Backpropagation scaling in parameterised quantum circuits

Joseph Bowles, David Wierichs, and Chae-Yeun Park

Xanadu, Toronto, ON, M5G 2C8, Canada

arXiv:2306.14962v4 [quant-ph] 19 Sep 2025

The discovery of the backpropagation algorithm ranks among one of the most important moments in the history of machine learning, and has made possible the training of large-scale neural networks through its ability to compute gradients at roughly the same computational cost as model evaluation. Despite its importance, a similar backpropagation-like scaling for gradient evaluation of parameterised quantum circuits has remained elusive. Currently, the most popular method requires sampling from a number of circuits that scales with the number of circuit parameters, making training of large-scale quantum circuits prohibitively expensive in practice. Here we address this problem by introducing a class of structured circuits that are not known to be classically simulable and admit gradient estimation with significantly fewer circuits. In the simplest case—for which the parameters feed into commuting quantum gates—these circuits allow for fast estimation of the gradient, higher order partial derivatives and the Fisher information matrix. Moreover, specific families of parameterised circuits exist for which the scaling of gradient estimation is in line with classical backpropagation, and can thus be trained at scale. In a toy classification problem on 16 qubits, such circuits show competitive performance with other methods, while reducing the training cost by about two orders of magnitude.

1 Introduction

Many tasks in optimisation and machine learning involve the minimisation of a continuous cost function over a set of parameters, and gradient based approaches are often the methods of choice to tackle such tasks. The efficiency with which gradients can be evaluated is crucial in these methods, since it sets a practical upper limit on the number of optimisation steps, determined by the time and cost of running the algorithm. The success of neural network models in machine learning for example is frequently credited to the possibility of fast gradient evaluation via backpropagation [1, 2], enabling the models to scale to the enormous sizes we see today.

Within quantum computation, much hope has been placed on parameterised quantum circuits to provide a new generation of powerful ansätze for optimisation and machine learning tasks [3, 4]. As such, a wide array of variational quantum algorithms have been designed that hinge on the ability to estimate gradients of quantum circuits. Despite this, classes of circuits for which fast gradient estimation is possible are still missing. Currently, the standard method to compute gradients is via the parameter-shift method [5, 6, 7, 8], which in the simplest case requires sampling from a pair of distinct circuits for every parameter in the model. For this reason, the cost of estimating the gradient can be significantly more expensive than evaluating the function itself. This is in stark contrast to automatic differentiation methods such as backpropagation [2], which can allow for exact gradient computation with roughly the same time and memory required to evaluate the function itself.

Furthermore, the stochasticity of quantum models results in an additional sampling cost that scales inversely with the square of the precision of the gradient estimate [6]. For problems with a large number of parameters, this quickly makes gradient estimation prohibitively expensive in practice. To give some concrete numbers, to perform one epoch of gradient descent for a dataset with 1000 data points using a quantum model with 1000 parameters to a gradient precision of 10^{-4} would require 2×10^{14} quantum cir-

Joseph Bowles: joseph@xanadu.ai

cuit shots. With a quantum computer operating continually at clock speed of 1MHz, this would take around six years.

All of this suggests that variational quantum algorithms–and in particular quantum machine learning algorithms–are in dire need of ansätze that allow for faster gradient estimation than with the parameter-shift rule, and a number of works have explored other methods [9, 10, 11, 12, 13, 14, 15, 16] and frameworks [17] to this end. For generic, unstructured parameterised quantum circuits, fast gradient estimation appears difficult to achieve however, since backpropagation-like scaling is known to be impossible given access to only single copies of an unknown input state [18]. In order to make progress in this direction, it therefore seems necessary to focus attention on specific circuit structures. In some sense this should not be surprising: building successful machine learning models has always been about finding well-designed structures that are tailored for fast optimisation, and one should not expect this to change when moving to quantum models.

In this work we present classes of parameterised quantum circuits that allow for faster gradient estimation than parameter-shift methods. In the simplest case (Sec. 3), these circuits feature mutually commuting parameterised gates, which we call *commuting-generator circuits*. Although gates in these circuits commute and can thus be simultaneously diagonalised, both the input state and final measurement need not be in this basis, which prevents classical simulability. The restriction to commuting gates allows the gradient estimation to be parallelised over the parameters of the circuit: whereas the parameter-shift method requires sampling from a pair of circuits for each parameter, our method returns the complete gradient vector to the same precision by sampling from a single circuit of the same number of qubits. This can lead to enormous savings in the number of circuit shots and in certain cases matches the scaling of backpropagation[1]. Surprisingly, this class of circuits even allows for parallelisation when estimating the Fisher information matrix, and all partial derivatives of a given degree (at the cost of exponentially increasing variance with degree). This is something which even neural network models cannot offer, and opens up the possibility for efficient higher-order optimisation methods.

We also show (Sec. 4) how one can go beyond the restriction to commuting gates and design classes of circuits with non-commuting parameterised gates that also allow for parallel gradient estimation, which we call *commuting-block circuits*. Here, the circuits must have a specific block structure, such that gates commute within blocks, with generators from distinct blocks having a fixed commuting or anticommuting relation. Due to this extra freedom, parameterised circuits from this class can be more expressive than their commuting counterparts. This raises the question of what are the ultimate limits of expressibility of these models, which we leave as an open question.

We then present two specific examples of commuting-generator circuits and derive the corresponding circuits needed for the evaluation of partial derivatives (Sec. 5). The first of these circuits is used as a basis to construct a quantum model whose layers are equivariant with respect to translation permutations of qubits. This model is compared against another translation-equivariant model that features non-commuting gates, and against a quantum convolutional neural network at a classification task with translation symmetry (Sec. 6). Simulating 16-qubit models, we show that the circuit with commuting generators achieves the lowest training cost and the best test accuracy of the three models, using orders of magnitude fewer circuit shots.

Going forward, we hope our work encourages a move away from viewing generic unstructured quantum circuits as viable ansätze for optimisation and machine learning, and highlights both the need for and potential of creating quantum circuit ansätze with well-designed and fit-for-purpose structures.

2 Backpropagation scaling in parameterised quantum circuits

The class of circuits we consider in this work take the form of classically parameterised quantum circuits:

$$C(\boldsymbol{\theta}) = \langle 0| U^\dagger(\boldsymbol{\theta}) \mathcal{H} U(\boldsymbol{\theta}) |0\rangle \tag{1}$$

[1] We note that our work does not represent a quantum version of the classical backpropogation algorithm, but is a different approach with comparable scaling guarantees.

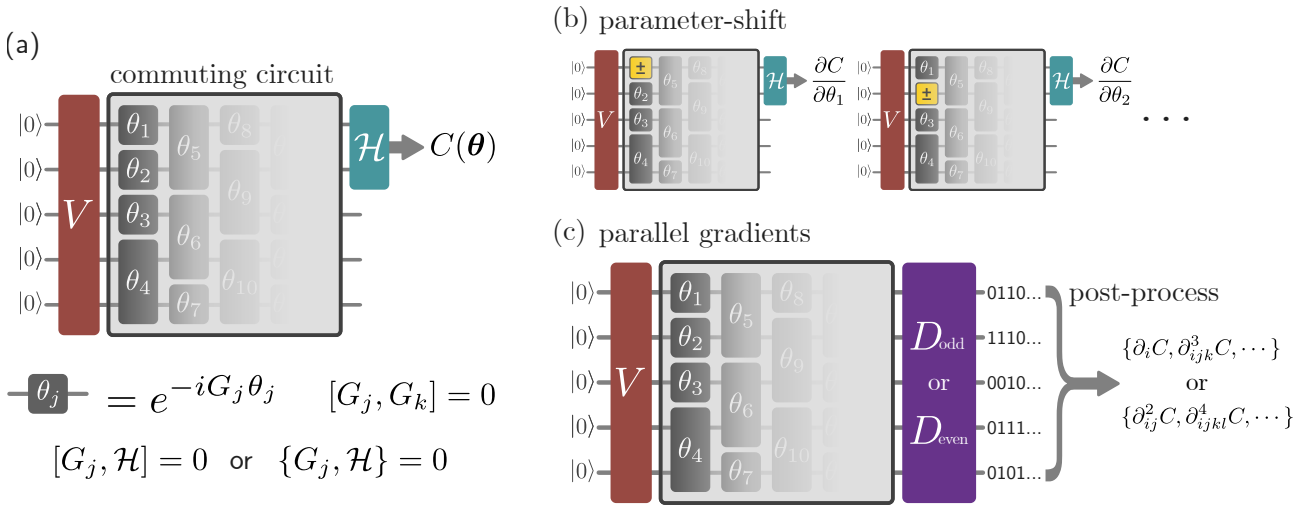


Figure 1: (a) The simplest type of circuit class we consider, which we call *commuting-generator circuits*. An arbitrary unitary V is applied, followed by a parameterised quantum circuit comprised of gates $\exp(-i\theta_j G_j)$ with mutually commuting generators G_j. An observable $\mathcal{H}$ is measured on some subset of the output qubits, and for each G_j either (i) $\mathcal{H}$ commutes with G_j or (ii) $\mathcal{H}$ anticommutes with G_j. (b) The standard method to estimate the gradient of quantum circuits is via parameter-shift rules. For gate generators with two distinct eigenvalues, the partial derivative with respect to each parameter is evaluated by estimating the difference between two circuits, where the parameter in question has been shifted. The resources required using this method can therefore be much larger than those required to estimate $C(\boldsymbol{\theta})$, which can be prohibitively expensive for circuits with thousands of parameters. (c) The corresponding circuit used for parallel gradient estimation. A unitary D_{odd} (or D_{even}) is applied to the original circuit and the circuit is sampled from in the computational basis M times. All odd (or even) partial derivatives can then be estimated to additive error $\epsilon = \mathcal{O}(\frac{1}{\sqrt{M}})$ by classically post processing the outcomes. In the case that all generators and the observable are stabiliser operators, the unitaries D_{even}, D_{odd} are Clifford unitaries and the post-processing consists of evaluating expectation values of subsets of qubits only.

where

$$U(\boldsymbol{\theta}) = \prod_j e^{-i\theta_j G_j} \tag{2}$$

is a unitary operator with Hermitian generators G_j and classical parameters $\boldsymbol{\theta}$, and $\mathcal{H}$ is some Hermitian observable. On hardware, by measuring $\mathcal{H}$ and collecting M shots, one obtains an unbiased estimate of $C(\boldsymbol{\theta})$ with variance scaling as $\mathcal{O}(1/M)$.

To compute the gradient $\nabla C(\boldsymbol{\theta})$ of such circuits, the most common approach is to use *parameter-shift methods* [19, 5, 6, 7, 8, 20]. In the simplest case of gate generators with two distinct eigenvalues, an unbiased estimate of the gradient can be obtained by evaluating two circuits for every parameter in the model, so that

$$\frac{\partial C}{\partial \theta_j} \propto C(\boldsymbol{\theta} + \boldsymbol{\alpha}_j) - C(\boldsymbol{\theta} - \boldsymbol{\alpha}_j), \tag{3}$$

with $\boldsymbol{\alpha_j}$ a vector with zeros on all components except the j^{th}. To obtain an estimate of the gradient to the same precision as $C(\boldsymbol{\theta})$, one therefore needs to take M shots from each circuit, so that each component has variance $\mathcal{O}(1/M)$. The cost of gradient estimation via this method is therefore larger than the cost of estimating $C(\boldsymbol{\theta})$ by a factor proportional to the number of trainable parameters.

In similar spirit to [18], we define a method to have *backpropagation scaling* if it returns an unbiased estimate of $\nabla C(\boldsymbol{\theta})$ to the same precision as $C(\boldsymbol{\theta})$, with a logarithmic overhead in the number of parameters compared to the evaluation of $C(\boldsymbol{\theta})$.

Definition 2.1 (backpropagation scaling) *Consider a parameterised quantum circuit of the form (1) with n parameters, which returns an unbiased estimate of $C(\boldsymbol{\theta})$ with variance $\mathcal{O}(\frac{1}{M})$ by sampling M shots from the circuit. Denote by TIME(C) and MEM(C) the time and space complexity of this procedure, and by TIME(∇C) and MEM(∇C) the time and space complexity of obtaining an unbiased estimate of the gradient with elementwise variance $\mathcal{O}(1/M)$. Then we say that a gradient method*

has backpropagation scaling if

$$TIME(\nabla C) \leq c_t\, TIME(C) \tag{4}$$

and

$$MEM(\nabla C) \leq c_m\, MEM(C) \tag{5}$$

with $c_t, c_m = \mathcal{O}(\log(n))$.

In this work, we will present families of circuits and gradient methods that achieve such scaling, even obtaining examples where c_t, c_m have constant scaling.

3 Commuting-generator circuits

The first class of circuits we consider consists of arbitrary input state preparation, mutually commuting parameterised gates, and measurement of an observable (see Fig. 1(a)). These circuits are thus given by

$$C(\boldsymbol{\theta}) = \langle 0| V^\dagger U^\dagger(\boldsymbol{\theta}) \mathcal{H} U(\boldsymbol{\theta}) V |0\rangle\,, \tag{6}$$

where V is an arbitrary unitary, $\mathcal{H}$ is an observable, and the parameterised unitary takes the form

$$U(\boldsymbol{\theta}) = \prod_{j=1}^{n} \exp(-iG_j\theta_j) \tag{7}$$

such that $[G_j, G_k] = 0\ \forall\ j,k \in [n]$, where we adopt the notation $[n] = \{1, \cdots, n\}$. We further restrict our attention to observables $\mathcal{H}$ such that for each G_j we have either $[\mathcal{H}, G_j] = 0$ or $\{\mathcal{H}, G_j\} = 0$; i.e. a given G_j either commutes or anticommutes with $\mathcal{H}$. This is the case, for example, if all generators and observables are tensor products of Pauli operators. We remark that many applications involve evaluating sums of observables of this form, since any Hermitian multi-qubit operator can be decomposed in the Pauli basis. Here we will consider just a single $\mathcal{H}$ for simplicity however; evaluating gradients for sums can be achieved in the usual way by summing the contributions of individual observables.

We will refer to such circuits as *commuting-generator circuits*. Note that this class differs from the 'commuting quantum circuits' [21, 22] commonly studied in complexity theory due to the arbitrary unitary V at the start of the circuit. Interestingly, circuits in this class allow for parallel gradient estimation in the following sense.

Theorem 1 *Consider a commuting-generator circuit* $C(\boldsymbol{\theta})$ *of the form* (6)*. Then an unbiased estimator of the gradient*

$$\nabla C(\boldsymbol{\theta}) = \left(\frac{\partial C}{\partial\theta_1}, \frac{\partial C}{\partial\theta_2}, \cdots, \frac{\partial C}{\partial\theta_n}\right) \tag{8}$$

can be obtained by classically post-processing a single circuit C' *with the same number of qubits as* C*. As the measurement statistics of* C' *are used to estimate all derivatives simultaneously, the variance of each derivative estimator scales as* $\mathcal{O}(1/M)$*, where* M *is the total number of shots used to evaluate* C'*.*

Proof—Since the generators commute, we have

$$\frac{\partial C}{\partial\theta_j} = i\,\langle\psi_0|\, U^\dagger(\boldsymbol{\theta})[G_j, \mathcal{H}]U(\boldsymbol{\theta})\,|\psi_0\rangle\,, \tag{9}$$

where $|\psi_0\rangle = V|0\rangle$. The gradient with respect to θ_j therefore corresponds to measuring the expectation value of the observable $O_j = i[G_j, \mathcal{H}]$ on $|\psi_0\rangle$, which vanishes if G_j and $\mathcal{H}$ commute. It follows that we only need to consider those generators that anticommute with $\mathcal{H}$. Consider two such generators G_j, G_k. Using anticommutativity we have $i[G_j, \mathcal{H}] = 2iG_j\mathcal{H}$ and so

$$\begin{aligned}[O_j, O_k] &= [i[G_j, \mathcal{H}], i[G_k, \mathcal{H}]] \\ &= -4(G_j\mathcal{H}G_k\mathcal{H} - G_k\mathcal{H}G_j\mathcal{H}) \\ &= 0.\end{aligned} \tag{10}$$

The operators O_j therefore mutually commute and can be simultaneously diagonalised. Denoting by $\{|\psi_i\rangle\}$ the basis in which the O_j are diagonal, we have

$$O_j = \sum_i \lambda_i(O_j)\,|\psi_i\rangle\langle\psi_i| \tag{11}$$

with $\{\lambda_i(O_j)\}$ the corresponding eigenvalues of O_j. To estimate the gradients in parallel, we use the same circuit as $C(\boldsymbol{\theta})$ but measure in the basis $\{|\psi_i\rangle\}$, i.e. we sample from the distribution

$$P(i) = |\,\langle\psi_i|\, U(\boldsymbol{\theta})V\,|0\rangle\,|^2. \tag{12}$$

From (11) and (9) we have

$$\frac{\partial C}{\partial\theta_j} = \mathbb{E}_{P(i)}[\lambda_i(O_j)]. \tag{13}$$

We estimate $\frac{\partial C}{\partial\theta_j}$ by sampling M shots $\{i_1, \cdots, i_M\}$ from $P(i)$ and estimating the expectation value (13) by

$$\frac{\partial C}{\partial\theta_j} \approx \frac{1}{M}\sum_{k=1}^{M} \lambda_{i_k}(O_j). \tag{14}$$

From the central limit theorem, since the variance of the sample mean of a bounded random variable scales as $\mathcal{O}(1/M)$, we have proven the claim. ■

Although the basis $\{|\psi_i\rangle\}$ in which the O_j are diagonal must exist, one may still encounter a difficulty in finding or implementing the unitary that rotates to this basis. In some cases this procedure may be simple however. For this reason we will often focus on circuits whose generators and observables are Pauli products (tensor products of operators in $\{\mathbb{I}, X, Y, Z\}$) and will call this class of circuits *commuting-Pauli-generator circuits*. For these circuits, the operators O_j are also Pauli products, and this implies that the diagonalizing unitary can be implemented efficiently.

Corollary 1 *Consider an N-qubit commuting-Pauli-generator circuit. Then the gradient can be estimated in parallel by a circuit C' with the same number of qubits as C and a depth at most $O\left(\frac{N}{\log N}\right)$ more than C.*

The above depth bound follows from [23, Cor 1.1]; note that one can also reduce the depth further to $O(\log N)$ at the expense of additional auxiliary qubits. Here, the diagonalising unitary D is Clifford, and partial derivatives therefore correspond to evaluating products of Pauli Z operators on subsets of qubits. Efficient algorithms to simultaneously diagonalise sets of stabiliser operators also exist [24, 25, 26], and for sets of operators with a lot of structure, suitable circuits can be found by hand (as we will see in Sec. 5.1).

Although the above depth bound scales with N, for some choices of generators and observables, the diagonalising unitary has constant depth. In these cases, the gradient evaluation therefore comes at the same computational cost as model evaluation, thus allowing for backpropagation scaling. In Sec. 5 we will see two examples of such circuits.

3.1 Higher order partial derivatives

By repeating the same process as in the proof of Theorem 1, one finds that the partial derivatives of any order can also be estimated in parallel.

Theorem 2 *Consider a commuting-generator circuit $C(\boldsymbol{\theta})$ as in Theorem 1 and a fixed integer t. Unbiased estimates of all t^{th} order partial derivatives of C can be obtained by classically post-processing a single circuit C' with the same number of qubits as C. The variance of each derivative estimator scales as $\mathcal{O}(1/M)$, where M is the total number of shots used to evaluate C'.*

Proof—We start by considering the case $t = 2$. Differentiating (9) again we find that the second-order partial derivatives are

$$\frac{\partial C^2}{\partial\theta_j\partial\theta_k} = -4\,\langle\psi_0|\,U^\dagger(\boldsymbol{\theta})G_jG_k\mathcal{H}U(\boldsymbol{\theta})\,|\psi_0\rangle \quad (15)$$

if both G_j and G_k anticommute with $\mathcal{H}$, and are zero otherwise. Note that in the non-zero case one has $[G_jG_k\mathcal{H}, G_lG_m\mathcal{H}] = 0$, so that these partial derivatives can also be measured in parallel by post-processing measurements performed in a single basis. Continuing this pattern and defining the multi-index $\boldsymbol{\alpha} \in [n]^t$, one finds that the t^{th} order partial derivatives are

$$\frac{\partial C^t}{\partial\theta_{\boldsymbol{\alpha}}} = i^t 2^t\,\langle\psi_0|\,U^\dagger(\boldsymbol{\theta})\left(\prod_{\ell=1}^{t} G_{\alpha_\ell}\right)\mathcal{H}U(\boldsymbol{\theta})\,|\psi_0\rangle \quad (16)$$

if the $G_{\alpha_1}, \cdots, G_{\alpha_t}$ all anticommute with $\mathcal{H}$, and zero otherwise. The non-zero t^{th} order derivatives can therefore be evaluated by measuring the observables

$$O_{\boldsymbol{\alpha}} = i^t 2^t \left(\prod_{\ell=1}^{t} G_{\alpha_\ell}\right)\mathcal{H}. \quad (17)$$

Once again, we find $[O_{\boldsymbol{\alpha}}, O_{\boldsymbol{\alpha}'}] = 0$ and so the t^{th} order partial derivatives can also be obtained in parallel with variances scaling inversely with the number of shots to C'. Due to the factor 2^t, the variance of the estimators will generally increase exponentially with the derivative order and thus in practice only reasonably low orders are within reach. However, this approach does enable approximate second-order optimisation with only a constant overhead in quantum resources compared to first-order methods.

Finally, we note that the form of (17) allows for an even stronger statement than Theorem 2.

Corollary 2 *Consider a commuting-generator circuit $C(\boldsymbol{\theta})$ as in Theorem 1. There exist two circuits C'_{even} and C'_{odd} with the same number of qubits as C such that all partial derivatives of even (resp. odd) order can be estimated by classically post-processing the circuit C'_{even} (resp. C'_{odd}).*

To see this, consider two observables $O_{\boldsymbol{\alpha}} \neq 0 \neq O_{\boldsymbol{\alpha}'}$ with $\boldsymbol{\alpha} \in [n]^t$ and $\boldsymbol{\alpha}' \in [n]^{t'}$ where t and t' need not coincide. Then we find $[O_{\boldsymbol{\alpha}}, O_{\boldsymbol{\alpha}'}] = 0$ if the parities of t and t' are the same ($t = t' \bmod 2$), and so all even (resp. odd) partial derivatives can be estimated in parallel.

3.2 Fisher information matrix

So far we have considered partial derivatives of cost functions as in Eq. (6). We now turn to the quantum Fisher information

$$\mathcal{F}_{jk} = \mathfrak{Re}\{\langle\psi_0| [\partial_j U^\dagger(\boldsymbol{\theta})]\partial_k U(\boldsymbol{\theta}) |\psi_0\rangle\} - \langle\psi_0| [\partial_j U^\dagger(\boldsymbol{\theta})]U(\boldsymbol{\theta}) |\psi_0\rangle \langle\psi_0| U^\dagger(\boldsymbol{\theta})\partial_k U(\boldsymbol{\theta}) |\psi_0\rangle \tag{18}$$

of the parameterised quantum state; it may be used for diagnoses of the prepared state itself [27, 28, 29], or to compute the quantum natural gradient which can be used for training [30]. For a commuting-generator circuit, we find that

$$[\partial_j U^\dagger(\boldsymbol{\theta})]\partial_k U(\boldsymbol{\theta}) = G_j G_k \tag{19}$$

$$U^\dagger(\boldsymbol{\theta})\partial_j U(\boldsymbol{\theta}) = -iG_j. \tag{20}$$

This means that the Fisher information reduces to the covariance matrix of the generators in the encoding state $|\psi_0\rangle = V|0\rangle$:

$$\mathcal{F}_{jk} = \mathrm{Cov}(G_j, G_k)_{|\psi_0\rangle}. \tag{21}$$

Note that all matrix entries can be measured in parallel because the generators commute. Furthermore, $\mathcal{F}$ does not depend on $\boldsymbol{\theta}$ but only on V, so that the same Fisher information matrix can be used throughout the optimisation. Finally, we remark that some choices of encoding unitaries V and gate generators even allow for a classical evaluation of $\mathcal{F}$, making the overhead of the quantum natural gradient purely classical. An example for this are the encoding and the generators used in Model A in Sec. 6. Overall, this makes estimating the natural gradient of commuting-generator circuits feasible in practice, provided that the number of parameters n allows for inversion of an $n \times n$ matrix on a classical computer.

3.3 Simulability

A natural question to ask is under what conditions commuting-generator circuits admit an efficient classical simulation. Since V is arbitrary, one cannot expect to sample from the output distributions of such circuits in general. However, this holds even for $V = \mathbb{I}$. For example, by choosing a commuting circuit with Pauli generators in the X basis, and an observable in the Z basis, one arrives at a parameterisation of the class of IQP circuits [21, 22], for which sampling is known to be hard (up to non-collapse of the polynomial hierarchy) [21].

However, for the weaker task of expectation value estimation, commuting-Pauli-generator circuits can become classically tractable under certain conditions. More precisely, the results of [22] imply the following.

Corollary 3 *Consider a commuting-Pauli-generator circuit with initial unitary V and observable*

$$\mathcal{H} = F^\dagger Z \otimes \mathbb{I} \otimes \cdots \otimes \mathbb{I} F \tag{22}$$

for some Clifford unitary F. Then if $FV|0\rangle = |x\rangle$ for some computational basis state $|x\rangle$, there is an efficient classical algorithm that estimates $C(\boldsymbol{\theta})$ to the same precision as the quantum circuit.

For a proof of the above, see App. A.

Note that the *dynamical Lie algebra (DLA)* (see Sec. 4.1 for a definition) of any commuting circuit with polynomial depth is Abelian and has polynomial size. Therefore, somewhat similar to the Corollary above, any input state and measurement observables with efficient classical representations that are compatible with a compressed representation of the DLA will lead to classical simulability as well [31, 32].

4 Commuting-block circuits

We now present a larger class of circuits which retains the property of fast gradient estimation. These circuits have a similar structure to the commuting generator circuits, however the parameterised part of the circuit consists of blocks of commuting gates (see Fig. 2). These blocks are such that while generators within a given block commute, generators between different blocks have a fixed commutation relation (either commutation or anticommutation). More precisely, if $\mathcal{G}_1 = \{G_j^1\}$ and $\mathcal{G}_2 = \{G_k^2\}$ are the generators from two distinct blocks j, k then one has either

$$\left[G_j^1, G_k^2\right] = 0 \ \forall j,k \quad \text{or} \quad \left\{G_j^1, G_k^2\right\} = 0 \ \forall j,k. \tag{23}$$

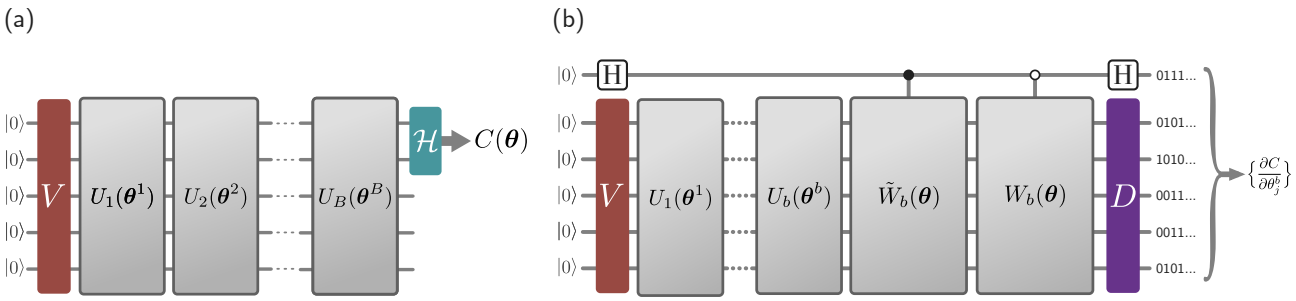


Figure 2: (a) The commuting-block circuit ansatz of Thm. 3. The blocks $U_j(\boldsymbol{\theta}_j)$ are commuting blocks with the same constraints as in Fig. 1. Generators between blocks have a fixed commutation relation: for any pair of blocks U_j, U_k either (i) all generators from block j commute with all generators from block k, or (ii) all generators from block j anticommute with all generators from block k. (b) The circuit used to estimate the partial derivatives of those generators G_j of block b that anticommute with $\mathcal{H}$. Here D is the unitary that diagonalises the operators $\{2iG_j\mathcal{H}\}$, and the unitaries W_b and $\tilde{W}_b$ are defined in (26) and (27). For generators G_j that commute with $\mathcal{H}$, one replaces W_b by iW_b and D diagonalises the operators $\{2G_j\mathcal{H}\}$.

A general circuit in this class with B blocks therefore has the form of (6) with

$$U(\boldsymbol{\theta}) = \prod_j \exp(-iG_j^B\theta_j^B)\cdots\prod_j \exp(-iG_j^1\theta_j^1) \tag{24}$$

with the blocks $\mathcal{G}_b = \{G_j^b\}$ respecting the aforementioned commutation structure. We will call such circuits *commuting-block circuits*. Gradients of these circuits can be evaluated with a number of circuits that scales with the number of blocks rather than the number of parameters, in the following sense.

Theorem 3 *Consider an N-qubit commuting-block circuit with B blocks. Then an unbiased estimate of the gradient $\nabla C(\boldsymbol{\theta})$ can be obtained by classically post-processing $2B-1$ circuits on $N+1$ qubits with increased depth. The variance of the estimator of each partial derivative scales as $\mathcal{O}(1/M)$ where M is the number of shots used to evaluate each circuit.*

We now explain the procedure to estimate the gradient; the full proof can be found in App. B. To estimate the gradient of the b^{th} block we use a technique based on linear combination of unitaries methods [33]. Since we focus on block b we will drop the block index on the generators, so that $G_j \equiv G_j^b$. Define

$$|\psi_b\rangle = U_b(\boldsymbol{\theta}^b)\cdots U_2(\boldsymbol{\theta}^2)U_1(\boldsymbol{\theta}^1)V|0\rangle\,, \tag{25}$$

i.e. the state after the b^{th} block, and

$$W_b(\boldsymbol{\theta}) = U_B(\boldsymbol{\theta}^B)\cdots U_{b+2}(\boldsymbol{\theta}^{b+2})U_{b+1}(\boldsymbol{\theta}^{b+1}). \tag{26}$$

We first define two sets of Hermitian observables: $\mathcal{O}_0 = \{O_j\} = \{2G_j\mathcal{H}\}$, for j such that G_j and $\mathcal{H}$ commute, and $\mathcal{O}_1 = \{O_j\} = \{2iG_j\mathcal{H}\}$, for j such that G_j and $\mathcal{H}$ anticommute. In contrast to commuting-generator circuits, the generators in $\mathcal{O}_0$ contribute to the gradient although they commute with $\mathcal{H}$, which is due to the circuit W_b applied after the b^{th} block. Note that all generators in either $\mathcal{O}_0$ or $\mathcal{O}_1$ mutually commute and so can be simultaneously diagonalised. We first focus on the partial derivatives corresponding to the indices j appearing in $\mathcal{O}_1$. One implements the circuit shown in Fig. 2 (b) to produce the quantum state $|\psi\rangle$, where D is the unitary that diagonalises the observables in $\mathcal{O}_1$, and $\tilde{W}_b$ is defined by

$$W_bG_j = G_j\tilde{W}_b. \tag{27}$$

For example, if generators in block b anticommute with all generators in W_b then $\tilde{W}_b$ has the same form as W_b with $\boldsymbol{\theta}$ replaced by $-\boldsymbol{\theta}$. For this circuit one finds

$$\langle\psi|\, Z\otimes DO_jD^\dagger\,|\psi\rangle = \partial_j C(\boldsymbol{\theta}). \tag{28}$$

Since D diagonalises the O_j, the partial derivatives are obtained by sampling the circuit in the computational basis via post-processing analogous to (14). For $\mathcal{O}_0$ the process is the same, however one replaces W_b by iW_b in the circuit and uses a unitary D that diagonalises the observables in $\mathcal{O}_0$. For each block we therefore require two circuits, however the generators that commute with $\mathcal{H}$ in the final block have zero gradient; hence $2B-1$ circuits suffice.

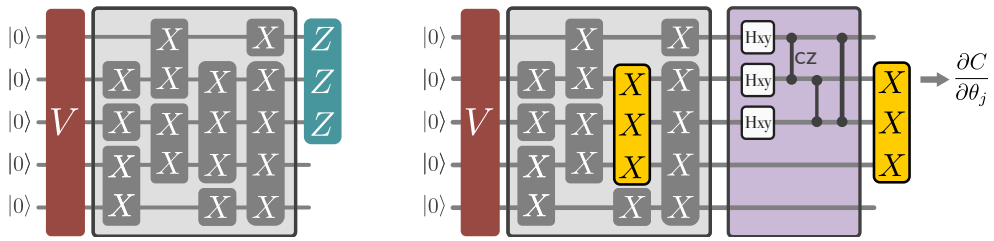


Figure 3: An instance of an X-generator circuit for 5 qubits (left) and the corresponding circuit used for gradient estimation (right). The parameterised gates are generated by products of Pauli X operators on different subsets of qubits and the observable is a product of Pauli Z operators on a subset of qubits. To estimate the gradient, an additional circuit D is added, which involves performing a controlled Z gate between all qubits on which the observable acts non-trivially. Here H_{xy} is the single-qubit Hadamard unitary that switches between the X and Y bases. To estimate the gradient of one of the gates (highlighted in yellow), one evaluates the expectation value of the generator at the output of the circuit. Since all generators are products of X, one can estimate the gradient in parallel by measuring each qubit in the X basis.

4.1 Increased expressivity with commuting-block circuits

A natural question to ask is whether commuting-block circuits have increased expressivity compared to the single block circuits of Fig. 1. This is indeed the case. To quantify expressivity we will use the *dynamical Lie algebra* [34] of the gate generators, defined as

$$\mathrm{DLA}[\{G_j\}] = \mathrm{span}_{\mathbb{R}}\langle\{iG_j\}\rangle_{\mathrm{Lie}} \tag{29}$$

where $\mathrm{span}_{\mathbb{R}}S$ is the real vector space spanned by the elements of S and $\langle\mathcal{S}\rangle_{\mathrm{Lie}}$ is the Lie closure: the set of nested commutators of operators in $\mathcal{S}$. As a simple example, a single-qubit ansatz with the Pauli X and Y operators as generators has the DLA $= \mathrm{span}_{\mathbb{R}}\{iX, iY, iZ\}$ since $[iX, iY] = -2iZ$. The class of unitaries that can be realised by sequential uses of gates with generators $\{G_j\}$ is given by [35, 36]

$$\mathcal{U} = \{e^X | X \in \mathrm{DLA}[\{G_j\}]\}. \tag{30}$$

Thus, the larger the dimension of the DLA, the larger the expressivity of the ansatz. Note that the dimension of the DLA cannot be larger than 4^N, since this is the dimension of the space of Hermitian operators acting on N qubits.

For a single block, since all generators commute, the dimension of the DLA is given by the dimension of the span of the generators, and therefore cannot be greater than 2^N. This dimension can be achieved for example by considering the 2^N generators comprised of tensor products of Z and $\mathbb{I}$. To show that we can go beyond this limit, consider a commuting-block ansatz consisting of sequential uses of the following two blocks. The first block is comprised of $2^N/2$ gates that are generated by $G_j \in \{\mathbb{I}, Z\}^N$, where we consider operators with an odd numbers of Zs only. The second block contains a single gate generated by $X^{\otimes N}$. These blocks satisfy the condition of mutual anticommutativity. A small calculation shows that the DLA is given by linear combinations of (i) the generators of the first block (of which there are $2^N/2$) and (ii) all full-weight Pauli tensors of X and Y operators (of which there are 2^N). Thus we have a DLA with dimension $\frac{3}{2}\cdot 2^N$, which is larger than the maximal DLA for a single block.

4.2 Gradient scaling

The ability to train parameterised quantum circuits under random initialisation of the parameters is closely connected to the phenomenon of barren plateaus [37], whereby the magnitudes of cost function partial derivatives decrease exponentially with the number of qubits. Recent work has proven that if either the initial density matrix or cost observable are in the DLA of the gate generators, then the scaling of the partial derivatives is closely tied to the inverse of the dimension of the algebra [34, 38].

For commuting-generator circuits, the DLA is simply the span of $\{G_j\} \cup \mathbb{I}$, which thus has a polynomially bounded dimension for polynomially sized circuits. Unfortunately, the results of [34, 38] cannot be applied to commuting generator circuits however, since the requirement that the input state or observable be in the DLA implies commutation with all generators, and the gradients of such circuits will therefore vanish.

The interesting cases for these circuits are therefore when both the input state and observable lie outside of the DLA, however this is not covered by the current theory.

For commuting-block circuits the DLA can contain non-commuting basis elements, and so non-trivial circuit structures that are free from barren plateaus can in principle be constructed, provided that the dimension of the DLA is kept polynomial (here, results from [39] may be useful). Nevertheless, one must be careful, since using an observable in the DLA can often render the circuit simulable [31]. One may therefore need to find a 'sweet spot' that avoids the criteria in [31] for efficient simulation, but retains non-vanishing gradients from the results of [34, 38].

Finally, we stress that although barren plateaus are often viewed as a practical barrier to training, their existence does not necessarily prohibit efficient training via strategies that do not initialise parameters uniformly at random. For example, alternative initialisation strategies [40, 41, 42, 43], layer-wise training [44] or classical pre-training [45, 46] may be effective approaches to optimizing such circuits in practice.

5 Explicit constructions of commuting circuits

In this section we present two constructions of commuting generator circuits, and detail the explicit unitaries required for gradient estimation. As we will see, both these constructions allow for backpropagation scaling for gradient estimation.

5.1 X-generator ansatz

For this ansatz the generators are diagonal in the X basis:

$$G_j = G_{j,1} \otimes \cdots \otimes G_{j,N} \in \{\mathbb{I}, X\}^{\otimes N} \tag{31}$$

and the possible observables are diagonal in the Z basis:

$$\mathcal{H} \in \{\mathbb{I}, Z\}^{\otimes N}. \tag{32}$$

Recall that to estimate the gradient we need to evaluate observables $O_j = 2iG_j\mathcal{H}$ for all G_j that anticommute with $\mathcal{H}$. We begin with the simple case where the observable is

$$\mathcal{H} = Z \otimes \mathbb{I} \otimes \cdots \otimes \mathbb{I}. \tag{33}$$

We need only consider generators that feature an X on the first qubit. The corresponding observables O_j take the form

$$O_j = 2Y \otimes G_{j,2} \otimes \cdots \otimes G_{j,N} \tag{34}$$

and mutually commute as expected. In this case, simultaneous diagonalisation is easy since the O_j are diagonal in the $Y \otimes X \otimes \cdots \otimes X$ product basis. To obtain estimates for all partial derivatives one samples the N qubits in this basis and then evaluates expectation values of the relevant subsets of qubits given by (34).

When there is more than one Z operator appearing in $\mathcal{H}$ the situation is a little more complicated. Let us assume that $\mathcal{H}$ is of the form

$$Z \otimes \cdots \otimes Z \otimes \mathbb{I} \otimes \cdots \otimes \mathbb{I}, \tag{35}$$

i.e. the Z operators appear on the first m qubits only. Due to the symmetry of the ansatz, other cases can be handled by permuting the qubits in the corresponding circuits.

We now need to diagonalise the operators $O_j = 2iG_j\mathcal{H}$ (for $\{G_j, \mathcal{H}\} = 0$), however the diagonalising unitary is no longer a tensor product of single-qubit unitaries. In App. D we show that the circuit of Fig. 3 diagonalises the O_j in the product X basis. This circuit involves applying an $x-y$ Hadamard on the first m qubits followed by a controlled-Z between every pair of the first m qubits. More precisely, the effect of the circuit is to map the O_j back to the G_j:

$$DO_jD^\dagger = 2G_j. \tag{36}$$

Thus, the partial derivative $\partial_j C(\boldsymbol{\theta})$ can be obtained by estimating the expectation value of G_j at the output of the circuit of Fig. 3. Since the $\{G_j\}$ take the form (31), one simply needs to measure each qubit in the X basis and construct the expectation value from the relevant subset.

The second-order derivatives are given by

$$\partial_{ij}C = \langle 0| V^\dagger U^\dagger(\boldsymbol{\theta}) O_{jk} U(\boldsymbol{\theta}) V |0\rangle \tag{37}$$

with $O_{jk} = -4G_jG_k\mathcal{H}$ such that both G_j and G_k anticommute with $\mathcal{H}$. The $\{O_{jk}\}$ can be diagonalised by a similar circuit to D: one applies a $z-y$ Hadamard on the first qubit, followed by D (see App. D). With this one finds

$$DH^{(1)}_{zy}O_{jk}H^{(1)}_{zy}D^\dagger = (-1)^{\beta_1}4X^{(1)}G_jG_k, \tag{38}$$

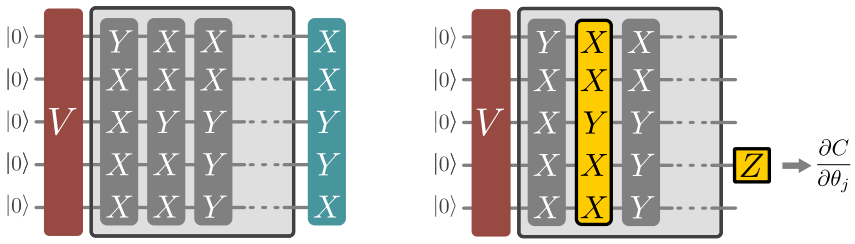


Figure 4: A commuting-generator circuit family with nonlocal generators and Hamiltonian (left). The similarity in structure between the Hamiltonian and the generators allows to measure all gradient entries in parallel without an additional basis change, simply by measuring in the computational basis and post-processing the samples into expectation values of products of Pauli Z operators (right).

where $\beta_1 = 0$ if $G_{j,1}G_{k,1} = X$ and 1 otherwise. The second-order partial derivatives can therefore be obtained by replacing D by $DH^{(1)}_{zy}$ in Fig. 3, measuring in the X basis and estimating the expectation values of the relevant subset of qubits given by (38).

Note that, if we restrict to local observables (i.e. m is upper bounded by a constant for all N), then the diagonalising unitaries for both the gradient and second order partial derivatives are of constant depth, and we achieve backpropagation scaling for the gradient estimation.

5.2 Circuits with nonlocal generators

Here we give another construction that uses fully nonlocal generators. While it generally comes at the cost of larger circuit depths, we will see that this structure allows the gradient of a sum of observables to be estimated from a single circuit. The generators we consider take the form

$$G_j \in \{X, Y\}^{\otimes N} \tag{39}$$

with an odd number of Y's in each generator. The observables take the same form

$$\mathcal{H} \in \{X, Y\}^{\otimes N} \tag{40}$$

but with an even number of Y's. One sees that the observables $O_j = 2iG_j\mathcal{H}$ are up to a sign factor given by products of Z operators on those qubits on which G_j and $\mathcal{H}$ differ. The gradient of $\langle\mathcal{H}\rangle$ can therefore be estimated by sampling each qubit in the Z basis and estimating the relevant expectation value. This holds true for any observable $\mathcal{H}_p$ of the form (40). Thus we can in fact estimate the gradient of any operator

$$\mathcal{H} = \sum_p h_p\mathcal{H}_p \tag{41}$$

by measuring the same circuit in the Z basis and postprocessing. Since the depth of the gradient evaluation is the same as the model, we achieve backpropagation scaling.

6 Numerical study: learning translationally invariant data

Although the circuits introduced in this work require much fewer circuit shots to optimise, it remains unclear if their limited structure makes them suitable in practice. In this section we investigate this by using a machine learning problem based on classifying 1D translationally invariant data. We first construct a 1D translation equivariant quantum model based on the X-generator ansatz of Sec. 5.1 that is tailored for learning data with this structure. We then train this model (Model A) and compare it against three other models: a 1D translation equivariant quantum model with non-commuting generators (Model B), inspired by a recently introduced permutation equivariant model [47]; a quantum convolutional neural network [48, 49] (Model C); and a simple separable quantum model (Model D). As we will see, the commuting circuit model (Model A) performs the best of the four models whilst requiring much fewer circuit shots to train. This suggests that commuting-generator circuits may generally perform as well or even better than other proposed models from the literature, and could therefore represent a powerful model class for machine learning tasks. Code to reproduce our numerical results is available online [50].

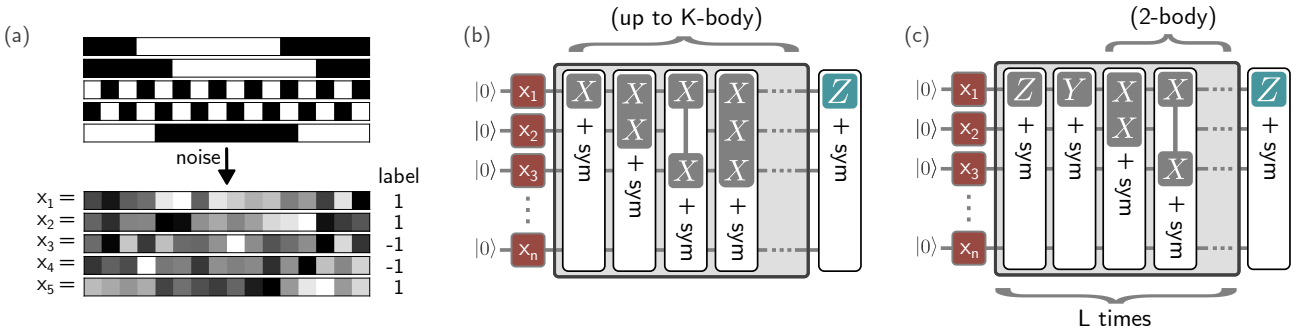


Figure 5: (a) We generate two types of 16-dimensional vectors corresponding to either bars (label 1) or dots (label 2). Independent Gaussian noise is added to generate the input data for the classification task. Note that the labels of the data set are invariant to translations of the elements of the vectors. (b–c) The two translation equivariant models we benchmark for this problem. Here '+ sym' denotes symmetrisation of the generator over 1D translations, e.g. $X_1 + \text{sym} = X_1 + X_2 \cdots + X_n$. The model in (b) features X generators only, and can therefore exploit the parallel gradient evaluation described in Sec. 5.1. The model in (c) is a translation equivariant version of the model in [47] and features non-commuting generators.

6.1 The learning problem: learning bars and dots

The dataset we consider can be seen as a simple 1D version of the bars and stripes data set that we call 'bars and dots' (see Fig. 5 (a)). Each data point $\boldsymbol{x}_i$ is a d-dimensional real vector, which corresponds to a bar (class 1) or dots (class 2). A bar vector is constructed by setting the values of $\lfloor d/2 \rfloor$ neighbouring elements of the vector to $+1$ and the remaining values to -1. A dots vector corresponds to alternating ± 1 values. The dataset is created by setting $d = 16$, sampling random bars and dots vectors, and then adding independent Gaussian noise (with variance 1) to each element of the vector. The learning problem is a simple binary classification problem to predict whether a given vector belongs to the bar or dots class (corresponding to a label $y_i = \pm 1$), and the figure of merit we consider is the usual class prediction accuracy on a test dataset consisting of 100 input vectors. During the training phase, we generate a set of 1000 input vectors and train each model with the adam optimiser using batches of 20 input vectors per update, an initial learning rate of 0.01 and gate parameters initialised uniformly random in $[0, 2\pi]$. All models are trained using the binary cross entropy loss

$$-\sum_i \log P(y_i|\boldsymbol{x}_i, \boldsymbol{\theta}) \tag{42}$$

where the probability $P(+1|\boldsymbol{x}_i, \boldsymbol{\theta})$ is given by $\sigma(6\langle \mathcal{H} \rangle)$, where σ is the logistic function and $\mathcal{H}$ is normalised such that $|\langle \mathcal{H} \rangle| \leq 1$. In practice, we observed that this gives better results than taking for example $P(+1|\boldsymbol{x}_i, \boldsymbol{\theta}) = (1 + \mathcal{H})/2$.

6.2 Quantum models

Each model corresponds to a quantum circuit with $N = 16$ qubits, where the data is encoded into the circuit via single-qubit Pauli Y rotations given by the data encoding unitary

$$V_{\boldsymbol{x}} = \bigotimes_{r=1}^{d} \exp\left(-\frac{i}{4} x^{(r)} Y_r\right) \tag{43}$$

with $x^{(r)}$ the r^{th} entry of $\boldsymbol{x}$. Note that we have chosen a specific scaling of the data in the above; in practice we found that this generally gives good results compared to other scalings, however we have not performed a detailed investigation into the effect of different data scalings on each model. After this data encoding layer, each model consists of a parameterised unitary followed by measurement of an observable $\mathcal{H}$, where we use $\text{sign}(\langle \mathcal{H} \rangle)$ as our class prediction label. The parameterised unitaries for each model are as follows.

Model A: Equivariant X-generator circuit— This model is an instance of the X-generator ansatz of Sec. 5.1, and is depicted in Fig. 5(b). To construct a model with equivariant layers, we need to consider gate generators that commute with the symmetry group that corresponds to the label invariance of the data [51, 52]. From our choice of data encoding, the translation symmetry of the data is represented by translations of the qubit subsystems; i.e. translating the elements of the input vector $\boldsymbol{x}$ results in a new data

encoding unitary $SV_{\boldsymbol{x}}$, where S is an operator that cyclicly permutes the qubits. We therefore consider generators that are symmetric with respect to this symmetry group. The simplest of these corresponds to the operator

$$X_1 + X_2 + \cdots X_d = \mathrm{sym}(X_1), \tag{44}$$

where $\mathrm{sym}(X_1)$ is the twirling operation [53, 51] that symmetrises a given operator with respect to 1D cylic permutations. Two-body generators are therefore given by $\mathrm{sym}(X_1X_r)$ for different choices of r. The ansatz uses all k-body generators up to some fixed K, which we leave unspecified for now. By using an observable that also respects this symmetry, we arrive at an equivariant model whose label prediction is invariant to the data symmetry. This follows from

$$\begin{aligned}\langle 0| V_{\boldsymbol{x}}^\dagger S^\dagger U(\theta)^\dagger \mathcal{H} U(\theta) S V_{\boldsymbol{x}} |0\rangle \\ = \langle 0| V_{\boldsymbol{x}}^\dagger U(\theta)^\dagger \mathcal{H} U(\theta) V_{\boldsymbol{x}} |0\rangle ,\end{aligned} \tag{45}$$

where we have used the fact that S commutes with $\mathcal{H}$ and $U(\theta)$ by construction. To this end, we consider the observable

$$\mathcal{H} = Z_1 + Z_2 + \cdots + Z_d. \tag{46}$$

Note that the gradient for each Z_r can be estimated in parallel via the method of Sec. 5.1, as will be described shortly in Sec. 6.3.

Model B: Equivariant model with non-commuting generators—This model (see Fig. 5(c)) can be seen as a translation equivariant analogue of the permutation equivariant model presented in [47]. The model consists of L layers where the generators in each layer are given by $\mathrm{sym}(Z_1)$, $\mathrm{sym}(Y_1)$ and $\mathrm{sym}(X_1X_r)$. The final observable $\mathcal{H} = \mathrm{sym}(Z_1)$ is the same as for Model A. Since this model is not of the form of a commuting-generator or commuting-block circuit, gradient evaluation will be assumed to be via the parameter-shift rule.

Model C: Quantum convolutional neural network–We also train a quantum convolutional quantum neural network [48, 49]. We construct the model based on the architecture in [49], using a 10-parameter convolutional circuit (circuit 7 of Fig. 2 therein). Here the observable is given by $\mathcal{H} = Z_d$. As with Model B, gradients are assumed to be evaluated via the parameter-shift rule.

Model D: Separable quantum model—As a sanity check, we also train a fully separable quantum model. This model has 3 trainable parameters per qubit that are used to optimise individual single-qubit rotations via the parameter-shift method. The observable is $\mathcal{H} = \mathrm{sym}(Z_1)$.

6.3 Calculating the required shots

Since we are interested in the number of shots used to train the models, we would ideally estimate gradients using a finite number of shots from each quantum circuit. However, the classical simulation of finite-shot gradients becomes very resource intensive at larger qubit numbers, making this approach inconveniently expensive. For this reason we use numerically exact statevector simulations and automatic differentiation, which returns the exact gradient and can in principle result in different behaviour compared to a finite-shot scenario. To account for this, we add independent Gaussian noise with zero mean and standard deviation $\epsilon = 0.1$ to the gradient vector before updating the parameters. This roughly approximates the shot noise that one would encounter from sampling $M = 1/\epsilon^2 = 10000$ shots from each quantum circuit involved in gradient evaluation.

We now proceed to calculate the number of shots needed to estimate the gradient for a fixed input $\boldsymbol{x}$ for each of the four models. For additional details of the calculation, also see App. F. For Model A, the gradient for each Z_r can be estimated using the parallel gradient method of Sec. 5.1. For example, for $r = 1$, if we consider the generator $\mathrm{sym}(X_1)$ with parameter θ_1 then

$$\begin{aligned}\frac{\partial C}{\partial \theta_1} &= \langle 0| V_{\boldsymbol{x}}^\dagger U^\dagger(\boldsymbol{\theta}) [i\,\mathrm{sym}(X_1), Z_1] U(\boldsymbol{\theta}) V_{\boldsymbol{x}} |0\rangle \\ &= \langle 0| V_{\boldsymbol{x}}^\dagger U^\dagger(\boldsymbol{\theta}) \left(O_1 + \cdots + O_d\right) U(\boldsymbol{\theta}) V_{\boldsymbol{x}} |0\rangle ,\end{aligned} \tag{47}$$

where $O_j = i[X_j, Z_1]$. It follows that each component of the gradient is obtained by summing contributions from operators O_j of the form (34), which we have seen can be performed in parallel. Since we need to do this for each Z_r in $\mathcal{H}$, the gradient evaluation requires $M \cdot d$ shots in total.

For Model B, similarly to above, considering the first generator $\mathrm{sym}(Z_1)$ with parameter θ_1 we

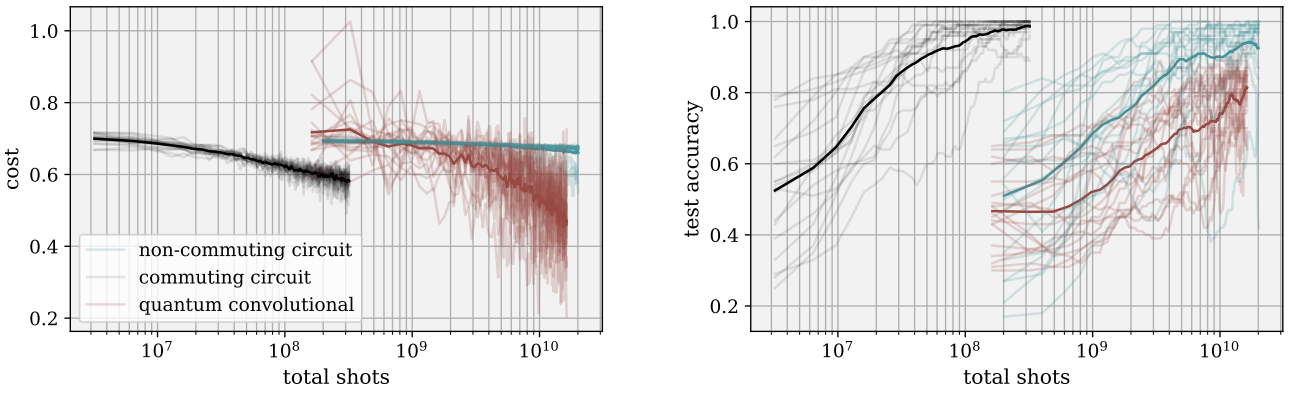


Figure 6: Numerical results for the 'bars and dots' learning problem using 16-qubit models and numerically exact gradient evaluation (simulated using PennyLane [54] and JAX [55]). All models are trained using the cross entropy loss and we assume $M = 10000$ shots per circuit used for gradient estimation. The plots show the results after 100 gradient steps using the adam optimiser, for 19 trials per model (average values given by thick lines). The highest average accuracy is achieved by the commuting-generator model, despite requiring two orders of magnitude fewer shots.

find

$$\frac{\partial C}{\partial \theta_1} = i \langle 0| V_{\boldsymbol{x}}^\dagger \mathrm{sym}(Z_1) U^\dagger(\boldsymbol{\theta}) \mathcal{H} U(\boldsymbol{\theta}) V_{\boldsymbol{x}} |0\rangle - i \langle 0| V_{\boldsymbol{x}}^\dagger U^\dagger(\boldsymbol{\theta}) \mathcal{H} U(\boldsymbol{\theta}) \mathrm{sym}(Z_1) V_{\boldsymbol{x}} |0\rangle \tag{48}$$

which is equivalent to a sum of d partial derivatives for gates with generators Z_r by expanding the sum in $\mathrm{sym}(Z_1)$. Via the parameter-shift rule we therefore need to evaluate two circuits per Pauli generator used in the circuit. Note that since one measures $\mathcal{H}$ directly here, we do not need to evaluate a different circuit for each Z_r appearing in $\mathcal{H}$, unlike in the parallel method. The total number of shots required is $M_B = M\left[d^2(L-1) + 3d(L+1) - 2\right]$ for L layers on d qubits (see App. F for details).

For Model C, we need to be more careful, because not all gates are generated by Pauli words, since the convolutional network also uses controlled Pauli rotations. We compute the required shots to be $M_C = M(56d - 80)$.

Finally, for Model D the gradient of each Z_r depends only on qubit r because the circuit is separable. Thus we can use the parameter-shift rule in parallel and a full gradient evaluation requires only $6M$ shots.

We see that the parallel method scales with d, whereas the parameter-shift rule scales with the number of Pauli generators: in a regime in which the number of parameters or the number of gates is much larger than the number of qubits, this can have a dramatic effect on shot efficiency. In our numerical experiment, Model B systematically suffers from this because it uses $\mathcal{O}(Ld^2)$ gates, whereas Model C merely suffers from a constant-factor overhead of 56, which is significantly larger than $d = 16$. This means that Model C asymptotically requires the same number of shots as Model A, namely $\mathcal{O}(Md)$, however at the price of being limited to $\mathcal{O}(d)$ parameters. In Table 1 we show the number of parameters and circuits for each model for the case $d = 16$ considered here.

Finally, we note that in the case of Model A, since gradients are estimated in parallel, correlations could exist between measurement outcomes that correlate the noise in the gradient vector in undesirable ways, although this will be less of a problem the higher the precision of the estimate. To investigate this, in App. E we present results for a 6-qubit version of Model A, using a complete finite shot analysis with 500 shots per circuit, and observe that the behaviour of the model closely matches that of the exact gradient.

7 Results

In Fig. 6 (a) we show the results for the 16-dimensional bars and dots problem, considering 19 trials for each model, and 100 training steps for each trial. The results for Model D can be found in Fig. 9 in the appendix; we do not include them here since the model performs very poorly compared to the others. For Model A, we consider a model containing all generators up to 3-body

Model	# parameters n	# circuits
A (commuting)	44	16
B (non-commuting)	40	1006
C (q. convolutional)	48	816
D (separable)	48	6

Table 1: Specifications and resource requirements for each model for the 16-qubit classification problem we consider. n is the number of trainable parameters, and # circuits is the number of unique circuits that need to be evaluated in order to estimate the gradient for a given input $\boldsymbol{x}$.

interactions (so $K = 3$), and for Model B we set the number of layers to $L = 4$. This ensures that the models have approximately the same number of trainable parameters. In Tab. 1 we show the number of parameters and the number of circuits required for gradient evaluation.

The horizontal axes of Fig. 6 show the (cumulative) number of shots used for the training, computed as described above, and assuming $M = 10000$ shots per circuit. As expected, Model A requires dramatically fewer shots, and completes the 100 gradient steps before the first gradient update of the other models is available. For this problem, this translates to a roughly two orders of magnitude reduction in circuit shots.

The training dynamics of the three models vary significantly, with Model C exhibiting a strongly fluctuating batch loss. Interestingly, this model achieves a lower average batch loss than the other models, however this is not in contradiction to the poorer test accuracy. Consider for example a model whose label prediction probabilities are close to $\frac{1}{2}$ on all inputs, but such that $\text{sign}\langle\mathcal{H}\rangle$ always returns the correct label. This model will have perfect accuracy, however the expected cross entropy on any dataset can be arbitrarily close to $-\log(\frac{1}{2}) \approx 0.69$. Essentially, a similar effect is at play here; Model C is more confident in its predictions than Model A and B (resulting in a lower cross entropy), however its prediction $\text{sign}\langle\mathcal{H}\rangle$ is more likely to be incorrect.

Both equivariant models (A and B) exhibit smoother training dynamics and achieve much higher test accuracies. This suggests that equivariant architectures of the form presented here may be better suited to learn translation invariant data than previous proposals of quantum convolutional models (however we stress that a more involved hyperparameter optimisation and longer training times would be required to make a conclusive statement). Most importantly, the restriction to commuting generators appears not to compromise the performance of Model A, which even achieves the highest average test accuracy of all models. We expect that the flatter training curve of Model B is a reflection of a flatter cost landscape for this model; this may be linked to the fact that model has a larger DLA than Model A.

8 Discussion

Although the results of Sec. 6 provide some indication that commuting-generator circuits can perform well at tasks of interest, it remains to be seen whether their reduced expressivity compared to general ansätze remains sufficient to expect advantages over classical models when tested at scale. An important route to clarify this issue would be to understand the limits of expressivity of commuting-block circuits. As we have seen in Sec. 4.1, increasing the number of blocks can result in a larger expressivity compared to single-block ansätze, however it remains unclear how far one can push this under the constraints on the circuit generators. Since these constraints involve mutual anticommutation between blocks, a natural starting point could be to consider maximal sets of mutually anticommuting Pauli tensors, of which it is known that there cannot be more that $2N + 1$ [56]. Such a set can be found for example by considering the Jordan-Wigner form of a set of Majorana operators, which we detail in App. C. We note that a recent work [57] has made significant progress in this direction, showing that the commuting-block generator construction offers and optimal trade-off between expressivity and gradient efficiency. A further work [58] increases expressivity by using probabilistic mixtures of commuting circuits.

An alternative route to increased expressivity would be to consider greedy layer-wise training. Here, one would train a sequence of layers in the order they are performed, where each layer contains mutually commuting generators. By viewing previously trained layers as part of the initial unitary V in a commuting-generator circuit, parameters can be trained in parallel. Note this goes beyond the commuting-block circuit ansatz

since we no longer require a fixed commutation relation between generators in differing layers. Although this strategy allows for universal quantum computation [59], it remains to be seen if layer-wise training is sufficient for good performance at scale.

Our results could also have significant impact across fields other than machine learning, in which quantum circuit optimisation plays a role. For example, can one construct problem-inspired ansätze based on our circuits in a similar spirit to QAOA? For the X-generator ansatz with the choice $V = \mathbb{I}$, it is known that no superpolynomial advantage with respect to the approximation ratio is possible relative to classical algorithms [60]. However, considering other choices of V or non-commuting layers via the commuting-block class may lead to faster variational algorithms for optimisation.

One other area that is in critical need of resource reductions is quantum chemistry, where the need to evaluate a large number of terms that comprise the Hamiltonian can render optimisation unfeasible in practice [61]. Much effort has been put into reducing this burden e.g. via clever groupings of mutually anticommuting terms [62, 63] or additional preprocessing steps of $\mathcal{H}$ [64], and our work can be seen as an additional method that attempts to find savings by exploiting commutation between circuit generators. It is nevertheless unclear whether our class of circuits can provide a useful structure for variational quantum chemistry: for example, parameterised gate sets that respect particle-number conservation in the Jordan-Wigner form do not form a commuting set and cannot obviously be put into a commuting-block form.

Finally, a key question with regards to quantum machine learning is to identify interesting symmetries that can be encoded into our circuits beyond that of the 1D translation symmetry considered here. Here, there is a possible link to quantum error correcting codes. Namely, suppose we have a data symmetry whose representation on the data-encoded quantum states corresponds to the logical operators of some error-correcting code. Then, since the code stabilisers commute with these operators, they can be used as generators to construct equivariant gates. Identifying data encodings and data symmetries that fit this framework may therefore lead to specific applications that leverage the backpropagation scaling of the quantum circuits proposed in this work.

9 Outlook

The overriding message we hope to convey with this work is that there is a strong need to move away from generic, unstructured models in quantum machine learning and towards specific circuit structures that are engineered to have desirable features. Indeed, works that consider unstructured quantum circuits as viable machine learning models often end up painting an overly pessimistic picture due to the disadvantages of working with an unreasonably large model class [37, 65, 66, 67]. With regards to gradient estimation, more focus on structured, non-universal circuits like those presented in this work may be necessary, since backpropagation scaling may be impossible for general models [18].

Although our work presents a potential route to efficient training for supervised problems such as classification or regression, we stress that more progress may be needed for generative problems, where access to log-probabilities—often required to train large classical generative models—is uncomfortably absent in quantum machine learning. In the long run, we hope that more focus on finding circuit architectures that can be trained at scale will lead to a 'neural network' moment in quantum machine learning, marked by the emergence of the specific building blocks necessary to construct scalable and powerful quantum learning models.

10 Acknowledgements

JB is grateful to Patrick Huembeli for initial discussions about this project. All authors thank Maria Schuld, Nathan Killoran and Josh Izaac for useful input and comments.

References

[1] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. "Learning representations by back-propagating errors". Nature **323**, 533–536 (1986).

[2] Atılım Günes Baydin, Barak A. Pearlmutter, Alexey Andreyevich Radul, and Jef-

frey Mark Siskind. "Automatic differentiation in machine learning: A survey". J. Mach. Learn. Res. **18**, 5595–5637 (2017).

[3] Marco Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, et al. "Variational quantum algorithms". Nature Reviews Physics **3**, 625–644 (2021). arXiv:2012.09265.

[4] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. "Parameterized quantum circuits as machine learning models". Quantum Science and Technology **4**, 043001 (2019).

[5] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. "Evaluating analytic gradients on quantum hardware". Phys. Rev. A **99**, 032331 (2019). arXiv:1811.11184.

[6] Javier Gil Vidal and Dirk Oliver Theis. "Calculus on parameterized quantum circuits" (2018). arXiv:1812.06323.

[7] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. "General parameter-shift rules for quantum gradients". Quantum **6**, 677 (2022).

[8] Oleksandr Kyriienko and Vincent E. Elfving. "Generalized quantum circuit differentiation rules". Phys. Rev. A **104**, 052417 (2021). arXiv:2108.01218.

[9] Panagiotis G Anastasiou, Nicholas J Mayhall, Edwin Barnes, and Sophia E Economou. "How to really measure operator gradients in ADAPT-VQE" (2023). arXiv:2306.03227.

[10] Afrad Basheer, Yuan Feng, Christopher Ferrie, and Sanjiang Li. "Alternating layered variational quantum circuits can be classically optimized efficiently using classical shadows" (2022). arXiv:2208.11623.

[11] Sachin Kasture, Oleksandr Kyriienko, and Vincent E. Elfving. "Protocols for classically training quantum generative models on probability distributions". Phys. Rev. A **108**, 042406 (2023). arXiv:2210.13442.

[12] Andrew Arrasmith, Lukasz Cincio, Rolando D Somma, and Patrick J Coles. "Operator sampling for shot-frugal optimization in variational algorithms" (2020). arXiv:2004.06252.

[13] Jonas M Kübler, Andrew Arrasmith, Lukasz Cincio, and Patrick J Coles. "An adaptive optimizer for measurement-frugal variational algorithms". Quantum **4**, 263 (2020).

[14] Charles Moussa, Max Hunter Gordon, Michal Baczyk, M Cerezo, Lukasz Cincio, and Patrick J Coles. "Resource frugal optimizer for quantum machine learning". Quantum Science and Technology **8**, 045019 (2023).

[15] Kosuke Ito. "Latency-aware adaptive shot allocation for run-time efficient variational quantum algorithms" (2023). arXiv:2302.04422.

[16] Zhiyan Ding, Taehee Ko, Jiahao Yao, Lin Lin, and Xiantao Li. "Random coordinate descent: a simple alternative for optimizing parameterized quantum circuits". Physical Review Research **6**, 033029 (2024).

[17] Guillaume Verdon, Jason Pye, and Michael Broughton. "A universal training algorithm for quantum deep learning" (2018). arXiv:1806.09729.

[18] 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). arXiv:2305.13362.

[19] Jun Li, Xiaodong Yang, Xinhua Peng, and Chang-Pu Sun. "Hybrid Quantum-Classical Approach to Quantum Optimal Control". Phys. Rev. Lett. **118**, 150503 (2017). arXiv:1608.00677.

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

[21] Michael J. Bremner, Ashley Montanaro, and Dan J. Shepherd. "Average-case complexity versus approximate simulation of commuting quantum computations". Phys. Rev. Lett. **117**, 080501 (2016). arXiv:1504.07999.

[22] Xiaotong Ni and Maarten Van Den Nest. "Commuting quantum circuits: Efficient classical simulations versus hardness results". Quantum Info. Comput. **13**, 54–72 (2013).

[23] Jiaqing Jiang, Xiaoming Sun, Shang-Hua Teng, Bujiao Wu, Kewen Wu, and Jialin Zhang. "Optimal space-depth trade-off of cnot circuits in quantum logic synthesis". In Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms. Pages 213–229. SIAM (2020).

[24] Ophelia Crawford, Barnaby van Straaten, Daochen Wang, Thomas Parks, Earl Campbell, and Stephen Brierley. "Efficient quantum measurement of Pauli operators in the presence of finite sampling error". Quantum **5**, 385 (2021).

[25] Tzu-Ching Yen, Vladyslav Verteletskyi, and Artur F Izmaylov. "Measuring all compatible operators in one series of single-qubit measurements using unitary transformations". Journal of chemical theory and computation **16**, 2400–2409 (2020). arXiv:1907.09386.

[26] Pranav Gokhale, Olivia Angiuli, Yongshan Ding, Kaiwen Gui, Teague Tomesh, Martin Suchara, Margaret Martonosi, and Frederic T Chong. "Minimizing state preparations in variational quantum eigensolver by partitioning into commuting families" (2019). arXiv:1907.13623.

[27] Johannes Jakob Meyer. "Fisher Information in Noisy Intermediate-Scale Quantum Applications". Quantum **5**, 539 (2021).

[28] Géza Tóth and Iagoba Apellaniz. "Quantum metrology from a quantum information science perspective". Journal of Physics A: Mathematical and Theoretical **47**, 424006 (2014).

[29] Kok Chuan Tan, Varun Narasimhachar, and Bartosz Regula. "Fisher information universally identifies quantum resources". Phys. Rev. Lett. **127**, 200402 (2021). arXiv:2104.01763.

[30] James Stokes, Josh Izaac, Nathan Killoran, and Giuseppe Carleo. "Quantum Natural Gradient". Quantum **4**, 269 (2020).

[31] Matthew L Goh, Martin Larocca, Lukasz Cincio, M Cerezo, and Frédéric Sauvage. "Lie-algebraic classical simulations for variational quantum computing" (2023). arXiv:2308.01432.

[32] Rolando D Somma. "Quantum computation, complexity, and many-body physics" (2005). arXiv:quant-ph/0512209.

[33] Andrew M Childs and Nathan Wiebe. "Hamiltonian simulation using linear combinations of unitary operations". Quantum Information and Computation **12**, 0901–0924 (2012). arXiv:1202.5822.

[34] Michael Ragone, Bojko N Bakalov, Frédéric Sauvage, Alexander F Kemper, Carlos Ortiz Marrero, Martin Larocca, and M Cerezo. "A unified theory of barren plateaus for deep parametrized quantum circuits" (2023). arXiv:2309.09342.

[35] Domenico d'Alessandro. "Introduction to quantum control and dynamics". CRC press. (2021).

[36] Martin Larocca, Piotr Czarnik, Kunal Sharma, Gopikrishnan Muraleedharan, Patrick J. Coles, and M. Cerezo. "Diagnosing Barren Plateaus with Tools from Quantum Optimal Control". Quantum **6**, 824 (2022).

[37] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. "Barren plateaus in quantum neural network training landscapes". Nature communications **9**, 4812 (2018).

[38] Enrico Fontana, Dylan Herman, Shouvanik Chakrabarti, Niraj Kumar, Romina Yalovetzky, Jamie Heredge, Shree Hari Sureshbabu, and Marco Pistoia. "The adjoint is all you need: Characterizing barren plateaus in quantum ansätze" (2023). arXiv:2309.07902.

[39] Roeland Wiersema, Efekan Kökcü, Alexander F Kemper, and Bojko N Bakalov. "Classification of dynamical lie algebras for translation-invariant 2-local spin systems in one dimension" (2023). arXiv:2309.05690.

[40] Edward Grant, Leonard Wossnig, Mateusz Ostaszewski, and Marcello Benedetti. "An initialization strategy for addressing barren plateaus in parametrized quantum circuits". Quantum **3**, 214 (2019).

[41] Stefan H Sack, Raimel A Medina, Alexios A Michailidis, Richard Kueng, and Maksym Serbyn. "Avoiding barren plateaus using classical shadows". PRX Quantum **3**, 020365 (2022).

[42] Kaining Zhang, Liu Liu, Min-Hsiu Hsieh, and Dacheng Tao. "Escaping from the barren plateau via gaussian initializations in deep variational quantum circuits". Advances in Neural Information Processing Systems **35**, 18612–18627 (2022).

[43] Tyler Volkoff and Patrick J Coles. "Large gradients via correlation in random parameterized quantum circuits". Quantum Science and Technology **6**, 025008 (2021).

[44] Andrea Skolik, Jarrod R McClean, Masoud Mohseni, Patrick van der Smagt, and Martin Leib. "Layerwise learning for quantum neural networks". Quantum Machine Intelligence **3**, 1–11 (2021).

[45] Manuel S Rudolph, Enrico Fontana, Zoë Holmes, and Lukasz Cincio. "Classical surrogate simulation of quantum systems with lowesa" (2023). arXiv:2308.09109.

[46] Lars Simon, Holger Eble, Hagen-Henrik Kowalski, and Manuel Radons. "Interpolating parametrized quantum circuits using blackbox queries" (2023). arXiv:2310.04396.

[47] Louis Schatzki, Martin Larocca, Frederic Sauvage, and Marco Cerezo. "Theoretical guarantees for permutation-equivariant quantum neural networks" (2022). arXiv:2210.09974.

[48] Iris Cong, Soonwon Choi, and Mikhail D Lukin. "Quantum convolutional neural networks". Nature Physics **15**, 1273 (2019).

[49] Tak Hur, Leeseok Kim, and Daniel K Park. "Quantum convolutional neural network for classical data classification". Quantum Machine Intelligence **4**, 3 (2022).

[50] Joseph Bowles, David Wierichs, and Chae-Yeun Park. code: XanaduAI/backprop_scaling_pqcs.

[51] Quynh T Nguyen, Louis Schatzki, Paolo Braccia, Michael Ragone, Patrick J Coles, Frederic Sauvage, Martin Larocca, and M Cerezo. "Theory for equivariant quantum neural networks" (2022). arXiv:2210.08566.

[52] Michael Ragone, Paolo Braccia, Quynh T Nguyen, Louis Schatzki, Patrick J Coles, Frederic Sauvage, Martin Larocca, and M Cerezo. "Representation theory for geometric quantum machine learning" (2022). arXiv:2210.07980.

[53] Johannes Jakob Meyer, Marian Mularski, Elies Gil-Fuster, Antonio Anna Mele, Francesco Arzani, Alissa Wilms, and Jens Eisert. "Exploiting symmetry in variational quantum machine learning". PRX Quantum **4**, 010328 (2023).

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

[55] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. "JAX: composable transformations of Python+NumPy programs" (2018).

[56] Rahul Sarkar and Ewout van den Berg. "On sets of maximally commuting and anticommuting pauli operators". Research in the Mathematical Sciences **8**, 14 (2021).

[57] Koki Chinzei, Shinichiro Yamano, Quoc Hoan Tran, Yasuhiro Endo, and Hirotaka Oshima. "Trade-off between gradient measurement efficiency and expressivity in deep quantum neural networks" (2024). arXiv:2406.18316.

[58] Brian Coyle, El Amine Cherrat, Nishant Jain, Natansh Mathur, Snehal Raj, Skander Kazdaghli, and Iordanis Kerenidis. "Training-efficient density quantum machine learning" (2024). arXiv:2405.20237.

[59] Seth Lloyd. "Quantum approximate optimization is computationally universal" (2018). arXiv:1812.11075.

[60] Juneseo Lee, Alicia B Magann, Herschel A Rabitz, and Christian Arenz. "Progress toward favorable landscapes in quantum combinatorial optimization". Physical Review A **104**, 032401 (2021).

[61] Jules Tilly, Hongxiang Chen, Shuxiang Cao, Dario Picozzi, Kanav Setia, Ying Li, Edward Grant, Leonard Wossnig, Ivan Rungger, George H. Booth, and Jonathan Tennyson. "The variational quantum eigensolver: A review of methods and best practices". Physics Reports **986**, 1–128 (2022). arXiv:2111.05176.

[62] Vladyslav Verteletskyi, Tzu-Ching Yen, and Artur F. Izmaylov. "Measurement optimization in the variational quantum eigensolver using a minimum clique cover". The Journal of Chemical Physics **152**, 124114 (2020). arXiv:1907.03358.

[63] Tzu-Ching Yen, Vladyslav Verteletskyi, and Artur F Izmaylov. "Measuring all compatible operators in one series of single-qubit measurements using unitary transformations". Journal of chemical theory and computation **16**, 2400–2409 (2020). arXiv:1907.09386.

[64] William J Huggins, Jarrod R McClean, Nicholas C Rubin, Zhang Jiang, Nathan Wiebe, K Birgitta Whaley, and Ryan Babbush. "Efficient and noise resilient measurements for quantum chemistry on near-term quantum computers". npj Quantum Information **7**, 1–9 (2021).

[65] Eric R Anschuetz and Bobak T Kiani. "Quantum variational algorithms are swamped with traps". Nature Communications **13**, 7760 (2022).

[66] Eric R Anschuetz. "Critical points in quantum generative models" (2021). arXiv:2109.06957.

[67] Ernesto Campos, Aly Nasrallah, and Jacob Biamonte. "Abrupt transitions in variational quantum circuit training". Physical Review A **103**, 032607 (2021).

A Proof of Corollary 3

The theorem follows from the results of [22] where it is shown that it is possible to estimate the expectation value

$$\langle x| U^\dagger (Z \otimes \mathbb{I} \otimes \cdots \otimes \mathbb{I}) U |x\rangle \tag{49}$$

for any circuit U that is comprised of exponentiated commuting Pauli operators $e^{-i\theta P}$ to precision ϵ in $\text{poly}(N, \frac{1}{\epsilon})$ runtime. Here $|x\rangle$ is any computational basis input. For a circuit with $\mathcal{H} = F^\dagger Z_1 F$ we need to estimate

$$C(\boldsymbol{\theta}) = \langle 0| V^\dagger U^\dagger(\boldsymbol{\theta}) F^\dagger Z_1 F U(\boldsymbol{\theta}) V |0\rangle \tag{50}$$

$$= \langle 0| V^\dagger F^\dagger \tilde{U}^\dagger(\boldsymbol{\theta}) Z_1 \tilde{U}(\boldsymbol{\theta}) F V |0\rangle \tag{51}$$

for some other commuting circuit $\tilde{U}(\boldsymbol{\theta})$ with Pauli generators $F G_j F^\dagger$. If $FV|0\rangle = |x\rangle$ our expression therefore takes the same form as (49). Since sampling the quantum circuit results in an ϵ-estimate of $C(\boldsymbol{\theta})$ in run time $\frac{1}{\epsilon^2}$, it follows that the same precision can be achieved by the classical algorithm with at most polynomial overhead in N.

We note that for 'strong' classical simulation of expectation values (i.e. exponential accuracy), classical simulability does not hold in general [22], however the quantum circuit itself cannot achieve this precision.

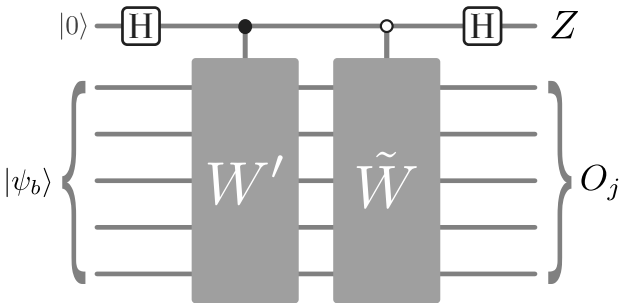


Figure 7: The circuit used to estimate the partial derivatives of a commuting block.

B Gradient estimation of commuting-block circuits

Here we prove Thm. 3. Consider a commuting-block ansatz with B blocks. The cost $C(\boldsymbol{\theta})$ can be written as

$$C(\boldsymbol{\theta}) = \langle \psi_b| W^\dagger \mathcal{H} W |\psi_b\rangle \tag{52}$$

where

$$|\psi_b\rangle = U_b(\boldsymbol{\theta}^b) \cdots U_2(\boldsymbol{\theta}^2) U_1(\boldsymbol{\theta}^1) V |0\rangle \tag{53}$$

is the state after the b^{th} block and

$$W = U_B(\boldsymbol{\theta}^B) \cdots U_{b+2}(\boldsymbol{\theta}^{b+2}) U_{b+1}(\boldsymbol{\theta}^{b+1}) \tag{54}$$

is the parameterised circuit after the b^{th} block. The partial derivatives of the b^{th} block are then

$$\frac{\partial C}{\partial \theta_j^b} = \langle \psi_b| \left(i G_j W^\dagger \mathcal{H} W - i W^\dagger \mathcal{H} W G_j \right) |\psi_b\rangle , \tag{55}$$

where G_j is the generator of the j^{th} unitary in block b. Since all generators in the block share the same commutation relation with other blocks, we have

$$WG_j = G_j\tilde{W} \quad \forall\, G_j \tag{56}$$

for some other unitary $\tilde{W}$. For example, if generators from all other blocks anticommute with those of block j then $\tilde{W}(\boldsymbol{\theta}) = W(-\boldsymbol{\theta})$ since

$$G_j e^{i\theta_k G_k} = e^{-i\theta_k G_k} G_j \tag{57}$$

for $\{G_k, G_j\} = 0$. We may now write (55) as

$$\frac{\partial C}{\partial \theta_j^b} = \langle\psi_b| \left(\tilde{W}^\dagger i G_j \mathcal{H} W - (-1)^{g_j} W^\dagger i G_j \mathcal{H} \tilde{W}\right) |\psi_b\rangle\,, \tag{58}$$

Where $g_j = 0$ or $g_j = 1$, depending on whether G_j commutes or anticommutes with $\mathcal{H}$. Defining $O_j = i^{g_j} G_j \mathcal{H}$, (58) takes the form

$$\begin{aligned}
\frac{\partial C}{\partial \theta_j^b} &= \langle\psi_b| \left(\tilde{W}^\dagger O_j W' + (W')^\dagger O_j \tilde{W}\right) |\psi_b\rangle && (59)\\
&= \frac{1}{2}\Big[\langle\psi_b| \left(\tilde{W}^\dagger + (W')^\dagger\right) O_j \left(W' + \tilde{W}\right) |\psi_b\rangle \\
&\quad + \langle\psi_b| \left(\tilde{W}^\dagger - (W')^\dagger\right) O_j \left(W' - \tilde{W}\right) |\psi_b\rangle\Big] && (60)\\
&= \frac{1}{2}\left[\langle O_j\rangle_{L_W^+|\psi_b\rangle} - \langle O_j\rangle_{L_W^-|\psi_b\rangle}\right] && (61)
\end{aligned}$$

where we first defined the modified unitary $W' = i^{1-g_j} W$ and then introduced the linear combinations of unitaries $L_W^\pm = \tilde{W} \pm W'$.

Consider the circuit depicted in Fig. 7, which applies controlled-W' instead of W in the original circuit, followed by a controlled-$\tilde{W}$ controlled on the first qubit being in state $|0\rangle$. The prepared state is

$$|\phi_b\rangle = \frac{1}{2}\left[|0\rangle L_W^+ |\psi_b\rangle + |1\rangle L_W^- |\psi_b\rangle\right]. \tag{62}$$

If we then measure the expectation value of $\tilde{O}_j = 2(Z \otimes O_j)$ on this state we obtain

$$\begin{aligned}
\langle\tilde{O}_j\rangle_{|\phi_b\rangle} =& \frac{1}{2}\Big[\langle 0| Z |0\rangle \langle\psi_b| (L_W^+)^\dagger O_j L_W^+ |\psi_b\rangle \\
&+ \langle 1| Z |1\rangle \langle\psi_b| (L_W^-)^\dagger O_j L_W^- |\psi_b\rangle\Big] \\
=& \frac{\partial C}{\partial \theta_j^b}. && (63)
\end{aligned}$$

Similar to the single-block case we may confirm that

$$[O_j, O_k] = 0 \quad \forall\, j, k. \tag{64}$$

for generators with the same value of g_j. The observables can therefore be simultaneously diagonalised and evaluated in parallel and we need to evaluate at most 2 circuits to estimate all partial derivatives of any given block, one for the generators that commute with $\mathcal{H}$ and one for those that anticommute. Since the derivative is estimated as a standard circuit expectation value, its variance scales as $1/M$ with M the number of shots.

For the last block we note that $\tilde{W} = W = \mathbb{I}$ so that we may use parallel estimation without any auxiliary qubits. Alternatively, this case can be interpreted as single-block commuting-generator circuit by absorbing all other blocks into V. The total number of circuits required for full gradient estimation of B blocks is therefore $2B - 1$.

C Maximal set of mutually anticommuting Pauli tensors

A maximal set of $2N + 1$ mutually anticommuting Pauli products can be found as follows. Here we give the construction for five qubits; the generalisation to N qubits is straightforward. In the Jordan-Wigner representation, the Majorana operators look like:

$$\begin{aligned}
P_1 &= X\ I\ I\ I\ I \\
P_2 &= Z\ X\ I\ I\ I \\
P_3 &= Z\ Z\ X\ I\ I \\
P_4 &= Z\ Z\ Z\ X\ I \\
P_5 &= Z\ Z\ Z\ Z\ X \\
P_6 &= Y\ I\ I\ I\ I \\
P_7 &= Z\ Y\ I\ I\ I \\
P_8 &= Z\ Z\ Y\ I\ I \\
P_9 &= Z\ Z\ Z\ Y\ I \\
P_{10} &= Z\ Z\ Z\ Z\ Y,
\end{aligned} \tag{65}$$

which can be seen to mutually anticommute. To these we may add $P_{11} = ZZZZZ$, giving the desired set of $2N+1 = 11$ mutually anticommuting operators.

D The diagonalising unitaries for the X-generator ansatz

Here we show that the circuits described in the main text diagonalise the operators $O_j = 2iG_j\mathcal{H}$ and $O_{jk} = -4G_jG_k\mathcal{H}$ in the product X basis for $\mathcal{H}$ given by (35). That is, we show

$$DO_jD^\dagger = 2G_j \tag{66}$$

and

$$DH^{(1)}_{zy}O_{jk}H^{(1)}_{zy}D^\dagger = (-1)^{\beta_1}4X^{(1)}G_jG_k. \tag{67}$$

We first prove (66). Let us denote the number of X operators of G_j on the first m qubits as m_j. Since we need only consider generators that anticommute with $\mathcal{H}$, we have that m_j is odd and the O_j take the form

$$O_j = 2s_jP_{j,1}\otimes\cdots\otimes P_{j,m}\otimes G_{j,m+1}\otimes\cdots\otimes G_{j,N}, \tag{68}$$

where $P_{j,r} \in Z, Y$ with an odd number of Ys. The sign factor takes the form

$$s_j = i^{m_j-1} = \begin{cases} 1 & \text{for } m_j \bmod 4 = 1 \\ -1 & \text{for } m_j \bmod 4 = 3 \end{cases}. \tag{69}$$

Since the O_j all contain X or $\mathbb{I}$ operators on the last $n-m$ qubits, D needs only to act on the first m qubits. Our task is therefore to diagonalise operators $\{P^j_1\otimes P^j_2\otimes\cdots\otimes P^j_m\}$ where $P^j_i \in \{Y, Z\}$ and there are an odd number of Y's.

To do this we first apply an X-Y Hadamard to each qubit, so we now have operators

$$(-1)^{m-m_j}P_{j,1}\otimes P_{j,2}\otimes\cdots\otimes P_{j,m} \tag{70}$$

with $P_{j,i} \in \{Z, X\}$ with an odd number of X's. We then apply the sequence of CZ operators between every pair of qubits. To make things clearer, we consider a specific operator which we show in Fig. 8. We call the qubits that have a Z/X operator 'Z qubits' or 'X qubits'. Since $CZ(Z\otimes Z)CZ = Z\otimes Z$, CZ operators that act of Z sites have no effect. Those that act between a given Z site and an X site (green lines in Fig. 8) have the effect of changing the Z site to an identity operator. This follows from the property

$$CZ(X\otimes\mathbb{I})CZ = X\otimes Z. \tag{71}$$

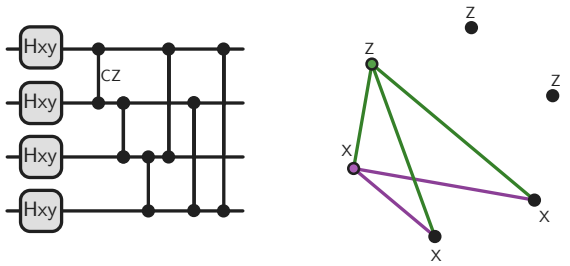


Figure 8: The diagonalising unitary (left) and a visual aid for the proof of App. D (right). Note that the order of operations in the diagonalising circuit matches the order in which we apply them to the original (non-diagonal) observable, because we implement $D^\dagger$ instead of D and because the operations are applied to the observable in reversed order compared to the quantum state.

Since there are an odd number of X operators, the combined effect is to multiply the Z site by an odd number of Z operators, thus converting it to an identity. We are thus left with all the CZ operators that act between two X sites (purple lines). From (71) and $XY = iZ$ we have

$$CZ(X\otimes X)CZ = Y\otimes Y, \tag{72}$$
$$CZ(Y\otimes X)CZ = -X\otimes Y. \tag{73}$$

Let us label the X sites as $1,\cdots,m_j$. From (72), if we apply all CZ operators between the first site and the rest, the effect of this is (i) alternate the operator at site 1 between X and Y an even number of times, thus ending as an X operator, and (ii) pick up a factor of $(-1)^{\lfloor(m_j-1)/2\rfloor} = -1$. In this process the remaining X sites now have a Y operator. Applying all CZs between site 2 and the rest now results in an odd number of flips (leaving site 2 with an X operator), and picking up a factor $(-1)^{\lfloor(m_j-2)/2\rfloor} = +1$. The remaining m_j-2 sites now have an X operator as before. Continuing this process we see that the X operators remain unchanged, but we pick up a factor

$$(-1)^{\left(\frac{m_j-1}{2}\right)^2} = \begin{cases} 1 & \text{for } m_j \bmod 4 = 1 \\ -1 & \text{for } m_j \bmod 4 = 3 \end{cases}. \tag{74}$$

This factor matches precisely the factor s_j in (69) which multiplies the operators. It follows that the action of the whole circuit D is given by (66).

We now prove (67). The possible O_{jk} take a similar form to (68):

$$O_{jk} = 4s_{jk}P_{j,1}\otimes\cdots\otimes P_{j,m}\otimes G_{j,m+1}\otimes\cdots\otimes G_{j,N},$$

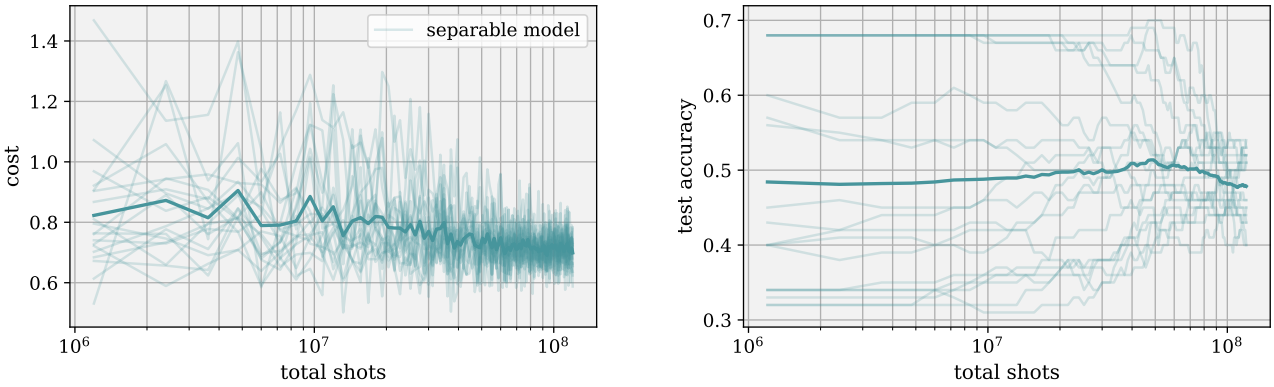


Figure 9: Results for the 16-qubit benchmarking problem of Sec. 6 using the separable model (Model D).

where now we have an even number m_{jk} of Y operators on the first m qubits and

$$s_{jk} = (-1)^{\frac{m_{jk}}{2}+1} = \begin{cases} 1 & \text{for } m_{jk} \bmod 4 = 2 \\ -1 & \text{for } m_{jk} \bmod 4 = 0 \end{cases}. \tag{75}$$

Note that if we apply an z-y Hadamard operator on the first qubit, we map any O_{jk} to an operator that now has an odd number of Y's on the first m qubits, which is precisely the case we had before. After applying D, from (74) and (75) we see that the sign factor we obtain matches s_{jk} if G_jG_k has an X operator on the first qubit; if not we pick up a factor -1. We therefore find $DH^{(1)}_{zy}O_{jk}H^{(1)}_{zy}D^\dagger = (-1)^{\beta_1}4X^{(1)}G_jG_k$ as claimed.

E finite-shot numerics

In Fig. 10 we show training and accuracy curves for Model A from Sec. 6 for a 6-dimensional bars and dots problem, using finite-shot analysis and 500 shots per quantum circuit used in the gradient evaluation. Here, the model has access to up to 6-body generators, i.e. $K = 6$. The black dotted curve corresponds to the learning trajectory if the exact gradient were provided instead of that given by the finite shots. As can be seen, the finite shot updates approximate well the exact behaviour.

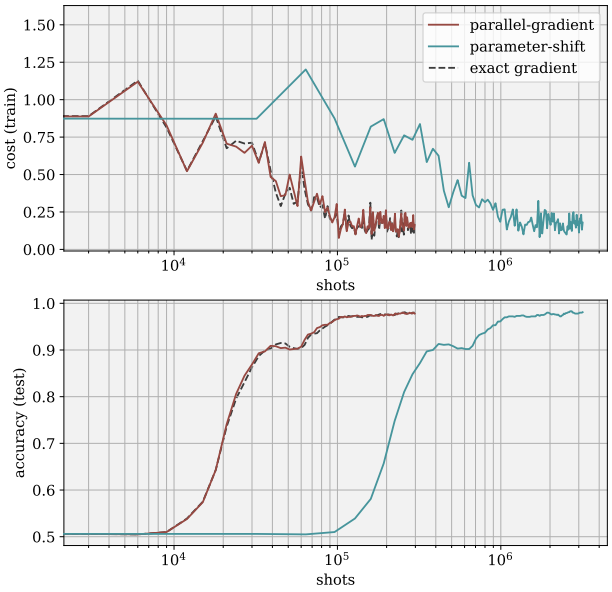


Figure 10: Results for the 6-qubit problem with finite shots described in App. E, using 500 shots per circuit.

F Details on the required shot counts

Here we present a few details on the numbers of shots required to compute a gradient for the four models discussed in Sec. 6.

For Model A, the calculation of the required shots given in the main text is already sufficiently precise and we have $M_A = M \cdot d$.

For Model B, the number of Pauli generators can be evaluated to be

$$\mathcal{P} = L\left(2d + \frac{d^2 - d}{2}\right) = L \cdot \frac{d}{2} \cdot (d+3), \quad (76)$$

where the parentheses in the first expression consists of the contributions for the two single-qubit layers and for the block of two-qubit gates. Each generator is a Pauli word and thus requires two parameter-shifted circuits, leading to a naïve count of $M_B = MLd(d+3)$. However, in the last layer of the circuit we may exploit commutativity within the block of two-qubit gates in order to execute the parameter-shift rule for $d/2$ of these gates simultaneously. This reduces the number of different circuits for the last two-body block from $d^2 - d$ to $\frac{d^2-d}{d/2} = 2(d-1)$. The resulting corrected shot count for a full gradient evaluation of Model B is then

$$\begin{aligned} M_B &= M\left[(L-1)d(d+3) + (6d-2)\right] \\ &= M\left[d^2(L-1) + 3d(L+1) - 2\right]. \end{aligned} \quad (77)$$

In Model C, we use $-1 + \sum_{j=1}^{\log_2 d} 2^j = 2d - 3$ two-qubit convolution blocks with 10 gates each, where we assumed that d is a power of two. Two of these gates are controlled Pauli rotations, which require four instead of two parameter-shifted circuits each. Thus we require $(2d-3) \cdot (8 \cdot 2 + 2 \cdot 4) = 24(2d-3)$ circuits for the gradient w.r.t. the convolution parameters. In addition, the model uses $\sum_{j=0}^{\log_2 d - 1} 2^j = d - 1$ pooling blocks, with two controlled Pauli rotations each. This leads to $(d-1) \cdot (2 \cdot 4) = 8(d-1)$ circuits to compute the gradient w.r.t. the pooling parameters. Overall, we find the shot cost for Model C to be

$$M_C = M\left[24(2d-3) + 8(d-1)\right] = M(56d - 80). \quad (78)$$

Lastly, the gradient for the separable Model D can be computed in parallel between the different qubits, as they do not interact at all. There are $3d$ Pauli generators, of which d can be treated in parallel, leading to shot cost of $M_D = M \cdot 2 \cdot 3d/d = 6M$.