Efficient Gradient Estimation of Variational Quantum Circuits with Lie Algebraic Symmetries

Mohsen Heidari MHEIDAR@IU.EDU
Department of Computer Sciences, Indiana University, Bloomington, IN, USA

Masih Mozakka MMOZAKKA@IU.EDU
Department of Computer Sciences, Indiana University, Bloomington, IN, USA

Wojciech Szpankowski SZPAN@PURDUE.EDU
Department of Computer Sciences, Purdue University, West Lafayette, IN, USA

Abstract

Hybrid quantum-classical optimization and learning strategies are among the most promising approaches to harnessing quantum information or gaining a quantum advantage over classical methods. However, efficient estimation of the gradient of the objective function in such models remains a challenge due to several factors including the exponential dimensionality of the Hilbert spaces, and information loss of quantum measurements. In this work, we study generic parameterized circuits in the context of variational methods. We develop a framework for gradient estimation that exploits the algebraic symmetries of Hamiltonian characterized through Lie algebra or group theory. Particularly, we prove that when the dimension of the dynamical Lie algebra is polynomial in the number of qubits, one can estimate the gradient with polynomial classical and quantum resources. This is done by a series of Hadamard tests applied to the output of the ansatz with no change to its circuit. We show that this approach can be equipped with classical shadow tomography to further reduce the measurement shot complexity to scale logarithmically with the number of parameters.

1. Introduction

Variational quantum algorithms (VQAs) are among the hybrid quantum-classical strategies most prominent for quantum optimization and learning [BLSF19, CAB+21]. First introduced as variational quantum eigensolver (VQE) [PMS+14], VQAs have been studied in a wide range of topics including optimization [FGG14], quantum chemistry [JEM+19, AWGP21,GEBM19,DAJ+21], and quantum machine learning [FN18,SK19,MNKF18,LW18, HCT+19, HGS22]. VQAs are implemented via an ansatz which is a parameterized quantum circuit (PQC). Several approaches suggest ways of creating a compact ansatz with shallow circuit depth and high accuracy. The ansatz can be layers of simple parametric gates acting on one or two qubits. More complex structures have been studied for Hamiltonian variational ansatzs (HVAs) [WHT15, KMT+17]. The unitary variant of coupled cluster theory truncated at single and double excitations (UCCSD) is the first of such an ansatz [PMS+14]. Compactness and robustness against barren plateaus are some of the benefits of such models [HWO+19, FHC+23, LCS+22].

In general, the ansatz is represented by a generic parameterized operator as

$$U(\overrightarrow{a}) = e^{iA(\overrightarrow{a})},$$

where $\overrightarrow{a} \in \mathbb{R}^p$ is the vector of adjustable model parameters, and p the number of the parameters. Whichever way one implements the parameters and the ansatz, the objective is to minimize a predefined loss evaluated via a (Hermitian) *observable* O for a specifically prepared input state as

$$\mathcal{L}(\overrightarrow{a}) := \operatorname{tr}\Big\{O\ U(\overrightarrow{a})\rho U(\overrightarrow{a})^\dagger\Big\}, \tag{1}$$

where ρ is the input (mixed) state *density operator*. To ensure computational tractability, it is assumed that the number of parameters $p = \mathsf{poly}(d)$, with d being the number of qubits. With this notion, the objective is to find $\min_{\overrightarrow{a}} \mathcal{L}(\overrightarrow{a})$. In the context, of quantum machine learning, ρ represents the expectation of the quantum samples, and O is the incurred prediction loss. In the problem of finding the ground state of a Hamiltonian, O represents the underlying Hamiltonian and the ρ often is the initial state, such as the Hartee-Fock state.

Gradient Estimation. Compared to derivative-free methods, gradient-based optimizers have shown a significant advantage in terms of convergence guarantees [HN21, SWM$^+$20] and have been widely used in the literature [SBG$^+$18, FGG14, FN18, SK19, MNKF18]. At each iteration, an unbiased estimate of $\nabla\mathcal{L}(\overrightarrow{a})$ is obtained, and the parameters are updated accordingly. One of the main challenges is to estimate the gradient efficiently in terms of the classical and quantum resources defined through the following measures:

(i) The number of unique circuit configurations and gate complexity;

(ii) The number of measurement shots (sample complexity);

(iii) Classical computational complexity.

The number of unique circuit configurations is motivated by the existing noisy intermediate-scale quantum (NISQ) implementations and the fact that quantum circuit reconfiguration tends to be a more costly task than generating measurement samples after executing each unique circuit [WIWL22]. The complexity of gradient estimation is negatively affected in quantum settings as the dimension of the associated Hilbert space is exponential in d and the terms in the Hamiltonian $A(\overrightarrow{a})$ are non-commuting. Moreover, other quantum properties that might contribute negatively are the stochasticity of quantum measurements, associated state collapse, and no-cloning.

The focus of this paper is on the efficient estimation of the gradient with polynomial complexity with respect to the above three measures. The well-known parameter shift rule (PSR) [SBG$^+$18,MNKF18] relies on the Hadamard test with Pauli operators to estimate the partial derivatives without any change to the ansatz. This is done via a simple circuit appended to the output [MNKF18]. However, PSR is restricted to gates with two distinct eigenvalues. Therefore, one may ask the following question: *is it possible to estimate the gradient using the same Hadamard tests as in PSR for general ansätze and without any change to their circuits?*

Existing works introduce alternative approaches to generalize PSR [BC21, WIWL22, The23]. Such methods require several changes to the ansatz circuit configuration or have high classical complexity. In this work, we give an affirmative answer to the above question when the Hamiltonian has certain algebraic symmetries characterized through group theoretic structures defined for Pauli matrices. We show that the gradient can be estimated in polynomial classical time and a linear number of Hadamard tests for Pauli strings.

1.1 Summary of the main results

We consider a generic parameterized unitary of the form $e^{i(A(\overrightarrow{a})+B)}$, where B is a fixed Hermitian component and A is the parameterized component with p parameters. Our approach is based on a series of Hadamard tests on the output of the ansatz followed by polynomial time classical post-processing. The Hadamard test corresponding to a Pauli string P_j measures the following quantity

$$D_j := i\,\mathrm{tr}\big\{O[P_j, \rho^{out}]\big\}, \tag{2}$$

where $\rho^{out} = U(\overrightarrow{a})\rho U(\overrightarrow{a})^\dagger$ is the output state of the ansatz. This observable can be implemented using a circuit with a few control rotation gates (see Figure 1). Mitarai *et al.* [MNKF18] used the Hadamard test to estimate the derivative when the ansatz has a simple form $U(\theta) = e^{i\theta P_i}$. However, it is not clear whether it works when the ansatz is the exponential of a generic Hamiltonian $A(\overrightarrow{a})$. The main difficulty is due to the non-commuting terms appearing in $A(\overrightarrow{a})$.

We make use of the powerful theory of Lie algebra that offers insightful perspectives in quantum physics and increasingly becoming important in quantum computing. It enables to capture of the essential features of the underlying symmetries and can be used to analyze the spectrum, eigenstates, and dynamics of quantum systems. Typically, a Hamiltonian is described by a linear combination of terms that correspond to a certain physical interaction. Such terms can be used to generate a Lie algebra, which is called the Hamiltonian algebra or dynamical Lie algebra (DLA) [SPS02, WKKB23]. Often, the DLA is a sub-algebra of $\mathfrak{su}(2^d)$ which is the vector space of all skew-Hermitian $2^d \times 2^d$ matrices with the standard commutator.

We first consider a slightly simpler Hamiltonian of the form $A(\overrightarrow{a}) = \sum_{i=1}^{p} a_i P_i$, where $a_i \in \mathbb{R}$ and P_i's are Pauli strings. We define a groupstructure on the Pauli operators P_i. We show that when the Pauli terms in the decomposition of $A(\overrightarrow{a})$ form a subgroup, then Hadamard tests with proper classical post-processing can be used to estimate the gradient $\nabla\mathcal{L}$. Our first result is abbreviated as follows.

Theorem 1 (abbreviated) *Suppose the Pauli strings appearing in $A(\overrightarrow{a})$ are closed under the commutation, that is $[P_i, P_j] = P_k$ for some Pauli string appearing in $A(\overrightarrow{a})$ up to a constant for all $i, j, k \in \{1, \cdots, p\}$. Then, there is an algorithm that estimates $\nabla\mathcal{L}(\overrightarrow{a})$ with an element wise additive error ϵ using $\tilde{O}(\frac{p}{\epsilon^2})$ Hadamard tests and $O(p^3 + pd)$ classical time*[1].

For a complete statement of the result see Theorem 5 and Theorem 6 in Section 3.2 and 3.3. A Hamiltonian with mutually commuting Pauli strings is a trivial example. However, the Pauli strings in the above theorem do not commute in general. Moreover, the ansatz is not necessarily local, though, a special example is a k-junta Hamiltonian, that is when A acts non-trivially on at most k out of d qubits.

Overview of the techniques. Our theoretical results rely on the interplay among various representations captured by Lie algebra, and group theoretic structures. We first make a connection between the partial derivatives of $\mathcal{L}(\overrightarrow{a})$ to the *adjoint* operator which characterizes the derivative of a differentiable operator exponential. Then, we represent the

[1] The notation $\tilde{O}$ hides logarithmic factors.

expressions using the stabilizer formulation that has been studied in quantum error correction [CRSS96, Got97]. This representation enables us to make a connection between the matrix differential in Lie algebra and specific group actions involving the vectors over the Klein four-group and $\mathbb{Z}_2 \oplus \mathbb{Z}_2$. With this framework, the partial derivatives of $\mathcal{L}(\overrightarrow{a})$ can be written as an infinite-length linear combination of expectation values of Hadamard tests on the output of the ansatz. Then, we show that the number of Hadamard tests in this decomposition is bounded by the dimensionality of the Hamiltonian algebra. Next, we show that such an infinite summation can be written as a classical matrix exponential with a size equal to the dimensionality of the Hamiltonian Lie algebra. Hence, when the DLA dimensionality is polynomial in d, the number of qubits, the partial derivative can be computed efficiently with d Hadamard tests and $\mathsf{poly}(d)$ classical time. The Hadamard tests are similar to those developed for single-gate ansätze. Moreover, the existing results on statistical estimations of quantum systems, such as shadow tomography can be used to measure the partial derivatives simultaneously and to reduce the costs.

Next, we study a more general Hamiltonian structure where $A(\overrightarrow{a})$ is decomposed into generic Hermitian terms. We show that when the Hamiltonian DLA has $\mathsf{poly}(d)$ dimensionality, then the gradient $\nabla\mathcal{L}$ can be estimated polynomially. This brings us to the following result:

Theorem 2 (abbreviated) *Suppose the variational terms in the Hamiltonian $A(\overrightarrow{a})$ belong to a sub-DLA with $\mathsf{poly}(d)$ dimensionality, then there is an algorithm that estimates $\nabla\mathcal{L}(\overrightarrow{a})$ with $\mathsf{poly}(d)$ tests and additional $\mathsf{poly}(d)$ classical time.*

The complete statement is given as Theorem 15 is Section 3.5. Hamiltonians with polynomial-size DLA are especially important in the context of avoiding the barren plateaus [LCS+22, FHC+23] corresponding to the exponential diminish of the gradient in the training of VQAs [MBS+18, CSV+21]. Recently it was shown that the variance of the gradient is inversely proportional to the dimension of the DLA [FHC+23], implying that VQAs with polynomial-size DLAs may not exhibit barren plateaus. This motivates us to restrict or attention to polynomial-size DLAs. Nevertheless, there still is a curiosity to understand the gradient estimation when DLA dimensionality grows exponentially with d. A classification of dynamical Lie algebras and bounds on their dimensionality has been thoroughly studied in [WKKB23]. Here, we provide an example for more intuition.

Example 1 *The transverse-field Ising model is an example of a Hamiltonian such that the dimensionality of the DLA is $O(d^2)$ with the Hamiltonian characterized as*

$$H = \sum_{i=1}^{d} Z_i Z_{i+1} + X_i,$$

where X_i, Z_i are the corresponding Pauli operators.

This paper's results are steps toward a unified framework for efficiently estimating the gradient of an arbitrary parameterized circuit without making changes to its unitary. Moreover, the Hadamard tests allow one to further enhance the estimations using existing methods for estimating the expectation values of observables. We highlight some of these implications below.

Shadow tomography. Turning the gradient estimation to a series of Hadamard tests has another benefit that can further reduce the number of shots to $O(\log p)$. This can be done using the classical shadow tomography [HKP20] — a procedure to estimate several observables with minimal sample complexity. Suppose the observable O in (1) is k-local, meaning that it decomposes into a finite sum of observables acting non-trivially on at most k qubits. Then, we can prove that the number of shots scales logarithmically with p.

Corollary 3 (abbreviated) *In Theorem 1, suppose additionally that the observable O is k-local. Then there exists an algorithm that estimates $\nabla\mathcal{L}(\overrightarrow{a})$ with $O(\frac{1}{\epsilon^2}4^k \log p)$ ansatz uses and additional $\tilde{O}(p^3 2^{\Theta(k)} + pd)$ classical time.*

This result relies on the observation that when O is k-local, then so are the Hadamard tests in (2). In that case, one can use classical shadow tomography for local observables. See Corollary 20 in Section 5.1 for detailed statement of the result.

Joint measurability. One benefit of performing the estimations via the proposed approach is the applicability of measuring groups of the partial derivatives jointly. The Hadamard tests in (2) are observables some of which are jointly measurable. For instance, the Ising model in Example 1 has several terms that are commuting with each other. Joint measurability helps reduce the ansatz use and sample complexity for estimating the gradient. For that, the Pauli strings can be grouped into commuting collections. Then, the strings in each collection can be measured simultaneously with a single reference measurement, as they all can be diagonalized by a single unitary.

Corollary 4 *Suppose $\mathcal{F}_1, \cdots, \mathcal{F}_m$ form the mutually compatible groups of the Pauli strings appearing in a parameterized Hamiltonian $A(\overrightarrow{a}) = \sum_i a_i P_i$. Then, the number of joint Hadamard tests in Theorem 1 can be reduced to $\tilde{O}(\frac{m}{\epsilon^2})$.*

Jointly measuring Pauli strings has been studied extensively in the literature and several methods have been introduced for grouping the Paulies [LBZ02,VYI20,CvSW$^+$20,BBRV01].

1.2 Comparison with related methods

We consider the three complexity measures we mentioned earlier. Firstly, the number of distinct circuits that need to be evaluated to obtain all the partial derivatives. Secondly, the overall number of measurements or circuit shots. Thirdly, the classical post-processing time. Although this measure is less restrictive, it is important to ensure it scales polynomially with the number of qubits. Our approach in the context of Theorem 1 and Corollary 3 requires no change to the ansatz, with $\log(p)$ Hadamard unique circuits, and $\mathsf{poly}(p, d)$ classical time. Below, we highlight some of the most relevant approaches for comparison to our work.

Stochastic PSR: This method is a generalization of PSR [BC21, WIWL22], where each partial derivative is written as an integral, and a Monte Carlo strategy is used to estimate it. For an ansatz with p parameters, it requires p unique circuits each with changes to the structure of the ansatz. In practice, such modifications require a re-evaluation of the schedule of the underlying quantum-control system and hence are at a disadvantage. Moreover, the stochastic PSR has a high estimation variance. This is because an integral is estimated by sampling values of its integrand with a finite-shot estimate. David *et al.* [WIWL22] presented a neat connection to the Discrete Fourier series and introduced a method that is efficient when the Hamiltonian A is promised to have equidistant eigenvalues. However, for a generic ansatz one first needs to compute the spectral decomposition of A to find the pattern of the parameter shifts. In that case, this process would take an exponential classical time as A is an exponentially large matrix.

Nyquist PSR: Recently [The23] proposed a shift rule for PQCs where only the parameters are shifted without any other modifications of the ansatz. The method relies on a beautiful connection between the Nyquist-Shannon Sampling theorem and the Fourier series that was observed earlier in [WIWL22, VT18]. The number of unique circuits for this estimation scales with p and the difference between the maximum and minimum eigenvalues of A — a quantity bounded by the operator norm $\|A\|$. As the authors reported, this method has low approximation error when the parameter value is large enough. More precisely, the approximation error is $O(\frac{1}{c^2})$ as long as $\theta = (1-\Omega(1))c$, where c is the maximum magnitude of a parameter value.

Approaches based on the special unitary group. This is another approach [WLW+24] based on Lie algebra and a nice connection to the geometry of $SU(2^d)$ matrices and the adjoint operator. The paper argues that the number of unique parameter shift rules is bounded by the dimensionality of the DLA that contains $A(\overrightarrow{a})$. However, this approach has a classical run time scaling as $p2^{\theta(d)}$ as it relies on finding the Jacobian matrix of $U(\overrightarrow{a})$.

2. Preliminaries and Model

Notation. For any $d \in \mathbb{N}$, let H_d be a the Hilbert space of d-qubits. By $\mathcal{B}(H_d)$ denote the space of all bounded (linear) operators acting on H_d. The identity operator is denoted by I_d. As usual, a quantum state, in its most general form, is denoted by a *density operator*; that is a Hermitian, unit-trace, and non-negative linear operator. A quantum measurement/observable is modeled as a positive operator-valued measure (POVM). A POVM $\mathcal{M}$ is represented by a set of operators $\mathcal{M} := \{M_v, v \in \mathcal{V}\}$, where $\mathcal{V}$ is the (finite) set of possible outcomes. The operators M_v are non-negative and form a resolution of identity, that is $M_v = M_v^\dagger \geq 0, \sum_v M_v = I_d$. The operator norm (infinity norm) for an operator A is defined as

$$\|A\| = \sup_{|\phi\rangle} \|A\,|\phi\rangle\|.$$

2.1 Variational Quantum Algorithms

VQAs are used for solving problems in optimization, learning, and simulation. They consist of an ansatz as a parameterized quantum operation to be tuned in a quantum-classical hybrid loop. The ansatz is modeled as a (unitary) $U(\overrightarrow{a})$, with $\overrightarrow{a} \in \mathbb{R}^p$ being the vector of adjustable model parameters, and p the number of the parameters. It can be constructed explicitly by concatenating multiple layers of smaller parametric units. This structure is sometimes referred to as a quantum neural network (QNN). To ensure computational tractability, it is assumed that $p = \mathsf{poly}(d)$, with d being the number of qubits. Recall that the objective of a VQA is to minimize the loss given in (1). Note that the loss is an expectation value and even the mere act of measuring the loss causes state collapse. Consequently, one expects that quantum optimization and learning problems are more sample-intensive compared to their classical analogs. This is particularly problematic when the quantum states are expensive to produce.

Making iterative progress in the direction of the steepest descent is one of the most popular optimization techniques in VQAs, as it has been in classical problems. Ideally, a gradient descent optimizer applies the following update rule at each iteration t:

$$\overrightarrow{a}^{(t+1)} = \overrightarrow{a}^{(t)} - \eta_t \nabla \mathcal{L}(\overrightarrow{a}^{(t)}) \tag{3}$$

where $\eta_t \in \mathbb{R}$ is the learning rate at iteration t. The above update rule is not realistic as the objective function $\mathcal{L}(\overrightarrow{a})$ is an expectation value, and the characteristics of ρ is either unknown or computationally not tractable. Hence, one needs gradient estimates — an extensively studied topic in the literature. A useful approach is one that is efficient in both computational and sample complexity. Several approaches have been introduced to efficiently estimate the gradient of the loss under various restrictions on the ansatz.

Product ansätze. A special variant of the ansatz is a concatenation of single parametric or non-parametric unitary circuits of the form

$$U(\overrightarrow{a}) = U_L(a_L)V_L \cdots U_1(a_1)V_1,$$

where V_l is non-parametric and each parametric layer is of the form

$$U_l(a_l) := e^{i a_{\mathbf{s}_l} \sigma^{\mathbf{s}_l}} \tag{4}$$

with $\sigma^{\mathbf{s}_l}$ being a Pauli string. There are several solutions for VQAs using product anästze.

Parameter Shift Rule. There have been several approaches to estimating the gradient [FN18, MNKF18, SWM+20, HGS22, HN21, SBG+18, MKF19, WLW+24]. The zeroth-order approach (e.g., finite differences) evaluates the objective function in the neighborhood of the parameters. Although it is a generic approach, recent studies showed their drawbacks in terms of convergence rate [HN21]. First-order methods (e.g., parameter shift rule) directly calculate the partial derivatives [SBG+18]. When the ansatz is of the form $U(\overrightarrow{a}) = \prod_{j=1}^{p} e^{-i a_j G_j}$, with G_j the parameter shift rule implies that:

$$\frac{\partial \mathcal{L}}{\partial a_j} = \langle \mathcal{L}(\overrightarrow{a} + \frac{\pi}{4} e_j) \rangle - \langle \mathcal{L}(\overrightarrow{a} - \frac{\pi}{4} e_j) \rangle,$$

where $e_j \in \mathbb{R}^p$ is the jth canonical vector, that is $e_{j,j} = 1$ and $e_{j,r} = 0$ for all $r \neq j$. As shown in [MNKF18], the partial derivatives can be directly measured via a Hadamard test (see Fig. 1). More formally, the partial derivative of a product ansätze can be written in terms of commutators with associated Pauli operators. This statement is summarized below:

Fact 1 *Let $\rho_l^{out} = U_{\leq l} |\phi\rangle\langle\phi| U_{\leq l}^{\dagger}$ denote the density operator of the output state at layer l when the input is $|\phi\rangle$ with label y. Then, the derivative of the loss is given by:*

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{s}_l}} = \operatorname{tr}\Big\{ O U_{>l} \big[\sigma^{\mathbf{s}_l}, \rho_l^{out}\big] U_{>l}^{\dagger} \Big\}, \tag{5}$$

where $[\cdot, \cdot]$ is the commutator operation.

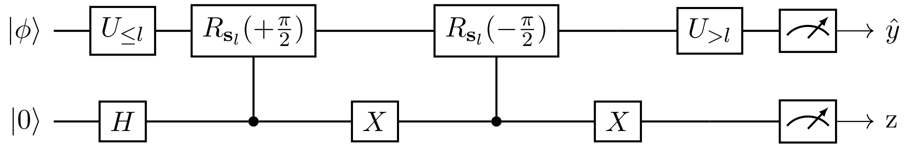
$|\phi\rangle$
$U_{\leq l}$
$R_{\mathbf{s}_l}(+\frac{\pi}{2})$
$R_{\mathbf{s}_l}(-\frac{\pi}{2})$
$U_{>l}$
$\hat{y}$
$|0\rangle$
H
X
X
z

Figure 1: Hadamard test for measuring the partial derivative of product with respect to a parameter $a_{\mathbf{s}_l}$ appearing at layer l. Here $U_{\leq l}$ corresponds to the first l layers of the ansatz, and $U_{>l}$ to the remaining layers. Here, X is the X-gate and $R_{\mathbf{s}_l}$ is the controlled rotation around Pauli $\sigma^{\mathbf{s}_l}$.

The expression above can be implemented using a quantum circuit with measurements at the end. The circuit is shown in Fig. 1 which is a special example of the generalized Hadamard test [MKF19, HN21]. In this procedure, given a sample $|\phi\rangle$ and an ancilla qubit $|0\rangle$, we first apply the first l layers of the ansatz and then apply a special circuit with controlled rotations for taking the derivative of the loss. Then, the rest of the layers of the ansatz are applied and measurement is performed.

2.1.1 Generic PQCs

A generic PQC is an ansätze of the form

$$U(\overrightarrow{a}) = e^{i(A(\overrightarrow{a}))}, \tag{6}$$

where $A(\overrightarrow{a})$ is the parameterized (traceless) Hermitian operator. For instance, A can be decomposed in terms of Pauli strings as

$$A(\overrightarrow{a}) = \sum_{\mathbf{s}} f_{\mathbf{s}}(\overrightarrow{a})\sigma^{\mathbf{s}},$$

where $\mathbf{s} \in \{0,1,2,3\}^d$ and $f_{\mathbf{s}} : \mathbb{R}^p \to \mathbb{R}$ is a real-valued function on the space of the parameters. The Pauli decomposition states that

$$f_{\mathbf{s}}(\overrightarrow{a}) = 2^{-d}\,\mathrm{tr}\{A(\overrightarrow{a})\sigma^{\mathbf{s}}\}.$$

Note that the ansatz can be multi-layered, but, for simplicity, we consider only a single-layer PQC. Due to the non-commutativity of the Pauli products, Fact 1 does not apply. A brief summary of some of the relevant approaches is provided in Appendix C.

3. Main Results

In what follows we present the main results of the paper. We introduce an approach for estimating the gradient of the loss for a generic ansatz. We show that the Hadamard test followed by classical post-processing leads to an approximation of gradient. We prove in Theorem 5 of Section 3.2 that the partial derivative is a linear combination of several terms similar to the product ansätze as in Fact 1. The coefficients in the linear combination are calculated through a series of group actions that we define over the stabilizer representation of Pauli strings. Then, in Theorem 6 in Section 3.3 we show that under certain group structures, the linear combination collapses into $\mathsf{poly}(p)$ terms. In Section 3.5, we extend this result to Hamiltonians with polynomial size DLA. Before presenting the To present the main results of the paper we need to introduce a few concepts and notations.

3.1 The Stabilizer Formulation

We first define a few actions on the Pauli strings. Basic definitions of group and Lie algebra are given in Appendix A.

First, the d-qubit Pauli group, denoted by $\mathcal{P}_d$, is defined as the set of all Pauli strings of the form $c\sigma^{\mathbf{s}}$, where $c \in \{\pm, \pm i\}$ and $\mathbf{s} \in \{0,1,2,3\}^d$ equipped with the matrix product as the group action. Ignoring the scalar c, the group quotient is *isomorphic* to the Binary vector group $\mathbb{Z}_2^{2d}$. Therefore, the Pauli strings abide a compact representation which we present in the following.

The representation associates the identity and each Pauli operator X, Y, Z with an element of $\mathbb{Z}_2 \oplus \mathbb{Z}_2$ which is the binary vector space with dimension two:

$$I = (0|0), \qquad X = (0|1), \qquad Y = (1|0), \qquad Z = (1|1),$$

where we use $(\cdot|\cdot)$ to denote the elements of $\mathbb{Z}_2 \oplus \mathbb{Z}_2$. For the d-qubit Pauli strings, consider the binary vector space $\mathbb{Z}_2^{2d}$, where each element is denoted by $(\mathbf{a}|\mathbf{b}), \mathbf{a}, \mathbf{b} \in \mathbb{Z}_2^d$. With this formulation, we can represent each Pauli string as

$$\sigma^{\mathbf{s}} \equiv (\mathbf{s}^0|\mathbf{s}^1), \qquad \forall \mathbf{s} \in \{0,1,2,3\}^d, \quad \forall (\mathbf{s}^0|\mathbf{s}^1) \in (\mathbb{Z}_2 \oplus \mathbb{Z}_2)^d,$$

where s_j^0 and s_j^1 give the most significant and least significant bit of the binary representation of $s_j \in \{0,1,2,3\}$. Next, we define two operations for $(\mathbb{Z}_2 \oplus \mathbb{Z}_2)^d$: the element-wise binary addition and a symplectic inner product given by

$$\langle (\mathbf{a}|\mathbf{b}), (\mathbf{a}'|\mathbf{b}') \rangle = \sum_{j=1}^{d} a_j b'_j + b_j a'_j, \qquad \forall (\mathbf{a}|\mathbf{b}), (\mathbf{a}'|\mathbf{b}') \in (\mathbb{Z}_2 \oplus \mathbb{Z}_2)^d.$$

We need to define another group action.

Klein four-group representation. Let $\mathcal{K}_4$ be the *Klein four-group* over the set $\{0,1,2,3\}$ with 0 being the group's identity and $\circ$ denoting the group addition. The group operation for distinct elements $a, b, c \in \mathcal{K}_4 - \{0\}$ is defined as

$$0 \circ a = a \qquad a \circ a = 0, \qquad a \circ b = b \circ a = c.$$

It is known that $\mathbb{Z}_2 \times \mathbb{Z}_2 \cong \mathcal{K}_4$. Hence, we can view the $\circ$ as an operation over $\mathbb{Z}_2^2$. Next, by $\mathcal{K}_4^d$ denote the usual vector of $\mathcal{K}_4$ with the element-wise addition $\circ$.

Next, we define a special operation on $\mathcal{K}_4$. For a pair of $s, r \in \mathcal{K}_4$ define

$$s \odot r := \mod{}_2(\langle (s^0|s^1), (r^0|r^1) \rangle) = \mod{}_2(s^0 r^1 + s^1 r^0),$$

where (s^0, s^1) and (r^0, r^1) are, respectively, the binary representation of s and r. We extend this notation for vectors $\mathbf{s}, \mathbf{r} \in \mathcal{K}_4^d$ as

$$\mathbf{s} \odot \mathbf{r} := \mod{}_2\big(\langle (\mathbf{s}^0|\mathbf{s}^1), (\mathbf{r}^0|\mathbf{r}^1) \rangle\big) = (s_1 \odot r_1) \oplus \cdots \oplus (s_d \odot r_d)$$

Signed product. Lastly, we proceed by defining a signed product over $\mathcal{K}_4^d$. For any pair $i, j \in \mathcal{K}_4$, define

$$\varepsilon_{i,j} := \begin{cases} -1 & \text{if } (i,j) = (2,1), (3,1), (3,2) \\ 1 & \text{otherwise} \end{cases}.$$

With these notation, the signed product of any pair $\mathbf{s}, \mathbf{r} \in \mathcal{K}_4^d$ is given by

$$\mathbf{s} \circledast \mathbf{r} := (\mathbf{s} \odot \mathbf{r}) \prod_{j=1}^{d} \varepsilon_{s_j, r_j}.$$

Example 2 *As an example, consider the single qubit case, where the elements take values from $\{0,1,2,3\}$. Table 1 demonstrates the $\circledast$ operation between pairs of such members.*

s/r	0	1	2	3
0	0	0	0	0
1	0	0	1	1
2	0	-1	0	1
3	0	-1	-1	0

Table 1: Table of the product $s \circledast r$ for the single qubit case.

Now, given the definitions for $\circ$, $\circledast$ and $\odot$, we are ready to present our main results.

3.2 Partial Derivative

Our first result is the decomposition of the partial derivatives into the expectation of the Hadamard tests with Pauli strings as in the product ansätze.

Theorem 5 *Consider an ansatz of the form $U(\overrightarrow{a}) = \exp\{iA(\overrightarrow{a})\}$, where $A(\overrightarrow{a}) = \sum_{\mathbf{s}\in\mathcal{S}} a_{\mathbf{s}}\sigma^{\mathbf{s}}$ for some $\mathcal{S} \subseteq \mathcal{K}_4^d$. For any $\mathbf{t} \in \mathcal{K}_4^d$, define*

$$D_{\mathbf{t}} := i\,\mathrm{tr}\big\{O\big[\sigma^{\mathbf{t}}, \rho^{out}\big]\big\},$$

where ρ^{out} is the output state. Then, the derivative of the objective function $\mathcal{L}$ for this ansatz expands in terms of $D_{\mathbf{t}}$ as:

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{r}}} = \sum_{k=0}^{\infty} \frac{(-2)^k}{(k+1)!} \sum_{\mathbf{s}_1\in\mathcal{S}} \cdots \sum_{\mathbf{s}_k\in\mathcal{S}} \prod_{j=1}^{k} a_{\mathbf{s}_j}\big(\mathbf{s}_j \circledast (\mathbf{s}_1 \circ \cdots \circ \mathbf{s}_{j-1} \circ \mathbf{r})\big) D_{\mathbf{s}_1\circ\cdots\circ\mathbf{s}_k\circ\mathbf{r}}, \tag{7}$$

for any $\mathbf{r} \in \mathcal{S}$.

The proof of the theorem is delayed until Section 3.4. With this theorem, one can compute the derivative of the loss for a non-product ansatz by treating it as a single-parameter ansatz as in Fact 1 but with correction terms. In other words, the derivative of loss can be measured by applying the standard Hadamard test followed by classical corrections. To highlight this result we present an example concerning a single qubit ansatz.

Example 3 *Consider a general single-qubit unitary of the form*

$$U(\overrightarrow{a}) = \exp\{ia_1\sigma^1 + ia_2\sigma^2 + a_3\sigma^3\}.$$

Let O be a generic observable and consider the associated loss $\mathcal{L}(\overrightarrow{a})$ as in (1). As an example, consider taking the partial derivative of $\mathcal{L}$ with respect to a_1 at the point $a_1 = 0$ and $a_3 = 0$. In the context of Theorem 5, let D_j be the result of the Hadamard test with Pauli σ^j, where $j = 1, 2, 3$. Then, (7) in Theorem 5 simplifies to the following:

$$\frac{\partial \mathcal{L}}{\partial a_1}(\overrightarrow{a} = (0, a_2, 0)) = \sum_{k=0}^{\infty} \frac{(-2)^k}{(k+1)!} \prod_{j=1}^{k} a_2 \left(2 \circledast (\underbrace{2 \circ \cdots \circ 2}_{k-1 \ \ times} \circ 1) \right) D_{\underbrace{2 \circ \cdots \circ 2}_{k \ \ times} \circ 1},$$

where we used the fact that only terms with all $\mathbf{s}_j = 2$ are surviving. Note that for even k, $\underbrace{2 \circ \cdots \circ 2}_{k \ \ times} \circ 1 = 1$, and for odd k it equals to $2 \circ 1 = 3$. Therefore, we have that

$$2 \circledast (\underbrace{2 \circ \cdots \circ 2}_{k-1 \ \ times} \circ 1) = \begin{cases} 2 \circledast 3 = 1 & even\ k \\ 2 \circledast 1 = -1 & odd\ k \end{cases}$$

where we used Table 1. Plugging such evaluations in the first equation, we have

$$\prod_{j=1}^{k} a_2 \left(2 \circledast (\underbrace{2 \circ \cdots \circ 2}_{k-1 \ \ times} \circ 1) \right) = a_2^k(-1) \times 1 \times (-1) \times \cdots = \begin{cases} a_2^k(-1)^{k/2} & even\ k \\ a_2^k(-1)^{(k+1)/2} & odd\ k \end{cases}$$

$$\frac{\partial \mathcal{L}}{\partial a_1}(\overrightarrow{a} = (0, a_2, 0)) = \sum_{p=0}^{\infty} \frac{(-2)^{2p}}{(2p+1)!} a_2^{2p}(-1)^p D_1 + \sum_{q=0}^{\infty} \frac{(-2)^{2q+1}}{(2q+2)!} a_2^{2q+1}(-1)^{q+1} D_3$$

Hence, one needs to measure D_1 and D_3 to compute the above partial derivative. Next, by simplifying the summations, it is not difficult to show that

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial a_1}(\overrightarrow{a} = (0, a_2, 0)) &= \frac{-1}{2a_2} \left(\sum_p \frac{(-2a_2)^{2p+1}}{(2p+1)!}(-1)^p \right) D_1 + \frac{-1}{2a_2} \left(\sum_q \frac{(-2a_2)^{2q+2}}{(2q+2)!}(-1)^{q+1} \right) D_3 \\ &= \frac{-1}{2a_2}\big(\sin(-2a_2)D_1 + (\cos(-2a_2) - 1)D_3\big) \\ &= \frac{1}{2a_2}\big(\sin(2a_2)D_1 + (1 - \cos(2a_2))D_3\big). \end{aligned}$$

Notice the presence of D_3 which relates to the Pauli σ^3 and not appear in the ansatz expression. One can verify that this is indeed equal to the analytic gradient of this ansatz (see Appendix B for more details).

In this example, we can see how our method is akin to computing the partial derivative in a closed-form expression. Interestingly, under certain structural properties, one can find the closed-form expression of the summation in Theorem 5. The power of this approach lies in the group structure of the Pauli matrices, which produces predictable patterns. We highlight examples of such structures in the next section. Another point worth mentioning is that the terms D_i only make use of ρ^{out} and hence are easily measurable.

3.3 Ansätze with Pauli Subgroup Structures

We consider special cases of the Hamiltonian for which the gradient can be computed efficiently. Note that when $\mathcal{S}$ is a subgroup of $\mathcal{K}_4^d$, then the terms appearing in (7) remain inside the subgroup. In that case, the partial derivative can be written as follows.

Theorem 6 *Suppose the Hamiltonian parameter set $\mathcal{S}$ is a subgroup of $\mathcal{K}_4^d$. Then, there is an algorithm that computes $\nabla\mathcal{L}$ in $O(p^3+pd)$ time with $O(p)$ use of the Hadamard test for Pauli strings as in (2).*

Note that here the run time depends on the number of relevant qubits that the ansatz acts. When the ansatz is a k-junta then the the run time will be $O(p^3+pk)$. Such examples might not be interesting as the overall unitary can be efficiently simulated in a classical computer. There are several subgroups of $\mathcal{K}_4^d$ for which the Hamiltonian is not k-junta. A simple example is

$$A = a_1 X^{\otimes d} + a_2 Y^{\otimes d} + a_3 Z^{\otimes d},$$

where the subgroup is isomorphic to $\mathcal{K}_4$ inside $\mathcal{K}_4^d$.

Proof sketch: To prove this theorem, we define a matrix representation of the derivative expression. Let $\mathcal{S} \leq \mathcal{K}_4^d$ be a subgroup of size p with elements denoted by $\mathbf{s}_1, \cdots, \mathbf{s}_p$. Note that since $\mathcal{S}$ is a subgroup, the infinite sum in the theorem reduces to a linear combination of their form

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{r}}} = \sum_{\mathbf{s}\in\mathcal{S}} g_{\mathbf{s}}(\mathbf{r}) D_{\mathbf{s}},$$

where $g_{\mathbf{s}}(\mathbf{r}) \in \mathbb{R}$. Therefore, if we consider each $D_{\mathbf{s}}$ as a basis in a vector space $\mathbb{R}^p$, then $\frac{\partial \mathcal{L}}{\partial a_{\mathbf{r}}}$ is a vector in that space with the representation

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{r}}} \equiv (g_{\mathbf{s}_1}(\mathbf{r}), \cdots, g_{\mathbf{s}_p}(\mathbf{r})) \in \mathbb{R}^p.$$

Moreover, with this representation each $D_{\mathbf{s}_j} \equiv \mathbf{e}_j$, where $\mathbf{e}_j \in \mathbb{R}^p$ is the jth canonical basis vector. Now define the $p \times p$ matrix V where the ith column is given as

$$\mathbf{v}_i := 2\sum_{j=1}^{p} a_{\mathbf{j}}(\mathbf{s}_j \circledast \mathbf{s}_i)\ \mathbf{e}_{j\circ i}, \tag{8}$$

where $j \circ i$ is the shorthand for the index of $\mathbf{s}_j \circ \mathbf{s}_i$. Note that the above summation can be computed in $O(pd)$ time. Moreover, noting that $\mathbf{v}_i = V\mathbf{e}_i$ and given (7), the vector representation of the partial derivative of $\mathcal{L}$ equals to

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{s}_i}} \equiv \sum_{k\geq 0} \frac{(-1)^k}{(k+1)!} V^k \mathbf{e}_i, \qquad \forall i \in [p].$$

Next, compute the following matrix

$$B = (I - e^{-V})V^{-1},$$

where V^{-1} is the generalized inverse of V. Then, it is not difficult to check that B gives the vector representation of the partial derivatives. With a proper matrix exponentiation algorithm, this matrix is computed in $O(p^3)$. To compute the gradient from the vector representation of the derivative, one first needs to estimate all $D_{\mathbf{s}_i}, i \in [p]$. This can be done with $O(p)$ runs of the derivative-taking circuit of Figure 1. Lastly, the gradient is computed by multiplying the matrix with the vectors of $D_{\mathbf{s}}$'s as

$$\nabla \mathcal{L} = B\overrightarrow{D}_{\mathbf{s}}.$$

This procedure is summarized in the following algorithm. An advantage of this method is that all the partial derivatives are estimated via the matrix exponentiation of V.

Remark 7 *Note that empirical estimations of $D_{\mathbf{s}}$ can be done via several measurement shots. In a straightforward approach each $D_{\mathbf{s}}$ is estimated independent measurement shots. Hence, with the standard concentration analysis (McDiarmid inequality) one can show that all the $D_{\mathbf{s}}$ in the subgroup can be estimated with probability at least $(1-\delta)$ up to an additive error ϵ with*

$$O\left(p\frac{1}{\epsilon^2}\log\frac{p}{\delta}\right)$$

measurement shots.

Algorithm 1 Subgroup Gradient Estimation

```
Input: Parameter subgroup $\mathcal{S}$,
Estimate the expectation value of Hadamard tests $D_{\mathbf{s}}, \mathbf{s} \in \mathcal{S}$ as in (2).
Compute the matrix $V$ where the column $i$ is computed as in (8).
Compute the matrix $B = (I - e^{-V})V^{-1}$, where $V^{-1}$ is the generalized inverse of $V$.
Return $\nabla \mathcal{L} \approx B\overrightarrow{D}_{\mathbf{s}}$.
```

Example 4 *As a sanity check. We derive again the the same derivative expression as in Example 3, this time using Algorithm 1. One can verify that for the general single-qubit unitary we have*

$$V = \begin{bmatrix} 0 & 0 & 2a_2 \\ 0 & 0 & 0 \\ -2a_2 & 0 & 0 \end{bmatrix}.$$

In Example 3, we had $a_1 = a_3 = 0$. Using the singular value decomposition, the matrix $B = (I - e^{-V})V^{-1}$ is computed as

$$B = \begin{bmatrix} \frac{\sin(2b)}{2b} & 0 & \frac{1-\cos(2b)}{2b} \\ 0 & 0 & 0 \\ \frac{\cos(2b)-1}{2b} & 0 & \frac{\sin(2b)}{2b}. \end{bmatrix}$$

Next, having the Hadamard tests D_1, D_2, D_3, the gradient is computed as

$$\nabla \mathcal{L}(a_1 = 0, a_2, a_3 = 0) = B \begin{bmatrix} D_1 \\ D_2 \\ D_3 \end{bmatrix} = \begin{bmatrix} \frac{\sin(2b)}{2b} D_1 + \frac{1-\cos(2b)}{2b} D_3 \\ 0 \\ -\frac{1-\cos(2b)}{2b} D_1 + \frac{\sin(2b)}{2b} D_3 \end{bmatrix}$$

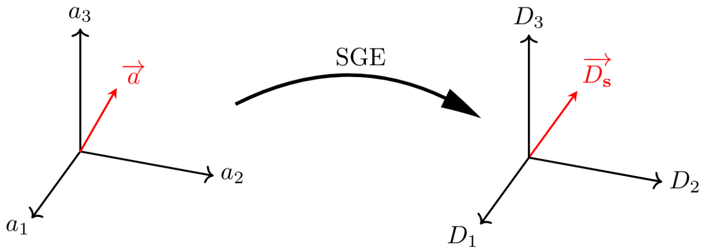

Figure 2: This figure illustrates the concept behind the subgroup gradient estimation algorithm. The expression in Theorem 6 points to an alternative representation of the problem. Instead of picturing the gradient as a function in the space of the parameters (the left picture) we can view it as a vector in the landscape of Hadamard tests $D_{\mathbf{s}}$.

The sum of which gives us the exact analytical expression for the derivative we got before. This example illustrates the connection between the space of $D_{\mathbf{s}}$ and the group nature of the Pauli operators.

3.4 Proof of Theorem 5

We start by taking the partial derivative of $\mathcal{L}(\overrightarrow{a})$. Noting that $\rho^{out} := U\rho U^\dagger$, the partial derivative of the loss with respect to $a_{\mathbf{s}}$ is

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{s}}} = \operatorname{tr}\left\{ O \frac{\partial}{\partial a_{\mathbf{s}}} \rho^{out} \right\},$$

where

$$\frac{\partial \rho^{out}}{\partial a_{\mathbf{s}}} = \frac{\partial U}{\partial a_{\mathbf{s}}} \left(\rho U^\dagger \right) + (U\rho) \frac{\partial U^\dagger}{\partial a_{\mathbf{s}}}, \tag{9}$$

and we used the fact that ρ^{out} is Fréchet differentiable with respect to $a_{\mathbf{s}}$.

Next, we need to introduce some ingredients in Lie algebra. We equip the space $\mathsf{GL}(2^d, \mathbb{C})$ of $2^d \times 2^d$ complex matrices with the Lie algebra and the standard Lie bracket defined as

$$[A, B] := AB - BA,$$

for any $A, B \in \mathsf{GL}(2^d, \mathbb{C})$. Such Lie algebra is denoted by $\mathsf{gl}(2^d, \mathbb{C})$ or g for shorthand. For more details on Lie algebra see [Ros06]. Now, we proceed by the adjoint representation. For any $X \in \mathsf{g}$, define the mapping $\mathsf{ad}_X : \mathsf{g} \to \mathsf{g}$ by $\mathsf{ad}_X(Y) = [X, Y]$. With this notation

$$\mathsf{ad}_X^k(Y) = [X, \cdots, [X, Y] \cdots].$$

The adjoint operator is connected to the derivative of the matrix exponential. In fact, from Theorem 5 of [Ros06], the differential of any exponential operator is given by

$$\frac{\mathrm{d} \exp\{X(\tau)\}}{\mathrm{d}\tau} = \exp\{X(\tau)\} \frac{1 - \exp\{-\mathsf{ad}_X\}}{\mathsf{ad}_X} \frac{\mathrm{d}X(\tau)}{\mathrm{d}\tau}. \tag{10}$$

where it is assumed that $X(\tau)$ is a differentiable (linear) operator with respect to a variable $\tau \in \mathbb{C}$.

Next, we prove the following lemma which is in a different ordering of the above equation.

Lemma 8 *Suppose $U = \exp\{iH(\tau)\}$ for some Hermitian operator valued function H. Then,*

$$\frac{\mathrm{d}U}{\mathrm{d}\tau} = -\frac{1-\exp\{i\mathsf{ad}_H\}}{\mathsf{ad}_H}(\frac{\mathrm{d}H(\tau)}{\mathrm{d}\tau})U,$$

where it is assumed that $H(\tau)$ is differentiable.

Proof First note that for any operator $X \in \mathsf{g}$

$$(\mathsf{ad}_X)^\dagger = \mathsf{ad}_{X^\dagger}.$$

We can write $U = (e^{-iH(\tau)})^\dagger$. Therefore, given that $\frac{\mathrm{d}X^\dagger}{\mathrm{d}\tau} = (\frac{\mathrm{d}X}{\mathrm{d}\tau})^\dagger$, from (10) we can write

$$\frac{\mathrm{d}U}{\mathrm{d}\tau} = \left(\frac{\mathrm{d}e^{-iH(\tau)}}{\mathrm{d}\tau}\right)^\dagger = \left(\exp\{-iH(\tau)\}\frac{1-\exp\{-\mathsf{ad}_{-iH}\}}{\mathsf{ad}_{-iH}}\frac{\mathrm{d}(-iH(\tau))}{\mathrm{d}\tau}\right)^\dagger$$

$$\left(\exp\{-iH(\tau)\}\frac{1-\exp\{+i\mathsf{ad}_H\}}{\mathsf{ad}_H}\frac{\mathrm{d}H(\tau)}{\mathrm{d}\tau}\right)^\dagger$$

$$= \left(\frac{1-\exp\{+i\mathsf{ad}_H\}}{\mathsf{ad}_H}\frac{\mathrm{d}H(\tau)}{\mathrm{d}\tau}\right)^\dagger U.$$

Note that for any $H \in \mathsf{g}$ we have the following equality by its convergent power series:

$$\frac{1-\exp\{-\mathsf{ad}_H\}}{\mathsf{ad}_H} = \sum_{k=0}^{\infty}\frac{(-1)^k}{(k+1)!}(\mathsf{ad}_H)^k. \tag{11}$$

Therefore, applying this equation for the operator $\frac{1-e^{i\mathsf{ad}_H}}{\mathsf{ad}_H}$, the derivative equals to the following

$$\left(-i\sum_{k=0}^{\infty}\frac{(i)^k}{(k+1)!}(\mathsf{ad}_H)^k(\frac{\mathrm{d}H}{\mathrm{d}\tau})\right)^\dagger U = i\sum_{k=0}^{\infty}\frac{(-i)^k}{(k+1)!}\left((\mathsf{ad}_H)^k(\frac{\mathrm{d}H}{\mathrm{d}\tau})\right)^\dagger U.$$

Note that for any $X, Y \in \mathsf{g}$ the following identity holds

$$(\mathsf{ad}_X(Y))^\dagger = -\mathsf{ad}^\dagger_X(Y^\dagger).$$

Therefore,

$$\frac{\mathrm{d}U}{\mathrm{d}\tau} = i\sum_{k=0}^{\infty}\frac{(i)^k}{(k+1)!}(\mathsf{ad}_H)^k(\frac{\mathrm{d}H}{\mathrm{d}\tau})U = -\frac{1-\exp\{+i\mathsf{ad}_H\}}{\mathsf{ad}_H}(\frac{\mathrm{d}H}{\mathrm{d}\tau})U.$$

This is the desired expression. ■

From this lemma, we have that

$$\frac{\partial U}{\partial a_{\mathbf{s}}} = -\frac{1-\exp\{i\mathsf{ad}_A\}}{\mathsf{ad}_A}(\frac{\partial A}{\partial a_{\mathbf{s}}})U.$$

This gives the first part of (9). Next, from (10), the partial derivative of $U^\dagger$ can be write as

$$\frac{\partial U^\dagger}{\partial a_{\mathbf{s}}} = U^\dagger\frac{1-\exp\{+i\mathsf{ad}_A\}}{\mathsf{ad}_A}(\frac{\partial A}{\partial a_{\mathbf{s}}}),$$

where we used the fact that A is Hermitian. Therefore, the partial derivative of ρ^{out} equals to

$$\begin{aligned}\frac{\partial \rho^{out}}{\partial a_{\mathbf{s}}} &= -\frac{1-\exp\{i\mathsf{ad}_A\}}{\mathsf{ad}_A}(\frac{\partial A}{\partial a_{\mathbf{s}}})U\rho U^\dagger + U\rho U^\dagger\frac{1-\exp\{i\mathsf{ad}_A\}}{\mathsf{ad}_A}(\frac{\partial A}{\partial a_{\mathbf{s}}})\\ &= \frac{1-\exp\{-i\mathsf{ad}_A\}}{\mathsf{ad}_A}(\frac{\partial A}{\partial a_{\mathbf{s}}})\rho^{out} + \rho^{out}\frac{1-\exp\{+i\mathsf{ad}_A\}}{\mathsf{ad}_A}(\frac{\partial A}{\partial a_{\mathbf{s}}}).\end{aligned}$$

By simplifying the terms in the right-hand side, the partial derivative can be written as the commutator:

$$\frac{\partial \rho^{out}}{\partial a_{\mathbf{s}}} = \Big[\rho^{out}, \frac{1-\exp\{i\mathsf{ad}_A\}}{\mathsf{ad}_A}(\frac{\partial A}{\partial a_{\mathbf{s}}})\Big].$$

Therefore,

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{s}}} = \mathrm{tr}\Big\{O\frac{\partial}{\partial a_{\mathbf{s}}}\rho^{out}\Big\} = \mathrm{tr}\Big\{O\Big[\rho^{out}, \frac{1-\exp\{i\mathsf{ad}_A\}}{\mathsf{ad}_A}(\frac{\partial A}{\partial a_{\mathbf{s}}})\Big]\Big\}.$$

Next, we use the definition of $\frac{1-e^{-\mathsf{ad}_X}}{\mathsf{ad}_X}$ in (11) to decompose the above summation. Setting, $X=-iA$ and from the fact that $\frac{\partial A}{\partial a_{\mathbf{s}}} = \sigma^{\mathbf{s}}$, the above quantity decomposes as

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{s}}} = -i\sum_{k=0}^{\infty}\frac{(i)^k}{(k+1)!}\,\mathrm{tr}\Big\{O\Big[\rho^{out}, (\mathsf{ad}_A)^k(\sigma^{\mathbf{s}})\Big]\Big\},$$

where the factor $-i$ appears because of the denominator.

It remains to write the above equation in terms of the group actions we defined. For that, we need to establish a few connections between the Lie bracket and the group actions.

Lemma 9 *The Lie bracket between any pair of Pauli strings $\sigma^{\mathbf{s}}, \sigma^{\mathbf{r}}$ is given by*

$$[\sigma^{\mathbf{s}}, \sigma^{\mathbf{r}}] = 2i(\mathbf{s} \circledast \mathbf{r})\sigma^{\mathbf{s}\circ\mathbf{r}},$$

The proof of this lemma is built upon the following lemma.

Lemma 10 *The product of any pair of Pauli strings equals the following*

$$\sigma^{\mathbf{s}}\sigma^{\mathbf{r}} = i^{\mathbf{s}\odot\mathbf{r}}\sigma^{\mathbf{s}\circledast\mathbf{r}}.$$

Proof We study single qubit products. Let $a \neq b \neq c$ be distinct non-zero elements of $\mathcal{K}_4$. Also without loss of generality suppose $a < b$. Then, it is not difficult to verify the following relations

$$\begin{aligned}
0 \circledast a &= 0, & 0 \circ a = a &\Rightarrow \sigma^0 \sigma^a = \sigma^a \\
a \circledast a &= 0, & a \circ a = 0 &\Rightarrow \sigma^a \sigma^a = \sigma^0 \\
a \circledast b &= 1, & a \circ b = c &\Rightarrow \sigma^a \sigma^a = i\sigma^c \\
b \circledast a &= -1, & a \circ b = c &\Rightarrow \sigma^a \sigma^a = -i\sigma^c,
\end{aligned}$$

where we used the fact that $i^{-1} = -i$. Hence, we established the lemma for $d = 1$. For general d, with the tensor product, we have that

$$\begin{aligned}
\sigma^{\mathbf{s}} \sigma^{\mathbf{r}} &= \bigotimes_j \sigma^{s_j} \sigma^{r_j} = \bigotimes_j i^{s_j \circledast r_j} \sigma^{s_j \circ r_j} \\
&= i^{\sum_j s_j \circledast r_j} \sigma^{\mathbf{s} \circ \mathbf{r}} = i^{\mathbf{s} \circledast \mathbf{r}} \sigma^{\mathbf{s} \circ \mathbf{r}}.
\end{aligned}$$

■

Hence, with the above lemma, we have that

$$[\sigma^{\mathbf{s}}, \sigma^{\mathbf{r}}] = i^{\mathbf{s} \circledast \mathbf{r}} \sigma^{\mathbf{s} \circ \mathbf{r}} - i^{\mathbf{r} \circledast \mathbf{s}} \sigma^{\mathbf{r} \circ \mathbf{s}}.$$

Hence the proof of Lemma 9 is complete.

To extend this result to arbitrary operators, first, we need to note the Pauli decomposition.

Remark 11 *Any bounded operator A on the Hilbert space of d qubits can be uniquely written as*

$$A = \sum_{\mathbf{s} \in \{0,1,2,3\}^d} a_{\mathbf{s}} \, \sigma^{\mathbf{s}},$$

where $a_{\mathbf{s}} \in \mathbb{C}$ are the Fourier coefficients of A and are given as

$$a_{\mathbf{s}} = \frac{1}{2^d} \operatorname{tr} \left\{ A \sigma^{\mathbf{s}} \right\}.$$

With this notation, we can present the following result on the adjoint operator in the Lie algebra g.

Lemma 12 *For any $A, B \in \mathsf{g}$, the corresponding adjoint decomposes as*

$$\mathsf{ad}_A(B) = 2i \sum_{\mathbf{r}, \mathbf{s} \in \mathcal{K}_4^d} a_{\mathbf{s}} b_{\mathbf{r}} (\mathbf{s} \circledast \mathbf{r}) \sigma^{\mathbf{s} \circ \mathbf{r}},$$

where

$$a_{\mathbf{s}} := \frac{1}{2^d} \operatorname{tr}\{A \sigma^{\mathbf{s}}\} \qquad b_{\mathbf{r}} := \frac{1}{2^d} \operatorname{tr}\{B \sigma^{\mathbf{r}}\}$$

are the Pauli coefficients of A and B, respectively.

Hence, from Lemma 12 for $\mathsf{ad}_A(\sigma^{\mathbf{s}})$, we can write the final expression for the partial derivative of $\mathcal{L}$

$$\frac{\partial \mathcal{L}(\overrightarrow{a})}{\partial a_{\mathbf{s}}} = -i \sum_{k=0}^{\infty} \frac{(-2)^k}{(k+1)!} \sum_{\mathbf{s}_1 \in \mathcal{S}} \cdots \sum_{\mathbf{s}_k \in \mathcal{S}} \prod_{j=1}^{k} \left(a_{\mathbf{s}_j} (\mathbf{s}_j \circledast \mathbf{r}_j) \right) \operatorname{tr}\left\{ O \left[\rho^{out}, \sigma^{\mathbf{s}_1 \circ \cdots \circ \mathbf{s}_k \circ \mathbf{r}} \right] \right\}.$$

From the definition of $D_{\mathbf{t}}$ in the statement of the Theorem, the proof is complete.

3.5 Extension to Polynomial Size DLA

Next, we extend the result of the previous section to general PQCs for which DLA has $\mathsf{poly}(d)$ dimensionality.

Definition 13 *For a PQC as in (6), with the unitary of the form $e^{iA(\overrightarrow{a})}$ where $A(\overrightarrow{a}) = \sum_i a_i G_i$ with G_i being traceless Hermitian operators, the set $\mathcal{G} = \{G_1, \cdots, G_p\}$ is called the generators of the PQC.*

Definition 14 (Dynamical Lie Algebra) *Given a PQC with generators $\mathcal{G}$ as in Definition 13, the Dynamical Lie Algebra (DLA) $\mathfrak{g}_A$ is the subalgebra of $\mathfrak{su}(2^d)$ spanned by the repeated nested commutators of the elements in $\mathcal{G}$, i.e.,*

$$i\mathfrak{g}_A := \operatorname{Span}_{\mathbb{R}} \langle iG_1, \cdots, iG_p \rangle_{Lie} \subseteq \mathfrak{su}(2^d),$$

where $\langle \cdot \rangle_{Lie}$ denotes the Lie closure, obtained by repeatedly taking the nested commutators, and the span is over the real numbers. We say that $\mathfrak{g}_A$ has polynomial size/dimensionality if $\dim(\mathfrak{g}_A) = \mathsf{poly}(d)$ as a vector space.

In general, computing the DLA is computationally expensive. One viable approach involves direct construction starting from a set of generators and iteratively commuting them to discover new elements until a basis of the DLA is attained (for example see Algorithm 1 in [LCS$^+$22]). The complexity of this approach in general is $O(2^d)$. However, this is not an issue if the dimension of the DLA to be constructed is promised to by $\mathsf{poly}(d)$.

Suppose that the dimension of $\mathfrak{g}_A$ as a vector space is polynomial in d.

Theorem 15 *Let a PQC as in (6) has the DLA $\mathfrak{g}_A$. Then, the partial derivative of $\mathcal{L}$ decomposes as*

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{r}}} = \sum_{G \in \mathfrak{g}_A} \alpha_G(r) i \operatorname{tr}\left\{ O[G, \rho^{out}] \right\},$$

where $\alpha_G(r) \in \mathbb{R}$ are the coefficients. Moreover, if $\mathfrak{g}_A$ has $\mathsf{poly}(d)$ dimensionality and its generators have a $\mathsf{poly}(d)$ size decomposition in Pauli strings, then there is an algorithm that estimates $\nabla \mathcal{L}(\overrightarrow{a})$ with $\mathsf{poly}(d)$ Hadamard tests and additional $\mathsf{poly}(d)$ classical time.

Proof The proof of the Theorem follows from a similar argument as in Theorem 6. With such an argument we can show that

$$\frac{\partial \mathcal{L}}{\partial a_j} = i \operatorname{tr}\left\{ O \left[\rho^{out}, \frac{1 - \exp\{i\mathsf{ad}_A\}}{\mathsf{ad}_A} \left(\frac{\partial A}{\partial a_j} \right) \right] \right\}.$$

From the definition of $\frac{1-e^{i\mathsf{ad}_A}}{\mathsf{ad}_A}$ and the fact that $\frac{\partial A}{\partial a_j} = G_j$, the above quantity decomposes as

$$\frac{\partial \mathcal{L}}{\partial a_j} = -i \sum_{k=0}^{\infty} \frac{(i)^k}{(k+1)!} \operatorname{tr}\Big\{ O\Big[\rho^{out}, (\mathsf{ad}_A)^k (G_j)\Big]\Big\},$$

Note that ad_A maps g_A unto itself as its *commutator ideal.* That is

$$\mathsf{ad}_A(G) = \sum_{j=1}^{m} c_j [X_j, Y_j],$$

where $X_j, Y_j \in \mathsf{g}_A$. Since g_A is closed under the Lie bracket, then $[X_j, Y_j] \in \mathsf{g}_A$. Hence, $\mathsf{ad}_A(G) \in \mathsf{g}_A$ and it equals to a finite sum of the basis elements E_l of g_A as

$$\mathsf{ad}_A(G_j) = \sum_{l=1}^{d_{\mathsf{g}}} \alpha_{l,j} E_l.$$

Therefore, the above summation can be rewritten as

$$\frac{\partial \mathcal{L}}{\partial a_j} = -i \sum_{k=0}^{\infty} \frac{(i)^k}{(k+1)!} \sum_{l=1}^{d_{\mathsf{g}_A}} \alpha_{l,j} \operatorname{tr}\big\{ O\big[\rho^{out}, E_l\big]\big\},$$

Note that $\mathsf{ad}_A(G_j)$ can be viewed as a vector $(\alpha_{1,j}, \cdots, \alpha_{d_{\mathsf{g}_A},j})$ in the E_l basis. Moreover, ad_A can be viewed as a linear transformation (matrix) T_A in the E_l basis. This matrix is $d_{\mathsf{g}_A} \times d_{\mathsf{g}_A}$ hence is $\mathsf{poly}(d)$ in size. With that taken into account, the partial derivative can be written as

$$\frac{\partial \mathcal{L}}{\partial a_j} \equiv \sum_{k \geq 0} \frac{(-i)^k}{(k+1)!} T_A^k G_j, \qquad \forall j \in [p].$$

Where G_j is understood as a vector in the E_l basis. Now let

$$B_{\mathsf{g}_A} = (1 - e^{-iT_A}) T_A^{-1}.$$

This matrix can be computed classically in $\mathsf{poly}(d_{\mathsf{g}_A}) = \mathsf{poly}(d)$ time. To compute the gradient from this vector representation, one first needs to estimate all $R_l := i \operatorname{tr}\big\{ O\big[\rho^{out}, E_l\big]\big\}$ and compute the following

$$\nabla \mathcal{L} = B_{\mathsf{g}_A} \overrightarrow{R}. \tag{12}$$

With that the proof is complete. ■

The following algorithm summarizes this procedure to estimate the gradient.

Algorithm 2 DLA Gradient Estimation

```
Input: DLA generators $E_1, \cdots, E_{d_{\mathfrak{g}_A}}$,
1: Estimate the expectation value of the tests $R_l := i\,\mathrm{tr}\big\{O[E_l, \rho^{out}]\big\}$ for $l \in [d_{\mathfrak{g}_A}]$.
2: Compute the $d_{\mathfrak{g}} \times d_{\mathfrak{g}_A}$ matrix $B_{\mathfrak{g}_A}$ as in (12).
3: Return $\nabla\mathcal{L} \approx B_{\mathfrak{g}_A}\overrightarrow{R}$.
```

4. Approximations for general case

When $\mathcal{S} \subset \mathcal{K}_4^d$ is not a subgroup, then the terms appearing in the partial derivative in Theorem 5 expand beyond $\mathcal{S}$. Indeed, they form a subgroup denoted by $\langle\mathcal{S}\rangle$ called the subgroup generated by $\mathcal{S}$ and is defined as the smallest subgroup containing $\mathcal{S}$. In that case, one can compute the gradient from Theorem 6 and Algorithm 1 for $\langle S\rangle$. This yields a run-time that scales with $|\langle\mathcal{S}\rangle|^3$. This is tractable as long as the size of $\langle\mathcal{S}\rangle$ is polynomial in d. Otherwise, one needs a different approach. In what follows, we introduce approximations to the gradient computation for general $\mathcal{S}$.

4.1 Truncation

In what follows, we propose to approximate the partial derivatives by truncating the infinite sum that appeared as (7) in Theorem 5.

Theorem 16 *Consider an ansatz of the form $U(\overrightarrow{a}) = \exp\{iA(\overrightarrow{a})\}$, where $A(\overrightarrow{a}) = \sum_{\mathbf{s}\in\mathcal{S}} a_{\mathbf{s}}\sigma^{\mathbf{s}}$, and $\mathcal{S} \subseteq \mathcal{K}_4^d$ is the index of the parameters. Then, the derivative of $\mathcal{L}(\overrightarrow{a})$ for this ansatz is approximated in terms of $D_{\mathbf{t}}$ as:*

$$\frac{\partial\mathcal{L}}{\partial a_{\mathbf{r}}} = \sum_{k=0}^{K} \frac{(-2)^k}{(k+1)!} \sum_{\mathbf{s}_1\in\mathcal{S}} \cdots \sum_{\mathbf{s}_k\in\mathcal{S}} \prod_{j=1}^{k} a_{\mathbf{s}_j}(\mathbf{s}_j \circledast (\mathbf{s}_1 \circ \cdots \circ \mathbf{s}_{j-1} \circ \mathbf{r}))D_{\mathbf{s}_1\circ\cdots\circ\mathbf{s}_k\circ\mathbf{r}} + \epsilon_K,$$

where $\epsilon_K = \exp\Big\{-K\log\frac{K}{e\pi p\|O\|}\Big\}$.

Corollary 17 *In the setting of the above theorem, with bounded $\|O\|$, setting $K = \Omega(p + \log\frac{1}{\epsilon})$ suffices to approximate $\frac{\partial\mathcal{L}}{\partial a_{\mathbf{s}}}$ upto an ϵ additive error.*

Proof The proof of the theorem follows from the fact that ∎

4.2 Unbiased Estimation via Randomization

The infinite sum can be estimated via a randomization technique. We show that the partial derivative in Theorem 5 can be written as an expectation value of a function of the Poisson random variable. Consider the Poisson probability mass function with rate $\lambda = 2$:

$$\mathbb{P}\Big\{X = k\Big\} = \frac{2^k}{k!}e^{-2}, \qquad k = 0, 1, \cdots$$

By sampling k from this distribution we can estimate the partial derivative expression of Theorem 5. Note that

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{r}}} = \sum_{k=0}^{\infty} \frac{(2)^{k+1}}{(k+1)!} X(k) \tag{13}$$

where

$$X(k) := \frac{(-1)^k}{2} \sum_{\mathbf{s}_1 \in \mathcal{S}} \cdots \sum_{\mathbf{s}_k \in \mathcal{S}} \prod_{j=1}^{k} \left(a_{\mathbf{s}_j}(\mathbf{s}_j \circledast (\mathbf{s}_1 \circ \cdots \circ \mathbf{s}_{j-1} \circ \mathbf{r}))\right) D_{\mathbf{s}_1 \circ \cdots \circ \mathbf{s}_k \circ \mathbf{r}}.$$

In that case,

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{r}}} = e^2 \mathbb{E}[X(K)].$$

Hence, by sampling from this distribution we can compute an unbiased estimate of the partial derivative.

4.2.1 Short-term Hamiltonian

The above procedure can be made significantly more efficient if one guarantees that the parameter range is small, that is $|a_{\mathbf{s}}| \le \epsilon$ for small enough $\epsilon > 0$. This is the case in short-term Hamiltonian simulation or a Hamiltonian with "weak interactions". In short-term simulation, the unitary is of the form $U = e^{i\Delta t H}$, where Δt is the time step. In that case, each partial derivative is approximated in $O(1)$ classical computation with one Hadamard test of Figure 1.

Corollary 18 *If* $\max_{\mathbf{s}\in\mathcal{S}} |a_{\mathbf{s}}| \le \frac{\epsilon}{p(1+2\epsilon)}$, *then the partial derivative can be approximated with* ϵ *additive error via* $\frac{\partial \mathcal{L}}{\partial a_{\mathbf{s}}} \approx D_{\mathbf{s}}$.

5. Sample Complexity of Gradient Estimation

Next, we study more sophisticated ways to estimate the gradient in the setting of Theorem 6.

5.1 Measurement with Shadow Tomography

Turning the gradient estimation to a series of Hadamard tests has another benefit that can further reduce the number of shots to $O(\log p)$. This can be done using the classical shadow tomography [HKP20] — a procedure to estimate several observables with minimal sample complexity. Suppose the observable O in (1) is k-local, meaning that it decomposes into a finite sum of observables acting non-trivially on at most k qubits. In that case, we can prove that the number of shots scales logarithmically with p.

We note that in general the above approach does not extend to existing works even when O is k-local. The reason is that the corresponding observables are not themselves k-local.

Shadow tomography with Pauli measurements. In shadow tomography [HKP20], a random unitary V is applied to the input state followed by a measurement in the computational basis. Then the measurement outcomes are used to generate a classical matrix called the shadow. Then the shadows are used classically to compute the expectation value of the observables of interest.

Different choices of V have been studied in [HKP20]. Here, we describe this process when V is the tensor product of randomly chosen Pauli operators. In other words,

$$V = V_1 \otimes \cdots \otimes V_d \in CL(2)^{\otimes d},$$

where each V_j is chosen randomly and uniformly from the Clifford group $CL(2)$. Given the input state ρ, the output after this rotation is given by $V\rho V^\dagger$. Measuring this state give a string of bits $\hat{b}_1, \cdots, \hat{b}_d \in \{0,1\}$ as the outcome. This process is equivalent to measuring each qubit of ρ randomly on the X, Y or Z basis. Given, these bits and with the choice of $V_1, \cdots, V_d$, the shadow matrix is computed as

$$\hat{\rho} := \bigotimes_{j=1}^{d} \left(3V_j^\dagger \left|\hat{b}_j\right\rangle\!\left\langle\hat{b}_j\right| V_j - I\right).$$

Repeating this procedure for n copies of ρ gives a set of n shadows denoted by $\hat{\rho}_1, \cdots, \hat{\rho}_n$. Suppose one is interested to estimate the expectation value of the observables $M_1, \cdots, M_m$ with respect to ρ. Then, for each M_j, one computes $\operatorname{tr}\{M_j \hat{\rho}_i\}, i \in [n]$ followed by a meadian of means estimator to estimate $\langle M_j \rangle_\rho$.

Theorem 19 (**[HKP20]**) *Suppose the observables $M_j, j \in [m]$ act non-trivially on at most k qubits. Then, the shadow tomography with random Pauli measurements estimates $\langle M_j \rangle_\rho$ for all $j \in [m]$ up to an additive error $\epsilon > 0$ provided that*

$$n = O\left(\frac{4^k}{\epsilon^2} \log m \max_j \|M_j\|_\infty^2\right),$$

copies of ρ. The algorithm runs in $O\big(2^{\Theta(k)} \log m\big)$ classical time.

Algorithm 3 Gradient Estimation With Shadow Tomography

```
Input: $n, k$, Parameter subgroup $\mathcal{S}$,
for $i = 1$ to $n$ do
    Construct a random Pauli measurement $V_j$ for the non-trivial qubits of $O$.
    Apply the ansatz to generate the output state and measure the nontrivial qubits in the selected basis.
    Let $\hat{b}_1, \cdots \hat{b}_k$ be the measurement outcomes. Compute the classical shadow $\hat{\rho}_i$ from the $k$ matrices $3V_j^\dagger \left|\hat{b}_j\right\rangle\!\left\langle\hat{b}_j\right| V_j - I$.
    Partition $\hat{\rho}_i$'s into $m$ groups and let $\tilde{\rho}_l$ be the empirical average of the $l$th group.
For each $\mathbf{s} \in \mathcal{S}$ estimate $D_{\mathbf{s}}$ by median of measns for $\hat{D}_{\mathbf{s},l} = i \operatorname{tr}\{O[\sigma^{\mathbf{s}}, \tilde{\rho}_l]\}, l \in [m]$.
Compute the matrix $V$ where the column $i$ is computed as in (8).
Compute the matrix $B = (I - e^{-V})V^{-1}$, where $V^{-1}$ is the generalized inverse of $V$.
Return $\nabla\mathcal{L} \approx B\overrightarrow{D}_{\mathbf{s}}$.
```

We will use this procedure to estimate the Hadamard tests with reduced sample complexity. All the steps are summarized as Algorithm 20. Based on this theorem and Theorem 3, we have the following result for the case when O in the VQA formulation is k-local.

Corollary 20 *In the setting of Theorem 6, suppose the observable O is a finite sum $O = O_1 + \cdots + O_m$ of observables that act on at most k-qubits. Then there exists an algorithm that estimates $\nabla \mathcal{L}(\overrightarrow{a})$ with $O(\frac{1}{\epsilon^2} 4^k \log(mp) \ \max_j \|O_j\|_\infty)$ ansatz uses and an additional*

$$O\left(\frac{1}{\epsilon^2} 2^{\Theta(k)} pm \log(mp) \max_j \|O_j\|_\infty^2 + p^3 + pd\right)$$

classical time.

Proof We consider a more restricted assumption first. Suppose the observable O in the VQA formulation acts on at most k qubits. Note that being k-loval means that O is a finite sum of such observables. Since O acts on at most k-qubits then $O = \tilde{O} \otimes I_{d-k}$ for some k-qubit observable $\tilde{O}$. Then, the Hadamard tests appearing in the partial derivative formulation of Theorem 5 have locality equal to k. To see this, recall that

$$D_{\mathbf{s}} = i \operatorname{tr}\left\{O[\sigma^{\mathbf{s}}, \rho^{out}]\right\}.$$

Moreover, note from [MNKF18] the following property of the commutator for any operator B:

$$[\sigma^{\mathbf{s}}, B] = i\left(R_{\mathbf{s}}(\frac{\pi}{2}) B R_{\mathbf{s}}(\frac{\pi}{2})^\dagger - R_{\mathbf{s}}(-\frac{\pi}{2}) B R_{\mathbf{s}}(-\frac{\pi}{2})^\dagger\right),$$

where $R_{\mathbf{s}}(\theta) := e^{-i\frac{\theta}{2}\sigma^{\mathbf{s}}}$. With this observation, we obtain

$$\begin{aligned} D_{\mathbf{s}} &= -\operatorname{tr}\left\{O\left(R_{\mathbf{s}}(\frac{\pi}{2})\rho^{out} R_{\mathbf{s}}(\frac{\pi}{2})^\dagger - R_{\mathbf{s}}(-\frac{\pi}{2})\rho^{out} R_{\mathbf{s}}(-\frac{\pi}{2})^\dagger\right)\right\} \\ &= -\operatorname{tr}\left\{R_{\mathbf{s}}(\frac{\pi}{2})^\dagger O R_{\mathbf{s}}(\frac{\pi}{2})\rho^{out}\right\} + \operatorname{tr}\left\{R_{\mathbf{s}}(-\frac{\pi}{2})^\dagger O R_{\mathbf{s}}(-\frac{\pi}{2})\rho^{out}\right\}, \end{aligned}$$

where we used the cyclic property of the trace. Without loss of generality assume $\tilde{O}$ acts on the first k qubits. Note that since $R_{\mathbf{s}}$ is in a tensor product form, then we have

$$\begin{aligned} R_{\mathbf{s}}(\frac{\pi}{2})^\dagger O R_{\mathbf{s}}(\frac{\pi}{2}) &= \left(\Big(\bigotimes_{j=1}^{k} R_{s_j}(\frac{\pi}{2})^\dagger\Big)\tilde{O}\Big(\bigotimes_{j=1}^{k} R_{s_j}(\frac{\pi}{2})\Big)\right) \bigotimes \left(\Big(\bigotimes_{j>k}^{d} R_{s_j}(\frac{\pi}{2})^\dagger\Big) I \Big(\bigotimes_{j>k}^{d} R_{s_j}(\frac{\pi}{2})\Big)\right) \\ &= \Big(\bigotimes_{j=1}^{k} R_{s_j}(\frac{\pi}{2})^\dagger\Big)\tilde{O}\Big(\bigotimes_{j=1}^{k} R_{s_j}(\frac{\pi}{2})\Big) \bigotimes I_{d-k} \end{aligned}$$

This implies that $R_{\mathbf{s}}(\frac{\pi}{2})^\dagger O R_{\mathbf{s}}(\frac{\pi}{2})$ acts non trivially on k qubits. The same holds for $R_{\mathbf{s}}(-\frac{\pi}{2})^\dagger O R_{\mathbf{s}}(-\frac{\pi}{2})$. Hence, $D_{\mathbf{s}}$ can be written as the expectation value of an observable that acts on k qubits of ρ^{out}. As a result, one can use the classical shadow tomography with Pauli measurements and Theorem 19 to obtain estimates of $D_{\mathbf{s}}$ for all $\mathbf{s} \in \mathcal{S}$. Therefore, we need $O\left(\frac{4^k}{\epsilon^2} \log p \|O\|_\infty^2\right)$ copies of ρ^{out}. This estimation takes

$$O\left(\frac{1}{\epsilon^2} 2^{\Theta(k)} \log p \|O\|_\infty^2\right)$$

classical time. Lastly, according to Theorem 6, we need an additional $O(p^3 + pd)$ classical time to estimate the gradient from the estimated $D_{\mathbf{s}}, \mathbf{s} \in \mathcal{S}$.

The extension to k-local follows from the fact that O is a finite sum $O = O_1 + \cdots + O_m$ of observables that act on at most k-qubits. In this case $D_{\mathbf{s}}$ can be written as a finite sum $D_{\mathbf{s}} = D_{\mathbf{s},1} + \cdots + D_{\mathbf{s},m}$. We can use the above procedure to estimate each component. Hence, in total, we will have mp observables acting on at most k qubits. This adds up logarithmically in m to the number of shots and linearly to the time. ■

5.2 Joint Measurability

One benefit of estimating the gradient via Hadamard tests is that one can potentially group all the required Hadamard tests into mutually compatible collections. Joint measurability refers to the possibility of measuring several observables via one reference measurement.

Definition 21 *A set of observables $M_j = \{\Lambda_u^j : u \in \mathcal{U}\}, j = 1, 2, ..., k$ are called stochastically compatible if there exist a measurement $M_{ref} = \{A_w : w \in \mathcal{W}\}$ and stochastic kernel functions $q_j(w|u)$ on $\mathcal{W} \times \mathcal{U}$ such that*

$$\Lambda_u^j = \sum_{w \in \mathcal{W}} q_j(w|u) A_w$$

for all $u \in \mathcal{U}$ and all $j \in [k]$.

Suppose $\mathcal{F}_1, \cdots, \mathcal{F}_m$ are groups of the Hadamard tests corresponding to $D_{\mathbf{s}}, \mathbf{s} \in \mathcal{S}$ that are mutually compatible. In that case, for each group, we first perform a fine-grained measurement M_j on the sample and then extract the estimations from additional post-measurement classical processes. This approach helps reduce the sample complexity.

Corollary 22 *Suppose $\mathcal{F}_1, \cdots, \mathcal{F}_m$ form the mutually compatible groups of the Pauli strings appearing in a parameterized Hamiltonian $A(\overrightarrow{a}) = \sum_i a_i P_i$. Then, the number of joint Hadamard tests in Theorem 6 can be reduced to $\tilde{O}(\frac{m}{\epsilon^2})$.*

Jointly measuring Pauli strings has been studied extensively in the literature [LBZ02, VYI20, CvSW+20, BBRV01].

Concluding Remarks

This paper provides a framework to estimate the gradient of generic PQCs via Hadamard tests for Pauli operators followed by classical post-processing. It is shown that the proposed approach is polynomial in classical and quantum resources when the DLA of the associated Hamiltonian of the PQC has a dimensionality polynomial in the number of qubits. Moreover, this method does not change the ansatz structure and can be used to reduce the measurement shot complexity to scale logarithmically with the number of parameters. The results would be beneficial in various optimization or learning quantum algorithms that rely on the estimation of the gradient.

As a future work, one might be interested in extending the proposed framework to the estimation of the higher order derivatives and measures such as the Hessian or the

Fubini–Study metric defined for a parameterized quantum system. For that, one needs to define a second-order Hadamard test strategy for measuring the double derivatives. Another future work is to extend this strategy to multi-layered PQCs where each layer might be a black box unitary. Finally, deriving lower bounds on the classical and quantum resources needed to estimate the gradient or higher-order derivatives is another important direction.

Appendix A. Basic Definitions

A.1 Group

A group G is a set equipped with a binary operation (usually denoted by $*$) that satisfies the following properties:

1. **Closure**: For any two elements $a, b \in G$, the result of the operation $a * b$ is also an element of G.
2. **Associativity**: For all $a, b, c \in G$, $(a * b) * c = a * (b * c)$.
3. **Identity Element**: There exists an element $e \in G$ such that for all $a \in G$, $a * e = e * a = a$.
4. **Inverse Element**: For every element $a \in G$, there exists an element $a^{-1} \in G$ such that $a * a^{-1} = a^{-1} * a = e$, where e is the identity element.

A subgroup H of a group G is a subset of G that forms a group under the same binary operation as G. Formally, H is a subgroup of G if it satisfies the following properties:

1. **Closure**: For any two elements $h_1, h_2 \in H$, the result of the operation $h_1 * h_2$ is also an element of H.
2. **Identity Element**: The identity element of G is also in H.
3. **Inverse Element**: For every element $h \in H$, its inverse h^{-1} is also in H.

A.2 Lie Algebra

A Lie algebra is a vector space $\mathfrak{g}$ equipped with a binary operation called the Lie bracket, denoted by $[\cdot, \cdot]$, which satisfies the following properties for all $X, Y, Z \in \mathfrak{g}$:

1. **Bilinearity**: $[aX + bY, Z] = a[X, Z] + b[Y, Z]$ and $[X, aY + bZ] = a[X, Y] + b[X, Z]$ for all $a, b \in \mathbb{R}$.
2. **Antisymmetry**: $[X, Y] = -[Y, X]$.
3. **Jacobi Identity**: $[X, [Y, Z]] + [Y, [Z, X]] + [Z, [X, Y]] = 0$.

A Lie subalgebra $\mathfrak{s}$ of a Lie algebra $\mathfrak{g}$ is a subspace of $\mathfrak{g}$ that is itself a Lie algebra under the same Lie bracket operation. Formally, $\mathfrak{s}$ is closed under the Lie bracket, meaning that for all $X, Y \in \mathfrak{s}$, $[X, Y] \in \mathfrak{s}$. For example, in a d-qubit quantum system, the subspace

$$Span\{X^{\otimes d}, Y^{\otimes d}, Z^{\otimes d}\},$$

is closed under the Lie bracket. Moreover, it has dimensionality equal to 3 inside the 4^d dimensional Lie algebra.

Appendix B. Analytical Derivation of the Gradient in Example 3

Note that from (9) the partial derivative of the objective function can be written as

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{s}}} = \mathrm{tr}\Big\{ O \Big(\frac{\partial U}{\partial a_{\mathbf{s}}} \big(\rho U^\dagger\big) + (U\rho)\frac{\partial U^\dagger}{\partial a_{\mathbf{s}}} \Big) \Big\},$$

Given that $\frac{\partial U^\dagger}{\partial a_{\mathbf{s}}} = (\frac{\partial U}{\partial a_{\mathbf{s}}})^\dagger$. Then, by denoting $\tilde{U} = \frac{\partial U}{\partial a_{\mathbf{s}}}$ we have that

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{s}}} = \mathrm{tr}\Big\{ O\tilde{U}\rho U^\dagger + U\rho\tilde{U}^\dagger \Big\}.$$

Next, as $UU^\dagger = I$, we have

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial a_{\mathbf{s}}} &= \mathrm{tr}\left\{ O \left(\tilde{U}U^\dagger(U\rho U^\dagger) + (U\rho U^\dagger)U\tilde{U}^\dagger \right) \right\} \\ &= \mathrm{tr}\left\{ O \left(\tilde{U}U^\dagger \rho^{out} + \rho^{out}(\tilde{U}U^\dagger)^\dagger \right) \right\}, \end{aligned}$$

where ρ^{out} is the ansatz output. Note that the single qubit ansatz can also be written as

$$U(\vec{a}) = I\cos\theta + i\left(\sum \hat{a}_s\sigma^s\right)\sin\theta, \tag{14}$$

where $\theta = \sqrt{\sum a_s^2}$ is a normalizing parameter and $\hat{a}_s = \frac{a_s}{\theta}$. Now, we can differentiate U with respect to a single parameter a_s appearing in the sum:

$$\begin{aligned} \frac{\partial U(\vec{a})}{\partial a_s} = (-\frac{a_s}{\theta}\sin\theta)I + i\left(\sum_{s'\neq s}\frac{-a_s a_{s'}}{\theta^3}\sigma^{s'} + \frac{\theta^2 - a_s^2}{\theta^3}\sigma^s\right)\sin\theta \\ + i\left(\frac{a_s\cos\theta}{\theta}\sum_{s'}\hat{a}_{s'}\sigma^{s'}\right). \end{aligned} \tag{15}$$

Using (14) to find $U^\dagger$ and (15) for $\tilde{U}$, we can analytically find the derivative of the loss function at each point. To better visualize how we converge to the true derivative, consider the example where

$$U(\vec{a}) = \exp\big\{i(a_1\sigma^1 + a_2\sigma^2)\big\} \tag{16}$$

We will analytically find the derivative at $a_1 = 0$. The equations yield

$$\tilde{U}U^\dagger = i\frac{\sin a_2}{a_2}\left(\cos a_2\sigma^1 + \sin a_2\sigma^3\right)$$

Which when plugged into the derivative expression gives

$$\left.\frac{\partial L}{\partial a_1}\right|_{a_1=0} = \frac{\sin a_2}{a_2}\left(\cos a_2 D_1 + \sin a_2 D_3\right).$$

Now, we will derive the same expression using Theorem 5. Looking at individual terms in the summation, we have the following:

$$\frac{\partial L}{\partial a_1}\bigg|_{a_1=0} = D_1 + a_2 D_3 - \frac{4}{6} a_2^2 D_1 - \frac{8}{24} a_2^3 D_3 + \dots$$

Note the repeating nature of the D_is. We then rearrange the terms:

$$\begin{aligned}
\frac{\partial L}{\partial a_1}\bigg|_{a_1=0} &= D_1 \left(1 - \frac{4}{6} a_2^2 + \dots\right) + D_3 \left(a_2 - \frac{8}{24} a_2^3 + \dots\right) \\
&= D_1 \frac{1}{a_2} \left(\sum_p \frac{(-1)^p}{(2p+1)!} (2a_2)^{2p+1}\right) + D_3 \frac{1}{a_2} \left(\sum_p \frac{(-1)^p}{(2p+2)!} (2a_2)^{2p+2}\right) \\
&= \frac{1}{2a_2} \left(D_1 \sin 2a_2 + D_3 (1 - \cos 2a_2)\right) \\
&= \frac{\sin a_2}{a_2} \left(\cos a_2 D_1 + \sin a_2 D_3\right).
\end{aligned}$$

Which is exactly the expression we got analytically. In this example, we can see how our method is akin to approximating a function with a series expansion. The power of this approach lies in the group structure of the Pauli matrices, which produces predictable patterns. Thus, we are dealing with a repeating structure that makes use of a subset of D_is. Another point is that D_i only makes use of ρ_{out}. We can calculate these D_i and approximate the derivative to any arbitrary precision.

Appendix C. Summary of Related Works

Trotterization. In this approach the non-product unitary is approximated via the Suzuki-Trotter transformation [Suz76] which states that for any operators $A_1, A_2, \cdots, A_k$, that do not necessarily commute with each other, the following holds

$$\lim_{n\to\infty} \left(\prod_{j=1}^{k} \exp\{A_j/n\}\right)^n = \exp\Big\{\sum_{j=1}^{k} A_j\Big\}.$$

The Trotter formula has been used in literature to derive approximations for generic PQCs [YNJ+22, LXS+20, MOT21]. However, implementing the above approximation may lead to a high gate complexity and is not preferable in scenarios where direct implementation is available.

Stochastic PSR: Stochastic parameter shift rule was proposed in [BC21] to measure the derivative for unitary operators of the form $e^{i(\theta\sigma^{\mathbf{s}}+B)}$, where $\sigma^{\mathbf{s}}$ is a Pauli string and B is an arbitrary Hermitian operator. The authors showed that the derivative is equal to

$$\frac{\partial \mathcal{L}}{\partial \theta} = \int_0^1 C_+(\theta, t) - C_-(\theta, t) dt,$$

where $C_\pm(\theta, t) := \operatorname{tr}\Big\{O V_\pm(\theta, t) \rho V_\pm^\dagger(\theta, t)\Big\}$, with

$$V_\pm(\theta, t) := e^{it(\theta\sigma^{\mathbf{s}}+B)} e^{\pm i\sigma^{\mathbf{s}}} e^{i(1-t)(\theta\sigma^{\mathbf{s}}+B)}.$$

Then the authors present a Monte Carlo strategy to estimate this integral. A more general variant of the parameter shift rule has been introduced [WIWL22]. The considered general ansatz of the form $U(\theta) = e^{i(\theta A+B)}$ for generic Hermitian A and B. Given the spectral decomposition of A the paper provides an explicit formula for the derivative of the objective function. This is done via a Discrete Fourier series approach.

Such methods for generic ansatz $e^{i(\theta A+B)}$ require not only to perturb the parameters but also to change the unitaries involved. As a result, to modify not just the parameters of the PQC, but also change the unitaries that appear. In practice, such modifications require a re-evaluation of the schedule of the underlying quantum-control system and hence are at a disadvantage. Moreover, the stochastic PSR has a high estimation variance. This is because the above integral is estimated by sampling values of s uniformly in the interval $(0, 1)$ and then calculating the costs with a finite-shot estimate. In addition, this method leads to a bigger number of unique circuits to compute the derivative, increasing the compilation overhead for both hardware and simulator implementations.

Nyquist PSR: Recently [The23] proposed a "proper" shift rule for PQCs of the form $e^{i(\theta A+B)}$ where only the parameters are shifted without any other modifications of the ansatz. The method was called *Nyquist parameter shift rule* and relies on a beautiful connection between the Nyquist-Shannon Sampling theorem and the Fourier series that was observed earlier in [WIWL22,VT18]. This paper shows that if $f(x) = tr\{OU\rho U^\dagger\}$ with $U = e^{ixH+B}$ and K being the difference between the maximum and minimum eigenvalues of A, then the Fourier spectrum of f is contained in $[-K, K]$. Hence, the Nyquist-Shannon Sampling theorem can be used to estimate the derivative of the objective function. The proposed method, however, has a low approximation error when the parameter value is large enough. More precisely, the approximation error is $O(\frac{1}{c^2})$ as long as $\theta = (1 - \Omega(1))c$, where c is the maximum magnitude of a parameter value. Our method is suitable when the parameter value is small.

References

[AWGP21] Gian-Luca R Anselmetti, David Wierichs, Christian Gogolin, and Robert M Parrish. Local, expressive, quantum-number-preserving vqe ansätze for fermionic systems. *New Journal of Physics*, 23(11):113010, November 2021.

[BBRV01] Somshubhro Bandyopadhyay, P. Oscar Boykin, Vwani Roychowdhury, and Farrokh Vatan. A new proof for the existence of mutually unbiased bases. *quant-ph/0103162*, 2001.

[BC21] Leonardo Banchi and Gavin E. Crooks. Measuring analytic gradients of general quantum evolution with the stochastic parameter shift rule. *Quantum*, 5:386, January 2021.

[BLSF19] Marcello Benedetti, Erika Lloyd, Stefan Sack, and Mattia Fiorentini. Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, 4(4):043001, 2019.

[CAB$^+$21] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3(9):625–644, aug 2021.

[CRSS96] A. R. Calderbank, E. M Rains, P. W. Shor, and N. J. A. Sloane. Quantum error correction via codes over gf(4). August 1996.

[CSV$^+$21] M. Cerezo, Akira Sone, Tyler Volkoff, Lukasz Cincio, and Patrick J. Coles. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nature Communications*, 12(1), March 2021.

[CvSW$^+$20] 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. *arXiv:1908.06942*, 2020.

[DAJ$^+$21] Alain Delgado, Juan Miguel Arrazola, Soran Jahangiri, Zeyue Niu, Josh Izaac, Chase Roberts, and Nathan Killoran. Variational quantum algorithm for molecular geometry optimization. *Physical Review A*, 104(5):052402, November 2021.

[FGG14] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.

[FHC$^+$23] 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. September 2023.

[FN18] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. February 2018.

[GEBM19] Harper R. Grimsley, Sophia E. Economou, Edwin Barnes, and Nicholas J. Mayhall. An adaptive variational algorithm for exact molecular simulations on a quantum computer. *Nature Communications*, 10(1), July 2019.

[Got97] Daniel Gottesman. *Stabilizer codes and quantum error correction.* California Institute of Technology, 1997.

[HCT$^+$19] Vojtěch Havlíček, Antonio D. Córcoles, Kristan Temme, Aram W. Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, March 2019.

[HGS22] Mohsen Heidari, Ananth Y. Grama, and Wojciech Szpankowski. Toward physically realizable quantum neural networks. *Association for the Advancement of Articial Intelligence (AAAI)*, 2022.

[HKP20] Hsin-Yuan Huang, Richard Kueng, and John Preskill. Predicting many properties of a quantum system from very few measurements. *Nature Physics 16, 1050–1057 (2020)*, February 2020.

[HN21] Aram W. Harrow and John C. Napp. Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms. *Physical Review Letters*, 126(14):140502, apr 2021.

[HWO$^+$19] Stuart Hadfield, Zhihui Wang, Bryan O'Gorman, Eleanor Rieffel, Davide Venturelli, and Rupak Biswas. From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms*, 12(2):34, February 2019.

[JEM$^+$19] Tyson Jones, Suguru Endo, Sam McArdle, Xiao Yuan, and Simon C. Benjamin. Variational quantum algorithms for discovering hamiltonian spectra. *Physical Review A*, 99(6):062304, June 2019.

[KMT$^+$17] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature*, 549(7671):242–246, sep 2017.

[LBZ02] Jay Lawrence, Časlav Brukner, and Anton Zeilinger. Mutually unbiased binary observable sets onNqubits. *Physical Review A*, 65(3), feb 2002.

[LCS$^+$22] 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, September 2022.

[LW18] Jin-Guo Liu and Lei Wang. Differentiable learning of quantum circuit born machines. *Physical Review A*, 98(6):062324, December 2018.

[LXS$^+$20] Xuanqing Liu, Tesi Xiao, Si Si, Qin Cao, Sanjiv Kumar, and Cho-Jui Hsieh. How does noise help robustness? explanation and exploration under the neural sde framework. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 279–287, 2020.

[MBS$^+$18] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature Communications*, 9(1), November 2018.

[MKF19] Kosuke Mitarai, Masahiro Kitagawa, and Keisuke Fujii. Quantum analog-digital conversion. *Physical Review A*, 99(1):012301, January 2019.

[MNKF18] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii. Quantum circuit learning. *Physical Review A*, 98(3):032309, sep 2018.

[MOT21] Alexander Miessen, Pauline J. Ollitrault, and Ivano Tavernelli. Quantum algorithms for quantum dynamics: A performance study on the spin-boson model. *Physical Review Research*, 3(4):043212, dec 2021.

[PMS$^+$14] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5(1), jul 2014.

[Ros06] Wulf Rossmann. *Lie groups: An Introduction through Linear Groups*. Number 5 in Oxford graduate texts in mathematics. Oxford University Press, Oxford [u.a.], 1. publ. in paperback edition, 2006. Reprinted 2011.

[SBG$^+$18] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A 99, 032331 (2019)*, November 2018.

[SK19] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical Review Letters*, 122(4):040504, February 2019.

[SPS02] SG Schirmer, ICH Pullen, and AI Solomon. Identification of dynamical lie algebras for finite-level quantum control systems. *Journal of Physics A: Mathematical and General*, 35(9):2327, 2002.

[Suz76] Masuo Suzuki. Generalized trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Communications in Mathematical Physics*, 51(2):183–190, jun 1976.

[SWM$^+$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, aug 2020.

[The23] Dirk Oliver Theis. "proper" shift rules for derivatives of perturbed-parametric quantum evolutions. *Quantum*, 7:1052, July 2023.

[VT18] Javier Gil Vidal and Dirk Oliver Theis. Calculus on parameterized quantum circuits. *arXiv:1812.06323*, December 2018.

[VYI20] 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(12), March 2020.

[WHT15] Dave Wecker, Matthew B. Hastings, and Matthias Troyer. Progress towards practical quantum variational algorithms. *Physical Review A*, 92(4):042303, October 2015.

[WIWL22] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. General parameter-shift rules for quantum gradients. *Quantum*, 6:677, March 2022.

[WKKB23] 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.

[WLW$^+$24] Roeland Wiersema, Dylan Lewis, David Wierichs, Juan Carrasquilla, and Nathan Killoran. Here comes the su(n): multivariate quantum gates and gradients. *Quantum*, 8:1275, March 2024.

[YNJ$^+$22] Xiaodong Yang, Xinfang Nie, Yunlan Ji, Tao Xin, Dawei Lu, and Jun Li. Improved quantum computing with higher-order trotter decomposition. *Physical Review A*, 106(4):042401, oct 2022.