# Efficient Gradient Estimation for Parameterized Quantum Systems with Lie Algebraic Symmetries

Mohsen Heidari[1], Masih Mozakka[1], and Wojciech Szpankowski[2]

[1]Department of Computer Sciences, Indiana University, Bloomington, IN, USA
[2]Department of Computer Sciences, Purdue University, West Lafayette, IN, USA

Gradient estimation is a central challenge in training parameterized quantum circuits (PQCs) for hybrid quantum-classical optimization and learning problems. This difficulty arises from several factors, including the exponential dimensionality of the Hilbert spaces and the information loss in quantum measurements. Existing estimators, such as finite difference and the parameter shift rule, often fail to adequately address these challenges for certain classes of PQCs. In this work, we propose a novel gradient estimation framework that leverages the underlying Lie algebraic structure of PQCs, combined with the Hadamard test. By analyzing the differential of the matrix exponential, we derive an expression for the gradient as a linear combination of expectation values obtained via Hadamard tests. The coefficients in this decomposition depend solely on the circuit's parameterization and can be estimated using state-of-the-art shadow tomography techniques. Hence, our approach enables efficient gradient estimation, requiring a number of measurement shots that scales logarithmically with the number of parameters, and with polynomial classical and quantum time. This is an exponential reduction in the measurement cost and a polynomial speed-up in time compared to existing works.

## 1 Introduction

Hybrid quantum-classical strategies have emerged as a leading approach for quantum optimization and learning [2, 3], and have been extensively studied across a broad range of domains, including optimization [4], quantum chemistry [5, 6, 7, 8], and quantum machine learning from classical and quantum data [9, 10, 11, 12, 13, 14, 15, 16, 17]. Variational quantum algorithm (VQA) particularly has been a promising paradigm for quantum learning and inference, where a PQC (a.k.a ansatz) is trained in a classical-quantum loop. Gradient-based training methods have gained significant attention in the literature [18, 19, 20, 4, 9, 10, 11] and have demonstrated advantages in convergence rates compared to gradient-free methods.

However, estimation of the gradient can be computationally challenging due to several factors including the exponential dimensionality of the Hilbert spaces, the no-cloning, information loss of quantum measurements, and non-commutativity of Hamiltonian terms.

Mohsen Heidari: mheidar@iu.edu, https://homes.luddy.indiana.edu/mheidar/,
Portions of this work have been published in the 2025 Conference on Neural Information Processing Systems (NeurIPS 2025) [1]. This manuscript substantially extends the previous publication by adding new results on general Hamiltonians (see Section 7) and added omitted proofs of previous claims.

Therefore, each gradient estimation can have an exponential sample complexity leading to a high overhead and hence a bottleneck for the scalability of gradient-based VQAs.

Several approaches have been introduced to estimate the gradient [9, 11, 19, 21, 18, 20, 22, 23, 24]; but they often yield suboptimal gradient circuits for certain PQCs. Methods based on finite differences evaluate the objective function in the neighborhood of the parameters. They can be applied to general PQCs, but suffer from a slow convergence rate [18]. The well-known parameter shift rule (PSR) [20, 11] relies on the Hadamard test with Pauli operators to estimate the partial derivatives. The Hadamard test is an efficient method that directly measures the partial derivatives and does not have the numerical instability of indirect methods such as finite differences. However, PSR with Hadamard test is restricted to ansätze with two distinct eigenvalues. It can be adapted for more complex circuits via backpropagation, but it comes with high computational costs in terms of gate decomposition. Other existing methods often apply to more general circuits but have high overhead due to the extensive use of the ansatz with repeated measurements, and exponential classical computation [25, 26, 27].

Lie algebraic structures in PQCs have been increasingly important in analysis and design of hybrid quantum-classical strategies. Ansätze that have dynamical Lie algebra (DLA) with polynomial dimensionality may not exhibit any Barren plateaus, which are flat regions in the parameter landscape [28, 29, 30]. Moreover, Lie algebraic symmetries have be used for classical simulation of quantum models [31]. In this work, we build upon the Lie algebraic characterizations and we develop an efficient gradient estimation for general circuits based on the Hadamard test followed by post-processing steps. With that, we enable efficient applicability of the Hadamard test to generic PQCs without the need to change the ansatz structure and with low classical overhead.

### 1.1 Summary of the Main Results

We analytically derive an explicit expression for the gradient of generic PQCs in terms of the expectation values of the Hadamard tests corresponding to a certain set of Pauli strings. Then, we develop a gradient estimation method using a series of Hadamard tests at the output of the ansatz followed by classical post-processing techniques including classical shadow tomography (CST) [32]. A generic PQC on $n$ qubits can be represented as $U(\overrightarrow{a}) = e^{iA(\overrightarrow{a})}$, where $A$ is the parameterized Hamiltonian with $\overrightarrow{a} = (a_1, \cdots, a_p)$ as the vector of parameters. Typically, the Hamiltonian is written in term of Pauli strings as $A(\overrightarrow{a}) = \sum_{i=1}^{p} a_i P_i$, where $a_i \in \mathbb{R}$ and $P_i$'s are tensor products of Pauli operators. This formulation includes multi-layered PQCs (e.g., the hardware-efficient ansatz) and appears in a wide range of setups including quantum approximate optimization algorithm (QAOA), many-body quantum systems (e.g., Ising model), and adiabatic evolutions. Typically, the objective is minimizing a loss function $\mathcal{L}(\overrightarrow{a})$ depending on the parameterization of the PQC, the input state, and the measurement observable. Hence, a gradient-based optimization can be used given an estimation of $\nabla\mathcal{L}$.

Our gradient estimation method is based on a binary encoding of Pauli strings to capture the structural properties of their commutation relations. Similar techniques have been used before in the context of the stabilizer formulation in quantum error correction [33, 34]. Then, we make a connection between this binary encoding and the differential of the matrix exponential map, studied in Lie algebra [35]. We write the partial derivatives of the PQC as an infinite-length linear combination of expectation values of Hadamard tests for various Pauli strings. We show that when the Pauli strings in $A(\overrightarrow{a})$ are closed under the commutation, the infinite-length linear combination collapses to a finite number of terms that can be computed efficiently using the binary encoding. Such terms are written

as the expectation value of a set of observables that can be estimated using CST, as an estimation procedure to estimate several observables with sample complexity growing logarithmically with the number of observables [32].

The closedness condition means that for any pair of Pauli strings $P_i$ and $P_j$ appearing in $A(\overrightarrow{a})$, their commutator $[P_i, P_j]$ can be written as a linear combination of the same set of Pauli strings. This condition implies that the Hamiltonian $A(\overrightarrow{a})$ generates a dynamical Lie algebra (DLA) with a basis consisting solely of the Pauli strings present in $A(\overrightarrow{a})$ [36, 37]. When the closedness condition is not directly satisfied, one can consider the sub algebra generated by $P_i$ terms in $A(\overrightarrow{a})$ and apply the proposed gradient estimation method. In that case, the sample complexity and running time depend on the dimensionality of this sub-algebra. Therefore, the proposed estimation is efficient when the dimensionality is polynomial with $n$.

We also extend our results to a more general Hamiltonian structure where $A(\overrightarrow{a})$ is decomposed into generic Hermitian terms that correspond to certain physical interactions. We show that when the Hamiltonian DLA has $\mathsf{poly}(n)$ dimensionality, then the gradient $\nabla\mathcal{L}$ can be estimated polynomially.

The polynomial size assumption is already satisfied for several well-known Hamiltonian models, including variants of 2-local Ising model (e.g., the transverse-field) and Kitaev chain [37] used to model molecular dynamics. In addition, the polynomial dimensionality is an essential component to avoid the barren plateaus [38, 29, 30, 28]. This stems from the fact that the variance of the gradient is inversely proportional to the dimension of the DLA [29].

Nevertheless, there still is a curiosity to understand the gradient estimation when DLA dimensionality grows exponentially with $n$. For that, we present a class of efficient methods to approximate the gradient and analyze the error rate as a function of the approximation terms. Particularly, we present a truncation technique and a randomized algorithm for gradient estimation.

A more specific summary of our contributions is give below:

- Showing the gradient of the loss function can be written as $\nabla\mathcal{L} = \overrightarrow{D} f(V)$, where $f(z) = \frac{1-e^z}{z}$, $\overrightarrow{D}$ is the vector of the expectation values of Hadamard tests for a set of Pauli strings, and $V$ is a matrix constructed based on the parameters $\overrightarrow{a}$ (see Theorem 3).
- An algorithm that estimates $\nabla\mathcal{L}(\overrightarrow{a})$ using $\tilde{O}(p)$ Hadamard tests and $O(p^3 + pn)$ classical time.
- When the shadow norm of the observable is bounded, the gradient can be estimated with $O(\log p)$ copies and $\mathsf{poly}(n)$ time (see Section 5).
- A master's theorem for the general case where the Pauli terms are not closed under commutation (see Theorem 2).
- Generalization to Hamiltonians with generic Hermitian terms when the DLA has polynomial dimensionality (see Theorem 7).
- Approximation methods for gradient estimation when the DLA dimensionality is exponential (see Section 7.1).

| | Circuit Changes | Sample Complexity | Running Time* |
|---|---|---|---|
| SPSR[25, 26] | $p$ | $O(p)$ | $O(n^{a+b})$ |
| NPSR[27] | $\tilde{O}(p\|\|A\|\|)$ | $\tilde{O}(p\|\|A\|\|)$ | $O(n^{a+b})$ |
| $SU(N)$[23] | $p$ | $O(p)$ | $\exp(\Theta(n))$ |
| Backpropagation [39] | - | $O(\log^2 p)$ | $p\exp\left(\tilde{O}(n)\right)$ |
| Theorem 3, 5 | 0 | $O(p), O(\log p)^\star$ | $\tilde{O}(n^b + n^{3a})$ |

Table 1: Rough comparison of various methods for estimating the gradient of a PQC with $p$ parameters and DLA based on Pauli strings with poly$(n)$ dimensionality. * For presentation convenience of the runtime, it is assumed that $p = \Theta(n^a)$ and that each use of the ansatz takes $\Theta(n^b)$ quantum time, where $a$ and $b$ are arbitrary constants. $^\star$ Conditioned on bounded shadow norm of $O$ and corresponding polynomial time complexity.

### 1.2 Comparison With Related Methods

We consider three complexity measures: (1) sample complexity, (2) the classical post-processing time, and (3) the number of distinct circuits that need to be evaluated to obtain all the partial derivatives. Table 1 demonstrates a simplified comparison with existing works for gradient estimation. For a more intuitive comparison, it is assumed that the PQC acts on $n$ qubits with gate complexity $\Theta(n^b)$ and $p = \Theta(n^a)$ parameters, where $a$ and $b$ are arbitrary constants. The table shows that our approach provides an exponential advantage in terms of the sample complexity and a polynomial speed-up in classical running time. Below, we highlight some of the most relevant approaches for comparison to our work.

**Stochastic PSR:** This method is a generalization of PSR [25, 26], where each partial derivative is written as an integral, and a Monte Carlo strategy is used to estimate it. David *et al.* [26] also presented a generalization of the PSR using the Discrete Fourier series. Here, the parameters are jointly shifted depending on the spectrum of the ansatz.

This method is efficient when the Hamiltonian $A$ is promised to have equidistant eigenvalues. For a generic PQC one first needs to compute the spectral decomposition of $A$ to find the pattern of the parameter shifts. Therefore, this process in general takes $\exp\{\Theta(n)\}$ classical time as $A$ is an exponentially large matrix.

**Nyquist PSR:** Recently [27] 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 [26, 40]. 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 $\theta$ 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.

**Lie algebraic:** This is another approach [23] based on Lie algebra and a nice connection to the geometry of $SU(2^n)$ matrices and the adjoint operator. The gradient is calculated by finding the Jacobian matrix of the matrix representation of PQC. Hence, the running time scales as $p2^{\Theta(n)}$.

**Shadow tomography:** A recent work [39] proposes a quantum backpropagation method for PQC of the form $U(\overrightarrow{a}) = \prod_{j=1}^{p} e^{ia_j P_j} U_j$, where $U_j$ are fixed unitaries and $P_j$ are fixed Pauli words. Leveraging the shadow tomography of [41], the method achieves a sample complexity scaling as $\log^2 p$, but with the cost of an exponential classical memory requirement of $p2^{\tilde{O}(n)}$.

**Classical simulation:** Existing simulation methods such as g-sim [31] can efficiently simulate a quantum system and hence compute the gradient, under the assumption that both the Hamiltonian $A(\overrightarrow{a})$ and the observable $O$ have polynomial Lie dimensionality. In contrast, we only require $A(\overrightarrow{a})$ (rather than both) to have polynomial Lie dimensionality.

Assuming both input state $\rho$ and observable $O$ are efficiently classically simulatable, our work and that of [31] have time complexity scaling polynomially with the dimension of the Lie algebra. Our work is complementary to [31] and offers distinct advantages when $\rho$ and/or $O$ are not classically simulatable. This arises, for example, when $O$ does not lie in a Lie algebra of polynomial dimension, and $\rho$ is a physical quantum state obtained via an external process (e.g., quantum sensing) or generated by a unitary that does not have polynomial Lie dimensionality.

When $\rho$ is stored in a physical quantum state, [31] relies on computing expectation values of the Lie algebra basis elements ($B_\alpha^{(\lambda)}$ in the reference) under $\rho$. This is only efficient when certain classes of $\rho$ including product states or stabilizer states. In more general setting, such expectation values must be estimated in a quantum computer and the sample complexity of the estimation may not be polynomial in $n$. Furthermore, even when $\rho$ is classically simulatable, we expect significant (possibly exponential) separation from [31] when the observable does not have polynomial Lie dimensionality while it can be implemented in polynomial quantum time on a quantum device and has bounded shadow norm. For example, in fidelity estimation, $O = |\phi\rangle\langle\phi|$, where $|\phi\rangle = e^{iH}|0\rangle$ and $H$ is a Hamiltonian not admitting a polynomial Lie decomposition. Here, $O$ is a low-rank observable with bounded shadow norm, but exponential Lie dimensionality. Hence, classical simulation runs exponentially in time.

## 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, 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 $\|A\| = \sup_{|\phi\rangle} \|A|\phi\rangle\|$. The Hilbert-Schmidt norm of $A$ is defined as $\|A\|_2 := \sqrt{\mathrm{tr}\{A^\dagger A\}}$

### 2.1 Pauli Group

Together with the identity, they are denoted as $\{\sigma^0, \sigma^1, \sigma^2, \sigma^3\}$ with

$$\sigma^0 = I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \qquad \sigma^1 = X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma^2 = Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma^3 = Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

The single-qubit Pauli group $\mathcal{P}_1$ is the 16 element set $\{c\sigma^s : s = 0, 1, 2, 3,\ c = \pm 1, \pm i\}$.

For $n$ qubit systems, the Pauli tensor products are denoted as $\sigma^{\mathbf{s}} := \sigma^{s_1} \otimes \sigma^{s_2} \otimes \cdots \otimes \sigma^{s_d}$, for all $\mathbf{s} \in \{0,1,2,3\}^n$. The $n$-qubit Pauli group $\mathcal{P}_n$ is then defined as the group generated by $n$-fold tensor products of the Pauli matrices:

$$\mathcal{P}_n = \{c\sigma^{\mathbf{s}} : \mathbf{s} \in \{0,1,2,3\}^n,\ c = \pm 1, \pm i\}\,.$$

This group has $4^{n+1}$ elements and spans any operator on the space of $n$ qubits:

**Fact 1.** *Any bounded operator $A$ on $n$ qubits can be uniquely written as $A = \sum_{\mathbf{s}\in\{0,1,2,3\}^n} a_{\mathbf{s}}\ \sigma^{\mathbf{s}}$, where $a_{\mathbf{s}} = \frac{1}{2^n} \operatorname{tr}\{A\sigma^{\mathbf{s}}\}$.*

## 2.2 General Framework

The objective is to minimize a cost function defined as

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

where $O$ is a fixed observable, $\rho$ is the initial (mixed) state, and $U(\overrightarrow{a})$ is a parameterized quantum circuit with $\overrightarrow{a} = (a_1, \cdots, a_p)$ as the vector of parameters. As $U$ is unitary, we can always write $U(\overrightarrow{a}) = e^{iA(\overrightarrow{a})}$ for some Hamiltonian matrix $A$. To ensure computational tractability, it is assumed that the number of parameters $p = \mathsf{poly}(n)$, with $n$ being the number of qubits.

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 $t$:

$$\overrightarrow{a}^{(t+1)} = \overrightarrow{a}^{(t)} - \eta_t \nabla\mathcal{L}(\overrightarrow{a}^{(t)}), \tag{2}$$

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 $\rho$ are either unknown or computationally intractable.

There have been several approaches to estimating the gradient [9, 11, 19, 21, 18, 20, 22, 23, 24, 21]. 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 [18]. First-order methods (e.g., parameter shift rule) directly calculate the partial derivatives [20].

## 2.3 Hadamard Test

Given a unitary $U$, the Hadamard test, which is a special case of the phase estimation, is a quantum circuit that we can use to estimate the real or imaginary value of $\langle\psi|U|\psi\rangle$ for some state $|\psi\rangle$. The circuit consists of two Hadamard gates and a controlled version of $U$.

When the ansatz has a simple form $U(\theta) = e^{i\theta\sigma^{\mathbf{s}}}$ its derivative can be written as below and be directly estimated via a Hadamard test [11]:

$$D_{\mathbf{s}} := i \operatorname{tr}\Big\{O[\sigma^{\mathbf{s}}, \rho^{out}]\Big\}, \tag{3}$$

where $\rho^{out} = U(\overrightarrow{a})\rho U(\overrightarrow{a})^\dagger$ is the output of the ansatz on input $\rho$. The above equation is based on the fact that $\frac{\mathrm{d}U}{\mathrm{d}\theta} = i\sigma^{\mathbf{s}}U(\theta)$. Estimating the gradient through the Hadamard test can enhance computing efficiency, and allows for the use of measurement optimization techniques. Moreover, it can be used to compute higher-order partial derivatives used in higher-order optimization algorithms [42].

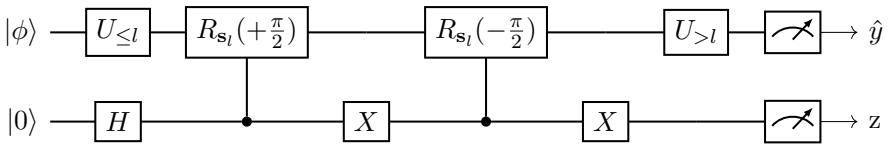


Figure 1: Hadamard test with backpropagation for measuring the partial derivative of product ansätze with respect to a parameter $a_{\mathbf{s}_l}$ appearing at layer $l$. Here $U_{\leq l}$ corresponds to the first $l$ layers of the ansatz, and $U_{>l}$ to the remaining layers. in addition, $X$ is the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

This approach can be extended to product ansätze that are 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 $U_l(a_l) := e^{ia_{\mathbf{s}_l}\sigma^{\mathbf{s}_l}}$ with $\sigma^{\mathbf{s}_l}$ being a Pauli string. More formally, the partial derivative of a product ansätze can be written in terms of commutators with associated Pauli strings. This statement is summarized below:

**Fact 2.** *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 $l$. Then, the derivative of the loss is given by:*

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{s}_l}} = \mathrm{tr}\Big\{OU_{>l}[\sigma^{\mathbf{s}_l}, \rho_l^{out}]U_{>l}^\dagger\Big\}, \tag{4}$$

*where $[\cdot,\cdot]$ is the commutator operation, and $U_{>l}$ denotes the unitary components from layer $l+1$ to $L$.*

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 [22, 18]. 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. Extending the Hadamard test for gradient estimation of more generic ansätze is one of the main focuses of this work.

### 2.4 Differential of The Matrix Exponential

The matrix exponential is defined as

$$\exp(X) = e^X = \sum_{k=0}^{\infty} \frac{X^k}{k!},$$

where $X$ is a square matrix. Due to non-commutativity of matrix product, the differential of the exponential map has a more complex formula compared to the exponential function. Suppose $X(\tau)$ is a differentiable matrix (linear operator) as a function of the variable $\tau \in \mathbb{R}$. The adjoint map is defined as the mapping

$$\mathsf{ad}_X(Y) = [X, Y] = XY - YX$$

for square $n \times n$ matrixes $X, Y \in \mathsf{GL}(n, \mathbb{C})$. Then, for any $k = 0, 1, ...$, we can define

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

The exponential map and the adjoint are fundamental concepts in the theory of Lie groups and Lie algebras, describing how a Lie group or Lie algebra acts on its own Lie algebra by conjugation (a standard text book on this topic is [35]). The adjoint operator is connected to the derivative of the matrix exponential.

**Theorem 1** ([35]). *Suppose $X(\tau)$ is a differentiable (linear) operator with respect to a variable $\tau \in \mathbb{R}$. Then, the differential of the matrix exponential is given by*

$$\frac{\mathrm{d}\exp\{X(\tau)\}}{\mathrm{d}\tau} = \exp\{X(\tau)\}\frac{1-\exp\{-\mathsf{ad}_X\}}{\mathsf{ad}_X}\frac{\mathrm{d}X(\tau)}{\mathrm{d}\tau}. \tag{5}$$

## 3 Binary Encoding of Pauli Operators

Representation of Pauli words with binary strings has been studied in quantum error correction [33, 34] to construct the Stabilizer codes. It is well-known that the Pauli group $\mathcal{P}_n$ is *isomorphic* to the *semi direct product* of $\mathbb{Z}_4$ and $\mathbb{Z}_2^{2n}$. This binary representation allows us to write the partial derivatives of $\mathcal{L}(\overrightarrow{a})$ as linear combination of terms related to the Hadamard tests applied to the ansatz output. In this work, we present an explicit form of such a mapping. Below, we start with a series of notations and definitions and present a binary encoding technique to study the products of various Pauli string terms.

The phase scalar $c \in \{\pm 1, \pm i\}$ in $\mathcal{P}_n$ can be written as $i^a$, where $a \in \mathbb{Z}_4$, the modulo-four group. As for the Pauli operators, consider the binary vector group $\mathbb{Z}_2 \times \mathbb{Z}_2 = \{(0|0), (0|1), (1|0), (1|1)\}$ with the element-wise modulo two addition:

$$(a^0|a^1) + (b^0|b^1) = (a^0 \oplus b^0|a^1 \oplus b^1),$$

where $\oplus$ is the binary addition. We use $(\cdot|\cdot)$ to distinguish between the first and the second components of elements of $\mathbb{Z}_2 \times \mathbb{Z}_2$. We associate the identity and each Pauli operator with elements of $\mathbb{Z}_2 \times \mathbb{Z}_2$ as

$$\sigma^0 \to (0|0), \qquad \sigma^1 \to (0|1), \qquad \sigma^2 \to (1|0), \qquad \sigma^3 \to (1|1).$$

Extending to $n$-qubits, $(\mathbb{Z}_2 \times \mathbb{Z}_2)^n$ is defined as the set of all $(\mathbf{a}^0|\mathbf{a}^1)$ for binary strings $\mathbf{a}^0, \mathbf{a}^1 \in \mathbb{Z}_2^n$ with the element-wise addition:

$$(\mathbf{a}^0|\mathbf{a}^1) \oplus (\mathbf{b}^0|\mathbf{b}^1) = (\mathbf{a}^0 \oplus \mathbf{b}^0|\mathbf{a}^1 \oplus \mathbf{b}^1).$$

Similarly, any Pauli string $\sigma^{\mathbf{s}}, \mathbf{s} \in \{0,1,2,3\}^n$ is associated with $(\mathbf{s}^0|\mathbf{s}^1)$ where $\mathbf{s}^0 = (s_1^0, \cdots, s_d^0), \mathbf{s}^1 = (s_1^1, \cdots, s_d^1)$ are binary strings. Therefore, we frequently switch between representations of s as a member of $\mathbb{Z}_4^n$ and $(\mathbb{Z}_2 \times \mathbb{Z}_2)^n$.

**Example 1.** *The Pauli string $X \otimes Y \otimes X$ which is associated with $\sigma^{\mathbf{s}}$ with $\mathbf{s} = (1,2,1)$ has the following binary encoding: $((0,1,0)|(1,0,1))$.*

The above association induces an isomorphism between the Pauli group $\mathcal{P}_n$ and the semi direct product of $\mathbb{Z}_4$ and $\mathbb{Z}_2^{2n}$:

**Remark 1.** *$\mathcal{P}_n$ is isomorphic to the central product of $\mathbb{Z}_4$ and $\mathbb{Z}_2^{2n}$. Ignoring the phase scalar $c$ in $\mathcal{P}_n$, the group quotient is isomorphic to the Binary vector group $\mathbb{Z}_2^{2n}$. On the other hand, the scalars $\{\pm 1, \pm i\}$ are isomorphic to the cyclic group of order 4 which is $\mathbb{Z}_4 = \{0,1,2,3\}$ with modulo-4 addition as the group action.*

We formalize the above mapping in the following definition.

**Definition 1.** *Any element of $\mathcal{P}_n$, written as $i^a\sigma^{\mathbf{s}}$, is represented as $(a, \mathbf{s})$ where $a \in \mathbb{Z}_4$ and $\mathbf{s} \in \mathbb{Z}_2^{2n}$. Such a representation is defined by the mapping $\phi : \mathcal{P}_n \to \mathbb{Z}_4 \times \mathbb{Z}_2^{2n}$, that sends $\phi(i^a\sigma^{\mathbf{s}}) = (a, \mathbf{s})$.*

**Encoding the products of Pauli words.** Inspired by the Levi-Civita symbol, we define a sign function on $i, j \in \mathbb{Z}_4$ (or $\mathbb{Z}_2 \times \mathbb{Z}_2$) as

$$\delta(i,j) := \begin{cases} -1 & \text{if } (i,j) = (1,3),(2,1),(3,2) \\ 1 & \text{if } (i,j) = (1,2),(2,3),(3,1) \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

For vectors $\mathbf{u}, \mathbf{v}$, define $\delta(\mathbf{u}, \mathbf{v}) = \sum_j \delta(u_j, v_j)$.

**Lemma 1.** *The product of any pair of Pauli strings $\sigma^{\mathbf{s}}, \sigma^{\mathbf{r}}$ equals*

$$\sigma^{\mathbf{s}}\sigma^{\mathbf{r}} = i^{\delta(\mathbf{s},\mathbf{r})}\sigma^{\mathbf{s}\oplus\mathbf{r}}.$$

*Proof.* It is not difficult to verify the lemma for single qubit case by testing it for all possible combinations. Building on the single qubit case, for general $n > 1$, with the tensor product, we have that

$$\begin{aligned} \sigma^{\mathbf{s}}\sigma^{\mathbf{r}} &= \bigotimes_j \sigma^{s_j}\sigma^{r_j} \\ &= \bigotimes_j i^{\delta(s_j,r_j)}\sigma^{s_j\oplus r_j} \\ &= i^{\sum_j \delta(s_j,r_j)} \bigotimes_j \sigma^{s_j\oplus r_j} \\ &= i^{\sum_j \delta(s_j,r_j)}\sigma^{\mathbf{s}\oplus\mathbf{r}}. \end{aligned}$$

where the second equality is due to the single qubit case and the third equality is due to the fact that the phase scalars are multiplicative. The last equality is due to the definition of $\oplus$ for binary strings. □

**Lemma 2.** *The mapping $\phi : \mathcal{P}_n \to \mathbb{Z}_4 \rtimes \mathbb{Z}_2^{2n}$ defined as $c\sigma^{\mathbf{s}} \mapsto \phi(c\sigma^{\mathbf{r}}) = (c, \mathbf{r})$ is an isomorphism of $\mathcal{P}_n$ onto the semi direct product of $\mathbb{Z}_4$ and $\mathbb{Z}_2^{2n}$.*

*Proof.* We define the semi direct product of $\mathbb{Z}_4$ and $\mathbb{Z}_2^{2n}$ by defining the action that maps $\mathbb{Z}_4$ to the group of automorphisms of $\mathbb{Z}_2^{2n}$. For $a \in \mathbb{Z}_4$, we define the automorphism $\alpha_a : \mathbb{Z}_2^{2n} \to \mathbb{Z}_2^{2n}$ as $\alpha_a(\mathbf{s}) = a\mathbf{s}$ where $a\mathbf{s}$ is defined as the element-wise product of $a$ and $\mathbf{s}$. Then, the semi direct product of $\mathbb{Z}_4$ and $\mathbb{Z}_2^{2n}$ is defined as the mapping that sends any pair of elements $(a, \mathbf{s}), (b, \mathbf{r})$ to $(mod_4(a + b + \sum_j \delta(s_j, r_j)), \mathbf{s} \oplus \mathbf{r})$. With this definition, Lemma 1 shows that $\phi(\sigma^{\mathbf{s}}\sigma^{\mathbf{r}}) = \phi(\sigma^{\mathbf{s}})\phi(\sigma^{\mathbf{r}})$. Adding the phase scalars is addressed by this definition. □

With this result, and by adding the phase scalars, for any pair $i^a\sigma^{\mathbf{s}}$ and $i^b\sigma^{\mathbf{r}}$ from the Pauli group $\mathcal{P}_n$, the product is characterized by the $\phi$ map in Definition 1 as

$$(i^a\sigma^{\mathbf{s}})(i^b\sigma^{\mathbf{r}}) = \phi^{-1}((\mathrm{mod}_4(a + b + \delta(\mathbf{s}, \mathbf{r})), \mathbf{s} \oplus \mathbf{r})).$$

**Example 2.** *Consider $P_1 = iX \otimes Y \otimes X$ and $P_2 = Z \otimes X \otimes Y$ that are encoded to $(1, ((010)|(101)))$ and $(0, ((101)|(110)))$, respectively. Then, $P_1P_2$ is associated with the binary encoding $(0, ((111)|(011)))$ which represents $Y \otimes Z \otimes Z$.*

**Example 3.** *For $n = 1$ as an example, the Cayley Table of product of Pauli matrices is given by the following table:*

| $\bullet$ | $(b, 01)$ | $(b, 10)$ | $(b, 11)$ |
|---|---|---|---|
| $(a, 01)$ | $((a+b)_4, 00)$ | $((a+b+1)_4, 11)$ | $(((a+b-1)_4), 10)$ |
| $(a, 10)$ | $((a+b-1)_4, 11)$ | $((a+b)_4, 00)$ | $((a+b+1)_4, 01)$ |
| $(a, 11)$ | $((a+b+1)_4, 10)$ | $((a+b-1)_4, 01)$ | $((a+b)_4, 00)$ |

The binary representation also characterizes the commutation relations between Pauli strings.

**Lemma 3.** *The commutator of any pair of Pauli strings $\sigma^{\mathbf{s}}, \sigma^{\mathbf{r}}$ is given by $[\sigma^{\mathbf{s}}, \sigma^{\mathbf{r}}] = 2i^{\delta(\mathbf{s},\mathbf{r})}\sigma^{\mathbf{s}\oplus\mathbf{r}}$.*

*Proof.* The proof is based on Lemma 1:

$$\begin{aligned}[\sigma^{\mathbf{s}}, \sigma^{\mathbf{r}}] &= \sigma^{\mathbf{s}}\sigma^{\mathbf{r}} - \sigma^{\mathbf{r}}\sigma^{\mathbf{s}} \\ &= i^{\sum_j \delta(s_j,r_j)}\sigma^{\mathbf{s}\oplus\mathbf{r}} - i^{\sum_j \delta(r_j,s_j)}\sigma^{\mathbf{r}\oplus\mathbf{s}}.\end{aligned}$$

Note that $\mathbf{r} \oplus \mathbf{s} = \mathbf{s} \oplus \mathbf{r}$ and that $\delta(r_j, s_j) = -\delta(s_j, j_j)$. Therefore, as $i^{-1} = -i$, the commutator equals to $[\sigma^{\mathbf{s}}, \sigma^{\mathbf{r}}] = 2i^{\sum_j \delta(s_j,r_j)}\sigma^{\mathbf{s}\oplus\mathbf{r}}$. □

## 4 Gradient Formulation for PQC with Pauli Decomposition

Recall that the objective function $\mathcal{L}(\overrightarrow{a})$ in (1) and the expectation values $D_{\mathbf{s}}$ in (3) can be estimated using Hadamard test. We show how the gradient of $\mathcal{L}$ can be evaluated in terms of $D_{\mathbf{s}}$ quantities. We present a master theorem writing the partial derivative is a linear combination of several terms similar to the product ansätze as in Fact 2. Then we show that under certain algebraic structures, the linear combination collapses into $\mathsf{poly}(p)$ terms.

**Theorem 2.** *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 $\mathcal{S} \subseteq \{0,1,2,3\}^n$. Then, for any $\mathbf{r} \in \mathcal{S}$,*

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

*where $D_{\mathbf{t}}$ is defined as in (3), and $\delta(\cdot,\cdot)$ is defined as in (6).*

The proof of the theorem is delayed until Section 6. With this theorem, the derivative of loss can be measured by applying the standard Hadamard test followed by classical corrections. As a sanity check, Appendix A presents an explicit derivation of the gradient in the single qubit case.

Next, we study the simplification of the partial derivative for Hamiltonians that have an algebraic structure with the goal to find an efficient estimator of the gradient.

We show that when $\mathcal{S}$ in the theorem is isomorphism to a subgroup of $(\mathbb{Z}_2)^{2n}$, the expression simplifies significantly. In that case,the set of Pauli strings $\sigma^{\mathbf{s}}, \mathbf{s} \in \mathcal{S}$ is closed under the commutation up to a global phase.

For that, we first consider a geometric representation of the gradient in terms of $D_{\mathbf{s}}$ terms. Let $\mathcal{S} = \{\mathbf{s}_1, \cdots, \mathbf{s}_p\}$, where $p$ is the number of parameters. Let $\mathbf{e}_1 = (1, 0, \cdots, 0)^T, \cdots, \mathbf{e}_p = (0, \cdots, 0, 1)^T$ be the canonical basis vectors in $\mathbb{R}^p$. We associate each $\mathbf{s}_j$ with $\mathbf{e}_j$ which is also denoted by $\mathbf{e}_{(\mathbf{s}_j)}$. Now consider the vector of the expectation terms $\overrightarrow{D} = (D_{\mathbf{s}_1}, \cdots, D_{\mathbf{s}_p})$ as a row vector in $\mathbb{R}^p$:

$$\overrightarrow{D} = (D_{\mathbf{s}_1}, \cdots, D_{\mathbf{s}_p}) = \sum_{j=1}^{p} D_{\mathbf{s}_j}\mathbf{e}_{(\mathbf{s}_j)}.$$

The following theorem shows that the gradient can be expressed as a vector derived from $\overrightarrow{D}$ by applying a matrix transformation. The matrix is defined in terms of the parameters and the group structure of the Pauli strings.

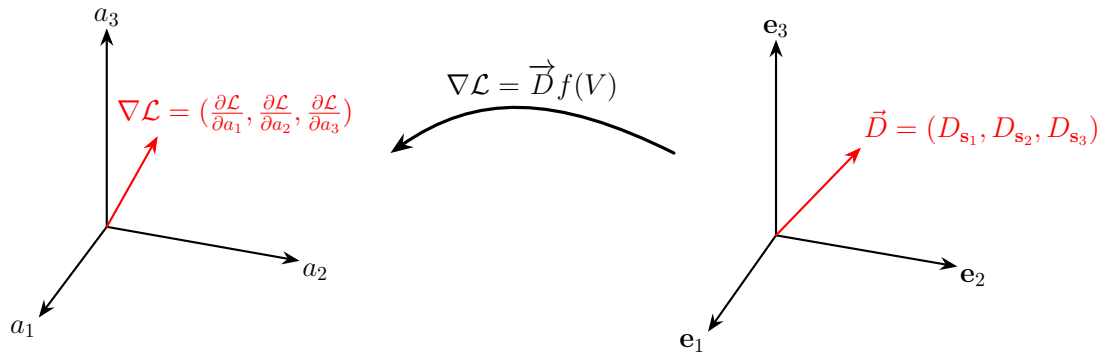


Figure 2: The left picture represents the parameter space of the ansatz with three parameters $\overrightarrow{a} = (a_1, a_2, a_3)$ corresponding to the Pauli strings $\sigma^{\mathbf{s}_1}, \sigma^{\mathbf{s}_2}, \sigma^{\mathbf{s}_3}$ for $\mathbf{s}_1, \mathbf{s}_2, \mathbf{s}_3 \in \mathcal{S}$. Here, the gradient is a vector in this space. When the corresponding Pauli strings are closed under commutation, the gradient can be expressed as a vector in the space of Hadamard test outcomes $D_{\mathbf{s}}$ as shown in the right picture. The transformation from the right to the left picture is given by the matrix $V$ defined in (8).

**Theorem 3.** *Consider the ansatz $U(\overrightarrow{a}) = \exp\{iA(\overrightarrow{a})\}$, with $A(\overrightarrow{a}) = \sum_{\mathbf{s}\in\mathcal{S}} a_{\mathbf{s}}\sigma^{\mathbf{s}}$, where $\mathcal{S}$ is closed under commutation and $|\mathcal{S}| = p$. Define a $p \times p$ matrix $V$ where the $j$th column is given by*

$$\mathbf{v}_j := 2i \sum_{\mathbf{s}\in\mathcal{S}} a_{\mathbf{s}} i^{\delta(\mathbf{s},\mathbf{s}_j)}\ \mathbf{e}_{(\mathbf{s}\oplus\mathbf{s}_j)}. \tag{8}$$

*Then, the gradient of the loss is given by*

$$\nabla\mathcal{L} = \overrightarrow{D} f(V), \tag{9}$$

*where $f(z) = \frac{1-e^z}{z}$ with $f(0) = 1$. Moreover there is an algorithm (Algorithm 1) that computes $\nabla\mathcal{L}$ in $O(p^3 + pn)$ time with $O(p)$ use of the Hadamard tests.*

*Proof.* First, as the Paulies $\sigma^{\mathbf{s}}, \mathbf{s} \in \mathcal{S}$ are closed under commutation, $\mathcal{S}$ is closed under the "$\oplus$" operation. Hence, each term $\mathbf{s}_1 \oplus \cdots \oplus \mathbf{s}_k \oplus \mathbf{r}$ in (7) remains in $\mathcal{S}$. As a result the infinite-length sum in Theorem 2 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 $g_{\mathbf{s}}(\mathbf{r}) \in \mathbb{R}$. This means that the partial derivative is always a linear combination of $D_{\mathbf{s}}, \mathbf{s} \in \mathcal{S}$ terms. However, the challenge is in computing the coefficients $g_{\mathbf{s}}(\mathbf{r})$ that are coming from an infinite-length sum. We present a method to address this issue.

Recall the association of $\mathbf{s}$ with the canonical basis vectors in $\mathbb{R}^p$ and the vector $\overrightarrow{D}$ in $\mathbb{R}^p$. Then $\frac{\partial\mathcal{L}}{\partial a_{\mathbf{r}}}$ is a vector in $\mathbb{R}^p$ with the representation

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

Recall the matrix $V$. By an induction argument, it is not difficult to check that

$$V^k \mathbf{e}_{(\mathbf{s}_i)} = (2i)^k \sum_{\mathbf{s}_1\in\mathcal{S}} \cdots \sum_{\mathbf{s}_k\in\mathcal{S}} \prod_{j=1}^{k} \left(a_{\mathbf{s}_j} i^{\delta(\mathbf{s}_j, \mathbf{s}_1\oplus\cdots\oplus\mathbf{s}_{j-1}\oplus\mathbf{r})}\right) \mathbf{e}_{\mathbf{s}_1\oplus\cdots\oplus\mathbf{s}_k\oplus\mathbf{r}},$$

and that

$$f(V)\mathbf{e}_{(\mathbf{r})} = \sum_{k=0}^{\infty} \frac{(2i)^k}{(k+1)!} \sum_{\mathbf{s}_1 \in \mathcal{S}} \cdots \sum_{\mathbf{s}_k \in \mathcal{S}} \Big( \prod_{j=1}^{k} a_{\mathbf{s}_j} i^{\delta(\mathbf{s}_j, \mathbf{s}_1 \oplus \cdots \oplus \mathbf{s}_{j-1} \oplus \mathbf{r})} \Big) e_{\mathbf{s}_1 \oplus \cdots \oplus \mathbf{s}_k \oplus \mathbf{r}}.$$

Therefore, the gradient of the loss is given by $\nabla \mathcal{L} = \overrightarrow{D} f(V)$.

This representation is the key to the efficient computation of the gradient from the vector representation of the $D_{\mathbf{t}}$. One first needs to estimate all $D_{\mathbf{s}_i}, i \in [p]$ which can be done with $O(p)$ Hadamard test. Creating the vector $\overrightarrow{D}_{\mathbf{s}} = (D_{\mathbf{s}_1}, \cdots, D_{\mathbf{s}_p})$ and applying the matrix $f(V)$ give the gradient as summarized in Algorithm 1. The runtime of the algorithm is $O(p^3 + pn)$ because each column $\mathbf{v}_i$ is computed in $O(pn)$ time. Also $f(V)$ can be computed by standard algorithms for matrix functions, such as Schur-based methods or scaling-and-squaring variants, in $O(p^3)$ arithmetic operations. □

**Example 4.** *Consider the ansatz $U(\overrightarrow{a}) = e^{i(a_1\sigma^1 + a_2\sigma^2 + a_3\sigma^3)}$. The set of Pauli strings in the Hamiltonian is $\mathcal{S} = \{1, 2, 3\}$ which is closed under commutation. Hence, the partial derivatives of the loss are linear combinations of $D_1$, $D_2$, and $D_3$. We can associate normalized $D_1$ with the first canonical vector $e_1 = (1, 0, 0)$ and normalized $D_2$ with $e_2 = (0, 1, 0)$ and normalized $D_3$ with $e_3 = (0, 0, 1)$. Then, the partial derivative is a vector in this space. The gradient is expressed as a vector derived from $\overrightarrow{D} = D_1\mathbf{e}_1 + D_2\mathbf{e}_2 + D_3\mathbf{e}_3$. Figure 2 illustrates this example.*

**Algorithm 1** Subgroup Gradient Estimation

**Input:** $\mathcal{S}$

1: **procedure** Subgroup Gradient Estimation
2: Estimate $D_{\mathbf{s}}, \mathbf{s} \in \mathcal{S}$ with Hadamard tests.
3: Compute the matrix $V$ where the column $i$ is computed as in (8).
4: Compute the matrix $f(V)$, where $f(z) = (1 - e^z)/z$ (extended continuously at $z = 0$).
5: **Return** $\widehat{\nabla \mathcal{L}} = \overrightarrow{\hat{D}} f(V)$.

A simple example of a Hamiltonian closed under commutation is $A = a_1 X^{\otimes n} + a_2 Y^{\otimes n} + a_3 Z^{\otimes n}$. The gradient in this case can be computed by a $3 \times 3$ matrix and 3 Hadamard tests, irrespective of the dimensionality of the quantum system. Another class is $K$-junta Hamiltonians, that acting non-trivially on $k$ qubits. In that case, all the Pauli strings $\sigma^{\mathbf{s}}, \mathbf{s} \in \mathcal{S}$, will be of the form $\sigma_k^{\mathbf{s}} \otimes I_{n-k}$, the set $\mathcal{S}$ will be closed under commutation.

**Remark 2.** *When the closedness condition is not directly satisfied, one can apply Theorem 3 to the DLA generated by the terms in $A(\overrightarrow{a})$. When the dimensionality of the DLA is $m$, Algorithm 1 computes $\nabla \mathcal{L}$ in $O(m^3 + dm)$ time with $O(m)$ use of the Hadamard tests. Hence, Algorithm 1 is efficient when the DLA has dimensionality polynomial in $n$.*

**Example 5.** *The following Hamiltonian has a DLA with dimensionality $O(n^2)$:*

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

*where $X_i, Z_i$ are the corresponding Pauli operators. In this case, applying Theorem 3 to the DLA generated by $H$, Algorithm 1 computes $\nabla \mathcal{L}$ in $O(n^6)$ time with $O(n^2)$ use of the Hadamard tests.*

## 5 Efficient Estimation 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 shadow tomography [41, 32, 17, 43, 44]. The components of the gradient $\nabla\mathcal{L}$ correspond to $p$ observables that only depend on $\rho^{out}$ without any reconfigurations. In that case, having several copies of $\rho^{out}$, we can efficiently estimate all the components of the gradient. Based on Theorem 3, the observables are $p$ different Hadamard tests denoted by $H_{\mathbf{s}_j}, j \in [p]$ for measuring $D_{\mathbf{s}_j}$. The following lemma gives an explicit characterization of these observables.

**Lemma 4.** *$D_{\mathbf{s}} = \mathrm{tr}\{H_{\mathbf{s}}\rho^{out}\}$, where $\rho^{out}$ is the ansatz output state and*

$$H_{\mathbf{s}} = R^{\dagger}_{\mathbf{s},-} O R_{\mathbf{s},-} - R^{\dagger}_{\mathbf{s},+} O R_{\mathbf{s},+}, \tag{10}$$

*where $O$ is the ansatz observable and $R_{\mathbf{s},\pm} = e^{-i\frac{\pi}{4}}$.*

*Proof.* Recall that

$$D_{\mathbf{s}} = i\,\mathrm{tr}\Big\{O[\sigma^{\mathbf{s}}, \rho^{out}]\Big\}.$$

Moreover, note from [11] the following property of the commutator for any operator $B$:

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

With this observation, we obtain

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

where we used the cyclic property of the trace. The last equation gives the expression for $H_{\mathbf{s}}$. □

In order to estimate the gradient, one needs to estimate all the expectation values $o^{\pm}_{\mathbf{s}} := \mathrm{tr}\Big\{R^{\dagger}_{\mathbf{s},\pm}OR_{\mathbf{s},\pm}\rho^{out}\Big\}$ for $\mathbf{s} \in \mathcal{S}$. We use CST to exponentially reduce the measurement shots for the gradient estimation when $O$ has a bounded Hilbert-Schmidt norm. Typically, for non-local bounded norm observables, one can use CST with random Clifford measurements. This approach has sample complexity $O(\log p)$ but generally when the observables are not classically simulatable it has an exponential computational complexity of $2^{\Theta(n)}$. This is because it performs operations (such as product and trace) on $2^n$ by $2^n$ matrices. Particularly, from Theorem 1 of [32] we have the following result for estimating the gradient.

**Theorem 4.** *Given an observable $O$, state $\rho^{out}$, and a set $\mathcal{S} \subseteq \{0,1,2,3\}^n$, the expectation values $o^{\pm}_{\mathbf{s}} := \mathrm{tr}\Big\{R^{\dagger}_{\mathbf{s},\pm}OR_{\mathbf{s},\pm}\rho^{out}\Big\}$ for $\mathbf{s} \in \mathcal{S}$ can be estimated with $\epsilon$ additive error using*

$$N = O\Big(\tfrac{1}{\epsilon^2}\log|\mathcal{S}|\max_{\mathbf{s}\in\mathcal{S}}\Big\|R^{\dagger}_{\mathbf{s},\pm}OR_{\mathbf{s},\pm}\Big\|^2_{shadow}\Big)$$

*copies of $\rho^{out}$, where $\|\cdot\|_{shadow}$ is the shadow norm.*

The shadow norm (Definition 4 in Appendix C) is closely related to the variance of the observable and the set of the unitary transformation used for taking the classical shadows. For random Clifford measurements, it is bounded by the Hilbert-Schmidt norm; whereas for random Pauli measurements, it is bounded by $4^k$, where $k$ is the locality of the observable, not the actual number of qubits [32].

**Theorem 5.** *Consider the ansatz* $U(\overrightarrow{a}) = \exp\{iA(\overrightarrow{a})\}$, *with* $A(\overrightarrow{a}) = \sum_{\mathbf{s}\in\mathcal{S}} a_{\mathbf{s}}\sigma^{\mathbf{s}}$, *where* $\mathcal{S}$ *is closed under commutation. Suppose* $\operatorname{tr}\{O^2\} \leq B$ *for a finite constant* $B$. *Then,* $\nabla\mathcal{L}(\overrightarrow{a})$ *can be estimated with an additive error less than* $\epsilon$ *using* $O(\frac{Be^{pc}}{pc\epsilon^2}\log\frac{|\mathcal{S}|}{\delta})$ *copies of* $\rho^{out}$.

*Proof.* We use CST with random Clifford measurements to estimate the observables $H_{\mathbf{s}}$ for all $\mathbf{s} \in \mathcal{A}$. From Theorem 1 of [32], the estimation of each $D_{\mathbf{s}_j}$ with an additive error $\epsilon$ and probability at least $(1-\delta)$ requires $O\Big(\frac{1}{\epsilon^2}\log\frac{1}{\delta}\max_{\mathbf{s}\in\mathcal{S}}\|H_{\mathbf{s}}\|^2_{shadow}\Big)$ copies of $\rho^{out}$. Therefore, it remains to analyze $\|H_{\mathbf{s}}\|^2_{shadow}$. For random Clifford measurements, the shadow norm is bounded by the Hilbert-Schmidt norm [32], and hence $\|H_{\mathbf{s}}\|^2_{shadow} \leq \operatorname{tr}\{H_{\mathbf{s}}^2\}$. We proceed with the following lemma to bound $\operatorname{tr}\{H_{\mathbf{s}}^2\}$ with the proof given in Appendix D.1.

**Lemma 5.** $\operatorname{tr}\{H_{\mathbf{s}}^2\} \leq 4\operatorname{tr}\{O^2\}$ *for any* $\mathbf{s} \in \{0,1,2,3\}^n$.

The lemma with the theorem's assumption imply that $\|H_{\mathbf{s}}\|^2_{shadow} \leq 4B$. Hence, the estimation of all $D_{\mathbf{s}}, \mathbf{s} \in \mathcal{S}$ with an additive error $\epsilon$ and probability at least $(1-\delta)$ requires $O\Big(\frac{B}{\epsilon^2}\log\frac{|\mathcal{S}|}{\delta}\Big)$ copies of $\rho^{out}$.

Theorem 3 connects the estimation of all $D_{\mathbf{s}}$'s to the estimation of $\nabla\mathcal{L}$ via the relation $\nabla\mathcal{L} = \overrightarrow{D}f(V)$. Hence, it remains to analyze the error propagation from the estimation of $D_{\mathbf{s}}$'s to the estimation of $\nabla\mathcal{L}$. We proceed with the following lemma to bound the operator norm of $f(V)$ with the proof given in Appendix D.2.

**Lemma 6.** *Let* $V \in \mathbb{C}^{p\times p}$ *be the matrix from Theorem 3, and* $f(z) := (1-e^{-z})/z$ *for* $z \neq 0$ *and* $f(0) := 1$. *Then,*

$$\|f(V)\| \leq \frac{e^{\|V\|}-1}{\|V\|}.$$

*Moreover, if* $|a_{\mathbf{s}}| \leq c$ *for all* $\mathbf{s} \in \mathcal{S}$, *then*

$$\|f(V)\| \leq \frac{e^{2pc}-1}{2pc}.$$

With the lemma, we can bound the error propagation from the estimation of $D_{\mathbf{s}}$'s to the estimation of $\nabla\mathcal{L}$. Note that $\nabla\mathcal{L} = \overrightarrow{D}f(V)$ and hence the estimation error of $\nabla\mathcal{L}$ is at most $\|f(V)\|$ times the estimation error of $\overrightarrow{D}$. Hence, $\nabla\mathcal{L}$ can be estimated with an additive error $\epsilon$ and probability at least $(1-\delta)$ using $O\Big(\frac{Be^{pc}}{pc\epsilon^2}\log\frac{|\mathcal{S}|}{\delta}\Big)$ copies of $\rho^{out}$. □

## 6 Proof of Theorem 2

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

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

where

$$\frac{\partial \rho^{out}}{\partial a_{\mathbf{s}}} = \frac{\partial U}{\partial a_{\mathbf{s}}}\left(\rho U^\dagger\right) + (U\rho)\frac{\partial U^\dagger}{\partial a_{\mathbf{s}}}, \tag{11}$$

and we used the fact that $\rho^{out}$ is Fréchet differentiable with respect to $a_{\mathbf{s}}$. To characterize the partial derivative of $U$ we need to introduce some ingredients in Lie algebra and the connection to the derivative of matrix exponential.

We equip the space $\mathsf{GL}(2^n, \mathbb{C})$ of $2^n \times 2^n$ complex matrices with the Lie algebra and the standard Lie bracket.

Next, we present the following lemma that characterizes the partial derivative of $U$ in terms of the adjoint operator.

**Lemma 7.** *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.* 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 (5) we can write

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

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

$$\frac{1-\exp\{-\mathsf{ad}_H\}}{\mathsf{ad}_H} = \sum_{k=0}^{\infty}\frac{(-1)^k}{(k+1)!}(\mathsf{ad}_H)^k. \tag{12}$$

Therefore, replacing $H$ with $-iH$ in this equation, the derivative equals to the following

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

Note that for any $X, Y \in \mathsf{g}$, $(\mathsf{ad}_X(Y))^\dagger = -\mathsf{ad}_{X^\dagger}(Y^\dagger)$. Therefore,

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

□

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 (11). Next, from (5), the partial derivative of $U^\dagger$ can be written 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 $\rho^{out}$ 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 of the loss can be written as the commutator:

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{s}}} = \operatorname{tr}\Big\{O\frac{\partial}{\partial a_{\mathbf{s}}}\rho^{out}\Big\} = \operatorname{tr}\Big\{O\Big[\rho^{out}, \frac{1-\exp\{i\mathsf{ad}_A\}}{\mathsf{ad}_A}(\frac{\partial A}{\partial a_{\mathbf{s}}})\Big]\Big\}. \tag{13}$$

Next, based on the Taylor expansion of the matrix exponential and from the fact that $\frac{\partial A}{\partial a_{\mathbf{s}}} = \sigma^{\mathbf{s}}$, the above quantity decomposes as

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{s}}} = -i\sum_{k=0}^{\infty}\frac{(i)^k}{(k+1)!}\operatorname{tr}\Big\{O\Big[\rho^{out}, (\mathsf{ad}_A)^k(\sigma^{\mathbf{s}})\Big]\Big\}. \tag{14}$$

Then, building on the binary encoding of Pauli operators, and Lemma 3, we can write the adjoint operator as below.

**Lemma 8.** *For any $A, B \in \mathsf{g}$, the corresponding adjoint decomposes as*

$$\mathsf{ad}_A(B) = 2\sum_{\mathbf{r},\mathbf{s}} a_{\mathbf{s}} b_{\mathbf{r}} i^{\delta(\mathbf{s},\mathbf{r})}\sigma^{\mathbf{s}\oplus\mathbf{r}},$$

*where*

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

*are the Pauli coefficients of $A$ and $B$, respectively.*

*Proof.* The proof follows from Lemma 3 and the linearity of adjoint:

$$\mathsf{ad}_A(B) = \sum_{\mathbf{s},\mathbf{r}} a_{\mathbf{s}} b_{\mathbf{r}}[\sigma^{\mathbf{s}}, \sigma^{\mathbf{r}}] = \sum_{\mathbf{s},\mathbf{r}} a_{\mathbf{s}} b_{\mathbf{r}} 2i^{\sum_j \delta(s_j, r_j)}\sigma^{\mathbf{s}\oplus\mathbf{r}}.$$

□

Hence, from Lemma 8 for $\mathsf{ad}_A(\sigma^{\mathbf{r}})$, we can write

$$\mathsf{ad}_A^k(\sigma^{\mathbf{r}}) = 2^k \sum_{\mathbf{s}_1\in\mathcal{S}}\cdots\sum_{\mathbf{s}_k\in\mathcal{S}}\left(\prod_{j=1}^{k} a_{\mathbf{s}_j} i^{\delta(\mathbf{s}_j, \mathbf{s}_1\oplus\cdots\oplus\mathbf{s}_{j-1}\oplus\mathbf{r})}\right)\sigma^{\mathbf{s}_1\oplus\cdots\oplus\mathbf{s}_k\oplus\mathbf{r}}.$$

Therefore, the expression for the partial derivative of $\mathcal{L}$ equals

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

Lastly, recalling that $D_{\mathbf{t}} := i\operatorname{tr}\{O[\sigma^{\mathbf{t}}, \rho^{out}]\}$, the above equation simplifies to

$$\frac{\partial \mathcal{L}(\vec{a})}{\partial a_{\mathbf{s}}} = \sum_{k=0}^{\infty}\frac{(2i)^k}{(k+1)!}\sum_{\mathbf{s}_1\in\mathcal{S}}\cdots\sum_{\mathbf{s}_k\in\mathcal{S}}\left(\prod_{j=1}^{k} a_{\mathbf{s}_j} i^{\delta(\mathbf{s}_j, \mathbf{s}_1\oplus\cdots\oplus\mathbf{s}_{j-1}\oplus\mathbf{r})}\right)D_{\mathbf{s}_1\oplus\cdots\oplus\mathbf{s}_k\oplus\mathbf{r}}.$$

# 7 Extensions and Approximations

In this section we present extensions and generalizations of the results from the previous sections.

## 7.1 Approximations for general case

When $\mathcal{S}$ in $A(\overrightarrow{a})$ is not a subgroup, then the terms appearing in the partial derivative in Theorem 2 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 3 and Algorithm 1 for $\langle S\rangle$. This yields a run-time that scales polynomially with $|\langle\mathcal{S}\rangle|$. 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 $\mathcal{S}$.

### 7.1.1 Truncation

We propose to approximate the partial derivatives by truncating the infinite sum that appeared as (7) in Theorem 2.

**Theorem 6.** *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 $\mathcal{S}$ 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:*

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

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

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

We present a numerical experiment to study the effect of the truncation on the approximation error of the gradient. For that we generated a Hamiltonian

$$H(\overrightarrow{a}) = \sum_{i=1}^{d} a_i Z_i Z_{i+1} + a'_i X_i,$$

where $Z_i$ and $X_i$ are the Pauli operators applied on the $i$'th qubit. Then we use our method with various truncation parameters $K$ to calculate the estimation error. Fig. 3 demonstrates the averaged approximation error measured as the norm distance between the true gradient and our approximation. The graphs are for randomly chosen parameters $a_i, a'_i$ as a function of $K$ for various qubit numbers $n$. The figure shows that with a few number of parameters, one can approximate the gradient with high precision. Shaded regions indicate one standard deviation over 10 trials with random observables $O$.

### 7.1.2 Unbiased Estimation via Randomization

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

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

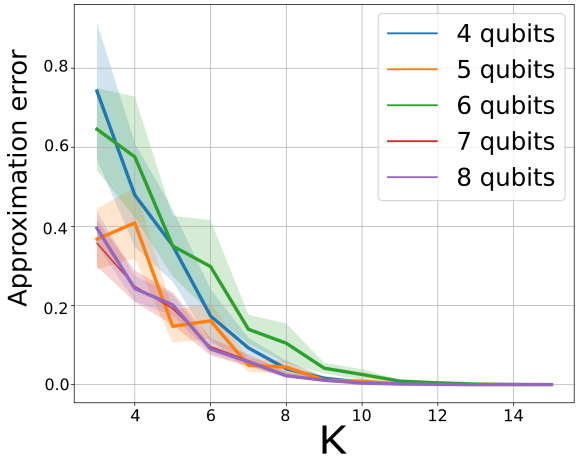


Figure 3: Estimation error for gradient approximation as a function of $K$ for various Hamiltonians with different numbers of qubits.

By sampling $K$ from this distribution we can estimate the partial derivative expression of Theorem 2. Note that

$$\frac{\partial \mathcal{L}}{\partial a_{\mathbf{r}}} = \sum_{k=0}^{\infty} \frac{(2)^{k+1}}{(k+1)!} X(k) \tag{15}$$

where

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

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.

### 7.2 Short-term Hamiltonian

The above procedure can be made significantly more efficient if one guarantees that the parameter range is small, that is $|a_{\mathbf{s}}| \leq \epsilon$ for small enough $\epsilon > 0$. This is the case in short-term Hamiltonian simulation or a Hamiltonian with "weak interactions". In the short-term simulation, the unitary is of the form $U = e^{i\Delta t H}$, where $\Delta t$ 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 2.** *If* $\max_{\mathbf{s} \in \mathcal{S}} |a_{\mathbf{s}}| \leq \frac{\epsilon}{p(1+2\epsilon)}$, *then the partial derivative can be approximated with* $\epsilon$ *additive error via* $\frac{\partial \mathcal{L}}{\partial a_{\mathbf{s}}} \approx D_{\mathbf{s}}$.

### 7.3 Extension to Polynomial Size DLA

Next, we extend our results to a more general class of PQCs for which DLA has poly$(d)$ dimensionality.

**Definition 2.** *For 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 3** (Dynamical Lie Algebra)**.** *Given a PQC with generators $\mathcal{G}$ as in Definition 2, the Dynamical Lie Algebra (DLA) $\mathsf{g}_A$ is the subalgebra of $\mathfrak{su}(2^n)$ spanned by the repeated nested commutators of the elements in $\mathcal{G}$, i.e.,*

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

*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 $\mathsf{g}_A$ has polynomial size/dimensionality if $\dim(\mathsf{g}_A) = \mathsf{poly}(n)$ 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 [38]). The complexity of this approach in general is $O(2^n)$. However, this is not an issue if the dimension of the DLA to be constructed is promised to by $\mathsf{poly}(n)$.

**Theorem 7.** *Consider an ansatz of the form $U(\overrightarrow{a}) = \exp\{iA(\overrightarrow{a})\}$, where $A(\overrightarrow{a}) = \sum_i a_i G_i$ with $G_i$ being traceless Hermitian operators. Let $\mathsf{g}_A$ be the DLA of the ansatz with a basis $\{E_1, \cdots, E_{d_{\mathsf{g}_A}}\}$ and $T_A$ be the matrix representation of the linear transformation $\mathsf{ad}_A(\cdot) = [A, \cdot]$ in the basis of $\mathsf{g}_A$. Then, $\nabla\mathcal{L}(\overrightarrow{a}) = \overrightarrow{R} f(T_A)$, where $f(z) := (1 - e^{-z})/z$ for $z \neq 0$ and $f(0) := 1$, and $\overrightarrow{R} = (R_1, \cdots, R_{d_{\mathsf{g}_A}})$ with $R_l = i\,\mathrm{tr}\{O[\rho^{out}, E_l]\}$*

*Moreover, if $\mathsf{g}_A$ has $\mathsf{poly}(n)$ dimensionality, there is an algorithm (Algorithm 2) that estimates $\nabla\mathcal{L}(\overrightarrow{a})$ with $\mathsf{poly}(n)$ copies of $\rho^{out}$ and additional $\mathsf{poly}(n)$ classical time.*

*Proof.* The proof of the theorem follows from a similar argument as in Theorem 2 and 3. First note that from (13)

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

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

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

Note that $\mathsf{ad}_A$ maps $\mathsf{g}_A$ unto itself as its *commutator ideal*. That is

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

where $V_j, W_j \in \mathsf{g}_A$. Since $\mathsf{g}_A$ is closed under the Lie bracket, then $[V_j, W_j] \in \mathsf{g}_A$. Hence, $\mathsf{ad}_A(G) \in \mathsf{g}_A$ and it equals to a finite sum of the basis elements $E_l$ of $\mathsf{g}_A$ 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, $\mathsf{ad}_A$ can be viewed as a linear transformation (matrix) $T_A(\overrightarrow{\alpha})$ in the $E_l$ basis. This matrix is $d_{\mathsf{g}_A} \times d_{\mathsf{g}_A}$. With that taken into account, the partial derivative can be written as

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

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

$$B(\overrightarrow{\alpha}) = f(T_A(\overrightarrow{\alpha})),$$

where $f(z) := (1 - e^{-z})/z$ for $z \neq 0$ and $f(0) := 1$. This matrix can be computed classically in $\mathsf{poly}(d_{\mathsf{g}_A}) = \mathsf{poly}(n)$ time. Note that the above summation is exactly the Taylor expansion of $f(T_A(\overrightarrow{\alpha}))$, and hence we have

$$\frac{\partial \mathcal{L}}{\partial a_j} = \sum_{l=1}^{d_{\mathsf{g}_A}} B_{(j,l)}(\overrightarrow{\alpha}) i \operatorname{tr}\Big\{ O[E_l, \rho^{out}]\Big\},$$

where $B_{(j,l)}(\overrightarrow{\alpha})$ is the $(j,l)$ entry of the matrix $B(\overrightarrow{\alpha})$. To compute the gradient from this vector representation, one first needs to estimate all $R_l := i \operatorname{tr}\{O[\rho^{out}, E_l]\}$ and compute the following

$$\nabla \mathcal{L} = \overrightarrow{R} B(\overrightarrow{\alpha}).$$

With that the proof is complete. □

The following algorithm summarizes this procedure to estimate the gradient.

**Algorithm 2** DLA Gradient Estimation

**Input:** DLA generators $E_1, \cdots, E_{d_{\mathsf{g}_A}}$,

1: Estimate the expectation value of the tests $R_l := i \operatorname{tr}\{O[E_l, \rho^{out}]\}$ for $l \in [d_{\mathsf{g}_A}]$.
2: Compute the $d_{\mathsf{g}} \times d_{\mathsf{g}_A}$ matrix $B(\overrightarrow{\alpha}) = f(T_A(\overrightarrow{\alpha}))$, with $f(z) := (1 - e^{-z})/z$ for $z \neq 0$ and $f(0) := 1$.
3: **Return** $\widehat{\nabla \mathcal{L}} = \overrightarrow{R} B(\overrightarrow{\alpha})$.

Extension to PQC with mixture of different DLA is also possible. For example, consider a PQC of the form $U(\overrightarrow{a}) = \exp\{iA_1(\overrightarrow{a})\} \exp\{iA_2(\overrightarrow{a})\}$ where $A_1(\overrightarrow{a}) = \sum_i a_i G_i$ and $A_2(\overrightarrow{a}) = \sum_j a'_j G'_j$ with $\{G_i\}$ and $\{G'_j\}$ being the generators of two different DLA's $\mathsf{g}_{A_1}$ and $\mathsf{g}_{A_2}$. If both $\mathsf{g}_{A_1}$ and $\mathsf{g}_{A_2}$ have polynomial size/dimensionality, then one can compute the gradient with polynomial number of measurement tests and additional polynomial classical time. The proof follows from a similar argument as in Theorem 7.

## 8 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.

Our results are steps toward a unified framework for efficiently estimating the gradient of an arbitrary parameterized circuit without making changes to its unitary. One limitation of this work is when the Hamiltonian has exponentially many Pauli terms. In that case, we can only approximate the gradient by truncation of the nested summations in Theorem 2 to a fixed number of terms. However, this will be a biased approximation.

As future work, one can extend the proposed framework to the estimation of higher-order derivatives 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.

## Acknowledgments

This work is partially supported by the NSF Center for Science of Information (CSoI) Grant CCF-0939370, and also by NSF Grants CCF-2006440 and and CCF-2211423.

## References

[1] Mohsen Heidari, Masih Mozakka, and Wojciech Szpankowski. "Hadamard test is sufficient for efficient quantum gradient estimation with lie algebraic symmetries". In The Thirty-ninth Annual Conference on Neural Information Processing Systems. (2025). url: https://openreview.net/forum?id=PdUdzvYzoH.

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

[3] 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**, 625–644 (2021).

[4] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. "A quantum approximate optimization algorithm" (2014). arXiv:1411.4028.

[5] Tyson Jones, Suguru Endo, Sam McArdle, Xiao Yuan, and Simon C. Benjamin. "Variational quantum algorithms for discovering hamiltonian spectra". Physical Review A **99**, 062304 (2019).

[6] 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**, 113010 (2021).

[7] 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** (2019).

[8] 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**, 052402 (2021).

[9] Edward Farhi and Hartmut Neven. "Classification with quantum neural networks on near term processors" (2018). arXiv:1802.06002.

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

[11] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii. "Quantum circuit learning". Physical Review A **98**, 032309 (2018).

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

[13] 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**, 209–212 (2019).

[14] Mohsen Heidari, Arun Padakandla, and Wojciech Szpankowski. "A theoretical framework for learning from quantum data". In 2021 IEEE International Symposium on Information Theory (ISIT). Pages 1469–1474. IEEE (2021).

[15] Mohsen Heidari and Wojciech Szpankowski. "New bounds on quantum sample complexity of measurement classes". In 2024 IEEE International Symposium on Information Theory (ISIT). Pages 1515–1520. IEEE (2024).

[16] Mohsen Heidari and Wojciech Szpankowski. "Learning k-qubit quantum operators via pauli decomposition". In Francisco Ruiz, Jennifer Dy, and Jan-Willem van de Meent, editors, Proceedings of The 26th International Conference on Artificial Intelligence and Statistics. Volume 206 of Proceedings of Machine Learning Research. PMLR (2023). url: https://proceedings.mlr.press/v206/heidari23a.

[17] Hsin-Yuan Huang, Michael Broughton, Masoud Mohseni, Ryan Babbush, Sergio Boixo, Hartmut Neven, and Jarrod R. McClean. "Power of data in quantum machine learning". Nature Communications**12** (2021).

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

[19] 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).

[20] Maria Schuld, Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran. "Evaluating analytic gradients on quantum hardware". Phys. Rev. A 99, 032331 (2019) (2018). arXiv:1811.11184.

[21] Mohsen Heidari, Ananth Y. Grama, and Wojciech Szpankowski. "Toward physically realizable quantum neural networks". Association for the Advancement of Articial Intelligence (AAAI) (2022). url: https://cdn.aaai.org/ojs/20647/20647-13-24660-1-2-20220628.pdf.

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

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

[24] Mohammad Aamir Sohail, Mohsen Heidari Khoozani, and S. Sandeep Pradhan. "Quantum natural stochastic pairwise coordinate descent" (2024). arXiv:2407.13858.
[25] Leonardo Banchi and Gavin E. Crooks. "Measuring analytic gradients of general quantum evolution with the stochastic parameter shift rule". Quantum **5**, 386 (2021).
[26] David Wierichs, Josh Izaac, Cody Wang, and Cedric Yen-Yu Lin. "General parameter-shift rules for quantum gradients". Quantum **6**, 677 (2022).
[27] Dirk Oliver Theis. ""proper" shift rules for derivatives of perturbed-parametric quantum evolutions". Quantum **7**, 1052 (2023).
[28] 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** (2021).
[29] 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.
[30] Jarrod R. McClean, Sergio Boixo, Vadim N. Smelyanskiy, Ryan Babbush, and Hartmut Neven. "Barren plateaus in quantum neural network training landscapes". Nature Communications**9** (2018).
[31] Matthew L. Goh, Martin Larocca, Lukasz Cincio, M. Cerezo, and Frédéric Sauvage. "Lie-algebraic classical simulations for quantum computing" (2023). quant-ph:2308.01432.
[32] 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) (2020). arXiv:2002.08953.
[33] A. R. Calderbank, E. M Rains, P. W. Shor, and N. J. A. Sloane. "Quantum error correction via codes over gf(4)" (1996). arXiv:quant-ph/9608006.
[34] Daniel Gottesman. "Stabilizer codes and quantum error correction" (1997). arXiv:http://arxiv.org/abs/quant-ph/9705052v1.
[35] Wulf Rossmann. "Lie groups: An introduction through linear groups". Oxford University PressOxford. (2002).
[36] S G Schirmer, I C H Pullen, and A I Solomon. "Identification of dynamical lie algebras for finite-level quantum control systems". Journal of Physics A: Mathematical and General **35**, 2327–2340 (2002).
[37] 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).
[38] 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).
[39] Amira Abbas, Robbie King, Hsin-Yuan Huang, William J. Huggins, Ramis Movassagh, Dar Gilboa, and Jarrod McClean. "On quantum backpropagation, information reuse, and cheating measurement collapse". In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, Advances in Neural Information Processing Systems. Volume 36, pages 44792–44819. Curran Associates, Inc. (2023). url: https://proceedings.neurips.cc/paper_files/paper/2023/file/8c3caae2f725c8e2a55ecd600563d172-Paper-Conference.pdf.
[40] Javier Gil Vidal and Dirk Oliver Theis. "Calculus on parameterized quantum circuits" (2018). arXiv:1812.06323.

[41] Scott Aaronson. "Shadow tomography of quantum states". In Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing. Page 325–338. STOC 2018New York, NY, USA (2018). Association for Computing Machinery.

[42] Dantong Li, Dikshant Dulal, Mykhailo Ohorodnikov, Hanrui Wang, and Yongshan Ding. "Efficient quantum gradient and higher-order derivative estimation via generalized hadamard test" (2024). quant-ph:2408.05406.

[43] Sitan Chen, Weiyuan Gong, and Qi Ye. "Optimal tradeoffs for estimating pauli observables". In 2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS). Pages 1086–1105. IEEE (2024).

[44] Robbie King, David Gosset, Robin Kothari, and Ryan Babbush. "Triply efficient shadow tomography". PRX Quantum **6**, 010336 (2025).

[45] 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**, 183–190 (1976).

[46] 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**, 042401 (2022).

[47] 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).

[48] 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**, 043212 (2021).

## A Derivation of The Gradient for a Single Qubit PQC

Consider a general single-qubit unitary of the form

$$U(\overrightarrow{a}) = \exp\Big\{i(a_1\sigma^1 + a_2\sigma^2 + a_3\sigma^3)\Big\}.$$

Let $O$ be a generic observable and consider the associated loss $\mathcal{L}(\overrightarrow{a})$ as in (1). As an illustrative example, we examine the partial derivative of $\mathcal{L}$ with respect to $a_1$, evaluated at the point $a_1 = 0$ and $a_3 = 0$. In the following we compute this derivative using two approaches: first, by applying Theorem 2, and second,through direct calculation.

**Closed-form expression based on Theorem 2.** In the context of Theorem 2, let $D_j$ be the result of the Hadamard test with Pauli $\sigma^j$, where $j = 1, 2, 3$. Then, (7) in Theorem 2 simplifies to the following:

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

where we used the fact that only terms with $\mathbf{s}_j = 2$ are surviving. Because $a_1 = a_3 = 0$. Note that for even $j$ we have $\underbrace{2 \oplus \cdots \oplus 2}_{j-1 \text{ times}} \oplus 1 = 3$, and for odd $j$ it is equal to $0 \oplus 1 = 1$. Therefore,

$$\delta(2, (\underbrace{2 \oplus \cdots \oplus 2}_{j-1 \text{ times}} \oplus 1)) = \begin{cases} \delta(2,3) = 1 & \text{even } k \\ \delta(2,1) = -1 & \text{odd } k \end{cases}$$

where we used (6). Plugging it in the first equation, we have

$$\prod_{j=1}^{k} a_2 \left( i^{\delta(2,(\overbrace{2 \oplus \cdots \oplus 2}^{k-1 \text{ times}} \oplus 1))} \right) = a_2^k (-i) \times i \times (-i) \times \cdots = \begin{cases} (a_2 i)^k (-1)^{k/2} & \text{even } k \\ (a_2 i)^k (-1)^{(k+1)/2} & \text{odd } k. \end{cases}$$

As a result, the partial derivative simplifies to

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

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 $\sigma^3$ and not appear in the ansatz expression. One can verify that this is indeed equal to the analytic gradient of this ansatz.

**Direct derivation.** Note that from (11) 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 $\rho^{out}$ is the ansatz output. Note that the single qubit ansatz can also be written as

$$U(\overrightarrow{a}) = I\cos\theta + i\left(\sum_{s\in\{0,1,2,3\}} \hat{a}_s\sigma^s\right)\sin\theta, \tag{16}$$

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:

$$\begin{aligned}\frac{\partial U(\overrightarrow{a})}{\partial a_s} = (-\frac{a_s}{\theta}\sin\theta)I + i\left(\sum_{s'\neq s}\frac{-a_s a_{s'}}{\theta^3}\sigma^{s'} + \frac{\theta^2 - a_s^2}{\theta^3}\sigma^s\right)\sin\theta \\ + i\left(\frac{a_s\cos\theta}{\theta}\sum_{s'}\hat{a}_{s'}\sigma^{s'}\right).\end{aligned} \tag{17}$$

Therefore,

$$\frac{\partial U(\overrightarrow{a})}{\partial a_1}\bigg|_{a_1=a_3=0} = i\sigma^1\frac{\sin a_2}{a_2}.$$

Using (16) to find $U^\dagger$, we have that

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

Which when plugged into the derivative expression gives

$$\frac{\partial L}{\partial a_1}\bigg|_{a_1=0} = \frac{\sin a_2}{a_2}\left(\cos a_2 D_1 + \sin a_2 D_3\right) = \frac{\sin 2a_2}{2a_2}D_1 + \frac{(1-\cos 2a_2)}{2a_2}D_3.$$

This is identical to the expression based on Theorem 2.

## B Summary of Related Works

**Parameter Shift Rule.** There have been several approaches to estimating the gradient [9, 11, 19, 21, 18, 20, 22, 23]. 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 [18]. First-order methods (e.g., parameter shift rule) directly calculate the partial derivatives [20]. When the ansatz is of the form $U(\overrightarrow{a}) = \prod_{j=1}^{p} e^{-ia_j\sigma^{\mathbf{s}_j}}$, 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 $j$th canonical vector, that is $e_{j,j} = 1$ and $e_{j,r} = 0$ for all $r \neq j$.

**Stochastic PSR:** Stochastic parameter shift rule was proposed in [25] 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 $C_\pm(\theta, t) := \mathrm{tr}\Big\{OV_\pm(\theta, t)\rho V_\pm^\dagger(\theta, t)\Big\}$, 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 [26]. 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 [27] 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 [26, 40]. 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.

**Trotterization.** In this approach the non-product unitary is approximated via the Suzuki-Trotter transformation [45] which states that for any operators $A_1, A_2, \cdots, A_k$, that do not necessarily commute with each other, the following holds

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

The Trotter formula has been used in literature to derive approximations for generic PQCs [46, 47, 48]. However, implementing the above approximation may lead to a high gate complexity and is not preferable in scenarios where direct implementation is available.

## C Classical Shadow Tomography

For completeness, in this section we briefly describe the classical shadow tomography procedure. For more details see [32]. Classical shadow tomography is a technique used in quantum computing to efficiently learn properties of a quantum state using only a few measurements. It was introduced to extract useful information from quantum states without requiring a full quantum state tomography, which is costly in terms of the number of measurements and computational resources.

More precisely, let $O_j, j \in [M]$ be a set of observables. The goal is to estimate the expectation value of these observable for measuring an unknown state $\rho$ in a Hilbert space $\mathcal{H}$. For that, several copies of $\rho$ are provided. CST is a procedure with minimal sample (copy) complexity.

**Theorem 8** ([32])**.** *Suppose the observables $O_j, j \in M$ are traceless, then the expectation values $\mathrm{tr}\{O_j\rho\}, j \in [M]$ can be approximated up to an additive error $\epsilon$ with probability $(1-\delta)$ given*

$$O\Big(\frac{1}{\epsilon^2} \log\frac{M}{\delta} \max_j \|O\|^2_{shadow}\Big)$$

*copies of $\rho$.*

The shadow norm is a measure that resembles the variance in the worst case state.

In what follows we describe the steps in this procedure for a generic state $\rho$ in a Hilbert space $\mathcal{H}$..

First, generate a unitary operator $U$ randomly from a class of choices $\mathcal{U}$ to be determined. Apply $U$ on the input state resulting in the state $U^\dagger \rho U$. Measure the resulted state in the canonical basis $|j\rangle, j \in [\dim_{\mathcal{H}}]$. From Born's rule the probability of getting the output $j$ is $p_j = \langle j|U^\dagger \rho U|j\rangle$. Given an outcome $j$, define $\omega_j = U|j\rangle\langle j|U^\dagger$. The expectation $\mathbb{E}_{\sim(J,U)}[\omega_J]$ over the measurement randomness $(p_j)$ and the choice of $U$ equals to $\mathcal{M}[\rho]$, where $\mathcal{M}$ is a mapping defined as

$$\mathcal{M}[O] := \mathbb{E}_U\Big[ \sum_{j\in[\dim_{\mathcal{H}}]} \langle j|U^\dagger O U|j\rangle \; U|j\rangle\langle j|U^\dagger \Big], \tag{18}$$

for any operator $O$ on $\mathcal{H}$. Observe that $\mathcal{M}$ is a linear mapping on $\mathcal{B}(\mathcal{H})$ and hence has an inverse denoted by $\mathcal{M}^{-1}$. We note that $\mathcal{M}^{-1}$ is the shadow channel $\mathcal{M}^{-1}$ introduced in [32]. We apply $\mathcal{M}^{-1}$ on $\omega_j$ resulting in the so called shadow

$$\hat{\rho} := \mathcal{M}^{-1}[U|j\rangle\langle j|U^\dagger]. \tag{19}$$

Note that $\hat{\rho}$ is a classical matrix and hence can be copied several times. Moreover, $\hat{\rho}$ is not a valid density operator as it is not necessarily a positive semi-definite matrix. However, it is an unbiased estimate of the original state.

When $\mathcal{U}$ is tomographically complete, the classical shadow $\hat{\rho}$ is unbiased, that is $\mathbb{E}_{U,J}[\hat{\rho}] = \rho$.

**Definition 4.** *The shadow norm of any operator $O$ on $\mathcal{H}$ is*

$$\|O\|_{\text{shadow}} := \max_{\sigma \in \mathcal{D}[H]} \Big( \sum_{j \in [dim_{\mathcal{H}}]} \langle j|U^\dagger \sigma U|j\rangle \;\; \langle j|U\mathcal{M}^{-1}[O]U^\dagger|j\rangle^2 \Big)^{1/2}.$$

## C.1 CST with Pauli Measurements

**Shadow tomography with Pauli measurements.** Suppose the observable $O_j$ act non trivially on at most $k$ qubits. For that $V$ in CST is the tensor product of randomly chosen Pauli operators:

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

where each $V_j$ is chosen randomly and uniformly from the Clifford group $CL(2)$. In this case, $\mathcal{U} = CL(2)^{\otimes d}$. Moreover, the shadow matrix is computed as

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

In that case, the shadow norm is bounded by the locality of the observables as

$$\|O_j\|_{shadow} \leq 2^k \|O_j\|_\infty$$

As a result the sample complexity of CST is given by

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

The CST algorithm runs in $\tilde{O}\Big(2^{\Theta(k)} m \log m\Big)$ classical time.

## C.2 CST with Clifford Measurements

The *Clifford group* is a set of unitary operations that map Pauli operators to other Pauli operators under conjugation. For a system of $d$ qubits, the Clifford group $\mathcal{C}_d$ consists of unitaries $U$ such that for any Pauli string $P$:

$$UPU^\dagger = P',$$

where $P'$ is another Pauli operator.

Clifford circuits are particularly useful because they can be efficiently simulated classically, and their structure allows for easy manipulation of Pauli observables, making them useful for shadow tomography. In that case, $V$ is chosen randomly from the Clifford group. The shadow norm is bounded as

$$\|O_j\|_{shadow} \leq \sqrt{3 \operatorname{tr}\{O^2\}},$$

when $\mathrm{tr}\{O^2\} < \infty$. The shadow matrix is given by

$$\hat{\rho} = (2^d + 1)V^\dagger \left|\hat{b}\right\rangle\!\left\langle\hat{b}\right| V - I.$$

As a result the sample complexity of CST is bounded as

$$n = O\Big(\frac{1}{\epsilon^2} \log m \max_j \mathrm{tr}\Big\{O_j^2\Big\}\Big),$$

and the CST algorithm runs $\tilde{O}\Big(2^{\Theta(d)} m \log m\Big)$ classical time.

# D Technical Lemmas and Proofs

In this section we present some of the technical lemmas and their proofs.

## D.1 Proof of Lemma 5

**Lemma 5.** *The partial derivative observables $H_{\mathbf{s}}$ satisfy $\mathrm{tr}\{H_{\mathbf{s}}^2\} \leq 4\,\mathrm{tr}\{O^2\}$ for any $\mathbf{s} \in \{0,1,2,3\}^n$.*

*Proof.* For the proof of the second claim note that

$$H_{\mathbf{s}}^2 = R_{\mathbf{s},+}^\dagger O^2 R_{\mathbf{s},+} + R_{\mathbf{s},-}^\dagger O^2 R_{\mathbf{s},-} - R_{\mathbf{s},+}^\dagger O R_{\mathbf{s}}(\pi) O R_{\mathbf{s},-} - R_{\mathbf{s},-}^\dagger O R_{\mathbf{s}}(-\pi) O R_{\mathbf{s},+}$$

Taking the trace of this operator gives

$$\mathrm{tr}\Big\{H_{\mathbf{s}}^2\Big\} = 2\,\mathrm{tr}\Big\{O^2\Big\} - \mathrm{tr}\{O R_{\mathbf{s}}(\pi) O R_{\mathbf{s}}(-\pi)\} - \mathrm{tr}\{O R_{\mathbf{s}}(-\pi) O R_{\mathbf{s}}(\pi)\}.$$

Note that $R_{\mathbf{s}}(\pi) = -i\sigma^{\mathbf{s}}$ and $R_{\mathbf{s}}(-\pi) = i\sigma^{\mathbf{s}}$. To see this, note that $\sigma^{\mathbf{s}} = \sum_b \lambda_b |b\rangle\!\langle b|$, where $|b\rangle$ is an eigenstate of $\sigma^{\mathbf{s}}$ and $\lambda_b \in \{-1, 1\}$ is the corresponding eigenvalue. As a result,

$$R_{\mathbf{s}}(\pi) = e^{-\frac{\pi}{2}\sigma^{\mathbf{s}}} = \sum_b e^{-i\frac{\pi}{2}\lambda_b} |b\rangle\!\langle b| = \sum_b (i)^{-\lambda_b} |b\rangle\!\langle b| = -i\sigma^{\mathbf{s}}$$

where we used the fact that $i^x = xi$ for $x = \pm 1$. Similarly, $R_{\mathbf{s}}(-\pi) = i\sigma^{\mathbf{s}}$. Therefore,

$$\mathrm{tr}\{O R_{\mathbf{s}}(\pi) O R_{\mathbf{s}}(-\pi)\} = \mathrm{tr}\{O R_{\mathbf{s}}(-\pi) O R_{\mathbf{s}}(\pi)\} = \mathrm{tr}\{O\sigma^{\mathbf{s}} O \sigma^{\mathbf{s}}\}.$$

The observable $O$ has a Pauli decomposition of the form $O = \sum_{\mathbf{t}\in\{0,1,2,3\}^d} a_{\mathbf{t}}\sigma^{\mathbf{t}}$ with $a_{\mathbf{t}} \in \mathbb{R}$. Then the above quantity is equal to

$$\mathrm{tr}\{O\sigma^{\mathbf{s}} O \sigma^{\mathbf{s}}\} = \sum_{\mathbf{t}} \sum_{\mathbf{r}} a_{\mathbf{t}} a_{\mathbf{r}}\, \mathrm{tr}\Big\{\sigma^{\mathbf{t}}\sigma^{\mathbf{s}}\sigma^{\mathbf{r}}\sigma^{\mathbf{s}}\Big\}.$$

Note that the trace term in the summation is zero unless $\mathbf{t} = \mathbf{r}$. Also note that for any $i, j \in \{0,1,2,3\}$

$$\mathrm{tr}\Big\{\sigma^i\sigma^j\sigma^i\sigma^j\Big\} = \begin{cases} -2 & \text{if } i \neq j \neq 0 \\ 2 & \text{otherwise.} \end{cases}$$

Therefore, the above summation equals to

$$\mathrm{tr}\{O\sigma^{\mathbf{s}} O \sigma^{\mathbf{s}}\} = 2^n \sum_{\mathbf{t}} (a_{\mathbf{t}})^2 (-1)^{|\{i : t_i \neq s_i \neq 0\}|}.$$

On the other hand from a Parseval-type identity we know that $\mathrm{tr}\{O^2\} = 2^n \sum_{\mathbf{t}} (a_{\mathbf{t}})^2$. Putting everything together we have the expression for $\mathrm{tr}\{H_{\mathbf{s}}^2\}$:

$$\begin{aligned}\mathrm{tr}\Big\{H_{\mathbf{s}}^2\Big\} &= 2^{n+1} \sum_{\mathbf{t}} (a_{\mathbf{t}})^2 \Big(1 - (-1)^{|\{i : t_i \neq s_i \neq 0\}|}\Big) \\ &\leq 2^{n+2} \sum_{\mathbf{t}} (a_{\mathbf{t}})^2 \\ &= 4\,\mathrm{tr}\Big\{O^2\Big\}.\end{aligned}$$

where the last inequality holds because $1 - (-1)^x \leq 2$ for any non-negative integer $x$. □

## D.2 Proof of Lemma 6

**Lemma 6.** *Let $V \in \mathbb{C}^{p \times p}$ be the matrix from Theorem 3, and $f(z) := (1 - e^{-z})/z$ for $z \neq 0$ and $f(0) := 1$. Then,*

$$\|f(V)\| \leq \frac{e^{\|V\|} - 1}{\|V\|}.$$

*Moreover, if $|a_{\mathbf{s}}| \leq c$ for all $\mathbf{s} \in \mathcal{S}$, then*

$$\|f(V)\| \leq \frac{e^{2pc} - 1}{2pc}.$$

*Proof.* The power series expansion of $f$ around 0 is given by

$$f(z) = -\sum_{k=0}^{\infty} \frac{z^k}{(k+1)!}.$$

Hence, for every matrix $V \in \mathbb{C}^{p \times p}$,

$$f(V) = -\sum_{k=0}^{\infty} \frac{V^k}{(k+1)!}.$$

On the other hand, from the power series expansion of the matrix exponential $e^{tV}$, we have

$$\int_0^1 e^{tV}\, dt = \int_0^1 \sum_{k=0}^{\infty} \frac{t^k V^k}{k!}\, dt = \sum_{k=0}^{\infty} \frac{V^k}{k!} \int_0^1 t^k\, dt = \sum_{k=0}^{\infty} \frac{V^k}{(k+1)!}.$$

Thus

$$f(V) = -\int_0^1 e^{tV}\, dt.$$

Taking operator norms and using the triangle inequality,

$$\|f(V)\| \leq \int_0^1 \|e^{tV}\|\, dt.$$

Moreover, by the power-series expansion and submultiplicativity of the operator norm,

$$\|e^{tV}\| \leq \sum_{k=0}^{\infty} \frac{\|tV\|^k}{k!} = e^{t\|V\|}.$$

Hence

$$\|f(V)\| \leq \int_0^1 e^{t\|V\|}\, dt = \begin{cases} \dfrac{e^{\|V\|} - 1}{\|V\|}, & \|V\| > 0, \\ 1, & \|V\| = 0. \end{cases}$$

It remains to bound $\|V\|$. Fix a column index $j$. Since the map $\mathbf{s} \mapsto \mathbf{s} \oplus \mathbf{s}_j$ is a permutation of $\mathcal{S}$, each coordinate of $\mathbf{v}_j$ has magnitude $2|a_{\mathbf{s}}|$ for exactly one $\mathbf{s} \in \mathcal{S}$. Therefore,

$$\sum_{r=1}^{p} |V_{rj}| = 2 \sum_{\mathbf{s} \in \mathcal{S}} |a_{\mathbf{s}}|.$$

This implies that the induced $\ell_1$-norm of $V$ equals $\|V\|_{\ell_1} = 2 \sum_{\mathbf{s} \in \mathcal{S}} |a_{\mathbf{s}}|$.

The same argument applied row-wise shows

$$\|V\|_{\ell_\infty} = \max_{j} \sum_{r=1}^{p} |V_{rj}| = \max_{r} \sum_{j=1}^{p} |V_{rj}| = 2 \sum_{\mathbf{s} \in \mathcal{S}} |a_{\mathbf{s}}|.$$

Thus, from the identity $\|V\| \leq \sqrt{\|V\|_{\ell_1} \|V\|_{\ell_\infty}}$, we have

$$\|V\| \leq 2 \sum_{\mathbf{s} \in \mathcal{S}} |a_{\mathbf{s}}|.$$

Therefore, assuming $|a_{\mathbf{s}}| \leq c$ for all $\mathbf{s} \in \mathcal{S}$, we have $\|V\| \leq 2pc$. Substituting this into the previous estimate gives the desired bound. □