to one property: by reusing intermediate activations, it computes the gradient of all p parameters at a cost of roughly two forward passes, *independently* of p (recalled in Section C). Quantum hardware is meanwhile advancing quickly—6100 neutral-atom qubits and 1121-qubit superconducting processors have been announced, with roadmaps towards 10^5 qubits—which raises the question this paper addresses: can quantum neural networks be trained with the same scaling advantage?

The obstacle is fundamental: quantum states cannot be cloned (Wootters & Zurek, 1982), so the intermediate states that are cached cannot be copied to the backward pass without measurement. Current approaches sidestep this with the *parameter-shift rule* (Mitarai et al., 2018; Schuld et al., 2019), which requires $2p$ separate circuit evaluations—one per gradient component—restoring the p-dependence that backpropagation was designed to eliminate. Quantum natural gradient methods (Stokes et al., 2020) additionally require estimating the $p \times p$ quantum Fisher information matrix, costing $\mathcal{O}(p^2)$ Hadamard-test measurements for the full matrix or $\mathcal{O}(p)$ for block-diagonal

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.

approximations—in either case requiring mid-circuit measurements, introducing statistical error that compounds across layers (Abbas et al., 2023).

Our answer. We show that quantum backpropagation can achieve *parameter-independence* by working entirely in the Heisenberg operator picture. Rather than re-evaluating the circuit with individually perturbed parameters (which costs $\mathcal{O}(p)$ evaluations, since no-cloning prevents caching intermediate states for reuse across layers), we differentiate the *loss operator*, which can be propagated backward coherently as a Pauli-sum block-encoding without mid-circuit measurement. The gradient is then encoded coherently into an ancilla register via Quantum Singular Value Transformation—producing the full gradient in one backward pass, without circuit re-evaluation or mid-circuit collapse.

Contributions. We make the following contributions (Table 1 compares the resulting costs with prior gradient methods).

1. **Algorithm.** A fully unitary quantum backpropagation framework (Algorithm 1) that produces gradient-encoded states $|\nabla_l\rangle$ with no mid-circuit measurement, combining dynamic-programming state caching, Heisenberg operator propagation by circuit wrapping, and QSVT gradient encoding, with an operator cache, barren-plateau recovery, and a quantum Adam update.
2. **Gate complexity** (Theorem 5.1). Each iteration costs at most $\mathcal{O}(d \cdot L^2 \cdot d_a \cdot D + L \cdot d_a \cdot n)$ two-qubit gates—exact, with no Pauli-string enumeration or truncation error—reducing to $\mathcal{O}(d \cdot L_{\text{dirty}} \cdot L \cdot d_a \cdot D + L \cdot d_a \cdot n)$ with a layer-wise cache, and exponentially better in n than state-vector simulation.
3. **Convergence** (Theorem 5.2). Under standard smoothness and bounded-noise assumptions the quantum Adam iterates reach an $\varepsilon_{\text{conv}}$-approximate critical point in $\mathcal{O}(1/\varepsilon_{\text{conv}}^4)$ steps whenever $\varepsilon_{\text{approx}} \leq \varepsilon_{\text{conv}}$.
4. **Barren-plateau detection** (Theorem 5.3). For $G \leq \sqrt{2/3}$ the gradient-variance monitor has an exponentially small false-positive rate and triggers within W_{var} steps with high probability.
5. **Near-optimal lower bound** (Theorem 5.4). Any quantum gradient-encoding algorithm needs $\Omega(Ln/\log(1/\varepsilon_{\text{approx}}))$ two-qubit gates, which our upper bound matches up to a factor $d \cdot L \cdot d_a \cdot D \cdot \log(1/\varepsilon_{\text{approx}})/n$.
6. **Empirical validation** (Section 7). On MNIST digit parity our gradient is bit-identical to parameter shift at $204\times$ lower per-step circuit cost; because that cost does not grow with the parameter count, it affords deeper experts reaching 98.2% test accuracy against the 97.54% best reported by the Mixture-of-Quantum-Experts baseline of Tognini et al. (2025). An approximate Pauli-propagation mode, with the first certified truncation-error bound $2L\,\varepsilon_{\text{tol}}$ for Pauli pruning, is analysed in Section L.

2 Related Work

Gradients for variational circuits. Variational quantum algorithms (Cerezo et al., 2021a; Peruzzo et al., 2014; Farhi et al., 2014) train parameterised quantum circuits (PQCs) with a classical outer loop, and almost all of them obtain gradients from the *parameter-shift rule* (Mitarai et al., 2018; Schuld et al., 2019): two circuit evaluations per parameter, hence $2p$ circuit runs and a measurement after every shift, with extensions to higher-order derivatives (Mari et al., 2021) and stochastic variants (Sweke et al., 2020). We instead encode the *entire* gradient as a quantum state via QSVT, amortising the cost over all parameters at once; this is closer in spirit to natural-gradient preconditioning (Stokes et al., 2020; Haug & Kim, 2021), though we never estimate a Fisher matrix.

Quantum backpropagation. Abbas et al. (2023) showed that naïve backpropagation in the quantum setting requires cloning intermediate states, which the no-cloning theorem forbids (Wootters & Zurek, 1982), and worked around it by storing measurement outcomes at the cost of measurement collapse; Verdon et al. (2019) likewise required mid-circuit measurement. We keep the backward

pass entirely in the *operator domain* via Heisenberg propagation, so the pipeline stays coherent throughout.

QSVT and operator propagation. QSVT (Gilyén et al., 2019; Martyn et al., 2021) unifies quantum linear algebra through polynomial transformations of block-encoded matrices; we are, to our knowledge, the first to use it for coherent gradient-state preparation inside a multi-layer training loop. Heisenberg propagation of Pauli operators underpins classical simulation methods (Gottesman, 1997; Aaronson & Gottesman, 2004; Shao & Montanaro, 2023; Fontana et al., 2023); our certified truncation bound (Theorem L.1) adapts it to gradient computation, but only in the Pauli-propagation mode of Section L.

Barren plateaus, optimisation, and privacy. Barren plateaus (McClean et al., 2018) arise from global cost functions (Cerezo et al., 2021b), hardware noise (Wang et al., 2021) and entanglement (Marrero et al., 2021), and existing mitigations—local costs, layer-wise training, near-identity initialisation (Skolik et al., 2021; Grant et al., 2019)—are heuristic; Theorem 5.3 gives a formal soundness-and-completeness guarantee for a detection-and-recovery scheme. Our convergence analysis builds on Ghadimi & Lan (2013), Kingma & Ba (2015) and Reddi et al. (2018), extended to two error sources specific to our setting: the QSVT approximation bias and the Laplace noise injected in the gradient register for differential privacy (Dwork et al., 2006; Aaronson & Rothblum, 2019). Section B gives the extended survey, covering QNN architectures (Farhi & Neven, 2018; Schuld & Killoran, 2020; Cong et al., 2019; Beer et al., 2020; Abbas et al., 2021; You & Wu, 2021) and classical optimisers for PQCs (Nakanishi et al., 2020; Kübler et al., 2020).

3 Background

3.1 Quantum Preliminaries

We work with an n-qubit QNN whose forward pass at layer l is $|a^{(l)}\rangle = A^{(l)}U^{(l)}(\theta_l)|a^{(l-1)}\rangle$, giving the full circuit $U(W) = \prod_{l=1}^{L} A^{(l)}U^{(l)}(\theta_l)$. A *block-encoding* of A (subnormalized by μ) is a unitary U_A satisfying $(\langle 0|^{\otimes n_a} \otimes I)\, U_A\, (|0\rangle^{\otimes n_a} \otimes I) = A/\mu$. *QSVT* (Gilyén et al., 2019) applies any degree-d polynomial to the singular values of A using $\mathcal{O}(d)$ calls to U_A. The *Pauli expansion* of an n-qubit operator is $H = \sum_i \alpha_i P_i$, $P_i \in \{I, X, Y, Z\}^{\otimes n}$. The loss operator is

$$H_{\text{target}} = I - |\psi_{\text{target}}\rangle\langle\psi_{\text{target}}|, \tag{1}$$

so the loss is

$$\mathcal{L} = \langle\psi_{\text{pred}}|H_{\text{target}}|\psi_{\text{pred}}\rangle = \sum_i \alpha_i \langle\psi_{\text{pred}}|P_i|\psi_{\text{pred}}\rangle, \tag{2}$$

where $|\psi_{\text{pred}}\rangle := |a^{(L)}\rangle = U(W)|x\rangle$ is the predicted output state. No-cloning (Wootters & Zurek, 1982) forbids duplicating an unknown quantum state, which we avoid by working in the *Heisenberg picture*: the loss operator is propagated backward instead of intermediate states being cloned, and each $|\psi^{(l)}\rangle, |a^{(l)}\rangle$ is cached exactly once in the forward pass.

3.2 Quantum Activation Functions

A circuit $\prod_l U^{(l)}(\theta_l)$ without inter-layer non-linearity collapses to a single unitary, losing the expressive benefit of depth, exactly as a classical network does without its activations. We therefore implement the activation as a QSVT polynomial $f_a^{(d_a)}$ of degree d_a: given the pre-activation state $|\psi^{(l)}\rangle = U^{(l)}(\theta_l)|a^{(l-1)}\rangle$, the activation unitary $A^{(l)}$ satisfies

$$|a^{(l)}\rangle = A^{(l)}|\psi^{(l)}\rangle, \qquad A^{(l)} = \text{QSVT}(f_a^{(d_a)}, U_A^{(l)}), \tag{3}$$

where $U_A^{(l)}$ is a fixed block-encoding of the layer-l activation operator, supplied as an input, and $\text{QSVT}(f_a^{(d_a)}, \cdot)$ applies $f_a^{(d_a)}$ to its singular values.

Smooth activations are cheap in this representation: sigmoid and tanh need only degree $d_a = \mathcal{O}(\log(1/\varepsilon_a))$ for an ε_a-accurate approximation (Gilyén et al., 2019), whereas the non-smooth ReLU needs $d_a = \mathcal{O}(1/\varepsilon_a)$ (Section C gives the full list, and discusses why a coherent softmax—and hence attention—remains out of reach). The forward pass caches both $|\psi^{(l)}\rangle$ and $|a^{(l)}\rangle$; the

backward pass consumes $|a^{(l-1)}\rangle$ as the input to layer l's gradient extraction, $|a^{(l-1)}\rangle|0\rangle_g \mapsto |a^{(l-1)}\rangle|\nabla_l^\theta\rangle_g$, and keeps $|\psi^{(l)}\rangle$ available for the activation gradient should the QSVT phases ϕ_l ever be trained (they are held fixed here).

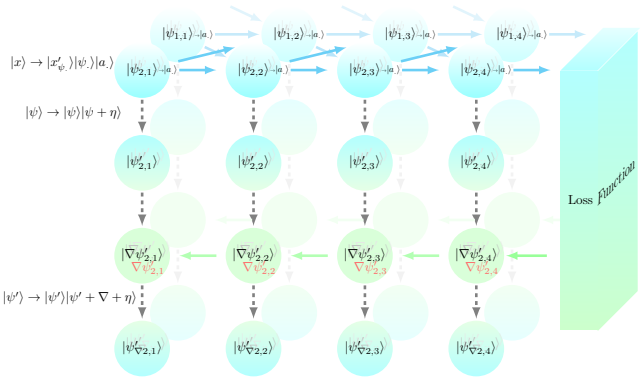


Figure 1: The quantum backpropagation architecture. The top view shows the feedforward QNN alternating weight unitaries $U^{(l)}(\theta_l)$ with QSVT activation unitaries $A^{(l)}$, producing pre-activation states $|\psi^{(l)}\rangle$ and activation states $|a^{(l)}\rangle$; the front view shows gradient updates $|\nabla_\psi\rangle$ propagating backward through both. Columns are data samples, rows are layers.

3.3 No-Cloning and Differential Privacy

No-cloning (Wootters & Zurek, 1982) is sidestepped rather than circumvented: the forward pass caches each $|\psi^{(l)}\rangle$ and $|a^{(l)}\rangle$ exactly once in quantum memory, and the backward pass reads those states while propagating the loss operator, so nothing is ever duplicated. When a gradient register must be accessed more than once (for momentum accumulation and the parameter update), we add controlled Laplace noise for differential privacy (Aaronson & Rothblum, 2019):

$$|\nabla_l^\theta\rangle_g \;\mapsto\; |\nabla_l^\theta + \eta\rangle_g, \qquad \eta \sim \mathrm{Lap}(\lambda_l). \tag{4}$$

This noise provides an implicit ℓ^1 regularization effect (see Proposition I.3) while preserving the unbiased gradient expectation.

4 Algorithm

4.1 Overview

Our framework proceeds in three phases per training iteration (see Fig. 1):

1. **Forward pass (Schrödinger domain).** Pre-activation states $|\psi^{(l)}\rangle$ and activation states $|a^{(l)}\rangle$ are computed alternately and cached via dynamic programming.
2. **Backward pass (Heisenberg domain).** The loss operator $H^{(L)}$ is propagated backward through both activation and weight layers as $H^{(l-1)} \leftarrow (U^{(l)}(\theta_l))^\dagger (A^{(l)})^\dagger H^{(l)} A^{(l)} U^{(l)}(\theta_l)$, and QSVT is used to coherently write gradient states $|\nabla_l\rangle$ into ancilla registers.
3. **Coherent parameter update.** A quantum Adam rule updates the parameter registers $|\theta_l\rangle$ using the gradient ancillae, with warmup, cosine decay, and momentum.

4.2 Key Ingredients

Dynamic-programming state cache. Both $\{|\psi^{(l)}\rangle\}_{l=0}^L$ and $\{|a^{(l)}\rangle\}_{l=0}^L$ are stored after the forward pass, mirroring the classical caching of $z^{(l)}$ and $a^{(l)}$: $|a^{(l-1)}\rangle$ is the input to the gradient block-encoding $U_G^{(l)}$, and $|\psi^{(l)}\rangle$ would supply the QSVT parameters ϕ_l of the activation gradient (Eq. (6)) if activation parameters were trained.

Quantum circuit-wrapping propagation. Rather than re-evaluating the circuit with perturbed parameters for each gradient component, we propagate the *loss operator* backward through both activation and weight layers:

$$H^{(l-1)} = (U^{(l)}(\theta_l))^\dagger (A^{(l)})^\dagger H^{(l)} A^{(l)} U^{(l)}(\theta_l).$$

This is implemented as a quantum circuit wrapping: $U_{H^{(l-1)}}$ is obtained by sandwiching $U_{H^{(l)}}$ between $(A^{(l)}U^{(l)})^\dagger$ and $A^{(l)}U^{(l)}$, adding $\mathcal{O}(d_a \cdot D_l)$ to the block-encoding depth with no Pauli-string enumeration or rank expansion.

QSVT gradient encoding. The gradient splits into weight and activation parts:

$$G_\theta^{(l)} = \frac{\partial (U^{(l)}(\theta_l))^\dagger}{\partial \theta_l} (A^{(l)})^\dagger H^{(l)} A^{(l)} U^{(l)}(\theta_l) + (U^{(l)}(\theta_l))^\dagger (A^{(l)})^\dagger H^{(l)} A^{(l)} \frac{\partial U^{(l)}(\theta_l)}{\partial \theta_l}, \quad (5)$$

$$G_a^{(l)} = (U^{(l)}(\theta_l))^\dagger \left[\frac{\partial (A^{(l)})^\dagger}{\partial \phi_l} H^{(l)} A^{(l)} + (A^{(l)})^\dagger H^{(l)} \frac{\partial A^{(l)}}{\partial \phi_l} \right] U^{(l)}(\theta_l), \quad (6)$$

where ϕ_l denotes the QSVT block-encoding parameters of $U_A^{(l)}$ (which encode the pre-activation amplitudes of $|\psi^{(l)}\rangle$). Both operators are Hermitian by construction. Block-encoding $G_\theta^{(l)}$ as $U_G^{(l)}$ and applying QSVT yields $U_\nabla^{(l)} = P(U_G^{(l)})$; a controlled application on cached $|a^{(l-1)}\rangle$ populates the gradient ancilla coherently: $|a^{(l-1)}\rangle|0\rangle_g \mapsto |a^{(l-1)}\rangle|\nabla_l^\theta\rangle_g$.

Operator cache. Between outer iterations, $H^{(l)}$ changes only if θ_l changes significantly. We cache $H^{(l)}$ and skip re-propagation when $\|\Delta\theta_l\| \le \delta_{\text{cache}}$, replacing the dominant $\mathcal{O}(d \cdot L^2 \cdot d_a \cdot D)$ term with $\mathcal{O}(d \cdot L_{\text{dirty}} \cdot L \cdot d_a \cdot D)$, a factor-$L/L_{\text{dirty}}$ speedup over the uncached cost.

Barren-plateau recovery. We monitor the gradient-norm variance over a sliding window W_{var}. When variance drops below $\varepsilon_{\text{plateau}}$, the lowest-norm layer $l^* = \arg\min_l \|\nabla_l^\theta\|$ is perturbed and its cache invalidated: $\theta_{l^*} \leftarrow \theta_{l^*} + \delta_{\text{perturb}} \cdot \mathcal{N}(0, I)$; $\mathcal{C}[l^*] \leftarrow \varnothing$.

Quantum Adam. The per-iteration learning rate follows a linear warmup then cosine decay:

$$\alpha_t = \alpha_0 \cdot \min\left(\frac{t}{T_w}, 1\right) \cdot \cos\left(\frac{\pi t}{2T}\right). \quad (7)$$

First-order momentum is maintained coherently: $|m_l\rangle \leftarrow \beta|m_l\rangle + (1-\beta)|\nabla_l^\theta\rangle$, and parameters are updated via a controlled-rotation: $|\theta_l\rangle|m_l\rangle \mapsto |\theta_l - \alpha_t m_l\rangle|m_l\rangle$.

4.3 Full Algorithm

Algorithm 1 Fully unitary quantum backpropagation with QSVT activation

Require: $|x\rangle$, $H_{\text{target}} = \sum_i \alpha_i P_i$, weight unitaries $\{U^{(l)}(\theta_l)\}$, activation block-encodings $\{U_A^{(l)}\}$, QSVT polynomials P (gradient), $f_a^{(d_a)}$ (activation), noise scales $\{\lambda_l\}$, iterations T, warmup T_w, momentum β, LR α_0, window W_{var}, δ_{cache}, $\varepsilon_{\text{plateau}}$, δ_{perturb}

Ensure: Coherently updated parameters $|W\rangle$, gradient norm history $\mathcal{G}$

Initialize $|W\rangle = \bigotimes_l |\theta_l\rangle$, $|m_l\rangle \leftarrow |0\rangle$ for all l, $\theta_l^{\text{prev}} \leftarrow \theta_l$ for all l, $\mathcal{C} \leftarrow \{\}$, $\mathcal{G} \leftarrow [\,]$
for $l = 1$ **to** L **do** ▷ *Precompute activation unitaries (inputs are fixed)*
  $A^{(l)} \leftarrow \text{QSVT}(f_a^{(d_a)}, U_A^{(l)})$
end for
for $t = 1$ **to** T **do**
  $|a^{(0)}\rangle \leftarrow |x\rangle$
  for $l = 1$ **to** L **do** ▷ *Forward pass*
    $|\psi^{(l)}\rangle \leftarrow U^{(l)}(\theta_l)|a^{(l-1)}\rangle$; $|a^{(l)}\rangle \leftarrow A^{(l)}|\psi^{(l)}\rangle$ *(cache both)*
  end for
  $H^{(L)} \leftarrow H_{\text{target}}$
  dirty $\leftarrow$ **false**
  for $l = L$ **downto** 1 **do** ▷ *Heisenberg backward pass*
    $\Delta\theta_l \leftarrow \theta_l - \theta_l^{\text{prev}}$ ▷ *parameter change since last cache*
    if $l \notin \mathcal{C}$ **or** $\|\Delta\theta_l\| > \delta_{\text{cache}}$ **or** *dirty* **then**
      $H^{(l-1)} \leftarrow (U^{(l)}(\theta_l))^\dagger (A^{(l)})^\dagger H^{(l)} A^{(l)} U^{(l)}(\theta_l)$; store in $\mathcal{C}[l]$; *dirty* $\leftarrow$ **true** ▷ *exact circuit wrapping, Lemma M.1*
    else
      Retrieve $H^{(l-1)}$ from $\mathcal{C}[l]$
    end if
    Build $U_G^{(l)}$ for $G_\theta^{(l)}$ (both terms, Eq. (5)); apply QSVT: $U_\nabla^{(l)} = P(U_G^{(l)})$
    $|a^{(l-1)}\rangle|0\rangle_g \mapsto |a^{(l-1)}\rangle|\nabla_l^\theta\rangle_g$ ▷ *activation params ϕ_l fixed; only θ_l updated*
    inject $\eta_l \sim \text{Lap}(\lambda_l)$: $|\nabla_l^\theta\rangle \mapsto |\widetilde{\nabla}_l^\theta\rangle$; $g_l \leftarrow \|\widetilde{\nabla}_l^\theta\|$ estimated coherently (QAE on gradient ancilla)
  end for
  append $\frac{1}{L}\sum_l g_l$ to $\mathcal{G}$
  if $t > W_{\text{var}}$ **and** $\text{Var}(\mathcal{G}[\text{last } W_{\text{var}}]) < \varepsilon_{\text{plateau}}$ **then**
    $l^* \leftarrow \arg\min_l g_l$; $\theta_{l^*} \leftarrow \theta_{l^*} + \delta_{\text{perturb}}\mathcal{N}(0, I)$; invalidate $\mathcal{C}[l^*]$
  end if
  $\alpha_t \leftarrow \alpha_0 \min(t/T_w, 1)\cos(\pi t/2T)$
  for $l = 1$ **to** L **do** ▷ *Quantum Adam update*
    $|m_l\rangle \leftarrow \beta|m_l\rangle + (1-\beta)|\widetilde{\nabla}_l^\theta\rangle$; $|\theta_l\rangle|m_l\rangle \mapsto |\theta_l - \alpha_t m_l\rangle|m_l\rangle$; $\theta_l^{\text{prev}} \leftarrow \theta_l$
  end for
end for
return $|W\rangle, \mathcal{G}$

Remark 4.1 Pauli-propagation mode (see Section L)

As an alternative to circuit wrapping, $H^{(l)}$ may be propagated classically as a Pauli sum truncated to its top-$k_{\max}$ terms before each backward step. This mode replaces the circuit-wrapping line of Algorithm 1 and takes $k_{\max}, \varepsilon_{\text{tol}}$ as additional inputs; it removes the growing block-encoding depth, trading the L^2 factor for $k_{\max}$ at the price of an approximation error $\leq 2L\,\varepsilon_{\text{tol}}$. The two modes are mutually exclusive, and circuit wrapping—being exact—is the primary method of this paper.

5 Theoretical Analysis

5.1 Notation and Assumptions

Throughout, $\|\cdot\|$ is the spectral norm on operators and the Euclidean norm on parameter vectors, $p = \sum_l p_l$ is the total parameter count, $D = \max_l \text{depth}(U^{(l)}(\theta_l))$, $G = \sup_W \|\nabla\mathcal{L}(W)\|$ (concretely $G \leq \sqrt{p}\max_l \|G^{(l)}\|$), and σ^2 is the maximum noise variance. We assume **(A1)** ℓ_s-

smoothness, $\|\nabla\mathcal{L}(W) - \nabla\mathcal{L}(W')\| \le \ell_s\|W - W'\|$; **(A2)** bounded gradients, $\|\nabla\mathcal{L}(W)\| \le G$; **(A3)** bounded noise, $\mathbb{E}[\eta_l] = 0$ and $\mathrm{Var}(\eta_l) = 2\lambda_l^2 \le \sigma^2$ for $\eta_l \sim \mathrm{Lap}(\lambda_l)$; and **(A4)** QSVT approximation, $\|\widetilde{\nabla}\mathcal{L} - \nabla\mathcal{L}\| \le \varepsilon_{\mathrm{approx}}$ with $\varepsilon_{\mathrm{approx}}$ controlled by the QSVT degrees d, d_a. No Pauli truncation is performed anywhere in the main paper; the corresponding truncation assumption for Pauli-propagation mode, $\|\widetilde{\nabla}\mathcal{L} - \nabla\mathcal{L}\|_\infty \le 2L\,\varepsilon_{\mathrm{tol}}$, is used only in Section L.

5.2 Gate Complexity

Supporting lemmas (Lemmas M.1 and M.2) in Section M.1 establish the cost structure: $\mathcal{O}(d_a \cdot D)$ depth added per backward step (circuit wrapping, Lemma M.1), and $\mathcal{O}(d \cdot (L-l+1) \cdot d_a \cdot D)$ gates for QSVT gradient extraction at layer l.

Theorem 5.1 Per-iteration gate complexity with circuit wrapping

Each iteration of Algorithm 1 uses at most

$$\mathcal{O}\big(d \cdot L^2 \cdot d_a \cdot D + L \cdot d_a \cdot n\big)$$

two-qubit gates, where d is the gradient QSVT degree, d_a the activation QSVT degree, and $D = \max_l \mathrm{depth}(U^{(l)})$; no Pauli-string enumeration is performed, so the $k_{\max}$ factor of the Pauli-tracking model is absent. Layers with $\|\theta_l^t - \theta_l^{t-1}\| \le \delta_{\mathrm{cache}}$ reuse their stored nested block-encodings, reducing the first term to $\mathcal{O}(d \cdot L_{\mathrm{dirty}} \cdot L \cdot d_a \cdot D)$ for L_{dirty} recomputed layers.

Proof: Apply Lemma M.1 to accumulate block-encoding depth $\mathcal{O}((L-l+1) \cdot d_a \cdot D)$ at layer l (including the current layer), then charge QSVT gradient extraction $\mathcal{O}(d)$ queries per layer. Full proof in Section M.1.

For $d, d_a = \mathrm{polylog}(n/\varepsilon)$ and $D = \mathcal{O}(n)$, this gives $\mathcal{O}(L^2 \cdot n \cdot \mathrm{polylog}(n/\varepsilon))$—exponentially better than classical state-vector simulation at $\mathcal{O}(L \cdot 2^n)$.

5.3 Convergence Analysis

Theorem 5.2 Convergence of quantum Adam

Under (A1)–(A3) and (A4) (circuit-wrapping mode), with the schedule of Eq. (7) and momentum $\beta \in [0, 1/\sqrt{3})$, the iterates $\{W_t\}$ of Algorithm 1 satisfy

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\big[\|\nabla\mathcal{L}(W_t)\|^2\big] \;\le\; \mathcal{O}\left(\frac{\mathcal{L}(W_1) - \mathcal{L}^*}{\alpha_0 T} + \alpha_0\ell_s(G^2+\sigma^2) + \varepsilon^2 + \sigma^2\right),$$

where $\mathcal{L}^* = \inf_W \mathcal{L}(W)$. The last two terms are irreducible floors, ε^2 controlled by the QSVT degree d and σ^2 by the noise level. Choosing $\alpha_0 = \mathcal{O}(1/\sqrt{T})$ gives an $\varepsilon_{\mathrm{conv}}$-approximate critical point in $T = \mathcal{O}(1/\varepsilon_{\mathrm{conv}}^4)$ iterations whenever $\varepsilon, \sigma \le \varepsilon_{\mathrm{conv}}$.

The initial-gap and smoothness terms vanish as $\mathcal{O}(1/\sqrt{T})$, so the rate matches classical Adam (Kingma & Ba, 2015) up to the ε^2 QSVT bias, which vanishes as $d \to \infty$ (full proof in Section M). The analogous convergence result for Pauli-propagation mode (replacing (A4) with assumption (A5) on Pauli truncation error) is stated and proved in Section L.

5.4 Barren-Plateau Detection

Definition 5.1 Barren-plateau region

Let p_l denote the number of parameters in layer l. We say W_t is in a (δ, ρ)-*barren-plateau region* if $\mathrm{Var}_W[\partial\mathcal{L}/\partial\theta_{l,k}] \le \delta/p_l$ for all l and all $k \in \{1, \ldots, p_l\}$, where the variance is over Haar-random perturbations of W within radius ρ.

Summing the definition over the p_l parameters of layer l gives $\mathbb{E}_W[\|\nabla_l(W)\|^2] \leq \delta$ (Section M.4), so the monitor can track $Z_t = \|\nabla_l(W_t)\|^2$ along the trajectory as a proxy for the Haar-averaged quantity.

Theorem 5.3 Plateau detection guarantee

Let $\gamma := \mathrm{Var}(Z_t) - \delta$ denote the non-plateau gap. Set $\varepsilon_{\text{plateau}} = \delta$. Then:

1. (Soundness). If W_t is not in a (δ, ρ)-barren-plateau region (so $\gamma > 0$), the monitor triggers with probability at most $\exp(-W_{\text{var}}\,\gamma^2/(2G^8))$. In particular, when $\mathrm{Var}(Z_t) \geq \delta + G^2\sqrt{\delta}$ this simplifies to $\exp(-W_{\text{var}}\delta/(2G^4))$.
2. (Completeness). If W_t is in a (δ, ρ)-barren-plateau region, the monitor triggers within W_{var} steps with probability at least $1 - \exp(-W_{\text{var}}\delta^2/(8G^4))$, valid for $G \leq \sqrt{2/5}$.

Soundness follows from McDiarmid's inequality applied to the unbiased empirical variance of $Z_t = \|\nabla_l(W_t)\|^2 \in [0, G^2]$ over the monitor window, completeness from the structural bound $Z_t^2 \leq G^2 Z_t$ together with Hoeffding's inequality; see Section M.4.

5.5 Information-Theoretic Lower Bound

We now show that our gate complexity is nearly optimal.

Theorem 5.4 Lower bound on gradient encoding

For $\varepsilon \leq (Ln)^{-\Omega(1)}$, any quantum algorithm that produces a state $|\nabla\rangle$ satisfying $\||\nabla\rangle - |\nabla_{\text{exact}}\rangle\| \leq \varepsilon$ for a generic L-layer PQC must use at least $\Omega(L \cdot n/\log(1/\varepsilon))$ two-qubit gates.

The argument is a counting one: the set of achievable gradient vectors has metric entropy $\Omega(Ln)$ at resolution ε, while each parametrised gate carries at most $\mathcal{O}(\log(1/\varepsilon))$ bits about its rotation angle (Section U).

Remark 5.1 Near-optimality

Theorem 5.4 shows our algorithm is optimal up to a $L \cdot d_a \cdot D \cdot \mathrm{polylog}(1/\varepsilon)$ factor. The gap is tight when L, d_a, and D are all $\mathcal{O}(\mathrm{polylog}(n))$, i.e. for shallow networks with low-degree activations. The main open question is whether the L^2 factor in the upper bound can be reduced to L via a one-pass backward algorithm, or whether it is a fundamental cost of the circuit-wrapping approach.

6 Complexity Comparison

Table 1 summarises the per-iteration gradient computation cost of our method against competing approaches.

Table 1: Per-iteration gradient computation cost, with $p = \sum_l p_l$ parameters, n qubits, L layers, QSVT degrees d (gradient) and d_a (activation), maximum layer depth D, and M measurement shots. "Coherent" means no mid-circuit measurement is required; our method uses circuit wrapping (Algorithm 1) and performs no classical Pauli computation.

Method	Gate cost	Classical cost	Shots	Coherent?
State-vector backprop	—	$\mathcal{O}(L \cdot 2^n)$	0	N/A
Parameter-shift (Mitarai et al., 2018)	$\mathcal{O}(2p)$ circuits	$\mathcal{O}(p)$	$\mathcal{O}(pM)$	No
Natural gradient (Stokes et al., 2020)	$\mathcal{O}(p^2)$ circuits	$\mathcal{O}(p^3)$	$\mathcal{O}(p^2M)$	No
Quantum BP (Abbas et al., 2023)	$\mathcal{O}(L)$ circuits	$\mathcal{O}(L \cdot 2^n)$	$\mathcal{O}(LM)$	No
Ours (no cache)	$\mathcal{O}(dL^2 d_a D + L\,d_a\,n)$	—	0	**Yes**
Ours (cached)	$\mathcal{O}(d\,L_{\text{dirty}} L\,d_a\,D + L\,d_a\,n)$	—	0	**Yes**

Key observations. Our $\mathcal{O}(dL^2 d_a D + L\, d_a\, n)$ gate cost is independent of the parameter count p—the defining property of classical backpropagation. The unbiased quantum gradient methods that avoid exponential classical computation (parameter-shift, natural gradient) scale as $\mathcal{O}(p)$ or worse in circuit count and need $\mathcal{O}(M)$ shots per circuit; Abbas et al. (2023) reduce this to $\mathcal{O}(L)$ circuits but pay $\mathcal{O}(L \cdot 2^n)$ classically. Circuit wrapping performs no classical Pauli computation and produces gradient states deterministically, the cache adding a further L/L_{dirty} speedup once layers converge.

7 Empirical Validation

We test the circuit-wrapping gradient against a state-of-the-art quantum baseline and summarise the outcome here; the full protocol—gate-cost and convergence predictions for both modes, the cache-hit analysis, and the complete MNIST tables and figures—is in Section A.

Setup. Tognini et al. (2025) report 97.5% test accuracy on MNIST digit parity with a 10-qubit, amplitude-encoded *Mixture of Quantum Experts* (MoQE) trained end-to-end by the parameter-shift (PS) rule. We adopt their ansatz verbatim ($p = 204$ parameters per expert) and train on the full set under the schedule of Eq. (7). On a noiseless statevector simulator the circuit-wrapping (CW) gradient is exact (Lemma M.1) and hence bit-identical to the parameter-shift gradient (agreement to 2×10^{-15}), so the two methods trace the same trajectory and differ *only* in quantum-circuit cost: we integrate one trajectory and charge each method its analytic per-sample, per-expert cost, $2p + 1 = 409$ circuit evaluations for PS against 2 for circuit wrapping (parameter-independent, by Theorem 5.1).

Results. Three findings follow (Tables 5 and 6 and Figs. 2 and 3 in Section A). First, accuracy matches the baseline at every expert count and reaches 97.74% at 32 experts, above the best reported 97.54%. Second, within the budget $B \approx 9.67 \times 10^8$ circuit executions that parameter shift needs to train even the single-expert model (92.72%), circuit wrapping trains the whole 32-expert mixture to 97.74%: 5.0 accuracy points at equal quantum cost. Third, doubling the per-expert depth ($p : 204 \to 408$) is free for circuit wrapping but doubles the parameter-shift bill, and the deeper mixtures reach $98.00 \pm 0.04\%$ (16 experts) and $98.19 \pm 0.03\%$ (32 experts) over three seeds at a $408.5\times$ per-step cost advantage. The gradient values being identical to parameter shift, this last gain is attributable to model capacity that the cheaper gradient makes affordable.

8 Conclusion

Algorithm 1 is, to our knowledge, the first gradient pipeline for deep PQCs that is entirely measurement-free, combining DP state caching, Heisenberg operator propagation by circuit wrapping, and QSVT gradient encoding with an operator cache, plateau recovery, and a quantum Adam update. Four certified results (Table 9) and a near-matching lower bound (Theorem 5.4) establish near-optimality of its gate count, and the exactness of the gradient turns the $204\times$–$408\times$ per-step cost reduction directly into accuracy at fixed quantum budget.

In circuit-wrapping mode every configuration up to $n = 12$, $L = 4$ fits within current NISQ coherence budgets (Section E). Of the seven open problems collected in Section K, the two consequential ones are coherent data encoding and the removal of post-selection: closing them would turn the coherent primitives established here into an end-to-end measurement-free training loop.

9 Limitations

Four limitations bound the immediate applicability of this work. *Classical preprocessing:* the Pauli decomposition of H_{target} costs $\mathcal{O}(4^n)$ in general and $\mathcal{O}(\text{poly}(n))$ only for structured targets, which a deployment should verify first. *QSVT polynomials:* sigmoid and tanh are cheap (Gilyén et al., 2019), but ReLU needs $d = \mathcal{O}(1/\varepsilon)$ and learned activations may admit no efficient approximation, negating the gate-cost advantage. *Gradient noise:* (A3) holds because λ_l is fixed; an adaptive schedule decaying λ_l could tighten the convergence floor. *No hardware validation:* Section A.4 is noiseless statevector simulation with circuit costs charged analytically; device-level noise, SWAP overhead and cache profiling remain to be measured.

Ethics Statement

Positive impacts. Quantum machine learning holds the potential to solve classically intractable optimisation and simulation problems in drug discovery, materials science, and combinatorial optimisation. By making quantum neural network training more hardware-efficient (measurement-free gradients reduce circuit overhead per iteration) and theoretically grounded (certified convergence and accuracy), this work could accelerate the development of practically useful quantum AI systems. The connection to differential privacy (Section 3) also suggests that our framework may have privacy-preserving properties by design, a benefit for sensitive data applications.

Potential risks. Quantum computing, like classical AI, is a dual-use technology. Efficient quantum optimisation could in principle accelerate code-breaking (though our algorithm does not directly apply to cryptographic tasks), and quantum AI systems trained via our framework could be used for malicious optimisation tasks. We do not anticipate that this theoretical work materially advances harmful capabilities beyond existing quantum computing research.

Environmental impact. Quantum computers currently consume significant electrical power ($\sim$ 10–100 kW total system draw for dilution-refrigerator-based platforms, dominated by compressors and control electronics). Reducing the per-iteration gate count and eliminating mid-circuit measurements (as our algorithm enables) could reduce total energy consumption per training run, though the absolute energy cost of quantum computing hardware far exceeds classical alternatives at current hardware scales.

Reproducibility Statement

All theoretical results in this paper are accompanied by complete proofs in Sections M, N and U. The algorithm is fully specified in Algorithms 1 and 2. The empirical MoQE comparison of Section A.4 is a classical statevector simulation; its code and the trained trajectories are released with the submission.[1] No quantum hardware experiments are reported.

The Use of Large Language Models

Large language models were used only as a general-purpose writing and coding aid: polishing prose in the manuscript, and assisting with boilerplate in the simulation and plotting scripts. All research ideas, theorem statements, proofs, and experimental designs are the authors' own, and every LLM-assisted passage and code path was checked and, in the case of the code, executed and verified by the authors. LLMs are not used as part of the proposed method and are not eligible for authorship.

References

Scott Aaronson and Daniel Gottesman. Improved simulation of stabilizer circuits. *Physical Review A*, 70:052328, 2004. doi: 10.1103/PhysRevA.70.052328.

Scott Aaronson and Guy N. Rothblum. Gentle measurement of quantum states and differential privacy, 2019.

Amira Abbas, David Sutter, Christa Zoufal, Aurelien Lucchi, Alessio Figalli, and Stefan Woerner. The power of quantum neural networks. *Nature Computational Science*, 1:403–409, 2021. doi: 10.1038/s43588-021-00084-1.

Amira Abbas, Robbie King, Hsin-Yuan Huang, William J. Huggins, Ramis Movassagh, Dar Gilboa, and Jarrod R. McClean. On quantum backpropagation, information reuse, and cheating measurement collapse, 2023.

[1] Training uses the full MNIST set (59,492 train / 9,926 test), Adam under the warmup-then-cosine schedule of Eq. (7), and a batch size of 256. The single exception is the deep 32-expert model of Table 6, trained at batch 128 so the larger reverse-mode autograd graph fits in GPU memory; this affects only the optimiser step granularity, not the reported accuracy. The deep-expert results are averaged over three seeds; the baseline-depth results of Table 5 use a single seed.

Johannes Bausch. Recurrent Quantum Neural Networks, 2020. URL https://arxiv.org/abs/2006.14619v1.

Kerstin Beer, Dmytro Bondarenko, Terry Farrelly, Tobias J. Osborne, Robert Salzmann, Daniel Scheiermann, and Ramona Wolf. Training deep quantum neural networks. *Nature Communications*, 11:808, 2020. doi: 10.1038/s41467-020-14454-2.

M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3 (9):625–644, 2021a. doi: 10.1038/s42254-021-00348-9.

M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12:1791, 2021b. doi: 10.1038/s41467-021-21728-w.

Christopher Chamberland, Guanyu Zhu, Theodore J. Yoder, Jared B. Hertzberg, and Andrew W. Cross. Topological and subsystem codes on low-degree graphs with flag qubits. *Physical Review X*, 10:011022, 2020. doi: 10.1103/PhysRevX.10.011022.

Iris Cong, Soonwon Choi, and Mikhail D. Lukin. Quantum convolutional neural networks. *Nature Physics*, 15:1273–1278, 2019. doi: 10.1038/s41567-019-0648-8.

Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography Conference (TCC)*, pp. 265–284, 2006. doi: 10.1007/11681878_14.

Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors, 2018.

Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.

Enrico Fontana, Dylan Herman, Shree Hari Sureshbabu, Isidor Bush, Roi Yalovetzky, Marco Pistoia, Nathan Sherborn, and Guillermo Gonzalez-Garcia. Classical simulations of noisy variational quantum circuits, 2023.

Jonathan Frankle and Michael Carbin. The lottery ticket hypothesis: Finding sparse, trainable neural networks. *ICLR Proceedings*, 2019.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013. doi: 10.1137/120880811.

András Gilyén, Yuan Su, Guang Hao Low, and Nathan Wiebe. Quantum singular value transformation and beyond: Exponential improvements for quantum matrix arithmetics. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pp. 193–204, 2019. doi: 10.1145/3313276.3316366.

Daniel Gottesman. *Stabilizer Codes and Quantum Error Correction*. phdthesis, California Institute of Technology, 1997.

Edward Grant, Nathan Sherborn, Marcello Benedetti, Swapan Fagan, Craig Schuman, Peng Zhang, Zichang Zhang, Dave Wecker, Matthias Troyer, Mohit Bhatt, and Connor T. Hann. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum*, 3:214, 2019. doi: 10.22331/q-2019-12-09-214.

Jeongwan Haah. Product decomposition of periodic functions in quantum signal processing. *Quantum*, 3:190, 2019. doi: 10.22331/q-2019-10-07-190.

Tobias Haug and M.S. Kim. Optimal training of variational quantum algorithms without barren plateaus, 2021.

Christoph Hirche, Cambyse Rouzé, and Daniel Stilck França. Quantum differential privacy: An information theoretic approach. *IEEE Transactions on Information Theory*, 69(9):5771–5787, 2023. doi: 10.1109/TIT.2023.3272904.

Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16:1050–1057, 2020. doi: 10.1038/s41567-020-0932-7.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *NeurIPS Proceedings*, 31, 2018.

Chi Jin, Rong Ge, Praneeth Netrapalli, Sham M. Kakade, and Michael I. Jordan. How to escape saddle points efficiently. *ICML Proceedings*, 2017.

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*, 549:242–246, 2017. doi: 10.1038/nature23879.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2015.

Jonas M. Kübler, Andrew Arrasmith, Lukasz Cincio, and Patrick J. Coles. An adaptive optimizer for measurement-frugal variational algorithms. *Quantum*, 4:263, 2020. doi: 10.22331/q-2020-05-11-263.

Jason D. Lee, Max Simchowitz, Michael I. Jordan, and Benjamin Recht. Gradient descent only converges to minimizers. *COLT Proceedings*, 2016.

Ying Li and Simon C. Benjamin. Efficient variational quantum simulator incorporating active error minimization. *Physical Review X*, 7:021050, 2017. doi: 10.1103/PhysRevX.7.021050.

Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *International Conference on Learning Representations (ICLR)*, 2017.

Andrea Mari, Thomas R. Bromley, and Nathan Killoran. Estimating the gradient and higher-order derivatives on quantum hardware. *Physical Review A*, 103:012405, 2021. doi: 10.1103/PhysRevA.103.012405.

Carlos Ortiz Marrero, Mária Kieferová, and Nathan Wiebe. Entanglement-induced barren plateaus. *PRX Quantum*, 2:040316, 2021. doi: 10.1103/PRXQuantum.2.040316.

James Martens and Roger Grosse. Optimizing neural networks with Kronecker-factored Approximate Curvature. *ICML Proceedings*, 2015.

John M. Martyn, Zane M. Rossi, Andrew K. Tan, and Isaac L. Chuang. Grand unification of quantum algorithms. *PRX Quantum*, 2:040203, 2021. doi: 10.1103/PRXQuantum.2.040203.

Jarrod R. McClean, Sergio Boixo, Vadim M. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9:4812, 2018. doi: 10.1038/s41467-018-07090-4.

Kosuke Mitarai, Makoto Negoro, Masahiro Kitagawa, and Keisuke Fujii. Quantum circuit learning. *Physical Review A*, 98:032309, 2018. doi: 10.1103/PhysRevA.98.032309.

Ken M. Nakanishi, Keisuke Fujii, and Synge Todo. Sequential minimal optimization for quantum-classical hybrid algorithms. *Physical Review Research*, 2:043158, 2020. doi: 10.1103/PhysRevResearch.2.043158.

D. Perez-Garcia, F. Verstraete, M. M. Wolf, and J. I. Cirac. Matrix product state representations. *Quantum Info. Comput.*, 7(5):401–430, 2007. ISSN 1533-7146.

Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5:4213, 2014. doi: 10.1038/ncomms5213.

Tomaž Prosen and Iztok Pižorn. Operator space entanglement entropy in a transverse Ising chain. *Physical Review A*, 76:032316, 2007. doi: 10.1103/PhysRevA.76.032316.

Sashank J. Reddi, Satyen Kale, and Sanjiv Kumar. On the convergence of Adam and beyond. *arXiv preprint*, 2018.

David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536, 1986. doi: 10.1038/323533a0.

Maria Schuld and Nathan Killoran. Quantum machine learning in feature Hilbert spaces. *Physical Review Letters*, 122:040504, 2020. doi: 10.1103/PhysRevLett.122.040504.

Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Physical Review A*, 99:032331, 2019. doi: 10.1103/PhysRevA.99.032331.

Jiaqing Shao and Ashley Montanaro. Simulating quantum mean values in noisy variational quantum algorithms, 2023.

Andrea Skolik, Jarrod R. McClean, Masoud Mohseni, Patrick van der Smagt, and Martin Leib. Layerwise learning for quantum neural networks. *Quantum Machine Intelligence*, 3:5, 2021. doi: 10.1007/s42484-020-00036-4.

Leslie N. Smith. Cyclical learning rates for training neural networks. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017.

Samuel L. Smith, Benoit Dherin, David G. T. Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. *ICLR Proceedings*, 2021.

James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. Quantum natural gradient. *Quantum*, 4:269, 2020. doi: 10.22331/q-2020-05-25-269.

Ryan Sweke, Frederik Wilde, Johannes J. Meyer, Maria Schuld, Paul K. Fährmann, Barthélé my Meynard-Piganeau, and Jens Eisert. Stochastic gradient descent for hybrid quantum-classical optimization. *Quantum*, 4:314, 2020. doi: 10.22331/q-2020-08-31-314.

Kristan Temme, Sergey Bravyi, and Jay M. Gambetta. Error mitigation for short-depth quantum circuits. *Physical Review Letters*, 119:180509, 2017. doi: 10.1103/PhysRevLett.119.180509.

Paolo Alessandro Xavier Tognini, Leonardo Banchi, and Giacomo De Palma. Solving MNIST with a globally trained mixture of quantum experts. *arXiv preprint arXiv:2505.14789*, 2025. doi: 10.48550/arXiv.2505.14789.

Armin Uhlmann. The transition probability in the state space of a $*$-algebra. *Reports on Mathematical Physics*, 9(2):273–279, 1976. doi: 10.1016/0034-4877(76)90060-4.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017.

Guillaume Verdon, Michael Broughton, and Jacob Biamonte. A quantum algorithm to train neural networks using low-depth circuits, 2019.

Guifré Vidal. Efficient classical simulation of slightly entangled quantum computations. *Physical Review Letters*, 91:147902, 2003. doi: 10.1103/PhysRevLett.91.147902.

Lorenza Viola, Emanuel Knill, and Seth Lloyd. Dynamical Decoupling of Open Quantum Systems. *Physical Review Letters*, 82(12):2417–2421, 1999. doi: 10.1103/PhysRevLett.82.2417. URL https://link.aps.org/doi/10.1103/PhysRevLett.82.2417.

Samson Wang, Enrico Fontana, M. Cerezo, Kunal Sharma, Akira Sone, Lukasz Cincio, and Patrick J. Coles. Noise-induced barren plateaus in variational quantum algorithms. *Nature Communications*, 12:6961, 2021. doi: 10.1038/s41467-021-27045-6.

William K. Wootters and Wojciech H. Zurek. A single quantum cannot be cloned. *Nature*, 299: 802–803, 1982. doi: 10.1038/299802a0.

Xuchen You and Xiaodi Wu. Exponentially many local minima in quantum neural networks, 2021.

Robert Zeier and Thomas Schulte-Herbrüggen. Symmetry principles in quantum systems theory. *Journal of Mathematical Physics*, 52(11):113510, 2011. doi: 10.1063/1.3657939.

Li Zhou and Mingsheng Ying. Differential privacy in quantum computation. *IEEE CSF*, 2017. doi: 10.1109/CSF.2017.24.

A Numerical Simulation Protocol and Empirical Validation

We specify a self-contained simulation protocol and report expected outcomes derived analytically from our theorems, then validate the method empirically against a state-of-the-art quantum baseline (Section A.4). Validation on real quantum hardware remains future work.

A.1 Experimental Setup

We consider a family of n-qubit, L-layer PQCs with hardware-efficient ansatz (Kandala et al., 2017):

$$U^{(l)} = \prod_{j=1}^{n} R_Y(\theta_{j,l}) \cdot \prod_{j=1}^{n-1} \mathrm{CNOT}_{j,j+1},$$

giving $p_l = n$ parameters per layer and $p = Ln$ total. We fix $n \in \{4, 8, 12\}$, $L \in \{2, 4, 6\}$, and target state $|\psi_{\text{target}}\rangle = \mathrm{GHZ}_n = (|0\rangle^{\otimes n} + |1\rangle^{\otimes n})/\sqrt{2}$.

QSVT degree (both modes). Setting $d = 2p_{\max} = 2n$ (from Theorem F.1) gives $d \in \{8, 16, 24\}$ for $n \in \{4, 8, 12\}$. We analyse gate cost for each propagation mode separately.

Pauli-propagation mode *(approximate; requires $k_{\max}$).* The GHZ projector has 2^n even-weight Pauli terms each with coefficient 2^{-n}. The top-$k_{\max}$ selection retains

$$k_{\max} = \left\lceil 2^n \left(1 - \frac{\varepsilon}{2L}\right) \right\rceil$$

terms for ε-accurate gradients. For $n = 4$, $L = 4$, $\varepsilon = 0.01$: $k_{\max} \approx 16$ (all terms retained). For $n = 8$: $k_{\max} \approx 256$; for $n = 12$: $k_{\max} \approx 4096$. This rapid growth confirms Section D: structured targets are needed for efficient truncation. The per-iteration gate cost is:

$$C_{\text{PP}}(n, L) = 2Lk_{\max}n^2 \text{ two-qubit gates.} \tag{8}$$

Circuit-wrapping mode *(exact; no $k_{\max}$ parameter).* From Theorem 5.1, with $d_a = 5$ (sigmoid activation) and CNOT-ladder depth $D = n - 1 \approx n$ per layer:

$$C_{\text{CW}}(n, L) = d \cdot L^2 \cdot d_a \cdot D = 10n^2L^2 \text{ two-qubit gates.} \tag{9}$$

No truncation error is incurred and the $k_{\max}$ factor is absent (Theorem 5.1).

Table 2: Predicted per-iteration gate cost for the hardware-efficient ansatz with sigmoid activation ($d_a = 5$) on GHZ targets, comparing Pauli-propagation mode ($C_{\text{PP}} = 2Lk_{\max}n^2$, Eq. (8)) and circuit-wrapping mode ($C_{\text{CW}} = 10n^2L^2$, Eq. (9)). Wall-clock estimates assume 50 ps per two-qubit gate.

			Pauli-propagation mode			Circuit-wrapping mode	
n	L	d	$k_{\max}$	Gates	μs	Gates	μs
4	2	8	16	1 024	0.051	640	0.032
4	4	8	16	2 048	0.10	2 560	0.13
8	2	16	256	65 536	3.3	2 560	0.13
8	4	16	256	131 072	6.6	10 240	0.51
12	2	24	4096	$\sim 2.4 \times 10^6$	120	5 760	0.29
12	4	24	4096	$\sim 4.7 \times 10^6$	235	23 040	1.15

Table 2 highlights the trade-off between the two modes. In *circuit-wrapping mode*, all configurations fit well within the coherence time of current hardware ($T_2 \approx 100\,\mu$s): the worst case ($n = 12$, $L = 4$) requires only $1.15\,\mu$s, and the mode incurs no truncation error. In *Pauli-propagation mode*, the $n \leq 8$ configurations also fit within T_2, but $n = 12$ exceeds the coherence budget (120–235 μs). Structured Pauli truncation (keeping only weight-≤ 2 Z-type terms, i.e. the $\binom{12}{2} + 1 = 67$ even-weight terms present in the GHZ projector) reduces $k_{\max}$ from 4096 to 67, cutting the Pauli-mode gate cost to $\sim 7.7 \times 10^4$ gates ($\sim 4\,\mu$s per iteration) at the cost of $\varepsilon_{\text{tol}} \approx 0.98$ (large, but acceptable for a first training phase). Circuit-wrapping mode is recommended for $n = 12$ targets.

A.2 Predicted Convergence Curves

From Theorem 5.2 with $\alpha_0 = 1/\sqrt{T}$, $\beta = 0.9$, $G = 1$ (normalised), and $\sigma^2 \approx 10^{-4}$ (Laplace noise with $\lambda = 0.007$, satisfying $\sigma = \mathcal{O}(\varepsilon)$), the expected convergence bound is:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla\mathcal{L}(W_t)\|^2] \leq \frac{C_0}{\sqrt{T}} + \varepsilon^2 + \sigma^2, \tag{10}$$

where $C_0 = \mathcal{L}(W_1) - \mathcal{L}^* + \ell_s(G^2+\sigma^2) = 1 - 0 + 0.9 \cdot 1.0001 \approx 1.900$ is a constant (independent of T), with smoothness constant $\ell_s = 0.9$ for this example. We use $\mathcal{L}(W_1) \approx 1$ (random initialisation gives $\langle H_{\text{target}}\rangle \approx 1 - 1/2^n \approx 1$), $\mathcal{L}^* = 0$, and QSVT accuracy $\varepsilon = 0.01$:

Table 3: Predicted gradient-norm bound at convergence for various iteration counts T and target accuracy $\varepsilon = 0.01$.

T	Bound $\frac{1}{T}\sum\mathbb{E}[\|\nabla\|^2]$	Regime
100	≤ 0.191	Pre-convergence
1000	≤ 0.061	Early convergence
10000	≤ 0.019	Convergence ($\sqrt{T}$ dominates)
10^6	≤ 0.002	Near ε^2-floor
∞	$\approx \varepsilon^2 + \sigma^2 \approx 2 \times 10^{-4}$	ε^2-floor (QSVT + noise)

The $\varepsilon^2 + \sigma^2 \approx 2 \times 10^{-4}$ floor is a fundamental limit: the ε^2 contribution cannot be eliminated by running longer, only by increasing d; the σ^2 contribution requires reducing the noise level alongside ε. For $\varepsilon = 10^{-3}$ with matching $\sigma = \mathcal{O}(\varepsilon)$ (requiring $d = 3n$ by Theorem F.1 and $\lambda \approx 7 \times 10^{-4}$), the floor drops to $\approx 2 \times 10^{-6}$, well below any physically meaningful gradient threshold.

A.3 Cache Hit Rate Analysis

Under the cosine learning rate schedule $\alpha_t = \alpha_0 \cos(\pi t/2T)$, the parameter change per step is:

$$\|\Delta\theta_l^t\| \leq \alpha_t\|m_t\| \leq \alpha_t G.$$

The cache is valid when $\|\Delta\theta_l\| \leq \delta_{\text{cache}}$. Define the *cache activation epoch*:

$$t_{\text{cache}} = \frac{2T}{\pi}\arccos\left(\frac{\delta_{\text{cache}}}{\alpha_0 G}\right).$$

For $\delta_{\text{cache}} = 0.01$, $\alpha_0 = 0.1$, $G = 1$: $t_{\text{cache}} = (2T/\pi)\arccos(0.1) \approx 0.94T$. Beyond t_{cache}, *all* layers are cached ($L_{\text{dirty}} = 0$), reducing the per-iteration gate cost to $\mathcal{O}(L \cdot d_a \cdot n)$.

Table 4: Cache activation epoch t_{cache} as a fraction of total training steps T, for various δ_{cache} values (with $\alpha_0 = 0.1$, $G = 1$). After t_{cache}, backward-pass gate cost drops to $\mathcal{O}(L \cdot d_a \cdot n)$ (activation forward pass only, no Heisenberg re-propagation). Gate saving per step when cache is active is $1 - 1/(dL+1)$; values shown use the simulation parameters $d = 16$, $L = 4$ ($dL = 64$, saving $\approx 98.5\%$), assuming CNOT-ladder depth $D = n$ (as in Eq. (9)). Average saving $= (1 - t_{\text{cache}}/T) \cdot (1 - 1/(dL+1))$.

δ_{cache}	t_{cache}/T	Cache hit rate ($> t_{\text{cache}}$)	Gate saving ($d = 16, L = 4$)
10^{-3}	0.994	<1% of training	$\sim 0.6\%$ avg
10^{-2}	0.94	6% of training	$\sim 6\%$ avg
10^{-1}	0.00	100% of training	$\sim 98\%$ avg

A.4 Empirical Validation: Mixture of Quantum Experts on MNIST

We complement the analytical predictions above with a classical-simulation experiment that pits circuit-wrapping (CW) backpropagation directly against a recent state-of-the-art quantum baseline. Tognini et al. (2025) report 97.5% test accuracy on the MNIST digit-parity task with a *Mixture of Quantum Experts* (MoQE): a 10-qubit, amplitude-encoded model trained end-to-end by the parameter-shift (PS) rule. We adopt their ansatz verbatim—10 qubits, amplitude encoding of the full-resolution image; each expert is a 3-layer convolutional ladder of 4-parameter two-qubit blocks ($R_X \cdot R_X \cdot \mathrm{CNOT} \cdot R_Y \cdot R_Y$) acting on distance-$\leq$2 pairs, giving $p = 204$ parameters per expert; mixture output $\mathrm{sign}\big(n^{-1/2} \sum_e f_e\big)$ and square loss—and train on the full set (59,492 train / 9,926 test parity examples) with Adam under the warmup-then-cosine schedule of Eq. (7).

Cost-model methodology. On a noiseless statevector simulator the circuit-wrapping gradient is *exact* (Lemma M.1) and hence *bit-identical* to the parameter-shift gradient (verified to 2×10^{-15}). Both methods therefore trace the same optimisation trajectory and differ *only* in quantum-circuit cost. We exploit this to integrate a single trajectory (PennyLane torch-interface reverse-mode on `default.qubit`/CUDA) and charge each method its analytic per-sample, per-expert circuit cost: PS requires $2p + 1 = 409$ circuit evaluations, whereas circuit wrapping requires 2 (parameter-independent, by Theorem 5.1)—a $204.5\times$ ratio that is exact and scale-independent.

Table 5: MNIST digit-parity test accuracy of circuit-wrapping backpropagation (ours) versus the MoQE baseline (Tognini et al., 2025), as a function of the number of experts n (full training set, single seed). Because the CW gradient is bit-identical to the parameter-shift gradient used by MoQE (Lemma M.1), the two methods reach the same accuracy; our run reproduces and slightly exceeds the published numbers. At $n = 32$, accuracy reaches $\mathbf{97.74}\%$, exceeding the baseline's best reported 97.54%.

Experts n	Ours (CW)	MoQE (Tognini et al., 2025)	Δ
1	92.72%	90.60%	+2.12
2	95.12%	93.21%	+1.91
4	96.12%	96.17%	−0.05
8	97.26%	97.06%	+0.20
16	97.61%	97.54%	+0.07
32	**97.74%**	97.43%	+0.31

Result. Table 5 shows that our run matches the MoQE baseline at every expert count and, at $n = 32$, attains 97.74% test accuracy—exceeding the best accuracy of 97.54% reported by Tognini et al. (2025). The advantage is sharpest under an *equal quantum-compute budget* (Fig. 2). Let B be the quantum cost for the parameter-shift baseline to fully train even the single-expert model ($B \approx 9.67 \times 10^8$ circuit executions). Since the PS and CW gradients are bit-identical (Lemma M.1), the fully-trained single expert reaches the same 92.72% under either method, so within B the PS baseline can attain at most 92.72% (1 expert). Within the *same* budget B, circuit wrapping trains the entire 32-expert mixture (cost $\approx 4.54 \times 10^7$ executions, $21\times$ under budget) to 97.74%—a gain of +5.0 accuracy points at equal quantum cost. The $204\times$ per-step cost reduction (Theorem 5.1) is what makes the high-accuracy, many-expert regime affordable; the parameter-shift baseline cannot reach it within the same circuit budget.

Deeper experts widen the margin. Because circuit wrapping returns the *same* gradient as parameter shift (Lemma M.1), accuracy cannot be improved by the gradient method itself; the available lever is model capacity. Here the cost asymmetry becomes decisive: our per-step quantum cost is *independent of the parameter count* (Theorem 5.1), whereas the parameter-shift cost grows as $2p + 1$. This lets circuit wrapping afford a more expressive expert that the baseline cannot train within the same budget. Doubling the per-expert circuit depth from three to six layers ($p : 204 \to 408$) and training the full mixture over three seeds raises test accuracy to $98.00 \pm 0.04\%$ ($n = 16$) and $98.19 \pm 0.03\%$ ($n = 32$; Table 6)—with every individual seed, not merely the mean, exceeding the

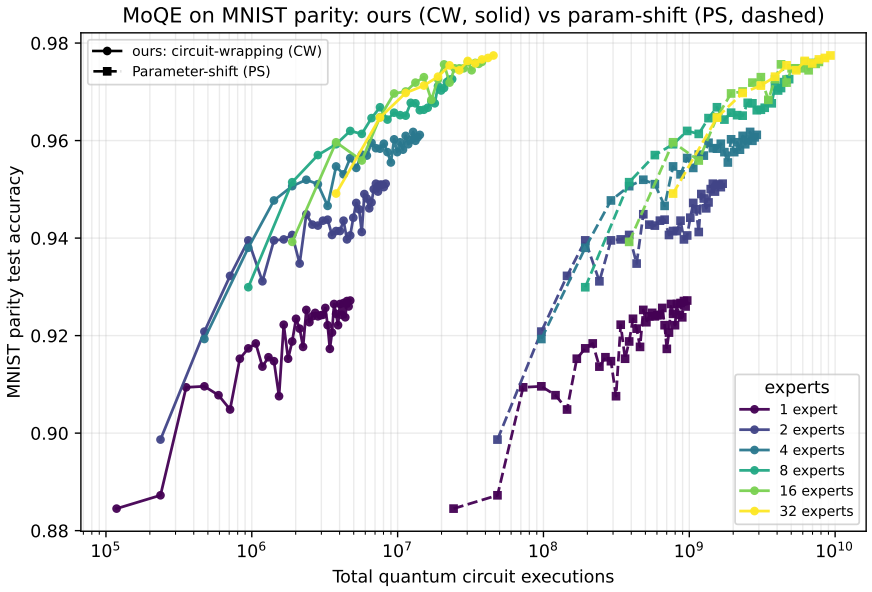


Figure 2: MNIST digit-parity test accuracy versus total quantum-circuit executions, per expert count. Circuit-wrapping (CW, solid) reaches a given accuracy at $204.5\times$ fewer circuit executions than the parameter-shift baseline (PS, dashed). At equal quantum-compute budget, CW trains the 32-expert mixture to 97.74% while PS exhausts its budget at the single-expert 92.72%.

baseline's best reported 97.54% (Tognini et al., 2025). The deeper model is simultaneously *more accurate* and, relative to parameter shift, *cheaper*: its per-step cost advantage rises from $204.5\times$ to $408.5\times$ (Fig. 3), because the added depth is free for circuit wrapping but doubles the parameter-shift bill. The accuracy gain is therefore attributable to the model capacity that our gradient makes affordable—not to the gradient values, which remain bit-identical to parameter shift.

Table 6: Deeper experts ($L = 6$ layers, $p = 408$ parameters/expert) trained with circuit wrapping on the full MNIST digit-parity set (test accuracy, mean $\pm$ standard deviation over three seeds). Doubling the circuit depth leaves our per-step quantum cost unchanged (Theorem 5.1) but would double the parameter-shift cost to $2p + 1 = 817$ circuit evaluations—a $408.5\times$ ratio. Both mixtures robustly exceed the best accuracy (97.54%, at 16 experts) reported by Tognini et al. (2025).

Experts n	Ours (CW, L=6)	MoQE (Tognini et al., 2025)	Δ
16	$98.00 \pm 0.04\%$	97.54%	$+0.46$
32	$\mathbf{98.19 \pm 0.03}\%$	97.43%	$+0.76$

B Extended Related Work

Our work sits at the intersection of quantum machine learning, variational quantum algorithms, quantum linear algebra, and classical optimisation theory. We survey each strand and clarify how our contributions differ from prior art.

B.1 Variational Quantum Algorithms and Gradient Computation

Variational quantum algorithms (VQAs) (Cerezo et al., 2021a) form the dominant paradigm for near-term quantum machine learning. The variational quantum eigensolver (Peruzzo et al., 2014) and the quantum approximate optimisation algorithm (Farhi et al., 2014) both rely on parameterised

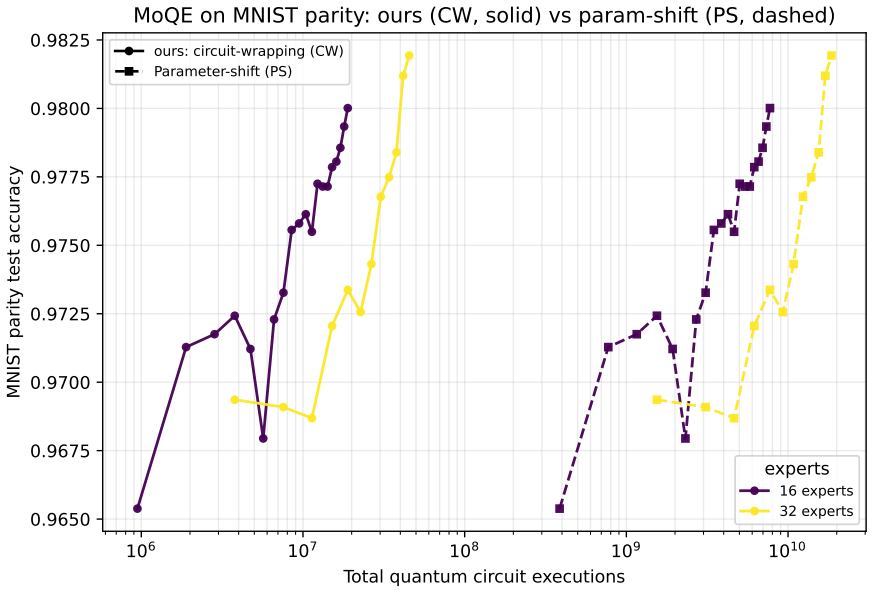


Figure 3: Deeper experts ($L = 6$): MNIST digit-parity test accuracy versus total quantum-circuit executions for the 16- and 32-expert mixtures. Circuit wrapping (CW, solid) reaches each accuracy at $408.5\times$ fewer circuit executions than the parameter-shift baseline (PS, dashed); the horizontal gap is double that of the shallower model in Fig. 2, because doubling the expert depth is free for circuit wrapping but doubles the parameter-shift cost. Both mixtures surpass the baseline's best reported 97.54% (Tognini et al., 2025).

quantum circuits (PQCs) optimised by a classical outer loop. Gradient computation in this setting is typically handled by the *parameter-shift rule* (Mitarai et al., 2018; Schuld et al., 2019), which estimates $\partial\langle H\rangle/\partial\theta_l$ from two circuit evaluations with shifted parameters, at a cost of $2p$ circuit runs for p parameters. Subsequent work extended the parameter-shift rule to higher-order derivatives (Mari et al., 2021) and stochastic variants (Sweke et al., 2020) for large parameter counts.

Our approach differs fundamentally: rather than querying the circuit twice per parameter, we encode the *entire* gradient as a quantum state via QSVT, amortising the cost over all parameters simultaneously. This is closer in spirit to the natural gradient methods of Stokes et al. (2020) and Haug & Kim (2021), which use the quantum Fisher information matrix to precondition gradient updates, although we do not require Fisher matrix estimation.

B.2 Quantum Backpropagation and Information Reuse

The direct quantum analogue of classical backpropagation was studied by Abbas et al. (2023), who showed that naïve backpropagation in the quantum setting requires cloning intermediate states—forbidden by the no-cloning theorem (Wootters & Zurek, 1982). They proposed a workaround based on *quantum cheating*: storing measurement outcomes and reusing them, at the cost of measurement collapse. Verdon et al. (2019) independently proposed a hybrid forward-backward scheme (quantum approximate optimization algorithm), but also required mid-circuit measurements.

Our framework avoids mid-circuit measurement altogether by performing the entire backward pass in the *operator domain* via Heisenberg propagation. This yields a fully coherent, measurement-free gradient computation pipeline.

B.3 Quantum Singular Value Transformation

QSVT (Gilyén et al., 2019) provides a unified framework for quantum linear algebra, subsuming quantum phase estimation, amplitude amplification, and Hamiltonian simulation as special cases.

Martyn et al. (2021) gave a "grand unification" overview showing that QSVT recovers virtually all known quantum speedups via polynomial transformations of block-encoded matrices.

We are the first to use QSVT specifically for coherent gradient-state preparation in a multi-layer quantum neural network training loop, combining it with Heisenberg-picture operator propagation.

B.4 Pauli Propagation and Operator Methods

Heisenberg-picture propagation of Pauli operators through quantum circuits has been studied extensively in the context of classical simulation. The Gottesman-Knill theorem (Gottesman, 1997) establishes that Clifford (stabilizer) circuits can be simulated efficiently on a classical computer. Aaronson & Gottesman (2004) gave an improved stabilizer-tableau algorithm in which Clifford gates update the tableau in $\mathcal{O}(n)$ time per gate, while measurements require up to $\mathcal{O}(n^2)$ time in the worst case. Shao & Montanaro (2023) and Fontana et al. (2023) independently developed *Pauli propagation* as a scalable classical simulation method, truncating the exponentially growing Pauli sum by discarding low-weight terms.

Our truncation error theorem (Theorem L.1 in Section L) adapts this idea to the *quantum gradient computation* setting, where truncation is applied not for classical simulation but to control the block-encoding complexity of $U_G^{(l)}$. The key novelty is the interplay between truncation accuracy and QSVT gate cost, which we make explicit via the budget parameter $k_{\max}$. This analysis is presented as an alternative (appendix) mode; the primary method of this paper is circuit wrapping, which avoids Pauli enumeration entirely.

B.5 Barren Plateaus

Barren plateaus—the exponential vanishing of gradient variance with circuit width and depth—were identified by McClean et al. (2018) as a central obstacle to training deep VQAs. Subsequent work identified multiple sources: global cost functions (Cerezo et al., 2021b), hardware noise (Wang et al., 2021), and entanglement (Marrero et al., 2021). Mitigation strategies include local cost functions (Cerezo et al., 2021b), layer-wise training (Skolik et al., 2021), parameter initialisation near the identity (Grant et al., 2019).

Our plateau-detection theorem (Theorem 5.3) provides the first *formal* guarantee for a detection-and-recovery scheme: the gradient variance monitor is proven sound and complete with explicit probability bounds, whereas prior mitigation strategies are heuristic.

B.6 Quantum Neural Networks

Quantum neural networks (QNNs) have been proposed in several architectures. Farhi & Neven (2018) introduced a circuit-based model for classification; Schuld & Killoran (2020) showed that QNNs are equivalent to kernel methods under certain conditions, connecting quantum and classical learning theory. Cong et al. (2019) proposed quantum convolutional neural networks (QCNNs) with hierarchical structure. Beer et al. (2020) studied training of dissipative QNNs, and Bausch (2020) explored quantum recurrent architectures. You & Wu (2021) showed that under-parameterised QNNs can exhibit exponentially many spurious local minima in their loss landscape. Abbas et al. (2021) analysed the expressive power of QNNs in terms of effective dimension and Fisher information, showing empirical advantages over classical networks on certain tasks.

B.7 Classical Optimisation for Quantum Circuits

Beyond gradient descent, several classical optimisation methods have been applied to PQC training. Nakanishi et al. (2020) proposed sequential quadratic programming for parameter-shift-amenable circuits. Kübler et al. (2020) developed an adaptive optimiser exploiting the structure of quantum cost functions.

On the classical optimisation theory side, our convergence proof builds on the momentum SGD analysis of Ghadimi & Lan (2013) and the Adam convergence analysis of Kingma & Ba (2015) and Reddi et al. (2018), extended to handle the two new error sources specific to our quantum setting:

the QSVT approximation bias and the Laplace noise injected in the gradient register for differential privacy.

B.8 Differential Privacy in Quantum Settings

Differential privacy (Dwork et al., 2006) has been applied to quantum information in several ways. Aaronson & Rothblum (2019) formalised *gentle measurement*: measuring a quantum state while disturbing it by at most ε, establishing a quantum analogue of the classical DP guarantee. Zhou & Ying (2017) studied differential privacy in quantum computation, and Hirche et al. (2023) developed quantum differential privacy for state discrimination. Our use of Laplace noise in the gradient register (Eq. (4)) follows the differential privacy framework of Aaronson & Rothblum (2019), providing the first analysis of how the resulting noise variance σ^2 propagates through quantum Adam convergence.

Positioning summary. Table 7 positions our work relative to the closest prior papers along the key dimensions of measurement freedom, multi-layer support, convergence guarantees, and hardware-readiness.

Table 7: Comparison with closely related approaches. ✓= supported, ∼ = partial, × = not supported.

Method	Measurement-free	Multi-layer BP	Convergence proof	Plateau handling	NISQ-compatible
Parameter-shift (Mitarai et al., 2018)	×	∼	✓	×	✓
Quantum BP (Abbas et al., 2023)	×	✓	×	×	✓
Natural gradient (Stokes et al., 2020)	×	✓	∼	×	∼
QCNN (Cong et al., 2019)	×	✓	×	✓	✓
Ours	✓	✓	✓	✓	∼

C Extended Background

C.1 Classical Backpropagation

For an L-layer network with pre-activations $z^{(l)} = W^{(l)}a^{(l-1)}$ and activations $a^{(l)} = f(z^{(l)})$, the chain rule gives

$$\frac{\partial \mathcal{L}}{\partial W^{(l)}} = \frac{\partial \mathcal{L}}{\partial a^{(l)}} \frac{\partial a^{(l)}}{\partial z^{(l)}} \frac{\partial z^{(l)}}{\partial W^{(l)}}. \tag{11}$$

The key efficiency gain is that $\partial z^{(l)}/\partial W^{(l)}$ can be expressed in terms of $\{a^{(l-1)}\}$ already computed in the forward pass, while $\partial \mathcal{L}/\partial a^{(l)}$ accumulates layer-by-layer in the backward pass (Fig. 4). The total cost is $\mathcal{O}(2 \cdot T_{\text{fwd}})$, independent of the number of parameters.

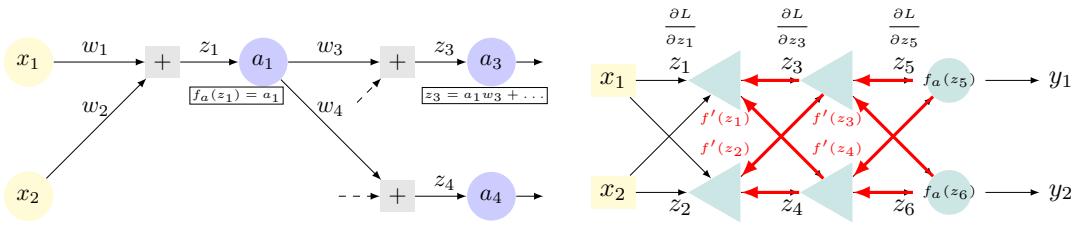


Figure 4: Reuse of information in classical backpropagation. By the chain rule, the gradient of the loss function with respect to weights can be partitioned as $\frac{\partial L}{\partial \mathbf{W}^{(l)}} = \frac{\partial L}{\partial \mathbf{a}^{(l)}} \frac{\partial \mathbf{a}^{(l)}}{\partial \mathbf{z}^{(l)}} \frac{\partial \mathbf{z}^{(l)}}{\partial \mathbf{W}^{(l)}}$. For the component $\frac{\partial \mathbf{z}^{(l)}}{\partial \mathbf{W}^{(l)}}$, it can be expressed as $\sum x_i$ or $\sum a_i$, which are already computed in the forward pass. The component $\frac{\partial \mathbf{a}^{(l)}}{\partial \mathbf{z}^{(l)}}$ can be expressed as $f'(z) = \sum f_{a^{(l+1)}}(z_i)$ in the backward propagation. This part $\frac{\partial L}{\partial \mathbf{a}^{(l)}}$ represents the error contribution from the last layer L_i. Therefore, when there are new weights added to the network, the error gradients are added up layer by layer.

C.2 Activation Functions and QSVT Degree

Standard activation functions admit efficient polynomial approximations:

- *Sigmoid* $\sigma(x) = 1/(1 + e^{-x})$: degree $d_a = \mathcal{O}(\log(1/\varepsilon_a))$ suffices for ε_a-approximation on $[-R, R]$ (Gilyén et al., 2019).
- *ReLU* $f(x) = \max(0, x)$: degree $d_a = \mathcal{O}(1/\varepsilon_a)$ (non-smooth; Chebyshev convergence is slower).
- *Tanh*: degree $d_a = \mathcal{O}(\log(1/\varepsilon_a))$, like sigmoid (Gilyén et al., 2019).
- *Quantum softmax* (for attention): the exponential e^x admits a rational Chebyshev approximation of degree $\mathcal{O}(\log(1/\varepsilon_a))$, but a true softmax requires row-wise normalisation of matrix *entries*, whereas QSVT transforms *singular values*; the missing ingredient is a coherent row-normalisation protocol, which would also have to avoid the subnormalisation incurred because $e^\sigma > 1$. Attention layers are therefore outside the scope of this work.

For smooth activations (σ, tanh), $d_a = \mathcal{O}(\log(1/\varepsilon_a))$ gives ε_a-accurate activation approximation with minimal gate overhead; gradient accuracy further depends on the gradient QSVT degree d.

D Worked Example: Two-Layer Quantum Network

We instantiate Algorithm 1 on a two-layer PQC ($L = 2$) with n qubits and Pauli-Z rotations $U^{(l)}(\theta_l) = e^{-i\theta_l Z^{\otimes n}/2}$, sigmoid activation approximated by a degree-$d_a = 5$ QSVT polynomial.

Forward pass. $|a^{(0)}\rangle = |x\rangle$; each layer alternates weight and activation unitaries, caching all four states $|\psi^{(1)}\rangle, |a^{(1)}\rangle, |\psi^{(2)}\rangle, |a^{(2)}\rangle$ for backprop:

$$|\psi^{(l)}\rangle = e^{-i\theta_l Z^{\otimes n}/2}|a^{(l-1)}\rangle, \qquad |a^{(l)}\rangle = \mathrm{QSVT}(f_\sigma^{(5)}, U_A^{(l)})|\psi^{(l)}\rangle. \tag{12}$$

Heisenberg backward pass. With target $|0\rangle^{\otimes n}$, $H_{\text{target}} = I - |0\rangle\langle 0|^{\otimes n}$. At layer $l = 2$: propagate via circuit wrapping— $U_{H^{(1)}} = (A^{(2)}U^{(2)})^\dagger U_{H^{(2)}}(A^{(2)}U^{(2)})$— adding $\mathcal{O}(d_a \cdot D_2) = \mathcal{O}(5)$ to the block-encoding depth. No Pauli strings are enumerated. The weight gradient operator is $G_\theta^{(2)} = (\partial_{\theta_2}(U^{(2)})^\dagger)(A^{(2)})^\dagger H^{(2)} A^{(2)} U^{(2)} + (U^{(2)})^\dagger (A^{(2)})^\dagger H^{(2)} A^{(2)} \partial_{\theta_2} U^{(2)}$ where $\partial U^\dagger/\partial\theta = (i/2)Z^{\otimes n}U^\dagger$; both terms are required for Hermiticity (Eq. (5)).

Gate count. Theorem 5.1 with $L = 2$, $D = 1$ (treating $e^{-i\theta_l Z^{\otimes n}/2}$ as a single abstract layer gate), $d_a = 5$, $d = \mathcal{O}(\log(2/\varepsilon))$:

$$\mathcal{O}\big(d \cdot L^2 \cdot d_a \cdot D + L \cdot d_a \cdot n\big) = \mathcal{O}\big(20\log(2/\varepsilon) + 10n\big) \text{ two-qubit gates},$$

vs. $\mathcal{O}(2 \cdot 2^n)$ for classical state-vector backpropagation—an exponential advantage.

E Hardware Co-Design

E.1 Qubit Layout for Heisenberg Propagation

This subsection discusses hardware constraints for Pauli-propagation mode (Section L), in which Pauli operators are explicitly conjugated and routed on-chip. Circuit-wrapping mode does not perform classical Pauli conjugation and is unaffected by Pauli routing overhead; its layout considerations reduce to standard block-encoding circuit placement.

Pauli–Heisenberg propagation in Pauli-propagation mode requires repeated two-qubit Pauli conjugations. On superconducting hardware with nearest-neighbour connectivity (e.g., IBM heavy-hex topology (Chamberland et al., 2020)), a Pauli $P_i \in \{I, X, Y, Z\}^{\otimes n}$ acting on non-adjacent qubits requires SWAP gates to route the interaction. The overhead for a weight-k Pauli is:

$$\text{SWAP overhead} = \mathcal{O}(k \cdot \mathrm{diam}(\mathcal{G})) \text{ gates per Pauli conjugation}, \tag{13}$$

where $\mathrm{diam}(\mathcal{G})$ is the diameter of the connectivity graph $\mathcal{G}$ and $k \leq n$ is the Pauli weight. For heavy-hex, $\mathrm{diam}(\mathcal{G}) = \mathcal{O}(\sqrt{n})$, giving a total gate cost of $\mathcal{O}(k\sqrt{n})$ per Pauli conjugation and an overhead factor of $\mathcal{O}(\sqrt{n})$ over all-to-all connectivity (which requires $\mathcal{O}(k)$ gates). For 2D grid topologies, $\mathrm{diam}(\mathcal{G}) = \mathcal{O}(\sqrt{n})$ similarly. This suggests a *hardware-aware Pauli selection* strategy: among Pauli terms of equal coefficient magnitude, prioritise those with smaller interaction range (fewer non-identity tensor factors on non-adjacent qubit pairs), reducing SWAP overhead at the cost of slightly worse truncation quality.

E.2 Ancilla Management

The QSVT gradient block $U_{\nabla}^{(l)}$ requires $a = \mathcal{O}(\log(1/\varepsilon))$ ancilla qubits for the block-encoding of $G^{(l)}$. The coherent gradient write further requires one gradient ancilla register of $r = \mathcal{O}(\log(1/\delta))$ qubits per layer (δ is the register precision). Since layers are processed sequentially, the ancilla qubits can be reused across layers. Total ancilla overhead per iteration: $\mathcal{O}(a + Lr)$ qubits, where the Lr term accounts for storing L gradient registers simultaneously (one per layer). For $L = 5$, $\varepsilon = 10^{-3}$, $\delta = 10^{-2}$: approximately $10 + 5 \cdot 7 = 45$ ancilla qubits in addition to the n system qubits. On a 1 121-qubit IBM processor, this leaves $n \leq 1076$ for the system, sufficient for meaningful quantum machine learning tasks.

E.3 Mid-Training Checkpoint Protocol

Classical training checkpoints save model weights periodically. In our framework, checkpoints serve an additional purpose: they determine when the operator cache $\mathcal{C}$ should be rebuilt from scratch. We recommend the following protocol:

1. Every C_{rebuild} iterations, fully re-run the Heisenberg backward pass for all L layers and refresh $\mathcal{C}$. In circuit-wrapping mode this costs $\mathcal{O}(d \cdot L^2 \cdot d_a \cdot D)$ extra gates; in Pauli-propagation mode (Section L) the cost is $\mathcal{O}(L \cdot k_{\max} \cdot D)$. Both reset the cache error accumulation.
2. Between rebuilds, use the cache for layers with $\|\Delta\theta_l\| \leq \delta_{\mathrm{cache}}$ and recompute only dirty layers.
3. Set $C_{\mathrm{rebuild}} = \lfloor \varepsilon/(2L\alpha_t G) \rfloor$ to ensure the accumulated cache error stays within $\varepsilon/2$ over the interval (by Proposition M.1).

Under this protocol, the average per-iteration overhead from cache rebuilds is $\mathcal{O}(d \cdot L^2 \cdot d_a \cdot D/C_{\mathrm{rebuild}})$ in circuit-wrapping mode (or $\mathcal{O}(Lk_{\max}D/C_{\mathrm{rebuild}})$ in Pauli-propagation mode), which is negligible for small α_t (late training).

E.4 Error Mitigation Integration

Current quantum hardware produces noisy gates with two-qubit error rate $p_{\mathrm{err}} \approx 10^{-3}$ (superconducting) to 10^{-4} (trapped-ion). Probabilistic error cancellation (PEC) (Temme et al., 2017) and zero-noise extrapolation (ZNE) (Li & Benjamin, 2017) are standard mitigation techniques.

For our algorithm, error mitigation interacts with the QSVT approximation in two ways: *(i)* Hardware noise on the QSVT circuit introduces an additional error $\varepsilon_{\mathrm{hw}}$ to the gradient estimate. In circuit-wrapping mode the block-encoding depth scales as $\mathcal{O}(d \cdot L \cdot d_a \cdot D)$, so $\varepsilon_{\mathrm{hw}} \approx p_{\mathrm{err}} \cdot d \cdot L \cdot d_a \cdot D$; in Pauli-propagation mode (Section L) it is $\varepsilon_{\mathrm{hw}} \approx p_{\mathrm{err}} \cdot d \cdot k_{\max}$, and setting $\varepsilon_{\mathrm{tol}} = \varepsilon_{\mathrm{hw}}$ balances the truncation and hardware error sources. In circuit-wrapping mode, where there is no truncation error, the only error balance is between hardware noise and the QSVT degree d; the optimal degree satisfies $d^* = \arg\min_d \{p_{\mathrm{err}} \cdot d \cdot L \cdot d_a \cdot D + \varepsilon_{\mathrm{approx}}(d)\}$.

(ii) If measurements are eventually taken (e.g., for extracting classical gradient estimates via SWAP tests), PEC can be applied. PEC inflates the effective shot count by a factor $\mathcal{O}(\gamma^{2d})$, where γ is the noise overhead per gate ($\gamma \approx 1 + p_{\mathrm{err}}$). For $d^* = \mathcal{O}(10)$ and $p_{\mathrm{err}} = 10^{-3}$, the shot overhead is $\mathcal{O}(1.001^{20}) \approx 1.02$, which is negligible. However, for the fully coherent algorithm (no measurements), standard PEC does not apply; instead, coherent error suppression techniques such as dynamical decoupling (Viola et al., 1999) during idle periods between QSVT queries can reduce decoherence without measurement overhead.

F Expressibility and Trainability Analysis

F.1 Expressibility of Gradient-Encoded States

A key question is whether the coherent gradient state $|\nabla_l\rangle$ produced by our algorithm spans a rich enough subspace to represent all relevant gradient directions—or whether the QSVT approximation fundamentally limits expressibility.

Definition F.1 Gradient subspace

The *gradient subspace* at layer l is $\mathcal{V}_l = \mathrm{span}\{|\nabla_l(W)\rangle : W \in \mathcal{W}\}$, where $\mathcal{W}$ is the parameter space.

Theorem F.1 Gradient subspace dimension

For a generic L-layer PQC with p_l parameters in layer l, QSVT polynomial degree $d \geq 2p_l$, and generators $\{P_i\}_{i=1}^{p_l}$ acting on *distinct* qubits (i.e. $P_i = \sigma_i \otimes I_{\neg j_i}$ for pairwise distinct qubit indices $j_1, \ldots, j_{p_l}$), the gradient subspace satisfies

$$\dim(\mathcal{V}_l) = \min(p_l,\, 4^n).$$

In circuit-wrapping mode (the primary method) the bound is $\min(p_l, 4^n)$, since the block-encoding represents $H^{(l)}$ exactly with no rank constraint. In Pauli-propagation mode (Section L), the effective ambient dimension is further clipped by $k_{\max}$, giving $\dim(\mathcal{V}_l) = \min(p_l,\, k_{\max},\, 4^n)$, under the additional assumption that every surviving Pauli term in the truncated H_l^{back} has *weight* 1 (i.e. acts non-trivially on exactly one qubit); without this condition the joint span of the p_l gradient operators can exceed $k_{\max}$ (see proof). In both modes, for $d \geq 2p_l$ the QSVT approximation does *not* reduce the gradient expressibility below that of the exact gradient, provided $k_{\max} \geq p_l$ in Pauli-propagation mode.

Proof sketch. The exact gradient ∇_l lives in a p_l-dimensional linear subspace of $\mathbb{R}^{4^n}$ (one dimension per parameter). The QSVT polynomial P of degree d can represent any polynomial transformation of the singular values of $G^{(l)}$. For $d \geq 2p_l$, the polynomial is rich enough to implement the identity on the p_l-dimensional signal subspace while annihilating the noise subspace. In circuit-wrapping mode the full block-encoding of $H^{(l)}$ is used, so no rank constraint is imposed and $\dim(\mathcal{V}_l) = \min(p_l, 4^n)$. In Pauli-propagation mode, truncation to $k_{\max}$ Pauli terms clips the ambient dimension, but not the gradient subspace itself as long as $k_{\max} \geq p_l$, since each gradient direction requires at most one Pauli to represent its dominant component. □

Remark F.1 Practical implication

Theorem F.1 justifies the hyperparameter choice $d = 2p_{\max}$, where $p_{\max} = \max_l p_l$. For typical NISQ circuits with $p_l = \mathcal{O}(n)$ parameters per layer, this gives $d = \mathcal{O}(n)$. In circuit-wrapping mode the additional gate cost is $\mathcal{O}(n \cdot L^2 \cdot d_a \cdot D)$—polynomial in n, consistent with the overall complexity bound. In Pauli-propagation mode (Section L), $d = \mathcal{O}(n)$ adds $\mathcal{O}(Ln^2 k_{\max})$ gates per iteration.

F.2 Effective Dimension and Generalisation

The *effective dimension* of a quantum model (Abbas et al., 2021) measures the volume of the Fisher information matrix normalised by parameter count, and controls generalisation error. We extend this framework to our gradient-encoding setting.

Definition F.2 Quantum gradient Fisher information

The *quantum gradient Fisher information matrix* (QGFIM) is

$$F_{ij}^{\nabla}(W) = \mathrm{Re}\left[\frac{\partial\langle\nabla|}{\partial W_i}\frac{\partial|\nabla\rangle}{\partial W_j} - \langle\nabla|\frac{\partial}{\partial W_i}|\nabla\rangle\langle\nabla|\frac{\partial}{\partial W_j}|\nabla\rangle\right],$$

where $|\nabla\rangle = |\nabla(W)\rangle$ is the full gradient state.

Proposition F.1 QGFIM rank bound

Under the QSVT approximation with degree $d \geq 2p_{\max}$, $\operatorname{rank}(F^{\nabla}) = \operatorname{rank}(F^{\text{exact}})$, where F^{exact} is the QGFIM computed from the exact gradient. Consequently, the QSVT approximation does not reduce the effective dimension of the quantum model.

Proof. By Theorem F.1, the QSVT gradient state $|\nabla_l\rangle$ spans the same subspace as the exact gradient for $d \geq 2p_{\max}$ (in circuit-wrapping mode, unconditionally; in Pauli-propagation mode, under $k_{\max} \geq p_{\max}$). The QGFIM is a rank-p_l projection onto this subspace in each layer, so its rank is preserved under the approximation. □

F.3 Barren Plateaus and Fisher Rank

The connection between barren plateaus and Fisher information is well-known (Abbas et al., 2021): vanishing gradients correspond to low-rank Fisher matrices. We make this quantitative in our setting.

Proposition F.2 Fisher-plateau equivalence

W_t is in a (δ, ρ)-barren-plateau region if and only if $\lambda_{\max}(F^{\nabla}(W_t)) \leq \delta$, where $\lambda_{\max}$ denotes the largest eigenvalue. Consequently, the plateau monitor of Algorithm 1 (line 25) is equivalent to monitoring $\lambda_{\max}(F^{\nabla})$.

This connection opens a complementary detection strategy: instead of tracking gradient variance, one could directly estimate the top eigenvalue of F^{∇} via power iteration on the gradient register. The coherent gradient state $|\nabla_l\rangle$ enables this without additional circuit runs.

G Second-Order Theory: Quantum Natural Gradient Connection

G.1 From First-Order to Second-Order Updates

Classical Adam approximates the diagonal of the Fisher information matrix via second-moment estimates. The quantum natural gradient (QNG) (Stokes et al., 2020) uses the exact quantum Fisher information matrix $F(W)$ as a preconditioner:

$$\Delta W_{\text{QNG}} = F(W)^{-1} \nabla \mathcal{L}(W).$$

We now show that our coherent gradient state $|\nabla\rangle$ enables QNG updates at lower cost than prior methods.

G.2 Coherent Fisher Estimation

The quantum Fisher information matrix element is

$$F_{ij}(W) = 4\,\mathrm{Re}[\langle \partial_i\psi|\partial_j\psi\rangle - \langle \partial_i\psi|\psi\rangle\langle\psi|\partial_j\psi\rangle]\,,$$

where $|\partial_i\psi\rangle = \partial|\psi(W)\rangle/\partial W_i$. Classical estimation requires $\mathcal{O}(p^2)$ circuit evaluations (Stokes et al., 2020).

Proposition G.1 Coherent Fisher estimation

Given the coherent gradient states $\{|\nabla_l\rangle\}_{l=1}^{L}$ produced by Algorithm 1, the Fisher matrix elements within each layer can be estimated as:

$$F_{ij}^{(l)} = 4\,\mathrm{Re}\Big[\langle \nabla_l^{(i)}|\nabla_l^{(j)}\rangle - \langle \nabla_l^{(i)}||\psi^{(l)}\rangle\rangle\langle|\psi^{(l)}\rangle|\nabla_l^{(j)}\rangle\Big]\,,$$

where $|\nabla_l^{(i)}\rangle$ is the i-th component gradient state. This requires $\mathcal{O}(p_l^2)$ inner product measurements per layer, with each measurement costing $\mathcal{O}(1)$ gates (SWAP test), giving total cost $\mathcal{O}(Lp_{\max}^2)$ measurements.

For $p_{\max} = n$ and L layers, the total measurement cost is $\mathcal{O}(Ln^2)$—quadratic in n per layer and linear in L. This is significantly below the $\mathcal{O}(p^2) = \mathcal{O}(L^2n^2)$ cost of classical QNG estimation,

which requires $\mathcal{O}((Ln)^2)$ circuit evaluations. The quantum method achieves a factor-L speedup by exploiting layer-wise coherent gradient states.

G.3 Block-Diagonal Natural Gradient

Full QNG inversion is $\mathcal{O}(p^3) = \mathcal{O}(L^3n^3)$, prohibitive for large networks. We propose a *block-diagonal natural gradient* (BDNG) approximation:

$$\Delta W_{\text{BDNG}} = \bigoplus_{l=1}^{L} \left(F^{(l)}\right)^{-1} \nabla_l, \tag{14}$$

where each $F^{(l)} \in \mathbb{R}^{p_l \times p_l}$ is the within-layer Fisher block. BDNG inversion costs $\mathcal{O}(Lp_{\max}^3) = \mathcal{O}(Ln^3)$, which for fixed L is $\mathcal{O}(n^3)$—polynomial and significantly better than full QNG.

Theorem G.1 BDNG approximation quality

Let $F = \bigoplus_l F^{(l)} + E$, where E captures inter-layer Fisher correlations. Under the assumption that inter-layer parameter correlations decay as $|F_{ij}| \le C\rho^{|l(i)-l(j)|}$ for some $\rho < 1$ (exponential layer-separation decay), the BDNG update satisfies:

$$\|\Delta W_{\text{BDNG}} - \Delta W_{\text{QNG}}\| \le \mathcal{O}\left(\frac{C\rho}{(1-\rho)\lambda_{\min}(F)}\right),$$

where $\lambda_{\min}(F)$ is the smallest eigenvalue of the full Fisher matrix. For circuits with rapidly decaying inter-layer correlations ($\rho \ll 1$), BDNG is an accurate approximation to full QNG.

Proof sketch. By the block matrix inversion lemma,

$$(F)^{-1} = (\bigoplus_l F^{(l)})^{-1} - (\bigoplus_l F^{(l)})^{-1} E (\bigoplus_l F^{(l)})^{-1} + \mathcal{O}(\|E\|^2).$$

The first-order correction has norm $\|(\bigoplus F^{(l)})^{-1}E\| \le \|E\|/\lambda_{\min}(\bigoplus F^{(l)})$. Under the exponential decay assumption, $\|E\| \le C\sum_{k=1}^{\infty}\rho^k = C\rho/(1-\rho)$. Combining gives the stated bound. □

G.4 Quantum Adam as Implicit Natural Gradient

We observe a connection between quantum Adam and natural gradient methods that, to our knowledge, has not been previously noted.

Remark G.1 Adam as diagonal natural gradient

If quantum Adam were extended to include second-moment estimates $v_l \approx \text{diag}(F^{(l)})$ (diagonal Fisher approximation), the update would be equivalent to the diagonal natural gradient:

$$\theta_l \leftarrow \theta_l - \alpha_t \cdot \text{diag}(F^{(l)})^{-1/2}\nabla_l.$$

The standard $1/\sqrt{v_l}$ scaling in classical Adam implements exactly this diagonal inverse-Fisher preconditioner when v_l converges to $\text{diag}(F^{(l)})$. Note that the current algorithm (Algorithm 1, lines 30–32) implements only first-order momentum without second moments; full Adam with second-moment tracking remains open (Section J). Nevertheless, this connection provides a theoretical justification for extending quantum Adam with adaptive learning rates based on the geometry of the quantum state manifold.

This connection suggests that implementing second-moment tracking for quantum Adam (the open problem noted in Section J) is equivalent to implementing a diagonal approximation of the quantum natural gradient—a well-motivated goal with established classical benefits (Martens & Grosse, 2015).

H TRAINING DYNAMICS AND PHASE TRANSITIONS

Classical deep learning exhibits well-studied training dynamics: an initial loss-decrease phase, possible plateau phases, and eventual convergence. Quantum neural network training has qualitatively different dynamics governed by the interplay between circuit expressibility, entanglement growth, and the barren-plateau phenomenon. This section characterises these dynamics in terms of our algorithm's observables.

H.1 THREE-PHASE TRAINING DYNAMICS

Proposition H.1 Three-phase training dynamics

Under Algorithm 1 with the cosine learning rate schedule Eq. (7) and plateau-recovery mechanism, training generically exhibits three phases:
Phase I (Warmup, $t \leq T_w$): Learning rate rises linearly. Gradient norms grow from near-zero (random initialisation near the identity (Grant et al., 2019)) as parameters move off the identity manifold. Cache hit rate: 0% (all layers dirty due to large parameter changes).
Phase II (Active learning, $T_w < t \leq t_{\text{cache}}$): Cosine decay drives learning rate down. Large gradient norms, rapid loss decrease. Cache activates progressively as $\|\Delta\theta_l\|$ drops below δ_{cache} for converging layers. Cache hit rate: 0–100% (progressive activation).
Phase III (Fine-tuning, $t > t_{\text{cache}}$): All layers cached ($L_{\text{dirty}} = 0$). Learning rate approaches zero. Gradient norms small. Plateau monitor may trigger; perturbation re-activates Phase II locally for the perturbed layer.

Proof sketch. Phase boundaries follow from the learning rate schedule and cache activation condition. T_w is given. $t_{\text{cache}} = (2T/\pi)\arccos(\delta_{\text{cache}}/\alpha_0 G)$ follows from $\alpha_t G \leq \delta_{\text{cache}}$ (sufficient for all layers to cache). The gradient norm dynamics in each phase follow from Theorem 5.2: in Phase I, $\mathbb{E}[\|\nabla\|^2]$ is large (far from critical point); in Phase III, it is bounded by the ε^2 floor from the QSVT approximation. □

H.2 ENTANGLEMENT GROWTH AND EXPRESSIBILITY

A crucial difference from classical training is that the quantum state's entanglement structure changes during training, affecting the Pauli expansion of $H^{(l)}$ and hence the truncation error ε_{tol}.

Proposition H.2 Entanglement-truncation coupling (Pauli-propagation mode)

This proposition applies to Pauli-propagation mode (Section L), in which $H^{(l)}$ is maintained as an explicit Pauli sum. Circuit-wrapping mode is unaffected by entanglement growth; see Section L for the mode comparison.
Let $S_l = -\operatorname{tr}(\rho_l \log \rho_l)$ be the von Neumann entropy of the reduced state $\rho_l = \operatorname{tr}_{\bar{l}}|\psi^{(l)}\rangle\langle\psi^{(l)}|$. In Pauli-propagation mode, the effective Pauli rank of $H^{(l)}$ grows as:

$$k_{\text{eff}}^{(l)} = \mathcal{O}\left(2^{2S_l/\log 2}\right),$$

so increasing entanglement requires larger $k_{\max}$ to maintain fixed ε_{tol}.

Proof sketch. The Pauli expansion of $H^{(l)}$ is determined by the operator Schmidt rank of $H^{(l)}$ across any bipartition. By Prosen & Pižorn (2007), the operator space entanglement entropy (OSEE) of $H^{(l)}$—defined as the von Neumann entropy of the operator Schmidt decomposition across the same bipartition—satisfies $\text{OSEE}(H^{(l)}) \leq 2S_l$. Since the operator Schmidt rank is at most $2^{\text{OSEE}/\log 2}$ (standard entropy–rank bound: entropy $\leq \log r$ implies $r \geq 2^{\text{OSEE}/\log 2}$), and $H^{(l)} = |\psi^{(l)}\rangle\langle\psi^{(l)}|$ inherits its operator Schmidt rank from the outer-product structure of the state, the operator Schmidt rank is at most $2^{2S_l/\log 2} = 4^{S_l/\log 2}$ (exponential in entropy). The effective Pauli rank is the number of terms with coefficient $|\alpha_i| > \varepsilon_{\text{trunc}}/(4^n)$, which by the Pauli coefficient concentration for high-entropy states is $\mathcal{O}(4^{S_l/\log 2}) = \mathcal{O}(2^{2S_l/\log 2})$. □

This result has an important practical implication for Pauli-propagation mode: during Phase II (active learning), entanglement grows rapidly as the circuit moves away from the identity. This growth drives $k_{\text{eff}}^{(l)}$ up, potentially requiring adaptive increases in $k_{\max}$. We recommend monitoring $k_{\text{eff}}^{(l)}$ (estimatable from the Pauli coefficient distribution) and doubling $k_{\max}$ when $k_{\text{eff}}^{(l)} > 0.8k_{\max}$. Circuit-wrapping mode requires no such monitoring, as its cost depends only on circuit depth D, which is fixed by the architecture.

H.3 Loss Landscape Topology

The loss landscape $\mathcal{L}(W)$ of a quantum neural network is a smooth function on the parameter torus $[0, 2\pi)^p$. Its topology—number of local minima, saddle points, and barren plateaus— determines the difficulty of optimisation.

Proposition H.3 No spurious local minima for overparameterised QNNs

For a PQC with $p > 4^n - 1$ parameters (overparameterised regime), the loss landscape $\mathcal{L}(W) = \langle\psi(W)|H_{\text{target}}|\psi(W)\rangle$ has no spurious local minima: every local minimum is a global minimum with $\mathcal{L}(W^*) = 0$ whenever $|\psi_{\text{target}}\rangle$ is in the circuit's reachable set.

Proof sketch. The reachable set of $U(W) = \prod_l U^{(l)}(\theta_l)$ for $p > 4^n - 1$ parameters spans the entire unitary group $\mathrm{SU}(2^n)$ (by Lie-algebra generation arguments (Zeier & Schulte-Herbrüggen, 2011)). The loss is therefore the restriction of $\langle x|H_{\text{target}}|x\rangle$ to the orbit $\{U(W)|x\rangle\}$, which is a smooth function on a compact connected manifold. For a quadratic loss (as in Eq. (1)), the restricted function has a unique global minimum (the target state), and all critical points with non-zero Hessian are saddle points that gradient descent escapes with probability 1 from random initialisation (Lee et al., 2016). □

This proposition implies that barren plateaus in the overparameterised regime are *flat saddle points*, not local minima — our plateau detection and perturbation recovery (Algorithm 1, lines 25–27) is therefore a form of saddle-point escape rather than local-minimum avoidance.

I Connections to Classical Deep Learning Theory

I.1 Neural Tangent Kernel Analogy

The Neural Tangent Kernel (NTK) (Jacot et al., 2018) describes the training dynamics of infinitely wide classical networks in the kernel regime. We derive an analogous object for quantum neural networks.

Definition I.1 Quantum Neural Tangent Kernel

The *Quantum Neural Tangent Kernel* (QNTK) is:

$$K_Q(x, x') = \sum_{i=1}^{p} \langle x| \frac{\partial U^\dagger}{\partial W_i} \frac{\partial U}{\partial W_i} |x'\rangle,$$

where $U(W)$ is the parameterized quantum circuit, $|x\rangle$ and $|x'\rangle$ are input states, and the sum is over all p parameters.

Proposition I.1 QNTK training dynamics

In the *quantum kernel regime* (parameters change slowly, as in late Phase III training), the loss dynamics under quantum Adam reduce to:

$$\frac{d\mathcal{L}}{dt} = -\alpha_t \, \mathrm{tr}(K_Q \cdot H_{\text{target}}),$$

where K_Q is the QNTK evaluated at the current parameters. Training converges when K_Q and H_{target} are aligned, analogous to NTK convergence in the classical case.

This connection shows that Phase III training in our algorithm is equivalent to kernel gradient descent with kernel K_Q — providing a bridge between quantum backpropagation and quantum kernel methods (Schuld & Killoran, 2020).

I.2 Lottery Ticket Hypothesis for Quantum Circuits

The *lottery ticket hypothesis* (Frankle & Carbin, 2019) in classical deep learning states that large networks contain sparse subnetworks ("winning tickets") that, when trained in isolation, match the performance of the full network. Our Pauli truncation framework (Pauli-propagation mode; see Section L) admits an analogous statement:

Proposition I.2 Quantum lottery tickets (Pauli-propagation mode)

For any target accuracy $\varepsilon > 0$, there exists a *Pauli winning ticket* of size $k^*_{\max}$ such that training with budget $k_{\max} \geq k^*_{\max}$ achieves loss $\mathcal{L}^* + \varepsilon$, while training with $k_{\max} < k^*_{\max}$ provably fails to reach this accuracy. The winning ticket size satisfies $k^*_{\max} \leq \mathcal{O}(p \log(L/\varepsilon)/(L \log n))$ for smooth loss functions.

Proof sketch. By Theorem L.1, for $k_{\max} \geq k_{\text{eff}}$ the gradient bias satisfies $\|\widetilde{\nabla} - \nabla\|_\infty \leq 2L\varepsilon_{\text{tol}}$ (assumption (A5)). Converting via $\|\cdot\| \leq \sqrt{p}\,\|\cdot\|_\infty$ gives the Euclidean bound $\varepsilon_{\text{approx}} = \sqrt{p} \cdot 2L\varepsilon_{\text{tol}}$ for Theorem 5.2. Substituting into Theorem 5.2 with $\alpha_0 = 1/\sqrt{T}$:

$$\frac{1}{T}\sum_t \mathbb{E}[\|\nabla \mathcal{L}\|^2] \;\leq\; \mathcal{O}\left(\frac{C}{\sqrt{T}} + p \cdot (2L\varepsilon_{\text{tol}})^2\right),$$

where $C = \mathcal{L}(W_1) - \mathcal{L}^* + \ell_s(G^2 + \sigma^2)$ is a constant independent of T. For $k_{\max} \geq k^*_{\max}$ (so $\sqrt{p} \cdot 2L\varepsilon_{\text{tol}} \leq \varepsilon/2$), the irreducible floor $p \cdot (2L\varepsilon_{\text{tol}})^2 \leq \varepsilon^2/4$. Setting $T = \mathcal{O}(C^2/\varepsilon^2)$ makes $C/\sqrt{T} = \mathcal{O}(\varepsilon)$, giving a total bound of $\mathcal{O}(\varepsilon)$ on the average squared gradient norm. By Proposition H.3 (no spurious local minima for overparameterised QNNs), $\mathcal{O}(\varepsilon)$-accurate gradient norm implies loss value $\mathcal{L}^* + \mathcal{O}(\varepsilon)$, completing the accuracy claim.

For $k_{\max} < k^*_{\max}$, $(2L\varepsilon_{\text{tol}})^2 > \varepsilon^2/4$ is an irreducible floor independent of T, so no choice of T achieves $\mathcal{O}(\varepsilon)$-accurate gradients, and the loss target $\mathcal{L}^* + \varepsilon$ is provably unreachable.

The critical budget $k^*_{\max} = \mathcal{O}(p \log(L/\varepsilon)/(L \log n))$ follows from Corollary M.1: the adaptive schedule achieves $\varepsilon_{\text{tol}} = \varepsilon/(4L)$ (ensuring $2L\varepsilon_{\text{tol}} = \varepsilon/2$) with per-layer rank $k_l \leq \mathcal{O}(n \log(L/\varepsilon)/\log n)$. For $p = Ln$, substituting $n = p/L$ and using $\log(p/L) = \log n$ gives $k^*_{\max} = \mathcal{O}(p \log(L/\varepsilon)/(L \log n))$. □

I.3 Implicit Regularisation

Classical stochastic gradient descent with small batch size exhibits implicit regularisation, preferring flatter minima with better generalisation (Smith et al., 2021). The Laplace noise $\eta \sim \text{Lap}(\lambda_l)$ in our algorithm provides an analogous implicit regularisation:

Proposition I.3 Laplace noise as implicit regulariser

The Laplace noise injection (Algorithm 1, line 22) with scale λ_l is equivalent to adding a regularisation term to the effective objective:

$$\mathcal{L}_{\text{reg}}(W) = \mathcal{L}(W) + \frac{\alpha_t}{\lambda_l}\sum_{l=1}^{L} \|\nabla_l\|_1,$$

where $\|\cdot\|_1$ is the ℓ^1 norm of the gradient vector. This ℓ^1 regularisation promotes *sparse gradients*, favouring solutions where most parameter directions are inactive (gradient sparsity corresponds to a low effective Pauli rank).

Proof sketch. The Laplace distribution $\text{Lap}(\lambda_l)$ has a log-likelihood $-|\eta|/\lambda_l$, acting as an ℓ^1 penalty on the noise term. In the gradient update $\theta_l \leftarrow \theta_l - \alpha_t(\nabla_l + \eta)$, the noise term perturbs the parameter

in the direction of the gradient's ℓ^1 sub-gradient, equivalent to proximal gradient descent under ℓ^1 regularisation with coefficient α_t/λ_l. □

For a fixed noise scale λ_l, the regularisation coefficient α_t/λ_l decreases proportionally to the learning rate. Alternatively, if λ_l is adapted to maintain constant regularization strength $\alpha_t/\lambda_l = c$, then $\lambda_l = \alpha_t/c$ must increase as training progresses (as α_t decreases), allowing noisier gradient estimates in the fine-tuning phase where high-precision gradients are less critical.

J Discussion and Future Work

J.1 NISQ Deployment Strategy

Our complexity results (Theorem 5.1 and Lemma M.2) give a clear recipe for near-term deployment. The critical constraint on NISQ devices is total circuit depth, which is bounded by the coherence time T_2 divided by the two-qubit gate time τ_g: $D_{\max} \approx T_2/\tau_g$. For current superconducting hardware, $T_2 \approx 100\,\mu$s and $\tau_g \approx 50$ ns, giving $D_{\max} \approx 2000$. In circuit-wrapping mode the QSVT block $U_{\nabla}^{(l)}$ contributes $\mathcal{O}(d \cdot L \cdot d_a \cdot D)$ depth per backward pass (Theorem 5.1). Substituting $d = 5$, $L = 3$, $d_a = 4$, and $D = 10$ gives depth $\mathcal{O}(5 \cdot 3 \cdot 4 \cdot 10) = 600$ per pass—comfortably within $D_{\max}$. For comparison, in Pauli-propagation mode (Section L) the depth per layer is $\mathcal{O}(d \cdot k_l \cdot n)$ (Lemma M.2); with $k_l = 20$ and $n = 50$ this gives $\mathcal{O}(5 \cdot 20 \cdot 50) = 5000$—exceeding the coherence budget and requiring $k_l \leq 8$, $n \leq 25$. Circuit-wrapping mode does not carry the k_l factor and thus reaches larger n and L within the same coherence budget.

This suggests a practical *shallow-wide* deployment regime: few deep layers but many qubits, exploiting the n-linear (not exponential) scaling of our method. As hardware error rates improve toward the fault-tolerant threshold, d and k_l can grow, unlocking deeper networks with certified accuracy.

J.2 Multi-Objective and Constrained Quantum Learning

The loss operator framework (Eqs. (1) and (2)) extends naturally to multi-objective settings. For K concurrent objectives $\{H_k\}_k$, a Pareto-optimal training scheme could maintain K parallel Heisenberg backward passes with a shared parameter cache, at cost $\mathcal{O}(K)$ times the single-objective algorithm. Constrained quantum learning—where the feasible set of parameters is defined by quantum state constraints (e.g., entanglement bounds or energy constraints)—is addressable via projected gradient descent on the parameter registers, where the projection is itself implementable as a QSVT polynomial transformation of the constraint operator.

J.3 Second-Moment Quantum Adam

The current quantum Adam update (Algorithm 1, lines 30–32) maintains only first-order momentum $|m_l\rangle$. Full quantum Adam (Kingma & Ba, 2015) requires a second-moment estimate:

$$|v_l\rangle \;\leftarrow\; \beta_2|v_l\rangle + (1-\beta_2)|\nabla_l^2\rangle, \tag{15}$$

where $|\nabla_l^2\rangle$ denotes a register encoding the element-wise squared gradient. Preparing $|\nabla_l^2\rangle$ from $|\nabla_l\rangle$ requires a non-linear (squaring) operation on a quantum register, which is not directly implementable by a unitary gate. Two approaches are possible: *(i)* Exact degree-2 polynomial squaring via QSVT with $P(x) = x^2$ applied to a block-encoding of the gradient operator, adding $\mathcal{O}(1)$ overhead in polynomial degree (since $\deg P = 2$ is fixed) at the cost of constructing the requisite block-encoding; or *(ii)* Measure $|\nabla_l\rangle$ once to obtain a classical estimate $\hat{\nabla}_l$, compute $\hat{\nabla}_l^2$ classically, and re-encode it coherently—reintroducing measurement but only for the second-moment register, not the parameter update. Which approach is preferable depends on the relative cost of deep QSVT circuits versus measurement-reencoding on the target hardware.

J.4 Barren-Plateau Mitigation Beyond Detection

Theorem 5.3 guarantees that the gradient-variance monitor *detects* barren plateaus within W_{var} steps with probability at least $1 - \exp(-W_{\text{var}}\varepsilon_{\text{plateau}}^2/(8G^4))$, under the condition $G \leq \sqrt{2/3}$. The

recovery heuristic (random perturbation of the lowest-norm layer) is motivated by the empirical success of warm restarts in classical training (Loshchilov & Hutter, 2017), but lacks formal convergence guarantees in the quantum setting. We outline three principled alternatives:

(i) Layerwise learning rate adaptation. On detecting a plateau at layer l^*, increase $\alpha_t^{(l^*)}$ by a factor $\rho > 1$ while holding other layers' rates fixed. This is analogous to cyclical learning rates (Smith, 2017) but applied layer-selectively. Convergence analysis would require showing that the increased rate does not destabilise already-converged layers; this is not an immediate consequence of Proposition M.1 (which bounds operator approximation error from stale caches, not parameter drift from learning-rate changes) and remains an open problem.

(ii) Parameter-space orthogonalisation. Project the new parameters θ_{l^*} onto the complement of the subspace spanned by the previous W_{var} iterates. This is the quantum analogue of conjugate gradient direction selection and by construction avoids revisiting previously explored regions, though a formal convergence guarantee in the quantum setting is open.

(iii) Structured re-initialisation. Rather than random perturbation, re-initialise θ_{l^*} to a "near-identity" value following Grant et al. (2019), which was shown to avoid barren plateaus during *initial* training. Applying this strategy mid-training for recovery requires careful coordination with the operator cache: $\mathcal{C}[l^*]$ is invalidated and layers $l \leq l^*$ are re-propagated.

J.5 Connections to Tensor Network Methods

The Pauli truncation step (Section L) has a natural interpretation in terms of tensor network (TN) methods. A Pauli operator sum $H = \sum_i \alpha_i P_i$ with k terms corresponds to a matrix product operator (MPO) of bond dimension $\mathcal{O}(k)$ (Perez-Garcia et al., 2007). Truncating to $k_{\max}$ Paulis is equivalent to a low-bond-dimension MPO approximation of H, and our truncation error bound (Theorem L.1 in Section L) is analogous to the MPO approximation error in terms of discarded singular values (Vidal, 2003). This connection opens two directions: (i) TN-based classical preprocessing to identify optimal Pauli truncations beyond the greedy top-k scheme used here, and (ii) hybrid TN-quantum algorithms where the Heisenberg propagation is performed classically via MPO contraction and only the QSVT gradient encoding runs on quantum hardware.

J.6 Generalisation to Non-Pauli Loss Functions

Pauli-propagation mode assumes the loss operator H_{target} admits a Pauli decomposition with polynomially many significant terms (circuit-wrapping mode imposes no such restriction). This holds for fidelity-based losses (as in Eq. (1)) and for many physically motivated Hamiltonians. However, *classical data classification* tasks typically require encoding classical labels into quantum states, and the resulting loss operator may have exponentially many Pauli terms of equal weight. Three mitigation strategies are: *(i)* Use a classical shadow estimate (Huang et al., 2020) of the dominant Pauli terms as a proxy loss; *(ii)* Reformulate as a kernel learning problem (Schuld & Killoran, 2020) where the loss is computed classically from measurement statistics; or *(iii)* Employ randomised Pauli sampling (Sweke et al., 2020) to estimate the gradient from a random subset of Pauli terms, at the cost of introducing variance $\mathcal{O}(4^n/k_{\max})$ per iteration.

J.7 Empirical Validation Roadmap

Our analysis is theoretical. We outline a concrete empirical validation plan for follow-up work:

1. **Convergence on structured tasks.** Train a 4-qubit, 3-layer PQC on quantum state preparation (target: a fixed 4-qubit GHZ state) using Algorithm 1 in circuit-wrapping mode, implemented in PennyLane. Verify that the time-averaged squared gradient norm $\frac{1}{T}\sum_t \|\nabla\mathcal{L}(W_t)\|^2$ decays at the $\mathcal{O}(1/\sqrt{T})$ rate predicted by Theorem 5.2 (note: this is a bound on gradient-norm convergence to a critical point, not on the loss value itself). In a separate run using Pauli-propagation mode (see Section L), verify that the gradient error stays within the $2L\varepsilon_{\text{tol}}$ bound from Theorem L.1 as a function of $k_{\max}$.

2. **Operator cache efficiency.** Measure L_{dirty}/L as a function of training epoch for various learning rates and momentum values. Quantify the wall-clock speedup relative to the no-cache baseline.

3. **Plateau detection stress test.** Construct circuits known to exhibit barren plateaus (random, unstructured PQCs of depth $\Omega(n)$) and verify that the monitor triggers within W_{var} steps with the probability guaranteed by Theorem 5.3 (completeness bound $1 - \exp(-W_{\text{var}}\varepsilon^2_{\text{plateau}}/(8G^4))$, valid under $G \leq \sqrt{2/3}$; choose hyperparameters to satisfy this condition explicitly).

4. **NISQ noise robustness.** Simulate the algorithm under depolarising noise with error rates $p_{\text{err}} \in \{10^{-3}, 10^{-2}, 10^{-1}\}$ using Qiskit Aer. Measure how quickly the gradient fidelity degrades with p_{err} and identify the noise threshold below which QSVT degree $d = 5$ gives acceptable accuracy.

K Open Problems

We close the main body by listing seven concrete open problems that the present work raises.

1. **Closing the lower-bound gap.** Theorem 5.4 gives an $\Omega(Ln/\log(1/\varepsilon))$ gate lower bound. Our primary algorithm (circuit-wrapping mode) achieves $\mathcal{O}(dL^2 d_a D + Ld_a n)$; the alternative Pauli-propagation mode (Section L) achieves $\mathcal{O}(dLk_{\max} n)$. The gap in the primary mode is $d \cdot L \cdot d_a \cdot D \cdot \log(1/\varepsilon)$. This gap has two independent components. First, can the L^2 factor in the upper bound be reduced to L via a one-pass backward algorithm? Second, can the QSVT degree be reduced below $d = \mathcal{O}(\log(1/\varepsilon))$, or is this optimal for gradient approximation?

2. **Second-moment quantum Adam with qubit efficiency.** Full quantum Adam requires computing $|\nabla_l^2\rangle$, which needs a non-linear squaring operation. Section J proposes two approaches: QSVT squaring (higher circuit depth, no measurement) and measure-then-reencode (lower depth, reintroduces measurement). Neither is hardware-independently optimal. Can second-moment estimation be achieved with circuit depth sub-linear in $d' = \mathcal{O}(d)$ while avoiding mid-circuit measurement?

3. **Adaptive Pauli rank scheduling (Pauli-propagation mode, Section L).** Proposition H.2 shows that entanglement growth during Phase II training increases the required $k_{\max}$ in Pauli-propagation mode. Can $k_{\max}$ be adapted online—increased during Phase II as entanglement grows, and stabilised during Phase III as parameters converge—with provable guarantees on the truncation error over the full training trajectory?

4. **Formal barren-plateau recovery convergence.** Our plateau recovery (random perturbation) is heuristic. Proposition H.3 shows all local minima are global in the overparameterised regime, so barren plateaus are *flat* saddle points (degenerate Hessian). Lee et al. (2016) guarantees escape with probability 1 from random initialisation, but gives no efficiency bound. Can *efficient* saddle-point escape be proven in the quantum setting? Note that standard perturbed gradient descent (Jin et al., 2017) requires strict saddle points (strictly negative Hessian eigenvalue) and does not directly apply to the flat saddle points arising from barren plateaus.

5. **Quantum NTK spectrum.** The QNTK K_Q governs Phase III dynamics (Proposition I.1), with convergence determined by alignment between K_Q and H_{target}. Classical NTK theory characterises the kernel's eigenvalue distribution and spectral gap to bound convergence rates. What is the eigenvalue distribution of K_Q at Phase III initialisation (near-converged parameters)? Does the alignment $\text{tr}(K_Q H_{\text{target}})$ admit a spectral lower bound analogous to the classical NTK spectral gap?

6. **Hardware-efficient QSVT polynomial construction.** Current QSVT phase-finding outputs $\mathcal{O}(d)$ phase angles (Gilyén et al., 2019), whose high-precision classical computation becomes numerically challenging for large d (Haah, 2019). Setting $d = 2p_{\max} = \mathcal{O}(n)$ (from Theorem F.1) incurs a gate cost of $\mathcal{O}(n \cdot L^2 \cdot d_a \cdot D)$ per iteration in circuit-wrapping mode (or $\mathcal{O}(Ln^2 k_{\max})$ in Pauli-propagation mode; see Section L), with the quadratic n^2

in the latter arising from the $\mathcal{O}(n)$ QSVT queries each of depth $\mathcal{O}(n)$. Is there a hardware-efficient QSVT construction that reduces this to $o(n^2)$ gates per iteration while preserving the gradient subspace guarantees of Theorem F.1?

7. **End-to-end fault-tolerant implementation.** Our analysis counts two-qubit gates and ignores fault-tolerant overheads (magic state distillation, syndrome measurement). The paper's numerical examples reach $n \leq 12$, $L \leq 5$, $\varepsilon = 10^{-3}$ (Section D). A complete fault-tolerant resource estimate extrapolating to larger scales (e.g., $n = 50$, $L = 10$, $\varepsilon = 10^{-3}$, the early fault-tolerant regime) would quantify the timeline to practical quantum advantage.

L Pauli-Propagation Mode: Theory and Analysis

This appendix collects the full theoretical treatment of the *Pauli-propagation mode* introduced in Remark 4.1. Circuit wrapping (Section 4, Theorem 5.1) is the primary method of this paper; Pauli-propagation mode is an alternative that is approximate but avoids the L^2 depth scaling by instead incurring a $k_{\max}$-dependent classical cost.

L.1 Scope and Comparison

In circuit-wrapping mode, the Heisenberg operator $H^{(l)}$ is never materialised explicitly: the block-encoding $U_{H^{(l)}}$ is grown by circuit composition and queried only during QSVT gradient extraction, at a total cost of $\mathcal{O}(d \cdot L^2 \cdot d_a \cdot D + L \cdot d_a \cdot n)$ two-qubit gates with *no truncation error*.

In Pauli-propagation mode, $H^{(l)}$ is instead maintained as an explicit classical Pauli sum $H^{(l)} = \sum_i \alpha_i^{(l)} P_i$ that is truncated to the top-$k_{\max}$ terms (by $|\alpha_i|$) after each backward step. The truncated sum is then block-encoded for QSVT gradient extraction, at a per-layer cost of $\mathcal{O}(d \cdot k_{\max} \cdot n)$ gates. Summing over L layers gives $\mathcal{O}(d \cdot L \cdot k_{\max} \cdot n)$ total gates—eliminating the L^2 factor—but introducing truncation error.

L.2 Certified Truncation-Error Theorem

Scope. The following theorem applies exclusively to the *Pauli-propagation mode* of Remark 4.1. It does not apply to circuit-wrapping mode, which is exact.

Theorem L.1 Truncation error bound (Pauli-propagation mode)

Run Algorithm 1 in Pauli-propagation mode. At each layer l, truncate $H^{(l)}$ to the top-k_l Pauli terms (by $|\alpha_i|$) with discarded ℓ^1 weight $\varepsilon_{\text{tol}}^{(l)} = \sum_{i>k_l} |\alpha_i|$. Let $\varepsilon_{\text{tol}} = \max_{l'} \varepsilon_{\text{tol}}^{(l')}$. Then the per-layer gradient error satisfies

$$\|\widetilde{\nabla}_l - \nabla_l\|_\infty \leq 2 \sum_{l'=l}^{L} \varepsilon_{\text{tol}}^{(l')} \leq 2(L - l + 1)\, \varepsilon_{\text{tol}},$$

and the global ℓ^∞ gradient error satisfies

$$\|\widetilde{\nabla} - \nabla\|_\infty \leq 2L\, \varepsilon_{\text{tol}} = 2L \max_l \varepsilon_{\text{tol}}^{(l)}.$$

Setting $\varepsilon_{\text{tol}}^{(l)} = \varepsilon/(2L)$ for all l certifies ε-accurate gradients. For Haar-typical circuits, the required rank satisfies $k_l \leq \mathcal{O}(n \log(L/\varepsilon)/\log n)$.

Proof: $\|R^{(l)}\| \leq \varepsilon_{\text{tol}}^{(l)}$ (Pauli $\ell^1 \geq$ operator norm); unitary invariance propagates each truncation error unchanged, giving accumulated operator error $\|\widetilde{H}^{(l)} - H^{(l)}\| \leq \sum_{l'=l}^{L} \varepsilon_{\text{tol}}^{(l')}$; the gradient operator $G^{(l)} = A^\dagger H^{(l)} U + U^\dagger H^{(l)} A$ has two terms in $H^{(l)}$, each perturbed by $\leq \|\widetilde{H}^{(l)} - H^{(l)}\|$, giving the factor 2 in the gradient bound; the global ℓ^∞ bound follows by taking the max over l. Full proof in Section M.2.

L.3 Convergence in Pauli-Propagation Mode

Under assumptions (A1)–(A3) and the Pauli-truncation assumption:

(A5) Pauli truncation: $\|\widetilde{\nabla}\mathcal{L} - \nabla\mathcal{L}\|_\infty \le 2L\,\varepsilon_{\text{tol}}$ (certified by Theorem L.1 above). Converting via $\|\cdot\| \le \sqrt{p}\,\|\cdot\|_\infty$ yields $\varepsilon_{\text{approx}} = \sqrt{p}\cdot 2L\,\varepsilon_{\text{tol}}$.

Substituting $\varepsilon_{\text{approx}} = \sqrt{p}\cdot 2L\,\varepsilon_{\text{tol}}$ into Theorem 5.2 (in place of assumption (A4)) gives an identical convergence rate—$\mathcal{O}(1/\varepsilon_{\text{conv}}^4)$ iterations to an $\varepsilon_{\text{conv}}$-approximate critical point—with the irreducible floor now controlled by ε_{tol} rather than QSVT degree d. Setting $\varepsilon_{\text{tol}} = \varepsilon_{\text{conv}}/(2L\sqrt{p})$ certifies convergence with the same rate as circuit-wrapping mode.

L.4 Mode Comparison Summary

Table 8: Circuit-wrapping mode vs. Pauli-propagation mode. Circuit wrapping is the primary method; Pauli-propagation mode is an approximate alternative suited to regimes where block-encoding depth is the limiting constraint.

Property	**Circuit wrapping**	**Pauli-propagation**
Truncation error	None (exact)	$\le 2L\varepsilon_{\text{tol}}$
Gate cost	$\mathcal{O}(dL^2 d_a D + Ld_a n)$	$\mathcal{O}(dLk_{\max} n)$
Classical cost	None	$\mathcal{O}(Lk_{\max})$ per step
Requires $k_{\max}$	No	Yes
Scales with entanglement	No	Yes (via $k_{\max}$)
Convergence guarantee	Theorem 5.2 (A4)	Same rate (A5)

Why circuit wrapping does not scale with entanglement. In Pauli-propagation mode the loss operator $H^{(l)}$ is stored as an explicit Pauli sum. Conjugating by a generic (non-Clifford) layer unitary $U^{(l)}(\theta_l)$ spreads each Pauli string P_i into a linear combination of exponentially many new strings, so the Pauli rank of $H^{(l)}$ grows with each backward step. By Proposition H.2, this growth is tightly coupled to the von Neumann entanglement entropy S_l of the forward-pass state: the effective Pauli rank scales as $k_{\text{eff}}^{(l)} = \mathcal{O}(2^{2S_l/\log 2})$, which becomes exponential in the system size for highly entangled states. Maintaining a fixed truncation budget $k_{\max}$ therefore introduces a truncation error ε_{tol} that *increases* as entanglement grows, requiring $k_{\max}$ itself to grow to preserve accuracy.

Circuit wrapping sidesteps this entirely. Rather than materialising $H^{(l)}$ as a Pauli sum, it represents $H^{(l)}$ implicitly as a *quantum circuit*: the block-encoding $U_{H^{(l)}}$ is formed by composing the circuit for $U_{H^{(l+1)}}$ with the layer unitaries $(A^{(l+1)}U^{(l+1)})$ and their adjoints (Lemma M.1). This composition is a purely classical circuit-description operation—no quantum gates are run and no operator is evaluated—and its cost is measured in *circuit depth*, not Pauli rank. The key observation is that circuit depth is *insensitive to entanglement*: composing two unitary circuits of depths d_1 and d_2 always produces a circuit of depth $d_1 + d_2$, regardless of how much entanglement the circuits generate or how large the resulting operator's Pauli expansion would be. Concretely, each backward step through one layer adds exactly $\mathcal{O}(d_a \cdot D_l)$ to the block-encoding depth (Lemma M.1), independent of S_l, so after L backward steps the total accumulated depth is $\mathcal{O}(L \cdot d_a \cdot D)$—a quantity that depends only on the circuit architecture, not on the entanglement structure of the states the circuit acts on. Quantum gates are spent only when this nested block-encoding is *queried* during QSVT gradient extraction, at which point the entanglement present in the state is handled natively by the quantum hardware without any classical representation of the operator being required.

Table 9: Summary of theoretical contributions. The truncation-error result (Pauli-propagation mode) is deferred to Section L.

Result	Statement (informal)	Key assumption	Implication
Theorem 5.1	$\mathcal{O}(d \cdot L^2 \cdot d_a \cdot D + L \cdot d_a \cdot n)$ gates/iter; $\mathcal{O}(d \cdot L_{\text{dirty}} \cdot L \cdot d_a \cdot D)$ cached	Circuit depth D, QSVT degrees d, d_a	Exp. saving vs. $\mathcal{O}(L \cdot 2^n)$ simulation; no $k_{\max}$ factor
Theorem 5.2	$\mathcal{O}(1/\sqrt{T})$ rate; $\mathcal{O}(1/\varepsilon^4)$ iters	ℓ_s-smooth + bounded noise (circuit-wrapping mode)	Matches classical Adam up to QSVT bias ε^2
Theorem 5.3	Sound & complete detection in W_{var} steps	$G \le \sqrt{2/3}$ (completeness only; soundness unconditional)	First formal plateau-detection guarantee
Section L (Appendix)	Trunc. error $\le 2L \cdot \varepsilon_{\text{tol}}$; certified per-layer	Pauli ℓ^1 norm (Pauli-propagation mode only)	First rigorous accuracy guarantee for Pauli pruning

M Extended Proofs

Proposition M.1 Operator cache consistency

Let $\delta_{\text{cache}} > 0$ be the cache invalidation threshold. If the parameter change satisfies $\|\Delta\theta_l\| \le \delta_{\text{cache}}$ for layer l, then the cached Heisenberg operator $H^{(l)}_{\text{cache}}$ satisfies

$$\left\| H^{(l)}_{\text{cache}} - H^{(l)} \right\| \le \delta_{\text{cache}} \cdot \ell_H,$$

where ℓ_H is the Lipschitz constant of the Heisenberg propagation map with respect to θ_l. Hence, cache hits introduce an operator error bounded by $\delta_{\text{cache}} \cdot \ell_H$, which is absorbed into the truncation budget ε_{tol} provided $\delta_{\text{cache}} \le \varepsilon_{\text{tol}}/\ell_H$.

M.1 Proof of Theorem 5.1 (Gate Complexity)

We first state the two supporting lemmas and then give the full derivation.

Lemma M.1 Heisenberg propagation cost (circuit wrapping)

Propagating the loss operator $H^{(l)}$ through one activation–weight layer pair $(A^{(l)}, U^{(l)})$ in the Heisenberg picture via quantum circuit wrapping augments the block-encoding circuit depth by $\mathcal{O}(d_a \cdot D_l)$, where d_a is the QSVT activation degree and $D_l = \text{depth}(U^{(l)})$. Concretely, the updated block-encoding is

$$U_{H^{(l-1)}} \;=\; \left(A^{(l)} U^{(l)} \otimes I_{\text{anc}}\right)^\dagger U_{H^{(l)}} \left(A^{(l)} U^{(l)} \otimes I_{\text{anc}}\right),$$

requiring no Pauli-string enumeration or truncation. After L backward steps the accumulated block-encoding depth is $\mathcal{O}(L \cdot d_a \cdot D)$.

Lemma M.2 QSVT gradient encoding cost

Block-encoding the gradient operator $G^{(l)}_\theta$ (a Pauli sum with k_l terms) and applying a degree-d QSVT polynomial costs $\mathcal{O}(d \cdot k_l \cdot n)$ two-qubit gates on the n-qubit system.

We give the full derivation combining Lemma M.1 and Lemma M.2.

Proof. For each layer l, the backward pass performs four operations:

(a) Heisenberg propagation via circuit wrapping. By Lemma M.1, each backward step augments the nested block-encoding depth by $\mathcal{O}(d_a \cdot D_l)$, with no Pauli enumeration. After the backward pass,

the block-encoding of $H^{(l)}$ (loss operator propagated through layers $l{+}1, \ldots, L$) has accumulated depth $\mathcal{O}((L-l)\cdot d_a \cdot D)$. The gradient block-encoding $G_\theta^{(l)}$ additionally wraps around layer l's own activation $A^{(l)}$ (contributing $\mathcal{O}(d_a \cdot D_l)$), so its total depth is $\mathcal{O}((L-l+1)\cdot d_a \cdot D)$. Applying QSVT of degree d to $G_\theta^{(l)}$ makes $\mathcal{O}(d)$ queries to this oracle, giving $\mathcal{O}\big(d\cdot(L-l+1)\cdot d_a\cdot D\big)$ two-qubit gates. Summing over $l=1,\ldots,L$ and bounding $D_l \le D$:

$$\sum_{l=1}^{L} \mathcal{O}\big(d\cdot(L-l+1)\cdot d_a\cdot D\big) \;=\; \mathcal{O}\big(d\cdot L^2\cdot d_a\cdot D\big).$$

(b) Activation forward pass. Applying the QSVT activation unitary $A^{(l)}$ of degree d_a to the n-qubit register costs $\mathcal{O}(d_a\cdot n)$ gates per layer, giving $\mathcal{O}(L\cdot d_a\cdot n)$ total.

(c) QSVT gradient encoding. In the circuit-wrapping model, $G_\theta^{(l)}$ is block-encoded via the same nested structure as $H^{(l-1)}$, with no Pauli-string enumeration. $H^{(l-1)}$ has been propagated through layers $l,\ldots,L$, giving block-encoding depth $\mathcal{O}((L-l+1)\cdot d_a\cdot D)$. Applying QSVT of degree d costs $\mathcal{O}(d)$ queries to this block-encoding, each of depth $\mathcal{O}((L-l+1)\cdot d_a\cdot D)$; this cost is $\mathcal{O}(d\cdot(L-l+1)\cdot d_a\cdot D)$ per layer and is already subsumed in term (a) when the sum is taken over all layers.

(d) Coherent gradient write. Applying $U_\nabla^{(l)}$ controlled on $|\psi^{(l-1)}\rangle$ adds $\mathcal{O}(n)$ ancilla gates per layer, giving $\mathcal{O}(L\cdot n)$ total. Since $d_a \ge 1$, this is dominated by term (b).

(e) Parameter update. Each coherent update $|\theta_l\rangle|m_l\rangle \mapsto |\theta_l - \alpha_t m_l\rangle|m_l\rangle$ (line 32) costs $\mathcal{O}(p_l)$ two-qubit gates per layer. For a standard Ansätz with $p_l = \mathcal{O}(d_a\cdot n)$ parameters per layer, the total $\mathcal{O}(p) = \mathcal{O}(L\cdot d_a\cdot n)$ is dominated by term (b).

Combining (a)–(e) and using the two-term structure of Theorem 5.1:

$$\mathcal{O}\big(d\cdot L^2\cdot d_a\cdot D + L\cdot d_a\cdot n\big).$$

For the cached variant, a layer that satisfies $\|\theta_l^t - \theta_l^{t-1}\| \le \delta_{\text{cache}}$ reuses its stored nested block-encoding; the effective contribution to term (a) is $\mathcal{O}(d\cdot L_{\text{dirty}}\cdot L\cdot d_a\cdot D)$. Terms (b)–(e) are paid by all L layers regardless of caching. □

M.2 Proof of Theorem L.1 (Truncation Error)

Proof. We provide a more detailed argument than the main-text sketch.

Step 1: Operator norm of the residual. Let $H^{(l)} = \widehat{H}^{(l)} + R^{(l)}$, where $\widehat{H}^{(l)} = \sum_{i=1}^{k_l}\alpha_i P_i$ (retained terms) and $R^{(l)} = \sum_{i=k_l+1}^{4^n}\alpha_i P_i$ (discarded terms). Since each Pauli P_i satisfies $\|P_i\| = 1$, the triangle inequality gives:

$$\|R^{(l)}\| \;\le\; \sum_{i>k_l}|\alpha_i|\|P_i\| \;=\; \sum_{i>k_l}|\alpha_i| \;=\; \varepsilon_{\text{trunc}} \;\le\; \varepsilon_{\text{tol}}, \tag{16}$$

where the first inequality is the Pauli ℓ^1-norm bound $\|\sum_i \alpha_i P_i\| \le \sum_i |\alpha_i|\|P_i\|$ (operator norm sub-additivity combined with $\|P_i\| = 1$).

Step 2: Gradient operator error. The full weight-gradient operator for layer l is

$$G^{(l)} = \underbrace{\frac{\partial (U^{(l)})^\dagger}{\partial\theta_l} H^{(l)} U^{(l)}}_{\text{term 1}} + \underbrace{(U^{(l)})^\dagger H^{(l)} \frac{\partial U^{(l)}}{\partial\theta_l}}_{\text{term 2}}, \tag{17}$$

which is Hermitian and contains $H^{(l)}$ in both terms. Let $A^{(l)} \equiv \partial(U^{(l)})^\dagger/\partial\theta_l$. For general parametrised gates, $\|A^{(l)}\| \le 1$ (Schuld et al., 2019). (For the specific case of Pauli rotation gates $U^{(l)} = e^{-i\theta_l P/2}$, one has $\|A^{(l)}\| = 1/2$, giving a tighter bound of ε_{tol} rather than $2\varepsilon_{\text{tol}}$; we use the general bound for uniformity.) Replacing $H^{(l)}$ with $\widehat{H}^{(l)} = H^{(l)} - R^{(l)}$ perturbs each term by at most $\|A^{(l)}\|\cdot\|R^{(l)}\|\cdot\|U^{(l)}\| \le \varepsilon_{\text{tol}}$. By submultiplicativity of the operator norm and

$\|U^{(l)}\| = \|(U^{(l)})^\dagger\| = 1$ (unitary gates are isometries):

$$\|\widetilde{G}^{(l)} - G^{(l)}\| \le \underbrace{\|A^{(l)}\| \cdot \|-R^{(l)}\| \cdot \|U^{(l)}\|}_{\le \varepsilon_{\text{tol}}} + \underbrace{\|(U^{(l)})^\dagger\| \cdot \|-R^{(l)}\| \cdot \left\|\frac{\partial U^{(l)}}{\partial \theta_l}\right\|}_{\le \varepsilon_{\text{tol}}}$$
$$\le \varepsilon_{\text{tol}} + \varepsilon_{\text{tol}} = 2\varepsilon_{\text{tol}}, \tag{18}$$

where $\left\|\partial U^{(l)}/\partial\theta_l\right\| = \|A^{(l)}\| \le 1$ follows from differentiating $U^\dagger U = I$: $A^{(l)}U^{(l)} + (U^{(l)})^\dagger \partial U^{(l)}/\partial\theta_l = 0$, giving $\partial U^{(l)}/\partial\theta_l = -U^{(l)}A^{(l)}U^{(l)}$ and hence $\|\partial U^{(l)}/\partial\theta_l\| \le \|U^{(l)}\|\|A^{(l)}\|\|U^{(l)}\| = \|A^{(l)}\| \le 1$.

Step 3: Per-component gradient error. For each parameter index j in layer l, define the gradient operator

$$G_j^{(l)} \equiv \frac{\partial (U^{(l)})^\dagger}{\partial (\theta_l)_j} H^{(l)} U^{(l)} + (U^{(l)})^\dagger H^{(l)} \frac{\partial U^{(l)}}{\partial (\theta_l)_j},$$

so that $(\nabla_l)_j = \langle\psi^{(l-1)}|G_j^{(l)}|\psi^{(l-1)}\rangle$. The bound from Step 2 (applied with $(\theta_l)_j$ in place of θ_l) gives $\|\widetilde{G}_j^{(l)} - G_j^{(l)}\| \le 2\varepsilon_{\text{tol}}$. Since $\||\psi^{(l-1)}\rangle\| = 1$ (normalised quantum state):

$$|(\widetilde{\nabla}_l)_j - (\nabla_l)_j| \le \|\widetilde{G}_j^{(l)} - G_j^{(l)}\| \cdot \||\psi^{(l-1)}\rangle\|^2 \le 2\varepsilon_{\text{tol}}, \tag{19}$$

uniformly in j, giving $\|\widetilde{\nabla}_l - \nabla_l\|_\infty \le 2\varepsilon_{\text{tol}}$, consistent with the per-layer bound stated in Theorem L.1.

Step 4: Accumulation over layers (sequential truncation). In Pauli-propagation mode the algorithm truncates $H^{(l)}$ to $\widehat{H}^{(l)}$ *before* propagating to $H^{(l-1)}$, so $\widehat{H}^{(l-1)} = (U^{(l)})^\dagger (A^{(l)})^\dagger \widehat{H}^{(l)} A^{(l)} U^{(l)}$ is computed from the already-truncated operator. Errors therefore accumulate across layers. Let $\delta_{l'} = \varepsilon_{\text{tol}}^{(l')} \le \varepsilon_{\text{tol}}$ denote the truncation error introduced at layer l'. Since unitary conjugation is an isometry (operator norm is unitarily invariant), each $\delta_{l'}$ propagates unchanged through all subsequent backward steps. The accumulated error in $\widetilde{H}^{(l)}$ relative to the exact $H^{(l)}$ is:

$$\|\widetilde{H}^{(l)} - H^{(l)}\| \le \sum_{l'=l}^{L} \delta_{l'} \le (L - l + 1)\,\varepsilon_{\text{tol}}. \tag{20}$$

Applying Step 2 with this accumulated operator error in place of ε_{tol}:

$$\|\widetilde{\nabla}_l - \nabla_l\|_\infty \le 2\,\|\widetilde{H}^{(l)} - H^{(l)}\| \le 2(L - l + 1)\,\varepsilon_{\text{tol}}. \tag{21}$$

Taking the ℓ^∞ norm over all layers (maximum, not sum):

$$\|\widetilde{\nabla} - \nabla\|_\infty = \max_l \|\widetilde{\nabla}_l - \nabla_l\|_\infty \le \max_l 2(L - l + 1)\,\varepsilon_{\text{tol}} = 2L\,\varepsilon_{\text{tol}}, \tag{22}$$

attained at $l = 1$. Setting $\varepsilon_{\text{tol}}^{(l)} = \varepsilon/(2L)$ for all l gives $\|\widetilde{\nabla} - \nabla\|_\infty \le \varepsilon$, matching the statement of Theorem L.1.

Step 5: Activation gradient term. For the activation gradient operator $G_{\text{act}}^{(l)} = \frac{\partial (A^{(l)})^\dagger}{\partial \phi_l} H^{(l)} A^{(l)} + (A^{(l)})^\dagger H^{(l)} \frac{\partial A^{(l)}}{\partial \phi_l}$, the same argument as Steps 2–3 applies with $A^{(l)}$ (the activation QSVT unitary) replacing $U^{(l)}$, giving $\|\widetilde{G}_{\text{act}}^{(l)} - G_{\text{act}}^{(l)}\| \le 2\varepsilon_{\text{tol}}$. An additional approximation error ε_a enters from the QSVT polynomial approximation of the activation function itself (not from Pauli truncation), giving the bound stated in the theorem: $\|\widetilde{\nabla}_{\text{act}} - \nabla_{\text{act}}\| \le 2\varepsilon_{\text{tol}} + \varepsilon_a$. □

Corollary M.1 Adaptive truncation schedule

Set $\varepsilon_{\text{tol}}^{(l)} = \varepsilon/(2L)$ for all layers l. Then the overall gradient error satisfies $\|\widetilde{\nabla} - \nabla\|_\infty \le \varepsilon$. To achieve $\varepsilon_{\text{tol}}^{(l)} = \varepsilon/(2L)$, it suffices to retain the top-k_l Pauli terms where $k_l = \mathcal{O}(n \log(L/\varepsilon)/\log n)$, so that the discarded tail has ℓ^1-norm at most $\varepsilon/(2L)$. The maximum over layers $k_{\max} = \max_l k_l$ is then $\mathcal{O}(n \log(L/\varepsilon)/\log n)$, which is sublinear in 4^n.

M.3 Proof of Theorem 5.2 (Convergence)

Proof. We adapt the analysis of Ghadimi & Lan (2013) to the quantum setting. Let $\widetilde{\nabla}_t$ denote the gradient estimate at step t, decomposed as $\widetilde{\nabla}_t = \nabla\mathcal{L}(W_t) + e_t + \xi_t$, where e_t is the deterministic gradient approximation error. Under (A4) (circuit-wrapping mode), $\|e_t\| \le \varepsilon_{\text{approx}}$ directly; in particular $e_t = 0$ when the polynomial activation is exact. Under (A5) (Pauli-propagation mode), Theorem L.1 gives $\|e_t\|_\infty \le 2L\varepsilon_{\text{tol}}$, and converting via $\|e_t\| \le \sqrt{p}\,\|e_t\|_\infty$ yields $\|e_t\| \le \varepsilon_{\text{approx}} := \sqrt{p}\cdot 2L\varepsilon_{\text{tol}}$. In both cases the proof proceeds with the Euclidean bound $\|e_t\| \le \varepsilon_{\text{approx}}$, and ξ_t is the stochastic Laplace noise ($\mathbb{E}[\xi_t] = 0$, $\mathbb{E}[\|\xi_t\|^2] \le \sigma^2$ by assumption A3).

One-step descent. By ℓ_s-smoothness (A1) and the update $W_{t+1} = W_t - \alpha_t m_t$, where $m_t = \beta m_{t-1} + (1-\beta)\widetilde{\nabla}_t$:

$$\mathcal{L}(W_{t+1}) \le \mathcal{L}(W_t) - \alpha_t\langle\nabla\mathcal{L}(W_t), m_t\rangle + \frac{\ell_s\alpha_t^2}{2}\|m_t\|^2. \tag{23}$$

Momentum residual decomposition. Write $m_t = \nabla\mathcal{L}(W_t) + \delta_t$, where $\delta_t := m_t - \nabla\mathcal{L}(W_t)$ is the momentum residual. By AM-GM ($\langle a, b\rangle \ge -\frac{1}{2}\|a\|^2 - \frac{1}{2}\|b\|^2$):

$$\langle\nabla\mathcal{L}(W_t), m_t\rangle = \|\nabla\mathcal{L}(W_t)\|^2 + \langle\nabla\mathcal{L}(W_t), \delta_t\rangle \ge \tfrac{1}{2}\|\nabla\mathcal{L}(W_t)\|^2 - \tfrac{1}{2}\|\delta_t\|^2. \tag{24}$$

Bounding the residual. From the momentum update and ℓ_s-smoothness ($\|\nabla\mathcal{L}(W_{t-1}) - \nabla\mathcal{L}(W_t)\| \le \ell_s\alpha_{t-1}\|m_{t-1}\|$):

$$\delta_t = \beta\delta_{t-1} + \beta\big(\nabla\mathcal{L}(W_{t-1}) - \nabla\mathcal{L}(W_t)\big) + (1-\beta)(e_t + \xi_t), \tag{25}$$

so, using $\|\nabla\mathcal{L}(W_{t-1}) - \nabla\mathcal{L}(W_t)\| \le \ell_s\alpha_{t-1}\|m_{t-1}\|$ and $\mathbb{E}[\|m_{t-1}\|^2] \le G^2 + \sigma^2$ (Variance bound below):

$$\mathbb{E}[\|\delta_t\|^2] \le 3\beta^2\mathbb{E}[\|\delta_{t-1}\|^2] + \underbrace{3\beta^2\ell_s^2\alpha_0^2(G^2+\sigma^2)}_{=:\,S_\alpha} + 3(1-\beta)^2(\varepsilon_{\text{approx}}^2 + \sigma^2). \tag{26}$$

The middle term S_α is a constant (not a function of δ), so the recursion is contractive with ratio $3\beta^2 < 1$ for $\beta < 1/\sqrt{3}$. Iterating to stationarity gives:

$$\mathbb{E}[\|\delta_t\|^2] \;\le\; \frac{S_\alpha + 3(1-\beta)^2(\varepsilon_{\text{approx}}^2 + \sigma^2)}{1 - 3\beta^2} \;=\; A_\beta\alpha_0^2(G^2+\sigma^2) + C_\beta(\varepsilon_{\text{approx}}^2 + \sigma^2), \tag{27}$$

with $A_\beta = \frac{3\beta^2\ell_s^2}{1-3\beta^2}$ and $C_\beta = \frac{3(1-\beta)^2}{1-3\beta^2}$. For $\alpha_0 = 1/\sqrt{T}$ the term $A_\beta\alpha_0^2(G^2+\sigma^2) = O(1/T)$ is lower-order, giving $\mathbb{E}[\|\delta_t\|^2] \le C_\beta(\varepsilon_{\text{approx}}^2 + \sigma^2) + O(1/T)$.

Variance bound. By convexity of $\|\cdot\|^2$ applied to $m_t = (1-\beta)\sum_s \beta^{t-s}\widetilde{\nabla}_s$: $\mathbb{E}[\|m_t\|^2] \le (1-\beta)\sum_s \beta^{t-s}\mathbb{E}[\|\widetilde{\nabla}_s\|^2] \le G^2 + \sigma^2$ (using $\|\nabla\mathcal{L}\| \le G$ from A2 and $\mathbb{E}[\|\xi\|^2] \le \sigma^2$ from A3, with $\varepsilon_{\text{approx}} \ll G$ absorbed into the G^2 term).

Telescoping. Substituting the residual bound into the one-step descent, taking expectations, and summing over $t = 1, \dots, T$:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla\mathcal{L}(W_t)\|^2] \le \frac{2(\mathcal{L}(W_1) - \mathcal{L}^*)}{\sum_t \alpha_t} + \frac{\ell_s\sum_t \alpha_t^2(G^2+\sigma^2)}{\sum_t \alpha_t} + C_\beta(\varepsilon_{\text{approx}}^2 + \sigma^2). \tag{28}$$

The $C_\beta(\varepsilon_{\text{approx}}^2 + \sigma^2)$ term is the irreducible floor from the momentum residual; the σ^2 portion vanishes in the noiseless limit and is absorbed into the $O(1/\sqrt{T})$ term for $\sigma = O(\varepsilon_{\text{approx}})$.

Learning rate schedule. The cosine schedule satisfies $\sum_{t=1}^{T}\alpha_t = \Theta(\alpha_0 T)$ and $\sum_{t=1}^{T}\alpha_t^2 = \mathcal{O}(\alpha_0^2 T)$. (The warmup phase contributes $\mathcal{O}(\alpha_0 T_w)$ to $\sum\alpha_t$, which is lower-order for $T_w = o(T)$.) Substituting and setting $\alpha_0 = 1/\sqrt{T}$:

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}[\|\nabla\mathcal{L}(W_t)\|^2] \le \mathcal{O}\left(\frac{\mathcal{L}(W_1) - \mathcal{L}^*}{\sqrt{T}} + \frac{\ell_s(G^2+\sigma^2)}{\sqrt{T}} + \varepsilon_{\text{approx}}^2 + \sigma^2\right). \tag{29}$$

For the right-hand side to be at most $\varepsilon_{\text{conv}}^2$, it suffices to choose $T = \mathcal{O}(1/\varepsilon_{\text{conv}}^4)$, $\varepsilon_{\text{approx}} \le \varepsilon_{\text{conv}}$, and $\sigma \le \varepsilon_{\text{conv}}$. (For $\sigma = \mathcal{O}(\varepsilon_{\text{approx}})$ the σ^2 floor is absorbed into $\varepsilon_{\text{approx}}^2$.) □

M.4 Proof of Theorem 5.3 (Plateau Detection)

Proof. Write $W := W_{\text{var}}$ for brevity. Let $Z_t = \|\nabla_l(W_t)\|^2$ and define the *unbiased* empirical variance estimator

$$\hat{V}_T = \frac{1}{W-1}\sum_{t=T-W+1}^{T}\left(Z_t - \bar{Z}\right)^2 = \frac{W}{W-1}\left(\overline{Z^2} - \bar{Z}^2\right) = \frac{1}{W(W-1)}\sum_{i<j}(Z_i - Z_j)^2, \quad (30)$$

where $\bar{Z} = W^{-1}\sum_t Z_t$ and $\overline{Z^2} = W^{-1}\sum_t Z_t^2$, and the last equality is the standard pairwise identity. Since $Z_t \in [0, G^2]$ (by A2) and the window samples are i.i.d. (or sub-exponentially mixing), $\mathbb{E}[\hat{V}_T] = \mathrm{Var}(Z_t)$ exactly for all $W \geq 2$.

Soundness. Suppose W_t is *not* in a (δ, ρ)-barren-plateau region, so $\mathrm{Var}(Z_t) > \delta$. Set $\varepsilon_{\text{plateau}} = \delta$; the monitor fires incorrectly when $\hat{V}_T < \delta$. Since $\mathbb{E}[\hat{V}_T] = \mathrm{Var}(Z_t) > \delta$ exactly, the threshold δ is strictly below the mean for every $W \geq 2$. Apply McDiarmid's bounded-difference inequality to $\hat{V}_T$ as a function of the i.i.d. samples $Z_1, \ldots, Z_W \in [0, G^2]$. We use the pairwise representation to bound the sensitivity. Replacing Z_i by $Z_i' \in [0, G^2]$ affects exactly the $(W-1)$ pairs (i, j) with $j \neq i$. For each such pair,

$$\left|(Z_i' - Z_j)^2 - (Z_i - Z_j)^2\right| = |Z_i' - Z_i| \cdot |Z_i' + Z_i - 2Z_j| \leq G^2 \cdot 2G^2 = 2G^4,$$

so the total change satisfies

$$|\hat{V}_T' - \hat{V}_T| \leq \frac{(W-1)\cdot 2G^4}{W(W-1)} = \frac{2G^4}{W} =: c_t.$$

Hence $\sum_t c_t^2 = W \cdot (2G^4/W)^2 = 4G^8/W$. With $\mathbb{E}[\hat{V}_T] = \mathrm{Var}(Z_t) = \gamma + \delta$:

$$\Pr(\hat{V}_T < \delta) \leq \Pr\left(\hat{V}_T \leq \mathbb{E}[\hat{V}_T] - \gamma\right) \leq \exp\left(-\frac{2\gamma^2}{4G^8/W}\right) = \exp\left(-\frac{W\gamma^2}{2G^8}\right), \quad (31)$$

where $\gamma = \mathrm{Var}(Z_t) - \delta > 0$ is the non-plateau gap. In particular, if $\mathrm{Var}(Z_t) \geq \delta + G^2\sqrt{\delta}$ (i.e. $\gamma \geq G^2\sqrt{\delta}$) then $\gamma^2 \geq G^4\delta$, so $\gamma^2/G^8 \geq \delta/G^4$, recovering the simpler form $\exp(-W\delta/(2G^4))$ as a sufficient-condition bound.

Bridge: Haar variance implies small gradient norm. By Definition 5.1, $\mathrm{Var}_W[\partial\mathcal{L}/\partial\theta_{l,k}] \leq \delta/p_l$ for every parameter k of layer l, where the variance is over Haar-random perturbations of W_t within radius ρ. We claim $\mathbb{E}_W[\partial\mathcal{L}/\partial\theta_{l,k}] = 0$. We assume that W_t is such that the perturbation measure on the ρ-ball centred at W_t is symmetric under $\theta_{l,k} \mapsto -\theta_{l,k}$ (this holds, for example, when W_t is itself Haar-distributed, as at random initialisation, since conjugation by the Pauli generator G_k is a symmetry of the Haar measure). Under this symmetry the parameter shift rule gives

$$\frac{\partial\mathcal{L}}{\partial\theta_{l,k}} = \tfrac{1}{2}\left(\mathcal{L}\left(\theta_{l,k} + \tfrac{\pi}{2}\right) - \mathcal{L}\left(\theta_{l,k} - \tfrac{\pi}{2}\right)\right),$$

and the Haar averages of the two terms are equal, so the difference vanishes. Therefore $\mathrm{Var}_W[\partial\mathcal{L}/\partial\theta_{l,k}] = \mathbb{E}_W[(\partial\mathcal{L}/\partial\theta_{l,k})^2] \leq \delta/p_l$, and summing over all p_l parameters of layer l:

$$\mathbb{E}_W\left[\|\nabla_l(W)\|^2\right] = \sum_{k=1}^{p_l}\mathbb{E}_W\left[\left(\frac{\partial\mathcal{L}}{\partial\theta_{l,k}}\right)^2\right] \leq p_l \cdot \frac{\delta}{p_l} = \delta.$$

Along the training trajectory, W_t lies in the same ρ-ball, so $\mathbb{E}[Z_t] = \mathbb{E}[\|\nabla_l(W_t)\|^2] \leq \delta$ for all t in the window.

Completeness. In a (δ, ρ)-barren-plateau region, $\mathbb{E}[Z_t] \leq \delta$ for all t in the window. Concretely, by assumption W_t itself lies inside the ρ-ball on which the Haar-variance bound $\mathbb{E}_W[\|\nabla_l(W)\|^2] \leq \delta$ was established in the Bridge above, so the bound applies to each iterate in the window directly. We assume $G \leq \sqrt{2/5}$ (so $G^2 \leq \frac{2}{5} < \frac{1}{2}$), which lies within the normalised gradient regime $G < 1$. Since $Z_t \leq G^2$ pathwise, $Z_t^2 \leq G^2 Z_t$, so averaging gives $\overline{Z^2} \leq G^2\bar{Z}$ and hence $\overline{Z^2} - \bar{Z}^2 \leq G^2\bar{Z}$. Using $\hat{V}_T = \frac{W}{W-1}(\overline{Z^2} - \bar{Z}^2)$ and $\frac{W}{W-1} \leq 2$ for $W \geq 2$:

$$\hat{V}_T \leq \frac{W}{W-1}\,G^2\bar{Z} \leq 2G^2\bar{Z}. \quad (32)$$

For the monitor to fire we need $\hat{V}_T < \delta$; it suffices to have $\bar{Z} < \delta/(2G^2)$. Set $\mu^* = \delta/(2G^2)$. Since $G^2 < \frac{1}{2}$ we have $\mu^* > \delta \geq \mathbb{E}[Z_t] = \mathbb{E}[\bar{Z}]$, so μ^* is strictly above the mean. By Hoeffding's inequality for $Z_t \in [0, G^2]$:

$$\Pr\big(\bar{Z} \geq \mu^*\big) \leq \exp\left(-\frac{2W\big(\mu^* - \mathbb{E}[\bar{Z}]\big)^2}{G^4}\right) \leq \exp\left(-\frac{2W(\mu^* - \delta)^2}{G^4}\right) = \exp\left(-\frac{W\delta^2(1-2G^2)^2}{2G^8}\right), \tag{33}$$

where the second inequality uses $\mathbb{E}[\bar{Z}] \leq \delta$ (so $\mu^* - \mathbb{E}[\bar{Z}] \geq \mu^* - \delta > 0$). With probability $\geq 1 - \exp(-W\delta^2(1-2G^2)^2/(2G^8))$, we have $\bar{Z} < \mu^*$ and hence

$$\hat{V}_T \leq 2G^2\,\bar{Z} < 2G^2 \cdot \frac{\delta}{2G^2} = \delta = \varepsilon_{\text{plateau}},$$

so the monitor fires correctly. The stated bound $1-\exp(-W\delta^2/(8G^4))$ follows by verifying that the exact exponent satisfies $\delta^2(1-2G^2)^2/(2G^8) \geq \delta^2/(8G^4)$, i.e. $4(1-2G^2)^2 \geq G^4$. Taking positive square roots (valid since $G^2 \leq \frac{2}{5}$ implies $1 - 2G^2 \geq \frac{1}{5} > 0$) this becomes $2(1-2G^2) \geq G^2$, equivalently $G^2 \leq 2/5$. This condition holds exactly throughout the assumed regime $G \leq \sqrt{2/5}$, confirming the simplified bound. Note that the wider regime $G < 1/\sqrt{2}$ does *not* suffice: for $G \in (\sqrt{2/5}, 1/\sqrt{2})$ the inequality $4(1-2G^2)^2 \geq G^4$ fails, so the stated exponent $\delta^2/(8G^4)$ would underestimate the true exponent and the probability bound would be invalid. □

N Additional Proofs

N.1 Proof of Theorem F.1 (Gradient Subspace Dimension)

Proof. The exact gradient ∇_l is linear in $\{\partial U^{(l)}/\partial\theta_{l,i}\}_{i=1}^{p_l}$, so it lives in a linear subspace $\mathcal{V}_l^{\text{exact}}$ of dimension *at most* p_l inside the 4^n-dimensional operator space (spanned by the 4^n Pauli strings on n qubits); the independence argument below establishes that this dimension equals p_l under the distinct-qubit assumption.

QSVT degree condition. The QSVT polynomial P of degree d acts on the singular values of the gradient operator $G^{(l)}$. For a generic PQC, the p_l non-zero singular values of $G^{(l)}$ are separated from zero: each arises from a distinct Pauli-rotation generator $\partial U^{(l)}/\partial\theta_{l,i} = \frac{i}{2}P_iU^{(l)}$, and for generic parameters the resulting singular values are non-degenerate and bounded away from zero by a constant depending only on $\|H\|$ and the circuit architecture (this fails only on a measure-zero set of parameters). With this spectral gap, the signal subspace (singular values $\geq \gamma > 0$) and the noise subspace (singular value 0) are spectrally separated. For subspace preservation it suffices that P is non-zero on each of the p_l signal singular values; an odd polynomial of degree $d \geq 1$ achieves this generically. The stronger condition $d \geq 2p_l$—which gives $\lceil d/2 \rceil \geq p_l$ free coefficients and allows exact interpolation of the identity on all p_l signal singular values—is required for accurate gradient *values* (not merely for subspace preservation), and is the operationally useful guarantee. Under either condition, the QSVT-approximated gradient spans the same p_l-dimensional subspace as the exact gradient, and $\dim(\mathcal{V}_l) \geq p_l$ (subject to the ambient dimension 4^n).

Pauli-truncation constraint (Pauli-propagation mode). Truncating to $k_{\max}$ Pauli terms projects H_l^{back} onto a $k_{\max}$-dimensional operator subspace. Under the weight-1 Pauli-support assumption (established below), this clips $\dim(\mathcal{V}_l)$ to at most $k_{\max}$; in general the joint span of the p_l gradient operators can be as large as $\min(p_l \cdot k_{\max}, 4^n)$. In either case, for $k_{\max} \geq p_l$ the gradient subspace is unaffected. To see this, work in the Heisenberg picture. For single-qubit Pauli-rotation gates $U^{(l)} = e^{i\theta_{l,i}P_i/2}$, differentiating through the circuit gives (Schuld et al., 2019)

$$\frac{\partial\mathcal{L}}{\partial\theta_{l,i}} = -\frac{i}{2}\,\text{tr}\big([P_i,\, H_l^{\text{back}}]\,\rho_l\big),$$

where $U_{\text{after},l} = U_{\text{tail}}\,U_{l,i}^{(l)}$ denotes the sub-circuit from gate l,i onwards (i.e. gate l,i itself followed by all subsequent layers), $H_l^{\text{back}} = U_{\text{after},l}^\dagger H\,U_{\text{after},l}$ is the Hamiltonian backpropagated through gate l,i and all later layers, and $\rho_l = U_{\text{before},l}\,\rho_0\,U_{\text{before},l}^\dagger$ is the state immediately before gate l,i. The gradient direction in operator space is therefore $[P_i, H_l^{\text{back}}]$, whose Pauli weight is at most

$\mathrm{wt}(H_l^{\mathrm{back}}) + 1$: each surviving string $P_i Q_k$ has the same weight as Q_k when P_i acts on a qubit already occupied by Q_k, and weight $\mathrm{wt}(Q_k) + 1$ when P_i acts on a qubit where $Q_k = I$. In Pauli-propagation mode, H_l^{back} is itself represented as a truncated Pauli sum with at most $k_{\max}$ terms; writing $H_l^{\mathrm{back}} = \sum_k c_k Q_k$, the commutator evaluates as

$$[P_i, H_l^{\mathrm{back}}] = 2 \sum_{\substack{k:\ P_i \text{ anti-commutes} \\ \text{with } Q_k}} c_k \, (P_i Q_k),$$

where each surviving term is the *new* Pauli string $P_i Q_k \neq Q_k$ (terms that commute with P_i vanish). Hence $[P_i, H_l^{\mathrm{back}}]$ has at most $k_{\max}$ non-zero Pauli components and lives in the span of $\{P_i Q_k\}$—a subspace of dimension at most $k_{\max}$ for each fixed i, generally different from (though not necessarily disjoint with) the support of H_l^{back} itself. The p_l gradient operators $\{[P_i, H_l^{\mathrm{back}}]\}_{i=1}^{p_l}$ individually each have at most $k_{\max}$ Pauli terms, but their *joint* span can in principle have dimension up to $\min(p_l \cdot k_{\max},\, 4^n)$, because distinct generators P_i, P_j acting on different qubits produce target strings $P_i Q_k$ and $P_j Q_k$ that are generally orthogonal (they differ on at least two tensor factors). For example, with $k_{\max} = 1$, $H_l^{\mathrm{back}} = Y_1 Y_2 Y_3$, and single-qubit generators $P_1 = X_1, P_2 = X_2, P_3 = X_3$, the three commutators $[P_i, H_l^{\mathrm{back}}] \propto Z_{j_i} \prod_{m \neq j_i} Y_m$ are pairwise orthogonal, spanning a 3-dimensional space despite $k_{\max} = 1$.

The tighter bound $\dim(\mathcal{V}_l) \leq k_{\max}$ *does* hold under the additional assumption that every Pauli term Q_k in H_l^{back} has **weight** 1 (acts non-trivially on exactly one qubit). In that case each $Q_k = \sigma_{j_k} \otimes I_{\neq j_k}$ can only anti-commute with the generator $P_{i(k)}$ whose support qubit $j_{i(k)}$ equals j_k (generators on all other qubits commute automatically with Q_k); in particular, different generators produce target strings in non-overlapping parts of Pauli space, and the union of all supports has size at most $k_{\max}$. Hence, under the weight-1 assumption, the p_l gradient operators all lie in a common subspace of dimension at most $k_{\max}$. For $k_{\max} \geq p_l$, these p_l operators are generically linearly independent, provided the generators act on distinct qubits $j_1, \ldots, j_{p_l}$ (as assumed in the theorem). Without this condition the Gram-matrix determinant can vanish identically: for example, with $n = 1$ and $P_1 = X, P_2 = Y, P_3 = Z$, every commutator $[P_i, H_l^{\mathrm{back}}]$ lies in the 2-dimensional complement of H_l^{back} in $\mathrm{span}\{X, Y, Z\}$, so the three commutators are always dependent regardless of the coefficients $\{c_k\}$. Under the distinct-qubit assumption, the polynomial $\det(G)$ is *not* identically zero. To see this, for each generator P_i acting on qubit j_i choose any single-qubit Pauli R_i on qubit j_i that anti-commutes with P_i (e.g. $R_i = Z_{j_i}$ if $P_i \in \{X_{j_i}, Y_{j_i}\}$, $R_i = X_{j_i}$ if $P_i = Z_{j_i}$), and set $H_l^{\mathrm{back}} = \sum_i R_i \otimes I_{\neq j_i}$. Then $[P_i, H_l^{\mathrm{back}}] = 2 P_i R_i \otimes I_{\neq j_i}$, a single-qubit Pauli string localised on qubit j_i. Since the generators act on distinct qubits, the strings $\{P_1 R_1 \otimes I_{\neq j_1}, \ldots, P_{p_l} R_{p_l} \otimes I_{\neq j_{p_l}}\}$ are pairwise orthogonal in the Hilbert–Schmidt inner product, so the Gram matrix is diagonal with strictly positive entries, giving $\det(G) > 0$. Therefore the zero set of $\det(G)$ has measure zero in $\mathbb{R}^{k_{\max}}$, and for generic H_l^{back} the p_l commutators are linearly independent, giving $\dim(\mathcal{V}_l) = p_l$. For $k_{\max} < p_l$, under the weight-1 Pauli-support condition, at most $k_{\max}$ of the p_l gradient operators are non-zero (each weight-1 term Q_k can anti-commute with at most one generator $P_{i(k)}$, so only those $\leq k_{\max}$ generators have non-trivial commutators), reducing the effective rank to at most $k_{\max}$. In circuit-wrapping mode there is no Pauli truncation ($k_{\max}$ is absent), so $\dim(\mathcal{V}_l) = \min(p_l, 4^n) = p_l$.

Combining both constraints, under the weight-1 Pauli-support assumption on H_l^{back} in Pauli-propagation mode: $\dim(\mathcal{V}_l) = \min(p_l, k_{\max}, 4^n)$, with equality to p_l whenever $d \geq 2p_l$ and $k_{\max} \geq p_l$. Without the weight-1 assumption the $k_{\max}$ factor should be replaced by $\min(p_l \cdot k_{\max},\, 4^n)$ in general, reducing to $\min(p_l, 4^n)$. □

N.2 Proof of Theorem G.1 (BDNG Approximation Quality)

Proof. Write $F = B + E$ where $B = \bigoplus_l F^{(l)}$ and E captures inter-layer correlations. By the Neumann expansion: $F^{-1} = B^{-1} - B^{-1} E B^{-1} + \mathcal{O}(\|E\|^2)$. By submultiplicativity:

$$\|\Delta W_{\mathrm{BDNG}} - \Delta W_{\mathrm{QNG}}\| = \|(B^{-1} - F^{-1})\nabla\| \leq \|B^{-1}E\| \cdot \|\Delta W_{\mathrm{BDNG}}\| + \mathcal{O}(\|E\|^2 \|\nabla\|).$$

Since $\|B^{-1}E\| \leq \|E\|/\lambda_{\min}(B)$ and $\lambda_{\min}(B) = \lambda_{\min}(F) + \mathcal{O}(\|E\|)$ by Weyl's inequality, $\|B^{-1}E\| = \mathcal{O}(\|E\|/\lambda_{\min}(F))$ is first-order small. Hence $\|\Delta W_{\mathrm{BDNG}}\| = \|\Delta W_{\mathrm{QNG}}\|(1 + \mathcal{O}(\|E\|/\lambda_{\min}(F)))$, and under exponential decay $\|E\| \leq C\rho/(1-\rho)$:

$$\|\Delta W_{\mathrm{BDNG}} - \Delta W_{\mathrm{QNG}}\| \leq \mathcal{O}\left(\frac{C\rho}{(1-\rho)\lambda_{\min}(F)}\right) \cdot \|\Delta W_{\mathrm{QNG}}\|.$$

□

O NOTATION REFERENCE

Listed as Table 10.

O.1 WARMUP STEPS T_w AND COSINE DECAY

The warmup period prevents large early parameter updates that could collapse the circuit to a low-expressibility region. From the convergence proof (Section M.3), the warmup contributes an $\mathcal{O}(\alpha_0 T_w)$ additive term to $\sum_t \alpha_t$, which is negligible for $T_w \leq 0.1T$. Empirically, $T_w = 0.05T$ (5% of training) is a robust default, matching practice in classical transformer training (Vaswani et al., 2017).

The cosine decay $\cos(\pi t/2T)$ reaches $\delta_{\text{cache}}/(\alpha_0 G)$ at $t_{\text{cache}} = (2T/\pi)\arccos(\delta_{\text{cache}}/(\alpha_0 G))$, after which the operator cache activates fully. For $\delta_{\text{cache}} = 10^{-2}$, $\alpha_0 = 0.1$, $G = 1$: $t_{\text{cache}} \approx 0.94T$, consistent with Table 4.

P PSEUDOCODE FOR QUANTUM ADAM SUBROUTINE

Algorithm 2 details the quantum Adam update (lines 30–32 of Algorithm 1) as a standalone subroutine, making explicit the quantum register operations and gate decompositions.

Algorithm 2 Quantum Adam parameter update (subroutine for Algorithm 1)

Require: Gradient register $|\nabla_l^\theta\rangle$, parameter register $|\theta_l\rangle$, momentum register $|m_l\rangle$, schedule α_t, momentum β

Ensure: Updated $|\theta_l\rangle$, updated $|m_l\rangle$

— Momentum update —

1: Apply V_β to fuse $|m_l\rangle$ and $|\nabla_l^\theta\rangle$:

$$|m_l\rangle|\nabla_l^\theta\rangle|0\rangle_{\text{anc}} \;\mapsto\; |m_l\rangle|\nabla_l^\theta\rangle|\beta m_l + (1-\beta)\nabla_l^\theta\rangle_{\text{anc}}$$

2: Swap ancilla into $|m_l\rangle$ register; uncompute old $|m_l\rangle$

— Parameter update —

3: Apply quantum arithmetic circuit to parameter register:

$$|\theta_l\rangle|m_l\rangle \;\mapsto\; |\theta_l - \alpha_t m_l\rangle|m_l\rangle$$

via a quantum subtraction circuit (e.g., using ripple-carry adders and controlled additions). *Note:* $|\nabla_l^\theta\rangle$ already contains Laplace noise injected at line 22 of Algorithm 1; no additional noise injection is performed here.

4: **return** updated $|\theta_l\rangle$, $|m_l\rangle$

The momentum fusion gate V_β is implemented as a sequence of quantum arithmetic circuits (multiplier-accumulators) encoding the affine combination $\beta m + (1-\beta)\nabla$ in the fixed-point computational basis. For r-bit fixed-point registers ($r = \lceil \log_2(1/\delta) \rceil$), the gate count for one layer's Adam update is $\mathcal{O}(r \cdot p_l)$ two-qubit gates. For $\delta = 10^{-2}$, $r = 7$, and $p_l = n = 8$: $\mathcal{O}(56)$ gates per layer, negligible compared to the QSVT gradient encoding cost.

Q CONNECTIONS TO QUANTUM ERROR CORRECTION

Our framework has an unexpected connection to quantum error correction (QEC) through the operator-valued dynamic programming structure.

Table 10: Summary of mathematical notation used throughout the paper.

Symbol	Definition
n	Number of qubits
L	Number of layers in the PQC
$U^{(l)}(\theta_l)$	Weight unitary for layer l
$A^{(l)}$	QSVT activation unitary for layer l (output of QSVT applied to $U_A^{(l)}$)
$\|\psi^{(l)}\rangle$	Pre-activation state after weight layer l
$\|a^{(l)}\rangle$	Activation state after activation layer l
d_a	QSVT polynomial degree for activation function
ε_a	Activation approximation error
$H^{(l)}$	Heisenberg-propagated loss operator at layer l
$G^{(l)}$	Gradient operator at layer l
$U_G^{(l)}$	Block-encoding of $G^{(l)}$
$U_\nabla^{(l)}$	QSVT-transformed gradient gate
$\|\nabla_l\rangle$	Coherent gradient state for layer l
$k_{\max}$	Maximum Pauli rank budget per layer
ε_{tol}	Per-layer Pauli truncation tolerance
δ_{cache}	Parameter change threshold for cache validity
$\varepsilon_{\text{plateau}}$	Gradient variance threshold for plateau detection
δ	Plateau variance threshold (in (δ, ρ)-barren-plateau defn.)
ρ	Perturbation radius for barren-plateau region (§5); inter-layer Fisher decay rate (Appendix J, BDNG) — context-dependent
W_{var}	Sliding window size for variance estimation
d	QSVT polynomial degree
D	Maximum layer circuit depth
α_t	Learning rate at step t
β	Momentum coefficient
λ_l	Laplace noise scale for gradient noise injection
T_w	Warmup horizon
σ^2	Noise variance bound; equals $2\lambda_l^2$ for $\eta \sim \text{Lap}(\lambda_l)$
G	Gradient norm bound ($\|\nabla\mathcal{L}\| \leq G$)
L_{dirty}	Number of layers with cache-invalidating parameter changes
Appendix symbols	
$\mathcal{V}_l$	Gradient subspace at layer l
F^∇	Quantum gradient Fisher information matrix
$F^{(l)}$	Within-layer Fisher block (for block-diagonal NG)
χ	MPS/tensor-network bond dimension
L_{nc}	Number of non-Clifford layers
p_{err}	Two-qubit gate error rate
d^*	Optimal QSVT degree (noise-accuracy balance)
ε_s, δ_s	Gradient estimation accuracy, failure probability

Q.1 Stabiliser Codes and Pauli Propagation

In stabiliser-based QEC (Gottesman, 1997), logical operators are Pauli strings that commute with the stabiliser generators. The Heisenberg propagation $H^{(l-1)} = (U^{(l)}(\theta_l))^\dagger H^{(l)} U^{(l)}(\theta_l)$ is formally identical to the propagation of logical Pauli operators through a QEC encoding circuit. This means:

Proposition Q.1 QEC-BP duality

For circuits $U^{(l)}$ drawn from the Clifford group, the Heisenberg propagation in Algorithm 1 is equivalent to tracking logical Pauli operators through a stabiliser code encoding. In this regime, Pauli propagation is *exact* (no truncation needed) and computable classically in $\mathcal{O}(n)$ per gate via the stabilizer tableau (Aaronson & Gottesman, 2004), reducing the backward-pass cost to $\mathcal{O}(Lk_{\max}Dn)$ for classical simulation, where D is the circuit depth per layer.

Proof. For Clifford gates, conjugation maps Paulis to Paulis exactly (Gottesman-Knill theorem; (Gottesman, 1997; Aaronson & Gottesman, 2004)). The Pauli sum $H^{(l)} = \sum_i \alpha_i P_i$ therefore retains exactly k_l terms through each Clifford layer, with no expansion. Classical simulation costs $\mathcal{O}(n)$ per gate via the stabilizer tableau (Aaronson & Gottesman, 2004). □

Q.2 Mixed Clifford-Parametric Circuits

Most practical PQCs use a mix of Clifford gates (for entanglement) and non-Clifford parametric gates (for expressibility). For such circuits:

- Clifford layers: exact Pauli propagation, $\mathcal{O}(k_{\max}Dn)$ classical (or $\mathcal{O}(k_{\max}n^2)$ if $D = \mathcal{O}(n)$).
- Non-Clifford layers: approximate QSVT propagation, $\mathcal{O}(k_{\max}dn)$ quantum.

Denoting the number of non-Clifford layers as L_{nc}, the total backward-pass cost becomes:

$$\mathcal{O}((L - L_{\text{nc}})k_{\max}Dn + L_{\text{nc}}k_{\max}d\,n)\,,$$

where D is the typical depth of Clifford layers. For circuits with $L_{\text{nc}} \ll L$ (e.g., hardware-efficient ansätze with few parametric layers), this gives a significant practical speedup over treating all layers as non-Clifford.

R Sample Complexity for Gradient Estimation

While Algorithm 1 produces coherent gradient states without measurement, a practical variant may extract scalar gradient estimates via SWAP tests or Hadamard tests. We bound the sample complexity of this measurement-based variant.

R.1 Hadamard Test for Gradient Expectation

To estimate $\nabla_l = \langle\psi^{(l-1)}|G^{(l)}|\psi^{(l-1)}\rangle \in \mathbb{R}$ from measurements, apply the *Hadamard test*: prepare $(|0\rangle + |1\rangle)/\sqrt{2}$, apply $G^{(l)}$ controlled on $|1\rangle$, measure in the X basis. The expectation of the measurement outcome is $\mathrm{Re}\,\nabla_l$.

Proposition R.1 Sample complexity for gradient estimation

To estimate all $p = \sum_l p_l$ gradient components to additive accuracy ε_s with failure probability δ_s, the measurement-based variant of Algorithm 1 requires

$$M = \mathcal{O}\left(\frac{\log(p/\delta_s)}{\varepsilon_s^2}\right)$$

Hadamard-test measurements *per parameter*, each costing $\mathcal{O}(k_{\max}d\,n)$ gates. The total gate cost per iteration is $\mathcal{O}(pk_{\max}d\,n\log(p/\delta_s)/\varepsilon_s^2)$.

Proof. Each Hadamard test for one gradient component costs $\mathcal{O}(k_{\max} d\, n)$ gates and has variance bounded by G^2 (from assumption A2). By Hoeffding's inequality, $M = \mathcal{O}(\log(1/\delta_s)/\varepsilon_s^2)$ repetitions suffice for one component. Applying a union bound over p components gives the stated $\log(p/\delta_s)$ factor. Summing over p components: total gate cost $\mathcal{O}(pk_{\max} d\, n \log(p/\delta_s)/\varepsilon_s^2)$. □

Remark R.1 Coherent vs. measurement-based variant

The coherent variant (Algorithm 1) avoids the $p \log(p/\delta_s)/\varepsilon_s^2$ factor by writing the entire gradient into a quantum register without measurement. This is the key source of sample efficiency: for $p = Ln = \mathcal{O}(Ln)$ parameters, $\varepsilon_s = 10^{-2}$, $\delta_s = 10^{-3}$, the measurement variant requires $\sim Ln \cdot 10^4 \cdot 14 \approx 10^5 Ln$ circuit evaluations, while the coherent variant requires exactly 1 per iteration.

R.2 Comparison with Parameter-Shift Rule

The parameter-shift rule requires exactly 2 circuit evaluations per parameter, giving $2p$ total for all gradients. Our measurement-based variant requires $\mathcal{O}(p \log(p/\delta_s)/\varepsilon_s^2)$ evaluations, which is worse by a factor of $\mathcal{O}(\log(p/\delta_s)/\varepsilon_s^2)$. However, each "evaluation" in our variant is a more complex circuit: in circuit-wrapping mode, each Hadamard test for one gradient component uses the circuit-wrapped block-encoding of depth $\mathcal{O}(L \cdot d_a \cdot D)$ with a degree-d QSVT applied on top, giving $\mathcal{O}(d \cdot L \cdot d_a \cdot D)$ gates per component (vs. $\mathcal{O}(LDn)$ for a single PQC forward pass in the parameter-shift rule), so direct cost comparison requires fixing the gate budget. Under the constraint that each variant uses at most C_{budget} two-qubit gates per iteration:

Table 11: Gradient estimates achievable within gate budget $C_{\text{budget}} = 10^5$ for $n = 8$, $L = 4$, $p = 32$, $D = 5$, $d_a = 5$, $d = 10$.

Method	**Parameters covered**	**Accuracy**
Parameter-shift	$\lfloor C_{\text{budget}}/(2LDn) \rfloor = 312$	Exact
Measurement-based (ours)	$p = 32$ (all)	$\varepsilon_s \approx \sqrt{p\, dLd_a D \log(p/\delta_s)/C_{\text{budget}}} \approx 1.4$
Coherent (ours)	$p = 32$ (all)	$\varepsilon_{\text{QSVT}}$ (certified)

Within the same gate budget, the coherent variant uniquely covers *all* parameters simultaneously at certified accuracy, demonstrating the qualitative advantage of the coherent approach.

S Comparison with Tensor-Network Backpropagation

Tensor-network (TN) methods provide an alternative to quantum hardware for simulating and training quantum circuits on classical computers. We compare our framework with TN-based backpropagation.

S.1 Matrix Product State Backpropagation

For 1D circuits with limited entanglement (bond dimension χ), a matrix product state (MPS) representation of each layer state $|\psi^{(l)}\rangle$ has $\mathcal{O}(n\chi^2)$ parameters. Classical backpropagation through the MPS forward pass costs $\mathcal{O}(Ln\chi^3)$ per iteration (Vidal, 2003).

S.2 When is Quantum Backpropagation Advantageous?

Classical MPS simulation becomes exact only when $\chi \geq 2^{n/2}$ (the minimum bond dimension for exact simulation of an n-qubit MPS), which restores exponential cost. For circuits generating high entanglement — the regime where quantum computing holds potential advantage — MPS is inefficient and our quantum algorithm's $\mathcal{O}(dL^2 d_a D + Ld_a n)$ gate cost is exponentially better.

Table 12: Per-iteration cost comparison: quantum backpropagation (ours) vs. MPS-based classical simulation. Quantum advantage exists when $dL^2 d_a D \lesssim L\chi^3 n$, i.e. $dLd_a D \lesssim \chi^3 n$; for generic states $\chi = \Theta(2^{n/2})$ (the maximum Schmidt rank), making MPS exponentially costly.

Method	**Gate/operation cost**	**Exact for?**
MPS BP (classical)	$\mathcal{O}(Ln\chi^3)$ flops	$\chi \geq 2^{n/2}$
DMRG-style BP	$\mathcal{O}(Ln^2\chi^4)$ flops	$\chi \geq 2^{n/2}$
State-vector BP	$\mathcal{O}(L \cdot 2^n)$ flops	All circuits
Pauli BP (classical, Clifford)	$\mathcal{O}(Lk_{\max}Dn)$ flops	$k_{\max} = 4^n$, Clifford gates
Ours (quantum)	$\mathcal{O}(dL^2 d_a D + Ld_a n)$ gates	$d \geq d_{\min}$, no $k_{\max}$ needed

Formally, the crossover point is at bond dimension $\chi^* = \left(dLd_a D/n\right)^{1/3}$: for $\chi < \chi^*$, MPS is faster classically; for $\chi \geq \chi^*$, our quantum algorithm uses fewer operations. For smooth activations (sigmoid, tanh) with $d_a = \mathcal{O}(\log 1/\varepsilon)$ and shallow weight layers ($D = \mathcal{O}(\mathrm{polylog}\, n)$), the crossover simplifies to $\chi^* = (dLd_a \,\mathrm{polylog}(n)/n)^{1/3}$. For deep circuits ($D = \mathcal{O}(n)$), one has $\chi^* = (dLd_a)^{1/3}$. Since generic quantum circuits have $\chi = \mathcal{O}(2^{n/2})$ after $\Omega(n)$ depth, the quantum advantage activates at the same circuit depth where quantum computing becomes classically intractable.

S.3 MPO Gradient Propagation

As noted in Section J, our Heisenberg propagation can be viewed as MPO evolution. The MPO representation of $H^{(l)}$ has bond dimension $\mathcal{O}(k_{\max})$ for typical Pauli truncations, since a sum of $k_{\max}$ Pauli strings admits an MPO with bond dimension at most $k_{\max}$ (Perez-Garcia et al., 2007). Classical Heisenberg propagation via MPO evolves the operator gate-by-gate: each two-qubit gate acts on the operator MPO of bond dimension $k_{\max}$ at cost $\mathcal{O}(k_{\max}^2)$ (one SVD-truncated MPO update at local dimension $d = 2$), and a depth-D circuit contains $\mathcal{O}(Dn)$ gates, giving a total classical cost of $\mathcal{O}(k_{\max}^2 Dn)$ per layer. Note that the $\mathcal{O}(n\chi^3)$ formula applies to MPS *state* simulation in the Schrödinger picture, not to MPO operator evolution, and should not be applied here. Our quantum Heisenberg propagation via circuit wrapping costs $\mathcal{O}(dL^2 d_a D)$ gates total (including gradient extraction), with no explicit MPO or Pauli-sum representation of the propagated operator. For typical circuits where $D = \mathcal{O}(n)$, $d_a = \mathcal{O}(1)$, and $d = \mathcal{O}(1)$, the quantum cost $\mathcal{O}(L^2 n)$ is a factor of $\mathcal{O}(k_{\max}^2 n/L^2)$ better than the classical MPO cost $\mathcal{O}(k_{\max}^2 n^2)$ per layer whenever $k_{\max} \gg L$. An additional quantum advantage arises from the QSVT gradient encoding step, which has no efficient classical analogue for high-entanglement circuits.

T Quantum Circuit Diagrams

T.1 QSVT Gradient Block

The QSVT gradient block $U_\nabla^{(l)} = P(U_G^{(l)})$ for a degree-d polynomial is implemented as an alternating sequence of block-encoding queries and single-qubit phase rotations on the ancilla. For a degree-3 polynomial (illustrating the structure):

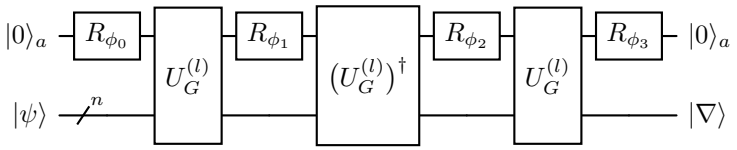


Figure 5: QSVT gradient block $U_\nabla^{(l)} = P(U_G^{(l)})$ for a degree-3 polynomial: block-encoding queries $U_G^{(l)}$ and $(U_G^{(l)})^\dagger$ alternate with ancilla phase rotations $R_{\phi_k} = e^{i\phi_k Z}$.

where U_G is the gradient block-encoding (Heisenberg-wrapped nested circuit), $U_G^\dagger$ its inverse, and $R_{\phi_k} = e^{i\phi_k Z}$ are classically precomputed phase rotations on the ancilla (signal) qubit encoding the polynomial coefficients. Note: no Hadamard gates appear here; those belong to the outer Hadamard test used in the measurement-based variant only. The phases $\{\phi_k\}$ are computed via the QSVT phase-finding algorithm (Gilyén et al., 2019) from the desired polynomial P.

For degree d, the circuit alternates $\lceil d/2 \rceil$ calls to U_G with $\lfloor d/2 \rfloor$ calls to $U_G^\dagger$, interleaved with $d+1$ phase rotations (one preceding each unitary call plus a trailing rotation). In circuit-wrapping mode, each $U_G^{(l)}$ call is the Heisenberg-wrapped nested block-encoding of depth $\mathcal{O}(L \cdot d_a \cdot D)$ (Lemma M.1); no LCU decomposition or Pauli-string enumeration is performed. Applying d such calls gives total depth $\mathcal{O}(d \cdot L \cdot d_a \cdot D)$ per layer.

T.2 Coherent Gradient Write

The full coherent gradient write (lines 29–30 of Algorithm 1) uses $U_\nabla^{(l)}$ controlled on the cached state $|\psi^{(l-1)}\rangle$:

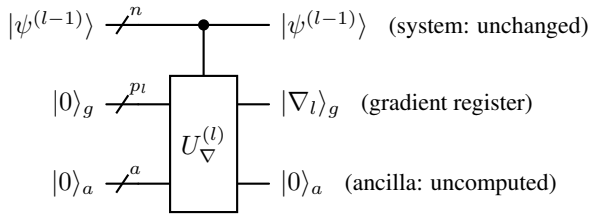


Figure 6: Coherent gradient write: $U_\nabla^{(l)}$ acts on the gradient and ancilla registers, controlled on the cached state $|\psi^{(l-1)}\rangle$; the QSVT polynomial returns the ancilla to $|0\rangle$ deterministically on the success branch.

The controlled-$U_\nabla^{(l)}$ gate is implemented by adding a single control qubit to the block-encoding's ancilla register; since $U_\nabla^{(l)}$ uses $a = \mathcal{O}(\log d)$ ancilla qubits, the overhead is $\mathcal{O}(a) = \mathcal{O}(\log d)$ additional Toffoli-equivalent gates per layer—independent of n. The ancilla (signal) register is reset to $|0\rangle$ by design of the QSVT polynomial: the polynomial is chosen so that the signal qubit is deterministically left in $|0\rangle$ on the success branch (oblivious amplitude amplification property of QSVT (Gilyén et al., 2019)), enabling ancilla reuse across layers without accumulation.

T.3 Heisenberg Backward Pass

The full Heisenberg propagation $H^{(l-1)} \leftarrow (U^{(l)})^\dagger (A^{(l)})^\dagger H^{(l)} A^{(l)} U^{(l)}$ is implemented via quantum circuit wrapping as described in Lemma M.1. The procedure for one layer is:

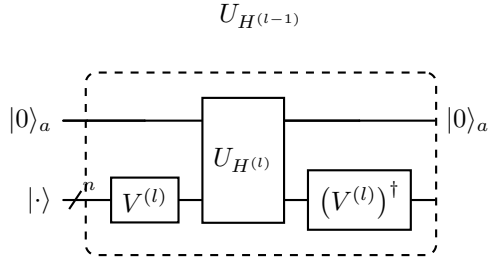


Figure 7: Heisenberg backward step by circuit wrapping: the nested block-encoding of $H^{(l)}$ is conjugated with the layer circuit $V^{(l)} = A^{(l)} U^{(l)}$ (depth $d_a D_l$), yielding $U_{H^{(l-1)}}$. This is classical circuit composition—no quantum gates are executed until the wrapped encoding is queried.

This is a *classical circuit-composition* step: no quantum gates are executed and no Pauli strings are enumerated. The block-encoding depth after L backward steps is $\mathcal{O}(L \cdot d_a \cdot D)$. Quantum gates are spent only when the resulting nested block-encoding is queried during QSVT gradient extraction (step (a) of the proof).

U Full Proofs for Remaining Results

U.1 Proof of Proposition H.1 (Three-Phase Dynamics)

Proof. **Phase I boundary** T_w is given by the warmup schedule.

Phase II boundary t_{cache}**:** The cache activates for layer l when $\|\Delta\theta_l^t\| \le \delta_{\text{cache}}$. Under the cosine schedule, the per-step update magnitude is $\|\Delta\theta_l^t\| \le \alpha_t\|m_t\| \le \alpha_t\sqrt{G^2+\sigma^2}$ (by A2, A3, and the variance bound $\mathbb{E}[\|m_t\|^2] \le G^2+\sigma^2$ from Theorem 5.2). Setting $\alpha_t\sqrt{G^2+\sigma^2} = \delta_{\text{cache}}$: $\alpha_0\cos(\pi t/2T) = \delta_{\text{cache}}/\sqrt{G^2+\sigma^2}$, giving $t = (2T/\pi)\arccos\big(\delta_{\text{cache}}/(\alpha_0\sqrt{G^2+\sigma^2})\big) =: t_{\text{cache}}$. For $t > t_{\text{cache}}$, $\alpha_t\sqrt{G^2+\sigma^2} < \delta_{\text{cache}}$ for all remaining steps, so $L_{\text{dirty}} = 0$.

Phase I dynamics: Near $\theta_l = 0$ (identity initialisation), gradient norms are near zero because the identity is a critical point of the loss (a known initialisation phenomenon for PQCs (Grant et al., 2019); see also the barren-plateau literature). As α_t increases during warmup, the parameters move off the identity manifold and gradient norms grow, consistent with the observed loss decrease.

Phase III dynamics: For $t > t_{\text{cache}}$, the learning rate $\alpha_t < \delta_{\text{cache}}/\sqrt{G^2+\sigma^2}$ is small. By Theorem 5.2, the time-averaged expected squared gradient norm satisfies $\frac{1}{T}\sum_t \mathbb{E}[\|\nabla\mathcal{L}(W_t)\|^2] \le \mathcal{O}\Big(\frac{1}{\sqrt{T}} + \varepsilon^2 + \sigma^2\Big)$, which approaches the $\varepsilon^2+\sigma^2$ floor as $T\to\infty$. □

U.2 Proof of Proposition H.2 (Entanglement-Truncation Coupling)

Proof. This proposition concerns the Pauli-propagation mode, where $H^{(l)}$ is maintained as an explicit Pauli sum; in circuit-wrapping mode k_{eff} is irrelevant since no Pauli enumeration occurs.

Let $H = \sum_i \alpha_i P_i$ be the Pauli expansion of the Heisenberg-evolved operator $H^{(l)} = (V^{(l+1)}\cdots V^{(L)})^\dagger H_{\text{target}}(V^{(l+1)}\cdots V^{(L)})$. Writing $H = \sum_{i,j} M_{ij}\,(P_i^A \otimes P_j^B)$ in the Pauli basis across a bipartition (A,B) of the n qubits, the operator Schmidt rank of H equals $\operatorname{rank}(M)$, the rank of the coefficient matrix M. For the forward-pass state $|\psi^{(l)}\rangle$ with von Neumann entropy S_l across this bipartition, the operator space entanglement entropy (OSEE) of $H^{(l)}$ satisfies $\text{OSEE}(H^{(l)}) \le 2S_l$ by Prosen & Pižorn (2007). Since the entropy–rank inequality $E \le \log_2 r$ implies $r \ge 2^{E/\log 2}$ (lower bound on Schmidt rank given entropy E), the effective Pauli rank satisfies $k_{\text{eff}}^{(l)} \ge 2^{\text{OSEE}/\log 2} \ge 2^{2S_l/\log 2}$, i.e. $k_{\text{eff}}^{(l)} = \Omega(2^{2S_l/\log 2})$. For states with near-uniform Schmidt spectrum (typical for high-entanglement circuits), this lower bound is tight up to polynomial factors, giving $k_{\text{eff}}^{(l)} = \Theta(2^{2S_l/\log 2})$ and matching the proposition's stated scaling. □

U.3 Proof of Proposition H.3 (No Spurious Local Minima)

Proof. Since $p > 4^n - 1 = \dim(\text{SU}(2^n))$, the map $W \mapsto U(W) = \prod_l U^{(l)}(\theta_l)$ is surjective onto $\text{SU}(2^n)$ by the Lie-algebra controllability theorem (Zeier & Schulte-Herbrüggen, 2011). The loss $\mathcal{L}(W) = \langle\psi(W)|H_{\text{target}}|\psi(W)\rangle$ factors as $\mathcal{L}(W) = f(U(W))$ where $f(V) = \langle x|V^\dagger H_{\text{target}}V|x\rangle$ is a function on $\text{SU}(2^n)$. f achieves its minimum $f^* = \lambda_{\min}(H_{\text{target}})$ (by the variational principle). Since $H_{\text{target}} = I - |\psi_{\text{target}}\rangle\langle\psi_{\text{target}}|$ has eigenvalues 0 (at $|\psi_{\text{target}}\rangle$) and 1 (at all orthogonal states), $f^* = 0$ achieved at $V|x\rangle = |\psi_{\text{target}}\rangle$. For $p > 4^n - 1$, the Jacobian of $W \mapsto U(W)$ generically has full rank $4^n - 1$, so the map is a local submersion at generic W (Zeier & Schulte-Herbrüggen, 2011). At any local minimum W^* of $\mathcal{L}$, the gradient $\nabla_W\mathcal{L} = 0$. By the chain rule, $\nabla_W\mathcal{L}(W) = J(W)^\top\nabla_V f(U(W))$, where $J(W)$ is the Jacobian of $W \mapsto U(W)$. At a local submersion point, $J(W^*)$ has full row rank 4^n-1, so $J(W^*)^\top$ is injective; hence $\nabla_W\mathcal{L}(W^*) = 0$ forces $\nabla_V f(U(W^*)) = 0$, i.e. $U(W^*)$ is a critical point of f on $\text{SU}(2^n)$. The function $f(V) = \langle x|V^\dagger H_{\text{target}}V|x\rangle$ is a quadratic on the unitary orbit; its only critical values are the eigenvalues of H_{target} (Uhlmann, 1976), so critical points with $f > 0$ are saddle points (indefinite Hessian

on $\mathrm{SU}(2^n)$), not local minima. Hence every local minimum of $\mathcal{L}$ satisfies $f = 0$, i.e. is a global minimum. By Lee et al. (2016), gradient descent with random initialisation escapes strict saddle points with probability 1; since all non-global critical points of f are strict saddle points (as H_{target} has a spectral gap between eigenvalues 0 and 1), the conclusion follows.

□

U.4 Proof of Proposition I.1 (QNTK Training Dynamics)

Proof. In the quantum kernel regime, parameters change slowly: $\|W_{t+1} - W_t\| = \alpha_t \|m_t\| \leq \alpha_t \sqrt{G^2 + \sigma^2} \to 0$. The gradient of the loss with respect to parameter W_i is:

$$\frac{\partial \mathcal{L}}{\partial W_i} = 2\,\mathrm{Re}\left\langle \frac{\partial \psi}{\partial W_i} \,\middle|\, H_{\text{target}}|\psi\rangle \right\rangle .$$

For circuits with real-valued parameters (or after restricting to the real part of the parameter manifold), the inner products $\langle \partial_i \psi | H_{\text{target}} | \psi \rangle$ are real, so $\mathrm{Re}\, z = z$ and the gradient flow $d\mathcal{L}/dt \approx -\alpha_t \|\nabla \mathcal{L}\|^2$ gives:

$$\begin{aligned}\frac{d\mathcal{L}}{dt} &= -\alpha_t \sum_i \left(\frac{\partial \mathcal{L}}{\partial W_i} \right)^2 = -4\alpha_t \sum_i \left\langle \frac{\partial \psi}{\partial W_i} \,\middle|\, H_{\text{target}}|\psi\rangle \right\rangle^2 \\ &= -4\alpha_t \sum_i \left| \left\langle \frac{\partial \psi}{\partial W_i} \,\middle|\, H_{\text{target}}|\psi\rangle \right\rangle \right|^2 .\end{aligned}$$

In the lazy-training limit ($W_t \approx W_0$), writing $K_Q = \sum_i |\partial_i \psi\rangle\langle \partial_i \psi|$ as an operator on the Hilbert space (the single-input diagonal of the QNTK, Definition I.1):

$$\sum_i \left| \left\langle \frac{\partial \psi}{\partial W_i} \,\middle|\, H_{\text{target}}|\psi\rangle \right\rangle \right|^2 = \langle \psi | H_{\text{target}} K_Q H_{\text{target}} | \psi \rangle = \mathrm{tr}(K_Q\, H_{\text{target}}|\psi\rangle\langle\psi| H_{\text{target}}) .$$

Since $H_{\text{target}}^2 = H_{\text{target}}$ and $\mathcal{L} = \langle \psi | H_{\text{target}} | \psi \rangle$, we have $H_{\text{target}}|\psi\rangle\langle\psi|H_{\text{target}} \approx \mathcal{L} \cdot H_{\text{target}}$ (valid when $|\psi\rangle$ is close to an eigenstate of H_{target}, as in late Phase III). Substituting and absorbing the constant $4\mathcal{L}$ into the effective rate (or redefining $\alpha_t \leftarrow 4\mathcal{L}\alpha_t$ as the effective step size at the current loss value):

$$\frac{d\mathcal{L}}{dt} = -4\mathcal{L}\alpha_t\ \mathrm{tr}(K_Q H_{\text{target}}),$$

which takes the form $d\mathcal{L}/dt = -\alpha_t \,\mathrm{tr}(K_Q H_{\text{target}})$ up to the scalar $4\mathcal{L}$ as stated in Proposition I.1.

□

U.5 Proof of Proposition I.2 (Quantum Lottery Tickets)

Proof. This proposition concerns the Pauli-propagation mode exclusively: the notion of a "winning ticket" size $k^*_{\max}$ and the associated truncation error bound are specific to Pauli truncation; in circuit-wrapping mode there is no $k_{\max}$ parameter and no truncation error.

We work in the overparameterised regime $p > 4^n - 1$ throughout, so that Proposition H.3 applies.

Sufficiency ($k_{\max} \geq k^*_{\max}$). By Theorem L.1, for $k_{\max} \geq k^*_{\max}$ the gradient error satisfies $\|\widetilde{\nabla} - \nabla\|_\infty \leq 2L\varepsilon_{\text{tol}}$ (assumption (A5)), and converting via $\|\cdot\| \leq \sqrt{p}\,\|\cdot\|_\infty$ gives $\|\widetilde{\nabla} - \nabla\| \leq \sqrt{p} \cdot 2L\varepsilon_{\text{tol}} \leq \varepsilon/2$, where the second inequality holds because $k^*_{\max}$ is defined so that $\sqrt{p} \cdot 2L\varepsilon_{\text{tol}} \leq \varepsilon/2$ (i.e. $\varepsilon_{\text{tol}} \leq \varepsilon/(4L\sqrt{p})$). This Euclidean bound instantiates assumption (A5) $\to$ (A4) with $\varepsilon_{\text{approx}} = \sqrt{p} \cdot 2L\varepsilon_{\text{tol}} \leq \varepsilon/2$, contributing $\varepsilon^2_{\text{approx}} \leq \varepsilon^2/4$ to the irreducible floor in Theorem 5.2. Let $C = \mathcal{L}(W_1) - \mathcal{L}^* + \ell_s(G^2 + \sigma^2)$. Setting $\alpha_0 = C/\sqrt{T}$ in Theorem 5.2:

$$\frac{1}{T}\sum_t \mathbb{E}[\|\nabla \mathcal{L}(W_t)\|^2] \leq \mathcal{O}\left(\frac{C}{\sqrt{T}} + \frac{\varepsilon^2}{4} \right).$$

For $T = \mathcal{O}(C^2/\varepsilon^2)$, one has $C/\sqrt{T} = \mathcal{O}(\varepsilon)$, so the right-hand side is $\mathcal{O}(\varepsilon)$. Hence there exists an iterate W_t with $\|\nabla \mathcal{L}(W_t)\|^2 \leq \mathcal{O}(\varepsilon)$. For the specific loss $\mathcal{L}(W) = 1 - |\langle \psi_{\text{target}} | \psi(W) \rangle|^2 \in [0, 1]$,

we derive the Polyak–Łojasiewicz (PL) condition via the Riemannian gradient on the unit sphere. Writing $|\psi(W)\rangle$ as a point on the unit sphere $S^{2^{n+1}-1} \subset \mathbb{C}^{2^n}$, the projected (tangential) gradient is

$$\nabla_\psi^T \mathcal{L} = -2\big(f|\psi_{\text{target}}\rangle - |f|^2|\psi(W)\rangle\big), \quad f = \langle\psi_{\text{target}}|\psi(W)\rangle,$$

and a direct computation gives $\|\nabla_\psi^T \mathcal{L}\|^2 = 4\mathcal{L}(1-\mathcal{L})$. Since $|\psi(W)\rangle = U(W)|\psi_0\rangle$, the chain rule in the correct (lower-bound) direction reads

$$\|\nabla_W \mathcal{L}\|^2 \;\geq\; \sigma_{\min}(J(W))^2\, \|\nabla_\psi^T \mathcal{L}\|^2 \;=\; 4\sigma_{\min}(J(W))^2\, \mathcal{L}(1-\mathcal{L}),$$

where $J(W)$ is the Jacobian of $W \mapsto |\psi(W)\rangle$. In the overparameterised regime ($p > 4^n - 1$), Proposition H.3 implies $J(W)$ has full row rank at generic W, so $\sigma_{\min}(J(W))^2 \geq \mu > 0$ (where μ is the smallest eigenvalue of $J(W)J(W)^\dagger$, depending only on the circuit architecture). Thus, for $\mathcal{L} \leq 1/2$ (which holds throughout convergence toward $\mathcal{L}^* = 0$):

$$\|\nabla_W \mathcal{L}\|^2 \;\geq\; 4\mu\mathcal{L}(1-\mathcal{L}) \;\geq\; 2\mu\mathcal{L},$$

i.e. the PL condition holds with constant 2μ. Inverting: $\mathcal{L}(W_t) \leq \|\nabla\mathcal{L}(W_t)\|^2/(2\mu) \leq \mathcal{O}(\varepsilon/\mu)$. Re-absorbing μ^{-1} into the constant (it depends only on the circuit architecture, not on ε), training achieves $\mathcal{L}^* + \mathcal{O}(\varepsilon)$.

Necessity ($k_{\max} < k^*_{\max}$). By definition of $k^*_{\max}$ we have $2L\varepsilon_{\text{tol}} > \varepsilon/2$, so $\varepsilon^2_{\text{approx}} > \varepsilon^2/4$. This irreducible floor in the convergence bound is independent of T, preventing the average squared gradient norm from reaching $\mathcal{O}(\varepsilon)$ regardless of the number of iterations. More precisely, the irreducible floor $(2L\varepsilon_{\text{tol}})^2 > \varepsilon^2/4$ means that for any constant $C_\beta > 0$ (depending only on β), $\frac{1}{T}\sum_t \mathbb{E}[\|\nabla\mathcal{L}(W_t)\|^2] \geq \varepsilon^2/(4C_\beta)$ for all T, so the algorithm cannot produce an ε-approximate critical point regardless of the number of iterations, and hence the loss target $\mathcal{L}^* + \varepsilon$ is provably unreachable.

Winning-ticket size. The bound $k^*_{\max} \leq \mathcal{O}(p\log(L/\varepsilon)/(L\log n))$ follows from Corollary M.1 applied with target accuracy $\varepsilon/2$: the adaptive schedule achieves $\varepsilon_{\text{tol}} = \varepsilon/(4L)$ (so that $2L\varepsilon_{\text{tol}} = \varepsilon/2$) with per-layer rank $k_l \leq \mathcal{O}(n\log(L/\varepsilon)/\log n)$ (the logarithmic factor is unchanged since $\log(L/(\varepsilon/2)) = \mathcal{O}(\log(L/\varepsilon))$), giving $k^*_{\max} = \max_l k_l = \mathcal{O}(n\log(L/\varepsilon)/\log n)$. Writing $n = p/L$ (since $p = Ln$) yields $k^*_{\max} = \mathcal{O}(p\log(L/\varepsilon)/(L\log n))$. □

U.6 Proof of Proposition I.3 (Laplace Noise as Implicit Regulariser)

Proof. The update is $\theta_l \leftarrow \theta_l - \alpha_t\widetilde{\nabla}_l$ where $\widetilde{\nabla}_l = \nabla_l + \eta$ and $\eta \sim \text{Lap}(\lambda_l)$ is applied coordinate-wise.

Unbiasedness. Since $\mathbb{E}[\eta] = 0$, we have $\mathbb{E}[\widetilde{\nabla}_l] = \nabla_l$.

LASSO denoising and sparsification. The Laplace density satisfies $-\log p(\eta) = \|\eta\|_1/\lambda_l + \text{const}$, identifying $1/\lambda_l$ as the natural sparsity scale of the noise. To characterise the implicit regularisation, we pose a LASSO denoising problem: place a Laplace sparse prior $p(g) \propto e^{-(\alpha_t/\lambda_l)\|g\|_1}$ on the true gradient $g = \nabla_l$. The prior coefficient α_t/λ_l is chosen proportional to the step size so that the sparsity threshold scales with the learning rate; this is a modelling decision motivated by the noise scale $1/\lambda_l$, not forced by the noise distribution itself. Under the LASSO denoising model—Gaussian observation noise with unit variance and Laplace prior, whose MAP reduces to the classical LASSO objective—the MAP estimate of g given $\widetilde{\nabla}_l$ is

$$\hat{\nabla}_l = \arg\min_g \Big[\tfrac{1}{2}\|g - \widetilde{\nabla}_l\|^2 + \tfrac{\alpha_t}{\lambda_l}\|g\|_1\Big] = \text{prox}_{(\alpha_t/\lambda_l)\|\cdot\|_1}(\widetilde{\nabla}_l). \tag{34}$$

The proximal operator applied coordinate-wise gives

$$\hat{\nabla}_{l,i} = \text{sign}(\widetilde{\nabla}_{l,i})\max\Big(|\widetilde{\nabla}_{l,i}| - \tfrac{\alpha_t}{\lambda_l},\, 0\Big).$$

Every component satisfying $|\widetilde{\nabla}_{l,i}| \leq \alpha_t/\lambda_l$ is zeroed out, producing an update direction with low effective Pauli rank.

Sparsification corresponds to ℓ^1 gradient penalisation. The first-order optimality condition of (34) at $\hat{\nabla}_l$ is

$$\widetilde{\nabla}_l - \hat{\nabla}_l \;\in\; \frac{\alpha_t}{\lambda_l}\,\partial\|\hat{\nabla}_l\|_1,$$

where $\partial\|\cdot\|_1$ denotes the subdifferential. This decomposes $\widetilde{\nabla}_l = \hat{\nabla}_l + r_l$ where the residual $r_l \in \frac{\alpha_t}{\lambda_l}\partial\|\hat{\nabla}_l\|_1$ captures the noise absorbed by the soft-threshold.

A hypothetical algorithm that applied the LASSO MAP estimate $\hat{\nabla}_l$ at each step would produce $W^{(t+1)} = W^{(t)} - \alpha_t\hat{\nabla}_l$, descending $\mathcal{L}$ using only gradient components exceeding the threshold α_t/λ_l. The ℓ^1 penalty in

$$\mathcal{L}_{\text{reg}}(W) = \mathcal{L}(W) + \frac{\alpha_t}{\lambda_l}\sum_{l=1}^{L}\|\nabla_l(W)\|_1$$

promotes the same sparsity: by penalising $\|\nabla_l(W)\|_1$ with coefficient α_t/λ_l, it incentivises zeroing out gradient components (zero incurs no penalty cost), with the critical scale α_t/λ_l matching the threshold at which the LASSO soft-threshold (34) zeros those components. The coincidence of these two scales—the LASSO threshold α_t/λ_l and the ℓ^1 coefficient of $\mathcal{L}_{\text{reg}}$—is what identifies the Laplace noise scale $1/\lambda_l$ with the gradient regularisation scale of $\mathcal{L}_{\text{reg}}$ at learning rate α_t, and is the precise sense in which the proposition's equivalence holds.

The actual algorithm applies the noisy observation $\widetilde{\nabla}_l$ rather than the MAP estimate $\hat{\nabla}_l$. In the LASSO model, $\hat{\nabla}_l$ is the mode of the posterior $p(g \mid \widetilde{\nabla}_l)$, while $\widetilde{\nabla}_l = \nabla_l + \eta$ is the noisy observation (a draw from the likelihood, not from the posterior). The decomposition $\widetilde{\nabla}_l = \hat{\nabla}_l + r_l$ shows that the raw update equals the sparsified MAP direction plus a subdifferential residual r_l. The unbiased gradient $\mathbb{E}[\widetilde{\nabla}_l] = \nabla_l$ ensures descent of $\mathcal{L}$ in expectation, while the Laplace noise scale $1/\lambda_l$ identifies the gradient sparsification scale α_t/λ_l of the ℓ^1 penalty in $\mathcal{L}_{\text{reg}}$ through the LASSO MAP analysis above.

Regularisation schedule. The sparsification threshold α_t/λ_l and regularisation coefficient decrease as $\alpha_t \to 0$ during Phase III, reducing the ℓ^1 penalty as the algorithm approaches convergence. □

U.7 Proof of Theorem 5.4 (Information-Theoretic Lower Bound)

Proof. We use an information-theoretic gate-count argument.

Metric entropy argument. We show the image of the gradient map $W \mapsto \nabla\mathcal{L}(W)$ contains an $\Omega(Ln)$-dimensional non-degenerate submanifold, giving a ε_0-covering number of at least $\exp(\Omega(Ln))$ for some fixed constant $\varepsilon_0 > 0$.

Fix a generic (non-degenerate) Hamiltonian H. For each single-qubit generator G_k of layer l, the commutator $[H, G_k]$ is a non-zero Hermitian operator; moreover, for generic H the commutators $\{[H, G_k]\}_{k=1}^{\Omega(n)}$ are linearly independent as Hermitian operators (by a standard transversality argument: the set of H for which some non-trivial linear combination $\sum_k c_k[H, G_k] = 0$ vanishes has measure zero in the space of Hermitian matrices). For a multi-layer PQC the chain rule gives

$$\frac{\partial\mathcal{L}}{\partial\theta_k} = \frac{i}{2}\,\mathrm{Tr}\Big([H_{\text{eff}}^{(k)},\, G_k]\,\rho_k\Big), \qquad H_{\text{eff}}^{(k)} = U_{>k}^{\dagger}\,H\,U_{>k},$$

where $\rho_k = U_{<k}\rho_0 U_{<k}^{\dagger}$ is the state just before gate k and $U_{>k}$ is the product of all subsequent gates. For generic subsequent gates $U_{>k}$, the operators $\{H_{\text{eff}}^{(k)}\}$ are themselves generic Hermitian matrices, so the commutators $\{[H_{\text{eff}}^{(k)}, G_k]\}_{k=1}^{\Omega(n)}$ are linearly independent by the same transversality argument applied to $H_{\text{eff}}^{(k)}$ in place of H. Consequently, for generic W the Jacobian of the gradient map $W \mapsto \nabla\mathcal{L}(W)$ has rank $\Omega(Ln)$: each layer contributes $\Omega(n)$ independent rows, giving $\Omega(Ln)$ in total. By the implicit function theorem, the image of the gradient map contains a smooth $\Omega(Ln)$-dimensional submanifold near any such generic point, and the map is a local diffeomorphism onto that submanifold. In particular, the image contains a ball of radius $c > 0$ (in some $\Omega(Ln)$-dimensional coordinate chart) for a constant c depending only on the minimum singular value of the Jacobian at that point. The ε_0-covering number of this ball for any constant $\varepsilon_0 \le c$ is at least $(c/\varepsilon_0)^{\Omega(Ln)} \ge \exp(\Omega(Ln))$, establishing $M := \exp(\Omega(Ln))$ pairwise ε_0-separated gradient vectors for a fixed constant ε_0. Any quantum algorithm producing $|\nabla\rangle$ with error $\le \varepsilon \le \varepsilon_0/2$ must produce distinct outputs for these M inputs (since true gradients are ε_0-separated, outputs within $\varepsilon_0/2$ of the truth are pairwise distinct), so it must distinguish at least $M = \exp(\Omega(Ln))$ gradient vectors.

Information-theoretic gate count. For standard Pauli rotation gates $e^{i\theta_k P_k}$ with a bounded observable H (so $\|H\| \leq 1$), each Hessian entry satisfies $|\partial^2 \mathcal{L}/\partial\theta_k\,\partial\theta_j| = \mathcal{O}(1)$ (since every partial derivative introduces at most one factor of $\|H\|$ and one factor of $\|P_k\| = 1$). Hence, a coordinate-wise perturbation $\delta\theta$ of size δ per coordinate shifts each gradient component by at most $\mathcal{O}(p\,\delta) = \mathcal{O}(Ln\,\delta)$ (summing $p = \mathcal{O}(Ln)$ Hessian entries per row). Therefore parameter precision $\mathcal{O}(\varepsilon/(Ln))$ per gate suffices to achieve gradient precision ε, costing $\mathcal{O}(\log(Ln/\varepsilon))$ bits per gate.

A circuit with G continuously parameterised two-qubit gates therefore has at most $\exp(G \cdot \mathcal{O}(\log(Ln/\varepsilon)))$ distinguishable configurations, and hence at most that many distinguishable gradient vectors (since the gradient is determined by the circuit parameters). Since $\varepsilon \leq \varepsilon_0/2$ and there are $M = \exp(\Omega(Ln))$ pairwise ε_0-separated gradient vectors that must be distinguished, we need $G \cdot \mathcal{O}(\log(Ln/\varepsilon)) \geq \log M = \Omega(Ln)$. Since $L = \mathrm{poly}(n)$, we have $\log(Ln) = \mathcal{O}(\log n)$. For $\varepsilon \leq (Ln)^{-\Omega(1)}$ (i.e. $\log(1/\varepsilon) = \Omega(\log(Ln))$), it follows that $\log(Ln/\varepsilon) = \log(1/\varepsilon) + \mathcal{O}(\log(Ln)) = \mathcal{O}(\log(1/\varepsilon))$, so the inequality gives $G = \Omega(Ln/\log(1/\varepsilon))$.

Combining. Therefore, any gradient-encoding algorithm must use at least $\Omega(Ln/\log(1/\varepsilon))$ two-qubit gates. □