Gradients parallelism and variance of quantum estimates

Francesco Preti,[1,2,*] Michael Schilling,[2,3] József Zsolt Bernád,[2,4] Tommaso Calarco,[2,3,5] F. A. Cárdenas-López,[2] and Felix Motzoi[2,3]

[1] *Jülich Supercomputing Center, Helmholtz AI*
[2] *Forschungszentrum Jülich, Institute of Quantum Control (PGI-8), D-52425 Jülich, Germany*
[3] *Institute for Theoretical Physics, University of Cologne, Zülpicher Straße 77, 50937 Cologne, Germany*
[4] *HUN-REN Wigner Research Centre for Physics, Budapest, Hungary*
[5] *Dipartimento di Fisica e Astronomia, Università di Bologna, 40127 Bologna, Italy*

(Dated: January 23, 2026)

Computation of observables and their gradients on near-term quantum hardware is a central aspect of any quantum algorithm. In this work, we first review standard approaches to the estimation of observables with and without quantum amplitude estimation for both cost functions and gradients, discuss sampling problems, and analyze variance propagation on quantum circuits with and without Linear Combination of Unitaries (LCU). Afterwards, we systematically analyze the standard approaches to gradient computation with LCU circuits. Finally, we develop a LCU gradient framework for the most general gradients based on n-qubit gates and for time-dependent quantum control gradient, analyze the convergence behaviour of the circuit estimators, and provide detailed circuit representations of both for near-term and fault-tolerant hardware.

INTRODUCTION

The fast developing field of quantum computation requires careful analysis of the sampling complexity of quantum algorithms [1]. NISQ quantum algorithms [2], in particular, can be used to encode specific optimization problems [3] that depend on classical parameters . In this context, the estimation of fast, reliable gradients of the output of quantum algorithms with respect to classical parameters has been studied extensively [4–6]. Quantum algorithms have a vast range of applications.

A quantum algorithm is usually implemented as a family of one or multiple quantum circuits [7], which in turn represent physical experiments on one or more of the available quantum computing platforms, such as superconducting quantum circuits [8, 9], trapped-ions [10] or Rydberg atoms [11]. In these models, a quantum state is first prepared, evolves under the action of unitary operations and is then measured. Qubits can be measured between unitary operations [12] (and subsequently reset if needed), so that more complex maps involving mixed states can also be implemented in quantum algorithms. By executing a quantum algorithm multiple times, we can collect data about the possible different outcomes. For example, in the case of variational quantum algorithms [13–15], the statistics of the measurement process is used to estimate a cost function, which is then optimized with respect to variational parameters using classical optimization methods.

One of the central aspects is the scaling of the number of measurements needed to estimate key circuit observables with precision ϵ. Usually, mean values of arbitrary observables are estimated by sampling from multiple quantum circuits, each one representing an element of an operator basis – e.g., the Pauli basis. The mean values of the single elements of the operator basis can be evaluated using multiple copies of the same circuit. In the most straightforward implementation the scaling is linear in the number of copies L, i.e., $O(L/\epsilon^2)$.

In shadow tomography models [16, 17] the scaling can be dramatically improved to reach $O(\log(L)/\epsilon^2)$, whereas using amplitude amplification the scaling becomes sublinear, at the cost of having to implement the amplitude amplification and the Jordan algorithm routine [18, 19] $O(\sqrt{L}/\epsilon)$. Once the mean values have been estimated, they are summed together with appropriate coefficients. This further increases the variance linearly in the number of terms [20].

In this paper, we analyze some specific estimators of quantum cost functions. Most of the proposals in NISQ circuits assume the use of a linear combination of measurements for the estimation of observables [20–23], which we refer to as Standard Estimator (SE). Such estimator is also used in cases in which mixtures of classical and quantum expectation values need to be computed, using, e.g., quasi-Monte Carlo approaches. For example. QML requires the computation of averages over data sets [24–29]. Quantum control and optimization, on the other hand, for example in the context of so-called robust [30] or adaptive control/meta-optimization [31, 32], require to compute averages of cost functions over a certain parameter space [31–35], in order to obtain control pulses that are less sensitive to parameter variations. Another example of estimators that use both classical and quantum sampling are (stochastic) parameter-shift rules [4, 36], which are used to evaluate gradients of quantum cost functions sampled using variational quantum circuits. Finally, applications of quantum algorithms such as quantum computational fluid dynamics (QCFD) require the (quantum or classical) summation of several estimates from quantum circuits to encode, e.g., the dynamics of relevant partial differential equations on quantum hard-

* f.preti@fz-juelich.de

ware [37].

Estimation of quantum cost functions can be also performed by implementing linear combinations of unitary operations (LCU) [14, 38–42] on quantum hardware. The question is whether this implementation can be beneficial in specific contexts. In this work, we compare LCU-based estimators to the standard estimator for quantum observables, which uses a different circuit for each non-zero basis element of the observable, and determine the conditions in which the implementation of the former is detrimental or beneficial, i.e., where it provides us with a speed-up over the classical counterpart, limited to when combined with amplitude estimation. We show that the LCU estimator allows for a $\sqrt{L}$ speedup over the standard estimator even for near-term amplitude estimation algorithms, which is in accordance with [18, 43]. We explore analytical derivations that confirm partial results and extend their validity to different types of sampling problems.

Furthermore, we analyze the problem of estimating gradients of quantum cost functions, which has been considered for both variational NISQ circuits and control circuits [4, 26, 27, 36, 44–50], as well as more advanced fault-tolerant algorithms [51]. More specifically, we extend the LCU framework to gradients of multi-qubit gates [44] and quantum control problems [52, 53]. This broader LCU framework enables us to differentiate a vast class of quantum cost functions for variational and control circuits.

The paper is structured as follows. In Section I we discuss the basics of observable and gradient estimation on quantum circuits. Afterwards, in Section II C we introduce basic estimators of quantum cost functions that make or do not make use of LCU methods and discuss their properties in accordance to the current literature. In Section (III) we introduce and outline the properties of amplitude estimation methods and show how their use affect the scaling of previously introduced estimators. We also show how different LCU methods can benefit differently from amplitude estimation compared to the standard LCU procedure. As a paradigmatic example, we also test the estimators on a typical QML regression task. In Section IV we introduce the topic of gradient estimation for quantum circuits and review some of the basic methods thereof. We then extend LCU gradient circuits to multi-qubit, multi-parametric quantum gates and quantum control gradients and discuss their relevant scaling. We focus in particular on studying the convergence behaviour of LCU gradients of multi-parametric gates for specific cost functions, such as those discussed in Ref. [54].

I. PROBLEM STATEMENT

Sampling from one or multiple quantum circuits involves computing linear combinations of binary counts corresponding to different outputs. An example is given by QUBO problems [3], where a quantity, which is in a quadratic form with binary arguments, needs to be sampled from various quantum systems. In the case of variational quantum eigensolvers [15], the goal is to minimize the energy of a Hamiltonian given a certain input state. The n-qubit observable as a whole is usually not available directly but it can be represented as a linear combination of Hermitian matrices P_i, e.g., Pauli strings, which can potentially be measured using quantum circuits. We consider the observable:

$$\mathcal{O} = \sum_{i=1}^{L} a_i P_i, \quad a_i \in \mathbb{R}, \tag{1}$$

with $L < d^2 = 4^n$. We limit ourselves w.l.o.g. to the case in which $\mathcal{O}$ can be decomposed by considering only one element of the generalized Pauli group, whereby the expression above becomes:

$$\mathcal{O} = \sum_{i=1}^{L} a_i U_i Z_{\text{prod}} U_i^\dagger, \tag{2}$$

where $Z_{\text{prod}} = \bigotimes_{i=1}^{n} \sigma_z^{(i)}$ is the n-qubit σ_z operator and $U_i, i = 1, ..., L$ are appropriate unitary matrices – e.g., they map from Z_{prod} to other elements of the Pauli basis. The choice of Z_{prod} is arbitrary: another possibility is to map the operator to a single-qubit σ_z operator $Z_{\text{prod}}^{(i)} = \mathbb{I} \otimes ... \otimes \sigma_z^{(i)} \otimes \mathbb{I}$ via CNOT operations, but any generalized Pauli operator can be used in principle, as matrices mapping generalized Pauli operator to each other can all be generated using CNOT, Hadamard and Phase gates [39, 57]. The mean value of the observable $\mathcal{O}$ is computed with respect to a density matrix ρ, such that for the expected value of $\mathcal{O}$ we can write

$$\langle \mathcal{O} \rangle = \text{tr}\{\rho \mathcal{O}\} = \sum_{i=1}^{L} a_i \, \text{tr}\Big\{\rho U_i Z_{\text{prod}} U_i^\dagger\Big\}. \tag{3}$$

Using this representation, we can implement the unitaries $U_i, i = 1, ..., L$ on different quantum circuits and then measure the register of qubits in the computational basis. We assume that the circuit input state ρ undergoes a parametric evolution generated by a variational unitary. As a result, the expression:

$$\langle \mathcal{O}(\boldsymbol{\theta}) \rangle = \sum_{i=1}^{L} a_i \, \text{tr}\Big\{V(\boldsymbol{\theta}) \rho V^\dagger(\boldsymbol{\theta}) U_i Z_{\text{prod}} U_i^\dagger\Big\}, \tag{4}$$

encodes an energy minimization problem in up to L different quantum circuits using N real parameters, i.e., $\boldsymbol{\theta} \in \mathbb{R}^N$ and a n-qubit variational quantum circuit $U(\boldsymbol{\theta}) \in \text{V}(d)$, $d = 2^n$. Let us first assume that any two different Pauli strings considered in Eq. (1) commute. If this is the case, they can be estimated within the same circuit run, which significantly reduces the amount of

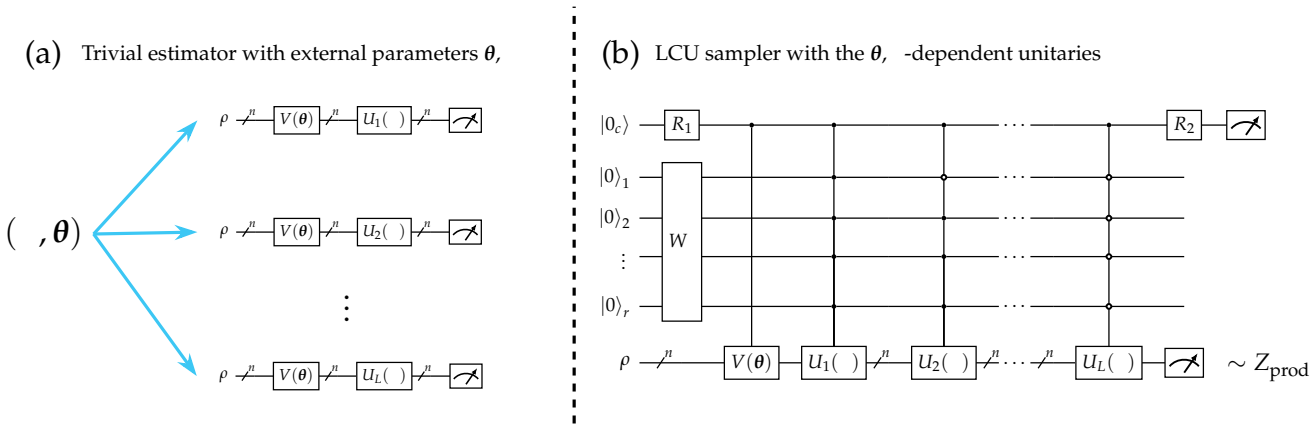


Figure 1. A representation of the two different approaches to observable sampling that are typical of variational quantum circuits: (a) summarizes the Standard Estimator (SE), which prepares L circuits with the same input density matrix and an arbitrary unitary operator $V(\ )$. The unitaries $U_1, ..., U_L$ (which are controlled by a parameter vector $\boldsymbol{\lambda}$) prepare, e.g., the different elements of an observable basis or a collection of L non-commuting operators. (b) summarizes the LCU sampler/estimator, which performs the same kind of estimation, but renormalized between, e.g., $I = (1, \ 1)$. The coefficients of the linear combination of L estimates are computed using classical methods (a) or loaded in the LCU register with $r = \lceil \log(L) \rceil$ qubits using the operator W_a that prepares the state $|a\rangle$ – see Eqs. (18) and (19) – using a suitable algorithm for state preparation [55, 56]. The unitary operations R_1 and R_2 control the type of cost function to estimate: $R_1 = H$, $R_2 = X$ estimates a cost function as in Eq. (3), whereas $R_1 = H$ and $R_2 = H$ estimates a cost function as in Eq. (89) – see also Ref. [38].

measurements needed – if all L of them commute, estimating their mean values scales as in $O(\lceil \log(L) \rceil / \epsilon^2)$ [18]. If they do not commute, up to L circuits need to be executed. Moreover, cost functions for variational circuits are also averaged over additional external parameters, where relevant parameters $\boldsymbol{\lambda} \sim P$ are sampled from a probability distribution P:

$$C(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{\lambda} \quad \boldsymbol{P}} \left[\langle \mathcal{O}(\boldsymbol{\theta}, \boldsymbol{\lambda}) \rangle \right]. \tag{5}$$

Therefore, there are two types of parameters: *meta-parameters*, denoted by $\boldsymbol{\lambda}$, which are sampled and averaged over – an example of this is given by Monte-Carlo sampling, where we want to average a value over a data set of parameters – and *variational parameters*, denoted by $\boldsymbol{\theta}$ which are generally used for numerical optimization in the context of variational algorithms.

Sampling using Eq. (4) is not the only option to evaluate the mean value of the observable. We can construct an estimator for $\langle \mathcal{O}(\boldsymbol{\theta}) \rangle$ by first constructing estimators for L different circuits. A different estimator can be constructed based on Linear Combination of Unitaries [38, 39] using a circuit that forks [58] the state evolution in different directions based on controlled operations. Our goal is to analyze the behaviour of such an estimator compared to the standard sequential procedure that uses L circuits. A similar approach can be defined also for gradients of quantum cost functions [26], where the properties of the gate Hamiltonians are exploited to estimate the gradient efficiently. As we discuss later in Section II, this procedure does not really bring any benefit in terms of sampling complexity: on the contrary, it delivers a $\log(L)$ increase in circuit complexity due to the multi-controlled operations. Therefore, in Section III we discuss how to use amplitude amplification to modify the sampling complexity of the estimators and reach an effective speedup using LCU methods.

In the context of variational algorithms, we are also interested not only in the (sampled) cost function $C(\boldsymbol{\theta})$, but also in its gradient. Gradients of quantum cost functions can be evaluated in terms of parameter-shift rules, i.e., trigonometric interpolation performed on the quantum cost function [4]. More specific parameter-shift rules can also be determined analytically for several classes of quantum gates [59] and are expressed as linear combinations of cost function values at different points:

$$\frac{\partial}{\partial \theta_i} C(\boldsymbol{\theta}) = \sum_{k=1}^{R} S_{ik} C(\boldsymbol{\theta} + \quad_{ik} \boldsymbol{e}_i), \tag{6}$$

where R is the number of shifts, S_{ik} and $\ _{ik}$ are suitable values that depend on the spectral properties of the gates implemented, and, depending on the type of gate, can be determined analytically [36, 59] or numerically [4]. Another possibility is given by Hadamard-like tests and LCU approaches [50], which offer a different solution to the gradient estimation problem. Yet a third approach to gradient estimation, which also finds use also in classical machine learning, is given by Monte Carlo sampling of the cost function gradient [60, 61], and also requires the evaluation of a linear combination of cost function samples. As gradient estimation is essentially a special case of estimation of the mean value of a quantum observable, we can make use of the general treatment of LCU vs.

standard methods to analyze the sampling complexity of different gradient estimators.

II. LINEAR COMBINATIONS OF ESTIMATES

A. Standard Estimator (SE): linear combinations of measurements

Our goal is to construct an estimator $\tilde{C}$ that, using the measurement outcomes collected from the quantum circuits, can successfully approximate C in Eq. (5). Let us consider a collection of circuits numbered 1 to L, each one implementing a unitary $V_1, ..., V_L$ that we use to perform a measurement of Z_{prod}. We refer to this estimator as the Standard Estimator (SE). This is a straightforward approach in most of the sampling problems in variational quantum circuits [15], so we use this name just for clarity. The principle is simple: we have different circuits that are initialized independently – see Fig. 1 (a). For each one of these circuits we prepare an identical initial state ρ. We first limit ourselves to the case in which the coefficients a_i are all non-negative (which we later generalize in Sec. II D). Formally, we consider first an estimator denoted by the pair $\left(M^{(i)}_{j_1 j_2 \ldots j_n}, \tilde{C}\right), i = 1, ..., L$ for a state ρ [62], where $M^{(i)}_{j_1 j_2 \ldots j_n}$ are projectors of the form:

$$M^{(i)}_{j_1 j_2 \ldots j_n} = U_i \left(\Pi_{j_1} \otimes \Pi_{j_2} \otimes \ldots \Pi_{j_n}\right) U_i^\dagger, \tag{7}$$

where $j_1, j_2, \ldots j_n \in \{0, 1\}$ and

$$\Pi_0 = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}, \quad \Pi_1 = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}. \tag{8}$$

This allows us to estimate $m_i = \text{tr}\left\{\rho U_i Z_{\text{prod}} U_i^\dagger\right\}$ for a given ρ and thus

$$\langle \mathcal{O} \rangle_\rho = \sum_{i=1}^{L} a_i \sum_{j_1, j_2, \ldots, j_n = 0,1} (\quad 1)^{\sum_{l=1}^{n} j_l} \text{tr}\left\{\rho M^{(i)}_{j_1 j_2 \ldots j_n}\right\}, \tag{9}$$

where a_i are the Pauli basis components of the observable given in Eq. (1). Each outcome of a circuit measurement corresponds to a binary string $x^{(i)}_{k_i)} \in 1, ..., 2^n$ and is weighted with a coefficient +1 or -1, depending on its binary Hamming weight

$$b(x) = \sum_{l=0}^{n} {}^{1} j_l(x), \tag{10}$$

where $x = \sum_l j_l(x) 2^l$ and $j_l(x) \in \{0, 1\}$ are the binary digits of x. Finally, $\tilde{C}$ is the map from the measurement data set to the real line. The SE estimator map is given by:

$$\tilde{C}_{\text{SE}} = \sum_{i=1}^{L} \frac{a_i}{n_s^{(i)}} \sum_{k_i=1}^{n_s{}^{i)}} (\quad 1)^{b(x^{i)}_{k_i})}. \tag{11}$$

We consider here the estimation of the expected values of L Pauli strings $P_1, ..., P_L$. The mean value of a Pauli string for a quantum state ρ has the following variance:

$$\sigma^2_{P_i} = \langle P_i^2 \rangle \quad \langle P_i \rangle^2, \tag{12}$$

and due to $P_i^2 = \mathbb{I}$ for any Pauli string, we have:

$$\text{Var}(P_i) = 1 \quad m_i^2, \tag{13}$$

where $m_i = \text{Tr}\{\rho P_i\} \in [\quad 1, 1]$. Eq. (13) can be considered as the variance of a Rademacher variable, which can be transformed into a (binary) Bernoulli variable with mean $p_i = \frac{1}{2}(m_i + 1)$ and variance $\text{Var}(P_i) = 4p_i(1 \quad p_i)$. Eq. (13) can be also seen as the variance of a projective measurement Π_i, with $p_i = \text{Tr}\{\Pi_i \rho\} \in [0, 1]$. Any SE draws measurement values $x^{(j)}_{k_1)}, ..., x^{(j)}_{k_L)}, 1 \leq j \leq n_s^{(i)}$ and $1 \leq i \leq L$ from circuits $V_1, ..., V_L$ acting upon the Hilbert space $\mathcal{H} = \mathcal{H}^{(1)} \otimes \mathcal{H}^{(2)} \otimes ... \otimes \mathcal{H}^{(L)}$ with density matrices $\bigotimes_{i=1}^{L} \rho$. Each circuit is used to estimate the mean value of Z_{prod} using $n_s^{(i)}$ shots from circuit i. Hence, SE has a variance of

$$\text{Var}(\tilde{C}_{\text{SE}}) = \sum_{i=1}^{L} \frac{4a_i^2}{n_s^{(i)}} p_i(1 \quad p_i). \tag{14}$$

Due to the absence of entanglement between the (presumed i.i.d.) circuits, there are no correlations between the estimates, so the variance factorizes in the sum of the variances, which can be written as $1 \quad m_i^2$ for Pauli observables and $p_i(1 \quad p_i)$ for projectors. If we observe that, for all $i = 1, ..., L$, $n_s^{(i)} \geq \min_{i=1,...,L} \left[n_s^{(i)}\right] =: n_s$ and use the Chebyshev inequality [63], we can lower-bound the number of shots per circuit as $n_s \geq \frac{L a_{\max}^2}{4\ ^2}$, where ϵ is the precision of the estimation. For a total of L circuits, this results in a circuit sampling complexity of

$$S = O(L^2 a_{\max}^2 / (4\epsilon^2)), \tag{15}$$

where $a_{\max} = \max_{i=1,..,L} [a_i]$ – see also Refs. [20–22].

B. Linear Combination of Unitaries

The Linear Combination of Unitaries [39] is a quantum algorithm that allows to implement sums and differences of unitaries on a quantum computer in a probabilistic fashion with a certain depth and success rate depending on the length of the sum to be implemented. In its

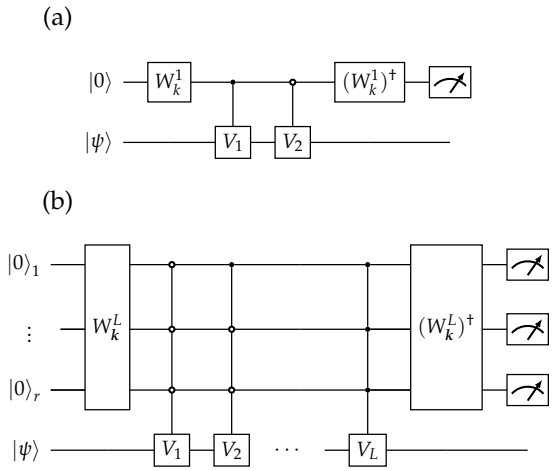


Figure 2. (a) Circuit implementing the sum of two unitaries V_1 and V_2 on a quantum computer using one control qubit and (b) circuit implementing the sum of L unitaries using up to $r = \lceil \log(L) \rceil$ qubits (both are based on the circuits given in Ref. [39]). Upon measuring the control qubit in either 0 or 1, the whole state collapses in a state proportional to either $V_1 + V_2$ or $V_1 \quad V_2$. The LCU can therefore be used to probabilistically implement arbitrary operators acting on a state $|\psi\rangle$, as those found, e.g., in Hamiltonian simulation. In its generalized implementation (b), the LCU generates all possible combinations using coefficients $\boldsymbol{k} = (k_1, ..., k_L)^{\mathrm{T}}$ of sums and differences of L unitaries. The linear combination with only positive terms is mapped to the zero state, however the probability of measuring it decreases with $1/L$ [39].

simplest form it uses the operator:

$$W_k = \begin{pmatrix} \sqrt{\frac{k}{k+1}} & \sqrt{\frac{1}{k+1}} \\ \sqrt{\frac{1}{k+1}} & \sqrt{\frac{k}{k+1}} \end{pmatrix}. \tag{16}$$

to create a superposition between control qubit states $|0\rangle$ and $|1\rangle$. Afterwards, conditional operations V_1 and V_2 are applied on an arbitrary state $|\psi\rangle$, followed by a second operation $W_k^\dagger$. This leads to a superposition of $V_1 + V_2$ and $V_1 \quad V_2$ with different probability amplitudes – see Fig. 2 (a). In particular, we see that the probability of finding the control qubit in its $|1\rangle$ state is given by

$$p_1 = \frac{k}{(k+1)^2} \| (V_1 \quad V_2) |\psi\rangle \|^2 \leq \frac{4k}{(k+1)^2}, \tag{17}$$

so that the algorithm implements $\frac{1}{\sqrt{a_+}}(V_1 + V_2)|\psi\rangle$ for $k \mapsto \infty$ and $\frac{1}{\sqrt{a_-}}(V_1 \quad V_2)|\psi\rangle$ for $k \mapsto 0$, where $a_\pm = \left| \langle\psi| (V_1 \pm V_2)^\dagger (V_1 \pm V_2) |\psi\rangle \right|$. The algorithm implements one or the other state probabilistically.

If our goal is to apply the sum of L operators, one either applies the circuit represented in Fig. 2 (a) recursively or uses a circuit with multi-controlled gates – see Fig. 2 (b) and also Appendix in Ref. [39], where typically $\log(L)$ qubits are needed. In this case the success probability will be always smaller than p_1.

The ancillas that implement the summation procedure can also be encoded in a larger number of qubits [64, 65]. If, e.g., L qubits instead of $\lceil \log(L) \rceil$ qubits are used, the implementation requires only single-qubit controlled gates and no multi-controlled gate. The overall depth of the circuit is then lower [57], but the number of control qubits scales linearly with the number of qubits implementing V_1 and V_2. The heralded nature of the LCU is particularly useful for quantum simulation. In the case of quantum estimation however, the encoding advantage of using $\lceil \log(L) \rceil$ qubits is traded off with the increased scaling of the variance, as we discuss in the next section.

C. Estimator with Linear Combination of Unitaries (LCU)

Variants of the LCU circuit have been proposed for estimation problems (with some claims of potential speedup) [38, 58]. Our goal here is to characterize the variance properties of these two estimators. We will see that for bounded variables, speedup is not intrinsically possible without further algorithmic improvements. Ref. [1] gives a more in-depth overview of different variants of the LCU algorithm, including continuous variables and integrals. We consider a circuit of the type given in Fig. 1 (b), where the unitary $W_{\boldsymbol{a}}$ generates the state

$$W_{\boldsymbol{a}} |0\rangle = |a\rangle \tag{18}$$

on the LCU register, which uses $O(\lceil \log(L) \rceil)$ qubits [57], where

$$|a\rangle = \frac{1}{\sqrt{\|\boldsymbol{a}\|_1}} \sum_{i=1}^{L} \sqrt{|a_i|}\, |i \quad 1\rangle, \tag{19}$$

for $\boldsymbol{a} = (a_1, \cdots, a_L)^{\mathrm{T}}$ and $\|\boldsymbol{a}\|_1 = \sum_{i=1}^{L} |a_i|$. The weights of the observable can in general be written in vector form $\boldsymbol{a} = \sum_{i=1}^{L} a_i \boldsymbol{e}_i$, where $\boldsymbol{e}_i$ are unit vectors in $\mathbb{R}^L$. We refer to the normalized probabilities associated with each amplitude of the state as w_i:

$$w_i = |\langle i|a\rangle|^2 = \frac{|a_i|}{\sum_{l=1}^{L} |a_l|}. \tag{20}$$

It is clear than any convex combination of such weights with coefficients $0 \leq p_i \leq 1$ lies itself between zero and one. The circuit measures the upper control qubit and the n-qubit system (analogously to the case of SE). We calculate the mean and variance of the circuit output measurements. We consider here the case in which measurements are drawn from the computational basis.

Theorem 1 (Mean and variance of the LCU estimator) *The expected value and variance of the observable*

$\Pi_{L\ \ U} = |0_c\rangle\langle 0_c| \otimes \mathbb{I}_L \otimes \Pi$ *where* Π *is an orthogonal projector that describes the measurement operation are given by*

$$\bar{p} = \sum_{i=1}^{L} w_i p_i, \tag{21}$$

$$\sigma_{\bar{p}}^2 = \sum_{i=1}^{L} w_i p_i \quad \left(\sum_{i=1}^{L} w_i p_i\right)^2 = \bar{p}(1 \quad \bar{p}), \tag{22}$$

with $p_i = \frac{1}{2}\mathrm{tr}\Big\{U_i \rho U_i^\dagger \Pi\Big\}$. *If instead the observable* $Z_{L\ \ U} = |0_c\rangle\langle 0_c| \otimes \mathbb{I}_L \otimes Z_{prod}$ *is measured then the corresponding expected value and variance are:*

$$\bar{m} = \sum_{i=1}^{L} w_i m_i, \tag{23}$$

$$\sigma_{\bar{m}}^2 = 1 \quad \left(\sum_{i=1}^{L} w_i m_i\right)^2 = 1 \quad \bar{m}^2, \tag{24}$$

with $m_i = \mathrm{tr}\Big\{U_i \rho U_i^\dagger Z_{prod}\Big\}$.

Proof. The initial input state of the LCU circuit – see Fig. 1 – is given by tensor product of the n-qubit input density matrix, the zero state of the LCU register and the zero state of the control qubit:

$$\rho_{\mathrm{in}} = |0_c\rangle\langle 0_c| \otimes \underbrace{|0\rangle\langle 0| \otimes ... \otimes |0\rangle\langle 0|}_{r} \otimes \rho, \tag{25}$$

where $r = \lceil \log L \rceil$. The evolved density matrix of the LCU circuit is given by

$$\rho_{\mathrm{out}} = \begin{pmatrix} A & B \\ B^\dagger & C \end{pmatrix}, \tag{26}$$

where A, C, B are given by [1]:

$$A = \sum_{j=1}^{L}\sum_{i=1}^{L} \frac{\sqrt{w_i}\sqrt{w_j}}{2} |0_c\rangle\langle 0_c| \otimes |i\rangle\langle j| \otimes \rho, \tag{27}$$

$$B = \sum_{j=1}^{L}\sum_{i=1}^{L} \frac{\sqrt{w_i}\sqrt{w_j}}{2} |1_c\rangle\langle 0_c| \otimes |i\rangle\langle j| \otimes (U_i\rho), \tag{28}$$

$$C = \sum_{j=1}^{L}\sum_{i=1}^{L} \frac{\sqrt{w_i}\sqrt{w_j}}{2} |1_c\rangle\langle 1_c| \otimes |i\rangle\langle j| \otimes U_i \rho U_j^\dagger. \tag{29}$$

The entries of A, B, C may change depending on the values of the single-qubit unitary gates R_1, R_2 – see Fig. 1 – acting on the first control qubit. The choice of $R_1 = H$ and $R_2 = X$ leads to the estimation of $\mathrm{tr}\{V\rho V^\dagger \mathcal{O}\}$, while, e.g., the choice of $R_1 = H$ and $R_2 = H$ or $R_2 = SH$ allows us to estimate $\mathrm{Re}\{\mathrm{tr}\{\rho V\mathcal{O}\}\}$ or $\mathrm{Im}\{\mathrm{tr}\{\rho V\mathcal{O}\}\}$, respectively. We consider here the first case, as it resembles Eq. (3). The second choice is useful whenever the target cost function is a real (or imaginary) overlap. Hence, the output probability distribution corresponding to the orthogonal projector Π acting on the subspace of the density matrix ρ and the measurement line is given by

$$\bar{p} = \mathrm{Tr}\{\rho_{\mathrm{out}}\Pi_{\mathrm{LCU}}\}, \tag{30}$$

where $\Pi_{\mathrm{LCU}} = \Pi_1 \otimes \mathbb{I}_L \otimes \Pi$, $\Pi_1 = |1_c\rangle\langle 1_c|$. The value of $\bar{p}$ is given by

$$\bar{p} = \sum_{i=1}^{L} w_i \frac{1}{2}\mathrm{Tr}\Big\{U_i\rho U_i^\dagger \Pi\Big\} = \sum_{i=1}^{L} w_i p_i. \tag{31}$$

Each mean value p_i represents the probability of success of a Bernoulli distribution, and it lies between 0 and 1, which introduces constraints on the values p_i that depend on the different unitaries U_i:

$$p_i = \frac{1}{2}\mathrm{Tr}\Big\{U_i\rho U_i^\dagger \Pi\Big\}. \tag{32}$$

The variance of the estimator is given by

$$\sigma_{\bar{p}}^2 = \mathrm{Tr}\big\{\rho_{\mathrm{out}}\Pi_{\mathrm{LCU}}^2\big\} \quad \mathrm{Tr}\{\rho_{\mathrm{out}}\Pi_{\mathrm{LCU}}\}^2, \tag{33}$$

which corresponds to the variance of a Bernoulli-type distribution, because $\Pi_{\mathrm{LCU}}^2 = \Pi_{\mathrm{LCU}}$.

Using Eq. (31), we have

$$\sigma_{\bar{p}}^2 = \bar{p}(1 \quad \bar{p}) = \sum_{i=1}^{L} w_i p_i \quad \left(\sum_{i=1}^{L} w_i p_i\right)^2. \tag{34}$$

The mean value of the estimator is bounded between zero and one, whereas the variance has its maximum at $\sigma_{\bar{x}}^2 = \frac{1}{4}$ when $\bar{p} = \frac{1}{2}$.

Now we turn to the estimation of Z_{prod} with $R_1 = X$ and $R_2 = \mathbb{I}$. If instead of the projective measurement, a measurement of Z_{prod} is carried out on the n-qubit subspace, on the whole Hilbert space we will be dealing with the measurement of the observable $Z_{\mathrm{LCU}} = \Pi_1 \otimes \mathbb{I}_L \otimes Z_{\mathrm{prod}}$. In this case we have $m_i = \mathrm{Tr}\Big\{U_i\rho U_i^\dagger Z_{\mathrm{prod}}\Big\}$ and the mean of the measurement is:

$$\bar{m} = \sum_{i=1}^{L} w_i m_i, \tag{35}$$

as well as the variance

$$\sigma_{\bar{m}}^2 = 1 \quad \left(\sum_{i=1}^{L} w_i m_i\right)^2, \tag{36}$$

where we used the property $Z_{\mathrm{LCU}}^2 = \Pi_1 \otimes \mathbb{I}_L \otimes \mathbb{I}_n$ and

$\sum_{i=1}^{L} w_i \operatorname{tr}\left\{U_i \rho U_i^\dagger\right\} = 1$. We see that the prefactors of the estimates, such as, e.g., $\frac{1}{2}$ in Eq. (32), heavily depend on the unitaries R_2 and R_1. Using $R_1 = X$ and $R_2 = \mathbb{I}$, we can also remove the factor $\frac{1}{2}$ also from $\bar{p}$. □

We can define a new estimator using the framework described before: Let $\bar{x}^{(j)}, j = 1, ..., n_s$ be shots of the LCU circuit that are sampled by measuring the control qubit in 0 and the corresponding Z_{prod} operator – or any other Pauli operator –, then

$$\tilde{C}_{\text{LCU}} = \frac{\|\boldsymbol{a}\|_1}{n_s} \sum_{j=1}^{n_s} (\quad 1)^{b(\bar{x}^{\ j})} \bar{x}^{(j)}, \tag{37}$$

with variance

$$\text{Var}(\tilde{C}_{\text{LCU}}) = \frac{\|\boldsymbol{a}\|_1^2}{n_s} (1 \quad \bar{m}^2) \leq \frac{\|\boldsymbol{a}\|_1^2}{n_s}. \tag{38}$$

Theorem 1, however, is not sufficient to bound SE and/or LCU variance, because in principle there could be covariances between the variables. Nonetheless, we show below that these are also bounded similarly to before.

Theorem 2 (LCU vs. Classically Correlated Bernoulli Samples) *Consider values $0 \leq p_i, w_i \leq 1$ $\forall i$ with $1 \leq i \leq L$ $\sum_{i=1}^{L} w_i = 1$. Let X_i be Bernoulli-distributed random variables with parameter p_i i.e. $\forall i = 1, ..., L$ we have $\mathbb{E}[X_i] = p_i$ and $\mathbb{E}\left[X_i^2\right] = \mathbb{E}\left[X_i\right] = p_i$ then we have*

$$\sum_{i=1}^{L} \sum_{j=1}^{L} w_i w_j \, \mathbb{E}[X_i X_j] \leq \sum_{i=1}^{L} w_i p_i. \tag{39}$$

Proof. By the Cauchy-Schwarz inequality, we know that

$$|\,\mathbb{E}[X_i X_j]| \leq \sqrt{\mathbb{E}[X_j]} \sqrt{\mathbb{E}[X_j]} = \sqrt{p_i} \sqrt{p_j}, \tag{40}$$

where we used the fact that $\mathbb{E}[X_i^2] = \mathbb{E}\left[X_i\right] = p_i$, and so

$$\begin{aligned} &\sum_{i=1}^{L} \sum_{j=1}^{L} w_i w_k \, \mathbb{E}[X_i X_j] \leq \\ &\leq \sum_{i=1}^{L} \sum_{j=1}^{L} w_i w_j \sqrt{p_i} \sqrt{p_j} = \left(\sum_{i=1}^{L} w_i \sqrt{p_i} \right)^2. \end{aligned} \tag{41}$$

Now we employ one of the generalized-mean inequalities [66], i.e., we use the fact that for $p < q$:

$$\left(\sum_{i=1}^{L} w_i p_i^p \right)^{\frac{1}{p}} \leq \left(\sum_{i=1}^{L} w_i p_i^q \right)^{\frac{1}{q}}, \tag{42}$$

setting $p = \frac{1}{2}$ and $q = 1$. After applying this equation to Eq. (41), we have

$$\sum_{i=1}^{L} \sum_{j=1}^{L} w_i w_j \, \mathbb{E}[X_i X_j] \leq \sum_{i=1}^{L} w_i p_i. \tag{43}$$

□

Therefore, we have that for such variables $X_1, ..., X_L$ the variance of their linear combination can be bounded from above as follows:

$$\sum_{i=1}^{L} \sum_{j=1}^{L} w_i w_j \text{Cov}(X_i, X_j) \leq \sum_{i,j=1}^{L} w_i w_j p_i (1 \quad p_j), \tag{44}$$

where $\text{Cov}(X_i, X_j) = \mathbb{E}[X_i X_j] \quad \mathbb{E}[X_i] \, \mathbb{E}[X_j]$ is the covariance. The Cauchy-Schwarz inequality provides us with the upper bound for the variance of the linear combination, i.e.:

$$\text{Var}\left(\sum_{i=1}^{L} w_i X_i \right) \leq \left(\sum_{i=1}^{L} w_i \sqrt{p_i (1 \quad p_i)} \right)^2. \tag{45}$$

If instead we consider shifted estimates Y_i in the interval $I = [\ 1, 1]$, i.e., $X_i = \frac{1}{2}(Y_i + 1)$, we obtain that the covariance is always bounded by one, which is the first term in Eq. (24). Thus, also in this case we have:

$$\sum_{i=1}^{L} \sum_{j=1}^{L} w_i w_j \text{Cov}(Y_i, Y_j) \leq 1 \quad \left(\sum_{i=1}^{L} w_i m_i \right)^2. \tag{46}$$

The maximum variance for this case can be derived by applying the same principle as the one used in Eq. (45).

We can see immediately that the variance of the LCU sampler is always larger than the variance of SE in any circumstance.

Theorem 3 *Both the LCU and SE samplers/estimators yield the same sampling complexity $S = O(L^2 a_{max}^2 / \epsilon^2)$.*

Proof. For the sampling of a bounded quantity, such the expected value of a Pauli string with respect to variational angles, the variance given in Eq. (38) of the LCU sampler is $O(L^2 a_{max}^2 / n_s) = O(1/n_s)$ and its sampling complexity is therefore $O(1/\epsilon^2)$. Conversely, the variance of SE for the mean is $O(L a_{max}^2 / n_s) = O(L/n_s)$, but L circuits need to be evaluated, so the overall sampling complexity given in Eq. (15) remains $O(1/\epsilon^2)$. □

The two algorithms are therefore equivalent again, although the LCU that uses logarithmic encoding has a slightly – $\log(L)$ – deeper circuit. This can be further brought down to $\log(\log(L))$ by using $\log(L)$ additional ancillas [64]. Unless classical correlations are present or are not negligible, some of the applications that have been considered for the LCU circuit [38, 58, 67] may be useful in specific contexts, but cannot naively provide a

speedup with respect to L in terms of sampling complexity (not even in terms of state preparation, as the variance of SE is quadratically smaller, which reduces the total number of shots needed from the L circuits).

D. Extension to real linear combinations

If the coefficients of the linear combination have both positive and negative values, we need to partially modify the encoding defined above and the derivations of the variance scaling. We first observe that we can separate the coefficients in positive and negative terms in the cost function:

$$C = \sum_{i=1}^{L} a_i m_i. \tag{47}$$

If we define

$$a_i^+ = \begin{cases} a_i, & \text{if } a_i \geq 0 \\ 0, & \text{otherwise,} \end{cases} \tag{48}$$

$$a_i \ = \begin{cases} a_i, & \text{if } a_i < 0 \\ 0, & \text{otherwise.} \end{cases} \tag{49}$$

We define therefore $L^{\pm}$ as the number of coefficients with positive and negative signs respectively, such that $L = L^+ + L \ $. In order to be able to construct an estimator similar for Eq. (4) using only positive weights, we have to first decompose it in its positive and negative terms:

$$C = \sum_{i=1}^{L} a_i^+ m_i \quad \sum_{i=1}^{L} a_i \ m_i = \|\boldsymbol{a}\|_1 \sum_{i=1}^{L} |w_i|(m_i^+ \quad m_i \ ), \tag{50}$$

where we used Eq. (20) for the definition of $|w_i|$ and $m_i^{\pm} = m_i a_i^{\pm}/|a_i|$. We assume now w.l.o.g. that the positive values appear all before L^+, i.e., at indices $i = 1, ..., L^+$ and the negative ones after L^+, i.e., at indices $i = L^+ + 1, ..., L$. Using the formulation of Eq. (3), we see that for each one of the positive or negative coefficients we have a unitary $U_i^{+/} \quad$ for the ith positive and negative coefficient, respectively. In order to use the LCU circuit to sample both negative and positive linear combinations, we use the first control qubit line in Fig. 1 (b) to obtain Fig. 3 (a). The unitaries $U_i^+, i = 1, ..., L^+$ are controlled on the value zero of the qubit and the unitaries $U_i \ , i = L^+ + 1, ..., L$ are controlled on value one, whereas the rest of the circuit remains unchanged. This leads to the modified output of the circuit – $R_1 = H$ and $R_2 = I$:

$$q_0 = \frac{1}{2}\left(\sum_{i=1}^{L^+} |w_i| m_i^+ + \sum_{i=L^+ +1}^{L} |w_i| \operatorname{Tr}\{\rho Z_{\text{prod}}\}\right), \tag{51}$$

$$q_1 = \frac{1}{2}\left(\sum_{i=L^+ +1}^{L} |w_i| m_i \ + \sum_{i=1}^{L^+} |w_i| \operatorname{Tr}\{\rho Z_{\text{prod}}\}\right). \tag{52}$$

The difference between q_0 and q_1 estimates the cost function C in Eq. (50) that uses positive and negative coefficients up to a bias, i.e., term $\langle Z_{\text{prod}}(0)\rangle = \operatorname{Tr}\{\rho Z_{\text{prod}}\}$. Since empirically using a single LCU register seems to always induce biases, we generalize in the next Section. An interesting case is one where the number of positive and negative terms in the sum is the same, i.e., $L^+ = L \ = \frac{L}{2}$ and the absolute value of each coefficient is $|w_i| = \frac{1}{L}$. In this case, the bias is the same for both q_0 and q_1 and is removed when subtracting them.
In the general case of uniform coefficients, i.e., $|w_i| = \frac{1}{L}$, we can write the expression as

$$q_0 = \frac{1}{2}\left(\frac{1}{L}\sum_{i=1}^{L^+} m_i^+ + \frac{L}{L}\langle Z_{\text{prod}}(0)\rangle\right), \tag{53}$$

$$q_1 = \frac{1}{2}\left(\frac{1}{L}\sum_{i=L^+}^{L} m_i \ + \frac{L^+}{L}\langle Z_{\text{prod}}(0)\rangle\right), \tag{54}$$

where $\langle Z_{\text{prod}}(0)\rangle = \operatorname{tr}\{\rho Z_{\text{prod}}\}$, which leaves us with

$$z = 2(q_0 \quad q_1) + \left(\frac{2L^+}{L} \quad 1\right)\langle Z_{\text{prod}}(0)\rangle, \tag{55}$$

with $z = \|\boldsymbol{a}\|_1 C$, which, as expected, reduces to the case $z = 2(q_0 \quad q_1)$ for $L^+ = L \ = \frac{L}{2}$. Therefore, this procedure allows us to estimate linear combinations of expected values that contain both positive and negative coefficients.

1. *Unbiased estimator*

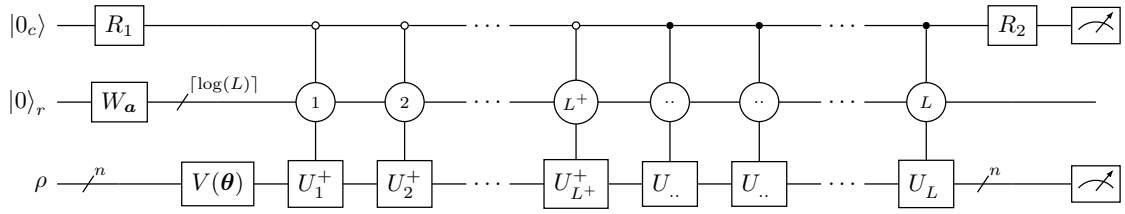
Now, let us analyze the variance in the case of this more general kind of linear combination of estimates. The bias will affect the variance of the estimator with a quadratic factor [71] $(2L^+/L \quad 1)^2 \leq 1$, which depends on the problem considered. The alternative to this biased estimator is an estimator that uses two LCU circuits, one for the positive and one for the negative coefficients and subtracts their average outcome. If we instead use a circuit with two LCU registers instead – see Fig. 3 (b) –, we can encode positive and negative coefficients $w_i^{\pm}$:

$$w_i^{\pm} = \frac{a_i^{\pm}}{\sum_{i=1}^{L^{\pm}} a_i^{\pm}} = \frac{a_i^{\pm}}{\|\boldsymbol{a}^{\pm}\|_1}, \tag{56}$$

	LCU [1, 38, 39, 58]	SE [20–22]	SA-LCU [1]	Classical Shadows [16]
(embarrassing) parallelization	no	yes	yes	yes
sampling complexity (i.i.d.)	$O(L^2/\epsilon^2)$	$O(L^2/\epsilon^2)$	$O(L^2/\epsilon^2)$	$O(L\log(L)/\epsilon^4)$
sampling complexity (AE)	$O(L/\epsilon)$	$O(L\sqrt{L}/\epsilon)$	$O(L^2/\epsilon^2)$	$O(\sqrt{L}\log(L)/\epsilon^3)$

Table I. A table representing the differences in terms of sampling complexity of the LCU and SE approaches. In case of normalized sums, all sampling complexities have to be re-scaled by L^2 for classical estimation and L for amplitude estimation (AE) [68], or one of its approximate versions [69, 70] – see also Section III A.

(a) LCU: positive and negative linear combinations with one qubit (biased estimator).



(b) LCU: positive and negative linear combinations with L^+ and L register.

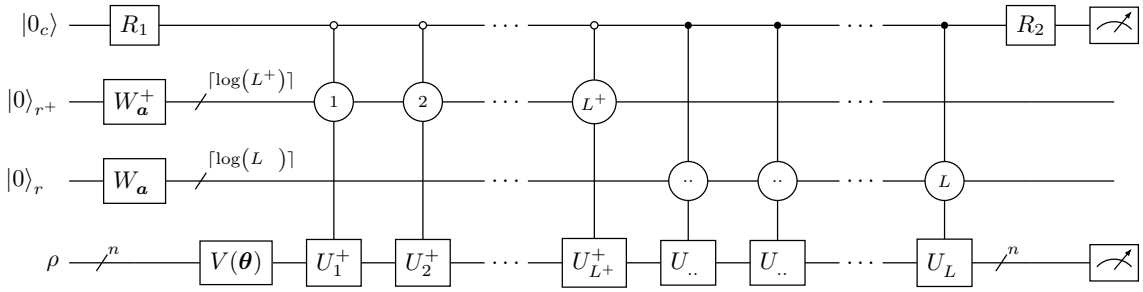


Figure 3. Representation of the circuits used to estimate real linear combinations of estimates. (a) Circuit that uses one single control qubit and one LCU registerwith $r = \lceil\log(L)\rceil$ and as a result gives a biased estimator – see Eq. (55). (b) Circuit that uses one control qubit and two LCU registers (one for positive and one for the negative terms). Circuit (b) provides us with an unbiased estimator of the real linear combination of estimates – see Eqs. (57) and (58). Here, the unitaries $U_1^+, U_1^+, ..., U_{L^+}^+$ and $U_{L^+\!+1}, U_{L^+\!+2}, ..., U_L$ refer to the positive and negative signs of the coefficients, respectively. The control values $1, 2, ..., L^+$ and $L^+ + 1, L^+ + 2, ..., L$, are implemented using the gates $W_{\boldsymbol{a}}^{\pm}$ and binary encoding with a total of $r = r^+ + r$ qubits, where $r^{\pm} = \lceil\log\ L^{\pm})\rceil$ but other types of qubit encoding for the multi-controlled gates are also possible.

in the first and second register with $r^{\pm} = \lceil\log(L^{\pm})\rceil$ qubits respectively, whose unitaries are defined as $W_{\boldsymbol{a}}^+$ and $W_{\boldsymbol{a}}$, s.t. $|a^{\pm}\rangle = W_{\boldsymbol{a}}^{\pm}|0\rangle_{L^{\pm}}$. For $R_1 = H$ and $R_2 = I$, if 0 and 1 are the outcome of the control qubit, the circuit estimates are:

$$e_0 = \frac{1}{2}\sum_{i=1}^{L^+}, |w_i^+|m_i^+ \tag{57}$$

$$e_1 = \frac{1}{2}\sum_{i=L^+\!+1}^{L} |w_i\ |m_i\ , \tag{58}$$

and, as a result, $C = 2(\|\boldsymbol{a}^+\|_1 e_0 \quad \|\boldsymbol{a}\ \ \|_1 e_1)$.

III. QUANTUM SAMPLING AND AMPLIFICATION

A. Amplitude ampli cation and estimation

Amplitude amplification (AA) [68] describes a class of algorithms based on Grover search [72] that allows to perform faster sampling on quantum computers [73, 74]. If applied to estimation problems in the context of quantum circuits, it is often referred to as amplitude estimation (AE) [68] and can be used to sample the information hidden inside of amplitudes of quantum states of the

type:

$$|\psi\rangle = \mathcal{V}\,|0\rangle_{n+1} = \sqrt{p}\,|1\rangle\,|\psi_1\rangle + \sqrt{1 \quad p}\,|0\rangle\,|\psi_2\rangle\,, \quad (59)$$

where p is a positive value that needs to be estimated and $|\psi_1\rangle, |\psi_2\rangle \in \quad ^{2^n}$ and $|\psi\rangle \in \quad ^{2^{n+1}}$ are general n-qubit quantum states. There exists a quantum algorithm that outputs to an estimator $\tilde{p}$ of p [68], which is bounded from above as follows given two integers k and n_q, where k is related to the success probability and n_q is the number of oracle queries:

$$|p \quad \tilde{p}| \leq 2\pi k \frac{\sqrt{p(1 \quad p)}}{n_q} + \frac{\pi^2 k^2}{n_q^2}. \quad (60)$$

The AE routine succeeds with probability $p_{\text{succ}} = 8/\pi^2$ if $k = 1$ and $p_{\text{succ}} \geq 1 \quad \frac{1}{2(k \quad 1)}$ if $k > 1$. The algorithm that performs multiple applications of the Grover operator for AE is simply given by [68]:

$$\mathcal{Q} = \quad \mathcal{V} S_0 \mathcal{V} \quad ^1 S_\chi, \quad (61)$$

where S_χ is the reflection operator that switches the sign of the state if the state is equal to the target, i.e., the state $\sqrt{p}\,|1\rangle\,|\psi_1\rangle$ in Eq. (59) (also known as the *good* state) [68, 69] and the operator S_0 flips the sign of the zero state. Generally, the routine for AE is defined for pure states and not for mixed inputs, such as those used in DQC1 (Deterministic Quantum Computation with one clean qubit [75]), a quantum complexity class which is closely related to LCU circuits – see also Fig. 11 (a) and (b) and Appendix A 2 for a discussion about DQC1. However, the routine that generates the mixed state can be expressed as tracing out a pure state of higher dimensionality than the one considered as input to the sampling circuit. As a result, to apply AE to one of our problems, we have to consider the entire algorithm, the one that generates the mixed state and the one that generates the state in Eq. (63). As an example and as described in Ref. [54], the n-qubit maximally mixed state $\rho = \frac{}{d}$ can be generated by creating a $2n$-qubit pure state

$$\left|\phi^+\right\rangle = \frac{1}{\sqrt{d}} \sum_{i=0}^{d} \,^1 |i\rangle \otimes |i\rangle\,, \quad (62)$$

and tracing out one of the n-qubit subsystems. So the $(n+1)$-qubit DQC1 circuit can be written as a $(2n+1)$-qubit circuit by including the generation of the mixed state in its original routine.

If we consider NISQ circuits, which are not fault-tolerant, we quickly realize that the depth of the Grover's algorithm, which lies at the core of the AE routine, cannot be easily implemented on such circuits. However, alternative versions of AE can be potentially applied on noisy quantum devices [69, 70, 76]. AE can be used to estimate means of data sampled from arbitrary distributions [77] and has a potentially vast range of industry-relevant applications, e.g., in finance. In this context, it can also be used to sample quadratically faster from stochastic processes [78].

1. *SE with amplitude ampli cation*

We consider here SE for the trace estimation problem. In this case we need to apply the AE scheme to each SE circuit with index $i = 1, ..., L$, after encoding the estimation problem appropriately. It is clear that for each term this bound is quadratically better than classical quasi-Monte Carlo sampling. We assume that each n-qubit quantum circuit corresponding to each SE term can be written as [69]:

$$\mathcal{V}_i\,|0\rangle_{n+1} = \sqrt{p_i}\,|1\rangle\,|\psi_{i,1}\rangle + \sqrt{1 \quad p_i}\,|0\rangle\,|\psi_{i,2}\rangle\,, \quad (63)$$

for $i = 1, ..., L$ and for two quantum states $|\psi_{i,1}\rangle$ and $|\psi_{i,2}\rangle$ that are determined by $\mathcal{V}_i$. This is always possible by extending the original circuit Hilbert space using an additional ancilla qubit.

For each amplitude encoded in a quantum circuit $\mathcal{V}_i$ we have:

$$|p_i \quad \tilde{p}_i| \leq 2\pi k \frac{\sqrt{p_i(1 \quad p_i)}}{n_q^{(i)}} + \frac{\pi^2 k^2}{[n_q^{(i)}]^2}, \quad (64)$$

where $\tilde{p}_i$ is the QAE estimator of p_i and $n_q^{(i)}$ are the queries for the ith AE problem. Our aim is to achieve an overall precision ϵ of the full estimation. For $i = 1, ..., L$, $n_q^{(i)} \geq \min_{i=1,...,L} \left[n_q^{(i)}\right] =: n_q$, we have:

$$\epsilon = |\sum_{i=1}^{L} p_i \quad \sum_{i=1}^{L} \tilde{p}_i| \leq \sum_{i=1}^{L} |p_i \quad \tilde{p}_i| \leq \quad (65)$$
$$\sum_{i=1}^{L} \frac{2\pi k}{n_q} \sqrt{p_i(1 \quad p_i)} + \frac{k^2 L \pi^2}{n_q^2},$$

where n_q is the number of oracle calls for each circuit $i = 1, ..., L$ and $k \in \mathbb{N}_{>0}$ and Δ denotes the use of the triangle inequality. We use here balanced weights ($a_i = 1$ for $i = 1, ..., L$), but the result can be generalized straightforwardly to the weighted case. The estimation is probabilistic, i.e., it is successful with probability p_{succ}. In the next steps we describe the working principles of the AE routine, see also Ref. [68]. The probability of success of the AE algorithm can be bound from below using the trigamma function $\psi^{(1)}(k) = \sum_{s=k}^{\infty} 1/s^2$ [68, 79]. For any $k \in \mathbb{N}_{>0}$ we have:

$$p_{\text{succ}} \geq 1 \quad \frac{1}{2}\psi^{(1)}(k). \quad (66)$$

As the trigamma function fulfills the identity [68]:

$$\psi^{(1)}(k) \leq \frac{1}{k \quad 1}, \quad (67)$$

the lower bound for the success probability can be written as

$$p_{\text{succ}} \geq 1 \quad \frac{1}{2(k \quad 1)}, \tag{68}$$

where p_{succ} is the success probability of one quantum AE algorithm running on one of the estimation problems. For $k = 1$ we have $p_{\text{succ}} = \frac{8}{\pi^2}$ and considering that the variance $p_i(1 \quad p_i)$ is always smaller than $\frac{1}{4}$ for $0 \leq p_i \leq 1$, we can use Eq. (65) to find a bound for the precision ϵ of the estimation:

$$\epsilon \leq \frac{\pi L}{n_q} + \frac{\pi^2 L}{n_q^2}, \tag{69}$$

and after solving the quadratic equation for small ϵ we obtain for the bound $\frac{\pi L}{n_q} \approx \epsilon$. Naively, the upper bound for the number of evaluations needed seems to be $O(L^2/\epsilon)$ – since we additionally consider the evaluation of L parallel AE routines – but a more detailed treatment – see Appendix B for the derivation from the point of view of Maximum Likelihood Quantum Amplitude Estimation (MLQAE) [69] – leads to a better bound which matches previous results obtained in other similar contexts [18]:

$$S = O(L\sqrt{L}/\epsilon), \tag{70}$$

that is, L circuits with n_q queries calls each, in analogy with SE – to have convergence with probability p_{succ}.

2. *LCU with amplitude ampli cation*

In the case of the circuit shown in Fig. 1, an amplification oracle can be constructed using the methodology given in Ref. [69, 80], which uses an ancilla qubit to encode the estimation of arbitrary observables. In classical estimation, the renormalization factor causes the variance of the LCU estimator to grow quadratically with the number of estimates. In the case of AE, it only increases the error linearly.

$$\mathcal{V}_{\text{LCU}} |0\rangle_b = \sqrt{\bar{p}} |1\rangle |\psi_{\text{LCU},1}\rangle + \sqrt{1 \quad \bar{p}} |0\rangle |\psi_{\text{LCU},2}\rangle , \tag{71}$$

where $b = (n+1)r$, $\bar{p} = \sum_{i=1}^{L} w_i p_i$ and for two quantum states $|\psi_{\text{LCU},1}\rangle$ and $|\psi_{\text{LCU},2}\rangle$ that are defined by the action of $\mathcal{V}_{\text{LCU}}$. An amplified LCU circuit $\tilde{C}^{\text{AE}}_{\text{LCU}}$ estimates the same value as C_{LCU} given in Eq. (37), but with a lower variance:

$$C = \mathbb{E}[\tilde{C}_{\text{LCU}}] = \mathbb{E}[\tilde{C}^{\text{AE}}_{\text{LCU}}] \tag{72}$$

$$\text{Var}(\tilde{C}^{\text{AE}}_{\text{LCU}}) = \|\boldsymbol{a}\|_1^2 \frac{\bar{p}(1 \quad \bar{p})}{n_q^2} = \epsilon^2, \tag{73}$$

where C_{full} is the corresponding non-renormalized cost function and n_q the number of queries. As usual the variance of the estimator determines the error threshold ϵ. The number of samples/queries is thus quadratically smaller than in standard sampling and as such $n_q \sim O(\frac{\|\boldsymbol{a}\|_1}{}) = O(L/\epsilon)$. We immediately see that the sampling complexity of this estimator scales as $O(L/\epsilon)$ for the non-renormalized estimation. We can briefly analyze a further important difference between the two algorithms. In SE case, we run L different AE routines, each one with a success probability $p_{\text{succ}} \geq \frac{8}{\pi^2}$. In the LCU case, we apply the routine to one single circuit. Clearly, the second approach has an advantage in terms of overall probability of failure. In fact, in the former case there are L independent algorithmic runs that can return the wrong outcome. In the latter case, however, only one circuit with a corresponding AE routine and success probability p_{succ} is used.

3. *Single-ancilla LCU*

Estimating a quantum cost function that depends on parameters $\boldsymbol{\theta}$ requires projective measurements of a density matrix whose dynamics is described by, e.g., a variational circuit. The estimation process can be accomplished by preparing independent circuits or by means of the LCU approach, which requires ancilla qubits to encode the linear combination. A third approach, the single-ancilla LCU (SA-LCU), is presented in Ref. [1]. It uses random sampling of unitaries from the set of weights that are used in the the linear combination. Ref. [1] shows that it provides us with the correct result in the case of projective measurements. In this case, given a collection of unitaries $\mathcal{U}_S = \{U_1, ...U_L\}$ and normalized weights $\{w_1, ..., w_L\}$, such weights induce a probability distribution $\pi_{\mathcal{U}}$ over the set of unitaries [1]. Randomly sampling a unitary according to the distribution of weights and implementing it on the circuit in Fig. 1 without using the linear combination register returns the expected value:

$$\bar{p} = \mathbb{E}_{U \sim \pi_{\mathcal{U}}} [p(U)] = \sum_{i=1}^{L} w_i p_i, \tag{74}$$

where p_i is the probability of finding the ith state, i.e., $U_i \rho U_i^\dagger$ in the eigenstate of Π. This is equivalent to preparing a mixed state ρ_{LCU} of the type:

$$\rho_{\text{LCU}} = \sum_{i=1}^{L} w_i U_i \rho U_i^\dagger, \tag{75}$$

and performing a measurement Π on it. In other words, if it is possible to efficiently generate this mixed state, then one can obtain the same kind of parallelization as with LCU. This approach resembles the one used for trace estimation in DQC1, where we prepare instead a maximally mixed state $\rho = \frac{}{d}$. If we assume a large number of samples and only consider the variance with respect to the

sampled random unitaries [81], we obtain the expression:

$$\mathrm{Var}_{U\sim\pi_{\mathcal{U}}}\left[p(U)\right] = \mathbb{E}_{U\sim\pi_{\mathcal{U}}}\left[p(U)^2\right] \quad \mathbb{E}_{U\sim\pi_{\mathcal{U}}}\left[p(U)\right]^2, \tag{76}$$

which gives

$$\mathrm{Var}_{U\sim\pi_{\mathcal{U}}}\left[p(U)\right] = \sum_{i=1}^{L} w_i p_i^2 \quad \bar{p}^2. \tag{77}$$

However, this derivation is not valid in the few-shot limit, as we need to consider the nested sampling of both U and the shots coming from the quantum circuit that we use to estimate $p(U)$. In the latter case, the variance is computed using the law of total variance [82]. By defining $\bar{x}$ as the random variable associated with a single shot from the circuit with unitary U, we have

$$\mathrm{Var}(\bar{x}) = \mathbb{E}_{U\sim\pi_{\mathcal{U}}}\left[\mathrm{Var}_q(\bar{x})\right] + \mathrm{Var}_{U\sim\pi_{\mathcal{U}}}\left[\mathbb{E}_q[\bar{x}]\right], \tag{78}$$

where $\mathrm{Var}_q(\bar{x})$ and $\mathbb{E}_q[\bar{x}]$ refer to the quantum statistics of the single-shot circuit sampling. Eq. (78) results in:

$$\mathrm{Var}(\bar{x}) = \bar{p}(1 \quad \bar{p}), \tag{79}$$

where $\bar{p} = \mathbb{E}_{U\sim\pi_{\mathcal{U}}}\left[p(U)\right]$, which is exactly the variance of the LCU circuit. The sampling complexity is therefore $O(\frac{1}{2})$ for the renormalized version and $O(\frac{L^2}{2})$ for the non-renormalized one. We observe that due to $p_i^2 \leq p_i$, Eq. (77) is always smaller than Eq. (76).

Let us now consider the same problem from the point of view of amplitude estimation, that is, we implement the SA-LCU estimator given in Eq. (74) and perform amplitude amplification on it. In this case, the speed up seems not to be present, due to the behaviour of the variance due to the law of total variance [82, 83]. We define the total number of queries n_q for AE and the number of random unitary samples N_U needed by this approach to reach a certain precision ϵ. This is not the same as the full LCU, as the sampling of the unitaries from $\pi_{\mathcal{U}}$ is performed classically. The estimator $\tilde{p}_{\mathrm{SA}}$ with $\mathbb{E}_U[\tilde{p}_{\mathrm{SA}}] = \bar{p}$ for the amplified ancilla-free LCU that outputs the amplified value $p_{\mathrm{AE}}(U)$ given a unitary $U \sim \pi_{\mathcal{U}}$ has a variance of:

$$\mathrm{Var}\left(\tilde{p}_{\mathrm{SA}}\right) = \frac{1}{N_U}\Bigg(\mathbb{E}_U[p_{\mathrm{AE}}(U)]\left(\frac{1}{n_q^2} \quad \mathbb{E}_U[p_{\mathrm{AE}}(U)]\right) + \tag{80}$$

$$+\left(1 \quad \frac{1}{n_q^2}\right)\mathbb{E}_U[p_{\mathrm{AE}}(U)^2]\Bigg) \leq \frac{\mathbb{E}_U[p_{\mathrm{AE}}(U)^2]}{N_U}.$$

For the case $M = 1$ we retrieve the variance of the ancilla-free and ancilla-based LCU estimator. It is therefore clear that there is no advantage in increasing the number of samples from the AE estimator, as the reduction in the variance is purely classical and controlled by the number of unitaries sampled from the corresponding distribution rather than the circuit shots. Interestingly, this means that any average of amplified quantum cost functions retains this property. As a consequence, such estimation problems can be tackled, in theory, by using exclusively *single-shot* estimation.

On the other hand, the estimator that encodes the sum of unitaries on the quantum circuit can be amplified to the Heisenberg limit. As the ancilla-free LCU can implement the mean value of any observable, we conclude that for any average value of an observable $\mathbb{E}_{\boldsymbol{\lambda}\sim P}\left[\langle\mathcal{O}(\lambda)\rangle\right]$ (for example, the average over a dataset in QML), single-shot estimators are a better choice, provided the number of data points available is high enough. This is true for both amplified and non-amplified nested estimation problems [83].

B. Example: Quantum machine learning

Quantum machine learning (QML) refers to the class of algorithms that allow to approximate functions on quantum computers using appropriate families of quantum circuits given a data set. The goal is to perform tasks such as classification, clustering and regression with a certain speedup compared to the classical case [28, 84]. In this context, we distinguish two main approaches: (1) Algorithms that make use of known quantum routines, such as the HHL algorithm [85] and Grover search [72] and (2) variational algorithms that encode a problem using quantum circuits and variational families of unitaries that are then optimized accordingly. The latter class of algorithms is then commonly sub-divided in explicit and implicit models [86]. In these models, we try to construct an approximator $f_{\boldsymbol{\theta}}$ with parameters $\boldsymbol{\theta} \in \mathbb{R}^N$ – see also Eq. (93). Explicit models are defined by a mapping with input $\boldsymbol{x} \in \mathbb{R}^m$:

$$f(\boldsymbol{\theta}, \boldsymbol{x}) = \mathrm{tr}\{\rho(\boldsymbol{x})\mathcal{O}(\boldsymbol{\theta})\}, \tag{81}$$

with $\mathcal{O}(\boldsymbol{\theta}) = V(\boldsymbol{\theta})\mathcal{O}V(\boldsymbol{\theta})^\dagger$, while implicit models are given by a linear combination of functions of the data points $\boldsymbol{x}_1, ..., \boldsymbol{x}_M \in \mathbb{R}^m$: $f_{,\mathcal{D}}(\boldsymbol{x}) = \sum_{m=1}^{M} {}_m k(\boldsymbol{x}, \boldsymbol{x}_m)$ where the kernel is defined as $k(\boldsymbol{x}, \boldsymbol{x}_m) = \mathrm{tr}\{\rho(\boldsymbol{x})\rho(\boldsymbol{x}_m)\}$ and $\boldsymbol{\alpha}$ is a vector of parameters. Both quantities can be sampled using the LCU circuit – see Fig. 1 (b). As an illustrative example, we consider the problem of regression for quantum cost functions. When dealing with (quantum) supervised learning, we are usually given a dataset $\mathcal{D}$ made of pairs of inputs $\boldsymbol{x}_1 \in \mathbb{R}^m, ..., \boldsymbol{x}_L \in \mathbb{R}^m$ and corresponding labels $y_1 \in [0, 1], ..., y_L \in [0, 1]$, i.e., $\mathcal{D} = \{(\boldsymbol{x}_i, y_i), i = 1, ..., L\}$. We consider here only two possible label options, i.e., the problem corresponds to binary classification and requires two output qubits. The generalization to the k-nary classification problem can be achieved by considering a vector $\boldsymbol{y} \in \{ \quad 1, 1\}^k$ and using $\log(k)$ qubits, where k is the number of different possible labels in the data. We assume that there exists a function $f : \mathbb{R}^N \times \mathbb{R}^m \mapsto [0, 1], (\boldsymbol{\theta}, \boldsymbol{x}) \shortmid \to y' = f(\boldsymbol{\theta}, \boldsymbol{x})$ that depends on parameters $\boldsymbol{\theta}$ that takes $\boldsymbol{x}_i$ as inputs and out-

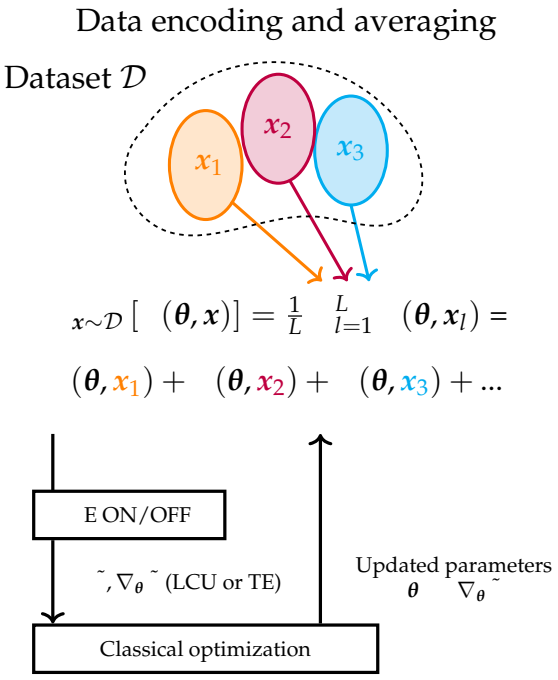


Figure 4. A representation of data encoding and sampling for a QNN [67] (explicit model). Classical data needs to be loaded on the quantum sampler/estimator. This procedure can be quite expensive due to the input data size and may require some classical pre-processing [86], but it can be realized with both LCU and SE approaches. In addition to the two estimators, (near-term) AE routines can be considered [68–70, 87]. The cost function is controlled by variational parameters that are optimized classically using gradient-based [50, 88, 89] or gradient-free [90, 91] algorithms. Different types of cost functions, such as the one given in Eq. (82), can be estimated using the LCU method given in Eq. (50) or variants thereof.

puts guess values y_i'. The regression problem is usually solved by minimizing the mean-squared-error loss [71], that is by minimizing the following cost function with respect to the model parameters $\boldsymbol{\theta}$:

$$C_{\mathrm{ML}}(\boldsymbol{\theta}) = \frac{1}{2L}\sum_{i=1}^{L}(y_i \quad f(\boldsymbol{\theta},\boldsymbol{x}_i))^2. \tag{82}$$

The solution $\boldsymbol{\theta}^*$ to the minimization problem is given by $\boldsymbol{\theta}^* = \mathrm{argmin}_{\boldsymbol{\theta}}\, C_{\mathrm{ML}}(\boldsymbol{\theta})$ and can be found using different optimization procedures [92]. Let us now consider the case in which the model is given by a QNN circuit. In this case, the function approximator f is given by a family of parametrized quantum circuits – see Eq. (81) – where the input-dependent density matrix $\rho(\boldsymbol{x})$ is constructed by applying a so-called feature map $V_{\boldsymbol{\phi}}(\boldsymbol{x})$ – $\boldsymbol{\phi}$ are specific feature encoding parameters – on the initial quantum state ρ such that $\rho(\boldsymbol{x}) = V_{\boldsymbol{\phi}}(\boldsymbol{x})\rho V_{\boldsymbol{\phi}}^{\dagger}(\boldsymbol{x})$. Compared to the standard cost functions of classical machine learning, this cost function is computed as an average over measurement outcomes. This type of encoding has been used extensively in QML and it is available in Ref. [67]. If we consider an observable $\mathcal{O} = Z_{\mathrm{prod}}$ and measure therefore the parity of the circuit output as a function of the input data, we see that we can simplify the cost function in Eq. (82) using the fact that the squared terms sum to one we are left with the expression [28]:

$$C_{\mathrm{ML}}(\boldsymbol{\theta}) = \frac{1}{2}\quad C_1(\boldsymbol{\theta}) + C_2(\boldsymbol{\theta}), \tag{83}$$

$$C_1(\boldsymbol{\theta}) = \frac{1}{L}\sum_{l=1}^{L} f(\boldsymbol{\theta},\boldsymbol{x}_i)y_i \tag{84}$$

$$C_2(\boldsymbol{\theta}) = \frac{1}{2L}\sum_{l=1}^{L} f(\boldsymbol{\theta},\boldsymbol{x}_i)^2. \tag{85}$$

It is clear that this is nothing else than a linear combination of estimates that contains both positive and negative coefficients. It can therefore be computed using both of the methodologies outlined in II A (standard approach, SE) and II C (LCU), respectively. A schematic diagram of a QML sampling process and optimization routine is given in Fig. 4.

The estimation of such quantity represents a possible application of the LCU approach. In our case, we make use of the simplified cost function:

$$C_1(\boldsymbol{\theta}) = 1 \quad \frac{1}{L}\sum_{i=1}^{L} f(\boldsymbol{\theta},\boldsymbol{x}_i)y_i. \tag{86}$$

The second part of the cost function can be sampled by either adding a control operation to the LCU circuit or by sampling with two different circuits. The first is a LCU circuit that implements Eq. (86). The second a is a circuit that implements $C_2(\boldsymbol{\theta})$. The latter can be realized by preparing the tensor product of the QNN unitary that is used to parameterize $f(\boldsymbol{\theta},\boldsymbol{x})$ conditioned on the LCU register. The LCU register implements the summation process and the tensor product allows us to estimate $f(\boldsymbol{\theta},\boldsymbol{x})^2$.

1. *Sampling cost functions from datasets*

We test the implementation of LCU and SE on a QML problem. We employ the Quantum Neural Network (QNN) given in QISKIT [67], which is composed of a ZZ feature map for encoding the input data $\boldsymbol{x}$ sampled from the data set and a variational circuit $V(\boldsymbol{\theta})$. In this context, we are interested in simply evaluating the average cost function over the $(\boldsymbol{x}_i, y_i)$ input values for $i = 1, ..., L$ and estimating its variance using the expressions introduced in Eq. (11) for SE and Eq. (24) for the LCU estimator applied on the cost function in Eq. (86). Since the linear combination contains both positive and negative values, we use the expression given in Eq. (55) and modify the variance accordingly. As before, in this

context we have the inner expectation value, corresponding to the average over the quantum statistics and the external one, corresponding to the average over the classical sampling.

For the plots in Fig. 5 we use four different datasets: (I) normally distributed variables $\boldsymbol{x} \sim \mathcal{N}(\mathbf{0}_d, \mathbb{I}_d)$, for which we have $y_i = f(\boldsymbol{x}_i) = 2 \cdot \text{sign}(\sum_j x_i^j) \quad 1$, i.e., the sign of the sum of vector components, – (II) – autocorrelated variables, where each variable $\boldsymbol{x}_i$ is defined by the following relation:

$$\forall i = 2, ..., L: \ \boldsymbol{x}_i = i \cdot \boldsymbol{x}_0, \ \boldsymbol{x}_0 \sim \mathcal{N}(\mathbf{0}_d, \mathbb{I}_d), \tag{87}$$

and again $y_i = f(\boldsymbol{x}_i) = 2 \cdot \text{sign}(\sum_j x_i^j) \quad 1$. The autocorrelated data is used to analyze the behaviour of SE variance in presence of correlations between different circuits. (III) data sampled from the IRIS [93] dataset – (IV) – data sampled from the MNIST data set [94] after performing a PCA transformation [95] to achieve dimensionality reduction [86]. The results of our simulations are shown in Fig. 5. We used 10000 shots for each circuit evaluation, and a number ranging from 2 to 100 of estimates of classical data that are summed together. In these plots we show different types of sampled data: in the first row we plot the values of the cost functions as a function of the number of samples for the four datasets (autocorrelated, uncorrelated, IRIS and MNIST, respectively) for both LCU and SE. The values for datasets I, II and IV are also averaged over 50 different sampling runs. Shaded regions show the standard deviation and the fourth moment of the estimates. We see that the standard deviation for the LCU estimator is always larger than the one of SE. The second row of plots shows the variance of the LCU estimators, SEs and also the maximum possible variance that hypothetic classically correlated (Rademacher) variables $X_1, ..., X_L$ with the same mean values as those used for SE. As we showed already, the maximum of such quantity cannot exceed the LCU variance, which is confirmed by the result in each plot.

Since the cost functions are initially renormalized, in order to obtain the sampling complexity we need to multiply them with an appropriate re-scaling parameter L^2 for the LCU estimator and L^3 for SE – because we need to account for the sampling of L different circuits, as shown in Eq. (15). As a result, we see that the sampling complexities of the two estimators are asymptotically the same. The maximum possible variance that SE can achieve is due to the Cauchy-Schwarz inequality – see also Eq. (40) (assuming classical correlations between the samples are possible). In this case, however, we have to consider that as a result of the correlations, some operations may commute, which would reduce the total number of copies needed to estimate their values. The extreme case is the estimation of the same (Pauli) observable multiple times: in this case the LCU and SE become the same estimator. Interestingly, this is also the (classical) case in which the so-called Poisson-Binomial probability distribution [96] and its Binomial approximation are the same – see the discussion in Appendix A 3. This aspect helps us shed light on the different properties of LCU and SE and how these differences affect the scaling of estimation on quantum circuits. It also shows that full LCU is mostly indicated for sampling and estimation tasks where full or at least near-term amplitude amplification algorithms can be implemented.

IV. GRADIENTS OF QUANTUM COST FUNCTIONS

In this Section, we consider the problem of sampling and estimating gradients of quantum cost functions. This problem has been discussed in several previous works [4, 26, 27, 36, 44–50, 59], both from the point of view of LCU-type approaches and SE. Here, we want to focus in particular on the circuit shown in Fig. 6 (b) and apply it to the following problems: the estimation of derivatives with forward propagation [99], see Section IV C, its application in the calculation of general SU(d) gradients [44], see Section IV D and Fig. 8, and its application in quantum control [100], see Section IV E. These altogether complete the description of gradient estimation circuits using LCU approaches and allow for the estimation of arbitrary gradients of unitary operations.

A. Quantum circuit gradients

For general gradients of quantum cost functions – see Eq. (4) – we can write:

$$\nabla_{\boldsymbol{\theta}} \langle \mathcal{O}(\boldsymbol{\theta}) \rangle = \text{Re}\{\text{tr} \ \nabla_{\boldsymbol{\theta}} V(\boldsymbol{\theta}) \rho V^\dagger \mathcal{O}\}, \tag{88}$$

which, as shown in Fig. 7, can be again implemented by combining LCU and a Hadamard test [50]. Let us consider the outcome of the so-called Power-of-Two-Qubits circuit or POTQ [54] circuit, a specific type of Hadamard test that outputs the following overlap between unitaries $V(\boldsymbol{\theta})$ and U^T and which is shown in Fig. 7 (a):

$$p_\pm(\boldsymbol{\theta}) = \frac{1}{2}\left(1 \pm \frac{1}{d}\text{Re}\{\text{tr}\{V(\boldsymbol{\theta})U^\dagger\}\}\right), \tag{89}$$

where $p_\pm(\boldsymbol{\theta})$ are the probability of finding the control qubit in the state 1 or 0, respectively. The cost function corresponding to the real trace overlap can then be estimated as $C_{\text{POTQ}}(\boldsymbol{\theta}) = 1 \quad p_+(\boldsymbol{\theta}) \quad p \ (\boldsymbol{\theta}) = 1 \quad \frac{1}{d}\text{Re}\{\text{tr}\{V(\boldsymbol{\theta})U^\dagger\}\}$. The gradient of this cost function is given by:

$$\nabla_{\boldsymbol{\theta}} C_{\text{POTQ}}(\boldsymbol{\theta}) = \quad \frac{1}{d}\text{Re}\{\text{tr}\{\nabla_{\boldsymbol{\theta}} V(\boldsymbol{\theta})U^\dagger\}\}. \tag{90}$$

Eq. (90) can be estimated using again a POTQ circuit where the unitary gradient $\nabla_{\boldsymbol{\theta}} V(\boldsymbol{\theta})$ is implemented.

We first consider the specific case in which the unitary

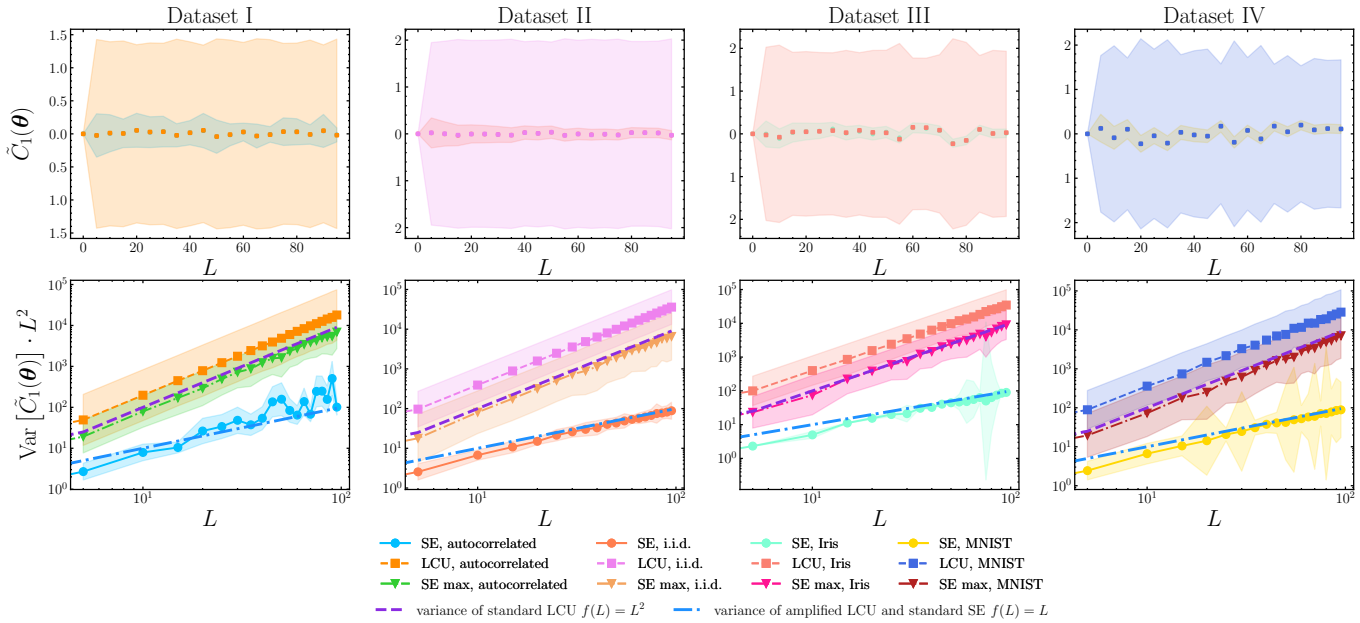


Figure 5. An example of estimation performed with (simulated) quantum circuits in QISKIT [67] for the regression cost function $C_1(\ )$ in Eq. (86), where we use 10000 shots per circuit and from 2 to 100 estimates. The values for datasets I, II and IV are also averaged over 50 different sampling runs. (Top line) Mean values and sampling complexity of the estimator $\tilde{C}_1(\ )$ of $C_1(\ )$ for a random $\sim \mathcal{N}(\mathbf{0}_N, \mathbb{I}_N)$ – see Eq. (86). The sampling complexity is defined as the asymptotical total number of queries needed to estimate a quantity up to a fixed precision ϵ. (Bottom line) sampling complexities of estimators according to Eqs. (14) and (24) for different types of datasets used as inputs to the QNN for both SE (blue line) and the LCU (orange line) estimators. In addition, the maximum possible SE variance is also shown (green line), and, as expected from Eq. (40), it always lies below the LCU variance. Shaded regions show the uncertainty for both mean (standard deviation) and variance (here we use an approximate estimate of the fourth moment for SE and LCU, while for the maximum of SE we use the fourth moment of LCU as an upper bound). Column (I) shows the results of sampling auto-correlated quantities as shown in Eq. (87). (II) shows the results for sampling i.i.d. Gaussian variables. (III) shows the results of sampling from the IRIS dataset and (IV) from the MNIST dataset (whose dimensionality is reduced first with a PCA transformation, see also the procedure used in Ref. [86]). We observe that in all cases the variance grows linearly compared to the variance of the LCU, which is always quadratic. In the auto-correlated case, the particular structure of the data seems to induce a superlinear behaviour, most likely due to the fact that estimates that are functions of the same unitaries (or highly correlated unitaries) are considered, see also the discussion in Appendix A 3.

$V(\boldsymbol{\theta})$ has a single parameter θ, i.e., $V(\theta) = \exp\{\ \ i\theta H\}$ for a Hamiltonian H. Assuming the Hamiltonian can be written as $H = \sum_{i=1} h_i P_i$ [26, 27], this gradient can be implemented using the circuit given in Fig. 1 (b), since $\dot{V}(\theta) = \ \ iHV(\theta)$. For a quadratic cost function, such as the standard gate infidelity – see Fig. (7) and Ref. [54]:

$$I(\boldsymbol{\theta}) = 1 \ \ \frac{1}{d^2}\left|\mathrm{tr}\{V(\boldsymbol{\theta})U^\dagger\}\right|^2, \tag{91}$$

we have instead a gradient of the type:

$$\nabla_{\boldsymbol{\theta}} I(\boldsymbol{\theta}) = \ \ \frac{2}{d^2}\,\mathrm{Re}\{\mathrm{tr}\{\ V(\boldsymbol{\theta})U^\dagger) \otimes \ \nabla_{\boldsymbol{\theta}} V(\boldsymbol{\theta})U^\dagger)\}\}, \tag{92}$$

which can be sampled by modifying the POTQ circuit with a register of $4n$ qubits and encoding the gradient of the unitary operation on the circuit appropriately [50].

B. Parameter-shift rules

Let us now consider the circuit product ansatz:

$$V_{\mathrm{CA}}(\boldsymbol{\theta}) = \prod_{i=1}^{N} V_i(\theta_i), \tag{93}$$

with $\boldsymbol{\theta} \in \mathbb{R}^N$. In simulation, the gradient of this circuit can be effectively computed using the GRAPE algorithm [100]. However, its evaluation on a real quantum device seems challenging, as it is not possible to naively store the results of intermediate unitaries on quantum experiments as we do in classical simulations, though some proposals for back-propagation on quantum circuits do exist [5, 6]. In the case where Eq. (93) is valid, we see that the gradient of the cost function given in Eq. (88) simplifies to:

$$\frac{\partial}{\partial\theta_i}\langle\mathcal{O}(\boldsymbol{\theta})\rangle = i\mathrm{Tr}\Big\{\mathcal{O}V^{(i)}_{(1)}(\boldsymbol{\theta})[H_i, \tilde{\rho}_i(\boldsymbol{\theta})]V^{(i)\dagger}_{(1)}(\boldsymbol{\theta})\Big\}, \tag{94}$$

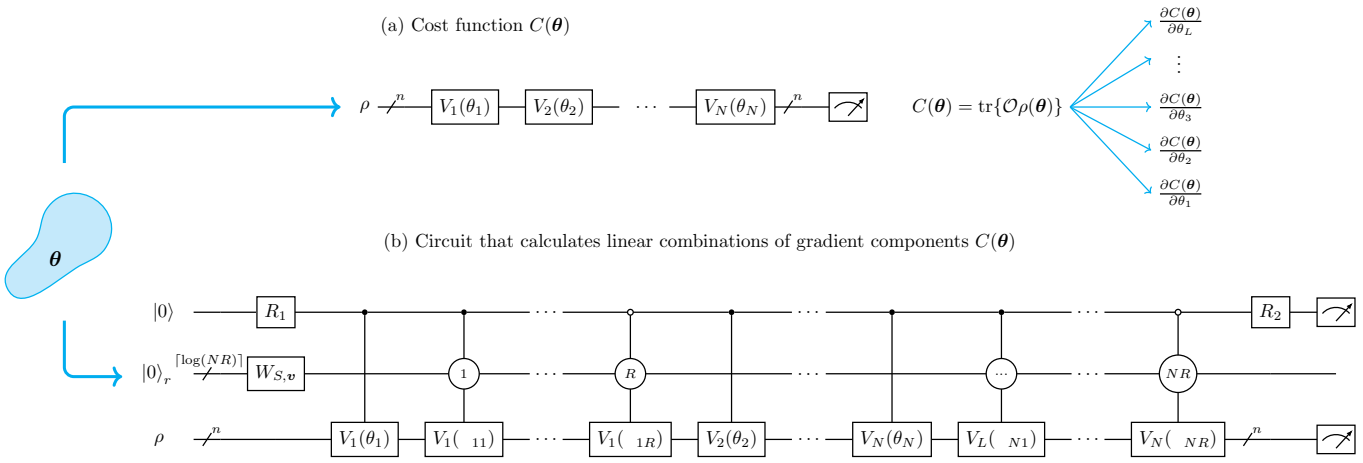


Figure 6. Schematic representation of the forward derivative circuit for a general quantum cost function given in Eq. (3). The cost function circuit is pictured in (a). In the case of parameter-shift rules, the cost function is evaluated multiple times by shifting the parameter vector along the derivative axis for instance according to Eq. (95), and where $_{ir}$, $i = 1, ..., N, r = 1, ..., R$ represent the shifts in the respective angles. The gradient circuit [38] (b), instead, uses a LCU register (or two registers for positive-negative coefficients) with controlled operations to perform either parameter-shift rules or finite-differences [4, 36, 47, 97, 98], or to encode the gradient operator itself [26, 50]. The gradient circuit uses multi-controlled gates on the values $1, ..., NR$ (assuming w.l.o.g. that each gate $V_1, ..., V_N$ requires a total of R shifts). Controlled operations on the values $1, ..., NR$ can be implemented using binary encoding and $\lceil \log(NR) \rceil$ qubits. The LCU unitary $W_{S,\boldsymbol{v}}$ encodes both the coefficient matrix S in Eq. (95) and the forward derivative coefficients $\boldsymbol{v}$ in Eq. (96). To obtain the standard gradient of the quantum cost function the circuit output needs to be evaluated N times with N different binary input vectors. The forward derivative, instead, can be evaluated directly by encoding a vector $\boldsymbol{v}$ in the LCU register with $r = \lceil \log(NR) \rceil$ qubits (we assume here for the sake simplicity only positive linear combinations). The forward gradient can be estimated by using a random input $\boldsymbol{v} \sim Q$, where Q is a suitable probability distribution, see Ref. [99].

where $\tilde{\rho}_i(\boldsymbol{\theta}) = V^{(N)}_{(i+1)}(\boldsymbol{\theta})\rho V^{(N)\dagger}_{(i+1)}(\boldsymbol{\theta})$ and $V^j_k(\boldsymbol{\theta}) = \prod_{l=N}^{L} V_l(\theta_l)$. By finding appropriate coefficients S^i_k and parameter shifts $_{ik}$, we can use the interpolation ansatz [4]:

$$\frac{\partial}{\partial \theta_i}\langle \mathcal{O}(\boldsymbol{\theta})\rangle = \sum_{k=1}^{R} S_{ik}\langle \mathcal{O}(\boldsymbol{\theta} + \ _{ik}\boldsymbol{e}_i)\rangle, \tag{95}$$

where R is the number of parameter shifts (in Ref. [4] it is set as twice the number of distinct eigenvalue differences contained in the spectrum of the gate generator) and $\boldsymbol{e}_i$, $i = 1, ..., N$ are unit vectors of the parameter space $\mathbb{R}^N$ – see Eq. (93). We can substitute symbolically any type of product ansatz into Eq. (95) to find a suitable formula for the parameter-shift vectors $\boldsymbol{\alpha}_1 = (\ _{11}, ..., \ _{1R})^{\mathrm{T}}, ..., (\ _{N1}, ..., \ _{NR})^{\mathrm{T}}$ and the coefficient matrix S_{ik}. This expression is the general parameter-shift rule for quantum gradients [4], written using a different notation. Moreover, even in the noiseless case, one needs $O(R^2N/\epsilon^2)$ circuit evaluations to compute every parameter variation. The same is true if the gradient is computed through, e.g., finite-difference methods [98], with the notable exception that the scaling for one finite-difference shift is $O\left[N/(\epsilon^2\delta^2)\right]$ [48], which is generally unfavorable due to the small size of δ. In this case, a shift $\delta \sim O(1)$ is often the optimal choice [44, 98].

It is clear that Eq. (95) can also be implemented using the LCU sampler – see Eq. (55). Pauli gates are a simple application, as they only have two distinct eigenvalues [36]. The n-qubit XY gate or Mølmer-Sørensen gate offer a more interesting use case with $n + 1$ and $\lceil n/2 \rceil + 1$ distinct eigenvalues [23, 59] respectively, which correspond to $O[\log(n)]$ control qubits in the LCU register. However, for arbitrary generators the eigenvalue structure may also scale more unfavourably.

For arbitrary generators with partially known spectra, several methods of reconstructing their landscape have been proposed. Often, these methods require performing linear transformations of a vector of estimates, such as computing their Discrete Fourier Transform (DFT). In this case, however, the coefficients need to be determined from the estimates classically, which means that loading them on the LCU register is less beneficial than approaching the problem using the method given in Ref. [26].

C. Directional derivatives and forward propagation

Another application of LCU is represented by forward propagation [99]. While back-propagation seems to be challenging to implement on quantum computers [5, 6], forward propagation on quantum circuits can instead be achieved by using the circuit shown in Fig. 10 (d) or by

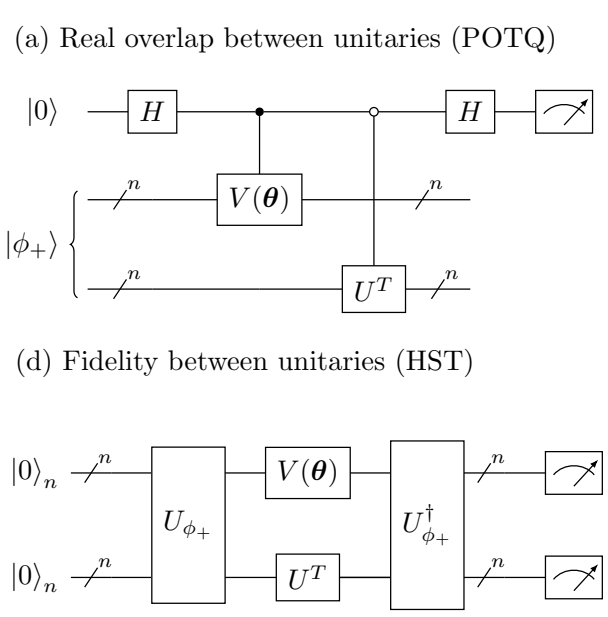


Figure 7. Relevant test circuits for quantum compilation introduced in Ref. [54]: (a) Power-of-Two-Qubits (POTQ) circuit (the original article uses one qubit more, but the result is analogous) and fidelity or Hilbert-Schmidt test circuit (b). The operation U_+ [54] prepares the $2n$-qubit generalized Bell state $|\phi^+\rangle$ given in Eq. (62) starting from $|0\rangle^{\otimes 2n}$. See Ref. [50] for a general review of gradients based on Hadamard tests in the context of parametrized circuits.

implementing it with the SE. The forward derivative of a cost function C that depends on parameters $\boldsymbol{\theta} \in \mathbb{R}^N$ is defined as:

$$\nabla_{\boldsymbol{v}} C(\boldsymbol{\theta}) = \sum_{i=1}^{N} v_i \frac{\partial C(\boldsymbol{\theta})}{\partial \theta_i}, \tag{96}$$

for a vector $\boldsymbol{v} \in \mathbb{R}^N$. And the corresponding forward gradient can be obtained by sampling in random directions [101, 102]:

$$\nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}) = \mathbb{E}_{\boldsymbol{v}\sim Q}\left[\nabla_{\boldsymbol{v}} C(\boldsymbol{\theta})\boldsymbol{v}\right], \tag{97}$$

where $\boldsymbol{v} \sim Q$ is a random vector with mean zero and bounded variance sampled from a probability distribution Q. In Ref. [102] Q is assumed to be a Rademacher distribution: values sampled from this distribution can be implemented in the LCU register using random graph states [103]. Since $\boldsymbol{v}$ can have both positive and negative values, the circuits given in Fig. 3 are needed. Despite its straightforward circuit-centric implementation, the scaling of forward propagation on quantum computers is significantly worse than the corresponding scaling on classical deterministic computers [99]. We consider here the product ansatz circuit given in Eq. (93). We assume that each gate can be written as $V_i(\theta_i) = \exp\{\ i\theta_i H_i\} = \exp\Big\{\ i\theta_i \sum_{k=1}^{L} a_{ik} P_{ik}\Big\}$, where P_{ik} are arbitrary Pauli operators and a_{ik} the coefficients of the Hamiltonian of the ith gate in the Pauli basis [26]. The relevant scaling parameters of forward propagation are the number N of gates in the product ansatz and the number of Pauli coefficients K in each gate according the LCU approach – see Eq. (1). Therefore, the LCU circuit can estimate:

$$g_{\boldsymbol{v}} = \ 2\sum_{i=1}^{N} v_i \sum_{k=1}^{K} a_{ik}\, \mathrm{Im}\{\mathrm{tr}\{P_{ik}\rho_{CA}(\boldsymbol{\theta})\mathcal{O}\}\}, \tag{98}$$

$$\nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}) = \mathbb{E}_{v\sim Q}[g_{\boldsymbol{v}}], \tag{99}$$

for $\rho_{CA}(\boldsymbol{\theta}) = V_{CA}(\boldsymbol{\theta})\rho V^\dagger_{CA}(\boldsymbol{\theta})$. Eq. (98) is the forward derivative in Eq. (96) of the cost function in Eq. (4) that uses the circuit defined in Eq. (93). Due to propagation of uncertainty, we obtain a $O(\frac{N^2K^2}{2})$ scaling for the forward gradient, which is worse than the PSR and LCU-gradient scaling with respect to N. Eq. (98) can be estimated by combining appropriate parameter shifts and forward propagation – see Fig. (6) (b) –, i.e., with a sampling complexity of $O(\frac{N^2R^2}{2})$. An amplified version of forward propagation computed on the LCU circuit would reduce the sampling complexity of forward propagation to $O(\frac{NK}{})$. In this case, we reach the same scaling as the PSR approach in N and quadratically better scaling in L. This analysis, however, does not cover the total variance of the estimation problem due to the number of shots and the variance with respect to $\boldsymbol{v} \sim Q$. In fact, forward propagation suffers from slowdowns in the optimization due to a curse of dimensionality that limits its effectivity even in classical settings [102].

D. Most general LCU gradient: SU(d) gradient circuit

In this case the circuit is a single multi-qubit circuit parametrized by:

$$V(\boldsymbol{\theta}) = e^{\ iH(\boldsymbol{\theta})} = \exp\Bigg\{\ i\sum_{k=0}^{K}{}^{1}\theta_k H_k\Bigg\}. \tag{100}$$

The circuit that provides an unbiased estimator for the adjoint is given in Ref. [50] and Fig. 8. Now we need to combine this circuit with the LCU itself. In the case of a general SU(d) gate, Eq. (90) becomes [44]:

$$\nabla_{\boldsymbol{\theta}} C_{\mathrm{POTQ}}(\boldsymbol{\theta}) = \ \frac{1}{d}\,\mathrm{Re}\big\{\mathrm{Tr}\{\Omega(\boldsymbol{\theta})V(\boldsymbol{\theta})U^\dagger\}\big\}, \tag{101}$$

with the operator

$$\Omega(\boldsymbol{\theta}) = \sum_{l=0}^{\infty} \frac{(\ \ i)^l}{(l+1)!}\|\boldsymbol{\theta}\|_1^l \mathrm{ad}^l_{\bar{H}(\boldsymbol{\theta})}\ \ \bar{\nabla}_{\boldsymbol{\theta}} H(\boldsymbol{\theta})\big)\,, \tag{102}$$

where $\mathrm{ad}^l_{\bar{H}(\boldsymbol{\theta})}\ \bar{\nabla}_{\boldsymbol{\theta}} H(\boldsymbol{\theta})\big)$ denotes the nested commutator of lth order

$$\mathrm{ad}^l_{\bar{H}(\boldsymbol{\theta})}\ \bar{\nabla}_{\boldsymbol{\theta}} H(\boldsymbol{\theta})\big) = \underbrace{\big[\bar{H}(\boldsymbol{\theta}), ..., \big[\bar{H}(\boldsymbol{\theta}),\ \bar{\nabla}_{\boldsymbol{\theta}} H(\boldsymbol{\theta})\big]\big]}_{l+1 \text{ elements}},$$

and $\bar{H}$ is the renormalized version of the Hamiltonian, i.e., $\bar{H}(\boldsymbol{\theta}) = H(\boldsymbol{\theta})/\|\boldsymbol{\theta}\|_1$ with $\|\boldsymbol{\theta}\|_1 = \sum_{k=0}^{K\ \ 1} |\theta_k|$, as defined in Eq. (100). The expression $\bar{\nabla}_{\boldsymbol{\theta}} = \frac{1}{\|\boldsymbol{\theta}\|_1}\nabla_{\boldsymbol{\theta}}$ is a rescaled nabla operator. As such, the derivative of the unitary for $k = 0, ..., K\ \ 1$ given in Eq. (100) can be written as an infinite sum of matrices:

$$\frac{\partial}{\partial\theta_k} V(\boldsymbol{\theta}) = \sum_{l=0}^{\infty} \mathcal{W}_l(\boldsymbol{\theta}), \tag{103}$$

where $\mathcal{W}_l(\boldsymbol{\theta}) = \frac{(\ \ i)^l}{(l+1)!}\|\boldsymbol{\theta}\|_1^l \mathrm{ad}^l_{\bar{H}(\boldsymbol{\theta})}(H_k)\, V(\boldsymbol{\theta})$. If we combine the estimator circuit given in Eq. (55) with the circuit given in Ref. [50], which allows for the computation of nested commutators – see Fig. 8 – we can compute any unitary gradient with a full quantum circuit up to a desired approximation order. A second LCU register is necessary to encode $\bar{H}(\boldsymbol{\theta})$ coherently on the quantum circuit. By inserting the expansion given in Eq. (102) into our gradient Eq. (101), we obtain

$$\nabla_{\boldsymbol{\theta}} C_{\mathrm{POTQ}}(\boldsymbol{\theta}) = \sum_{l=0}^{\infty} T_l, \tag{104}$$

with

$$T_l = \frac{(\ \ 1)}{(l+1)!}\|\theta\|^l \frac{1}{d}\,\mathrm{Re}\Big\{(\ \ i)^l\,\mathrm{Tr}\big[V \mathrm{ad}^l_H(G) U^\dagger\big]\Big\}, \tag{105}$$

where $G = \frac{1}{\|\boldsymbol{\theta}\|_1}\nabla_{\boldsymbol{\theta}} H(\boldsymbol{\theta}) = (H_0, ..., H_{K\ \ 1})^{\mathrm{T}}$ is a vector of matrices and the trace and matrix multiplications operations in Eq. (105) are carried out in a vectorized form. For an Lth order expansion

$$\nabla_{\boldsymbol{\theta}} C_{\mathrm{POTQ}}(\boldsymbol{\theta}) \approx \sum_{l=0}^{L} T_l, \tag{106}$$

the remainder is then given by $R_L = \sum_{l=L+1}^{\infty} T_l$. Here we focus specifically on the POTQ cost function. However, similar expressions can be derived for arbitrary cost functions of the type given in Eq. (4), whose gradient is given by Eq. (88), as well as for the infidelity gradient given in Eq. (92). We now turn to specific examples of multi-parameter, multi-qubit gates where Eq. (106) can be implemented using LCU methods. In particular, we analyze the behaviour of the SU(d) gradient approximation given in Eq. (106) on control problems with fixed and time-dependent control parameters.

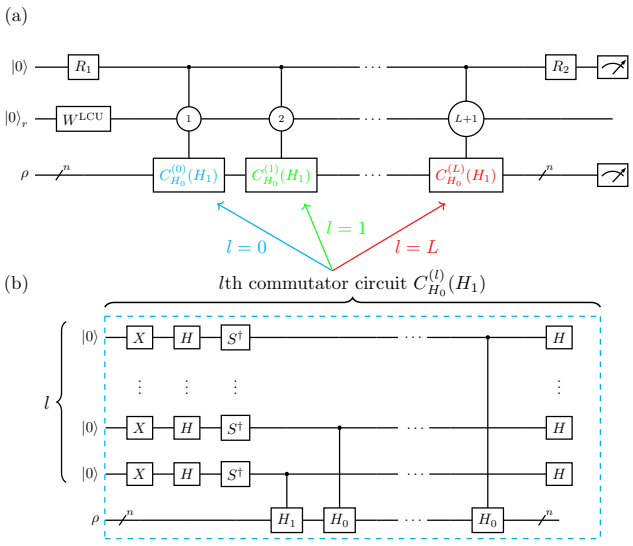


Figure 8. Representation of the SU(d) gradient circuit [44] using the nested commutator circuit from Ref. [50] for a control problem with Pauli gates H_0 and H_1. The circuit $C^{l)}_{H_0}(H_1)$ shown in (b) computes the expected value of the adjoint $\mathrm{ad}^l_{H_0}(H_1)$, i.e., the nested commutator term of order l. Each commutator term of order $l = 0$ (which is simply H_1), $l = 1$ (which is $[H_0, H_1]$), etc. is mapped to the corresponding entry in the LCU register with $r = \lceil \log(L) \rceil$ qubits. The LCU state is prepared by the unitary W^{LCU}. The circuit in (a) can be simplified further due to the redundancies in the composition of L multiply-controlled circuits of type (b), each one with a number of Hamiltonian terms that grows from 1 to L as in Eq. (106).

1. *Example*

We consider now a simplified unitary operation that is typical of control problems [104], i.e.:

$$V_c(\boldsymbol{\theta}) = \exp\{\ \ i(\theta_0 H_0 + \theta_1 H_1)\Delta t\} \tag{107}$$
$$V_0 = \exp\{\ \ i\theta_0 H_0 \Delta t\}, \tag{108}$$

where H_0 and H_1 are drift and control Hamiltonians respectively, θ_1 is a control parameter and Δt is a time interval. This is a special case of Eq. (100), where $\theta_k = 0$ for $k = 1, ..., K\ \ 1$ and an additional dynamical evolution of time Δt is applied. We define therefore the vector parameter $\boldsymbol{\theta} = (\theta_0, \theta_1)^T$. We can compute the control gradient of Eq. (107) by evaluating $\frac{\partial V_c(\boldsymbol{\theta})}{\partial\theta_1}\Big|_{\boldsymbol{\theta}=(1,0)^T}$ using Eq. (102):

$$\frac{\partial V_c(\boldsymbol{\theta})}{\partial\theta_1}\Big|_{\boldsymbol{\theta}=(1,0)^T} = \sum_{l=0}^{\infty} \mathcal{W}^c_l, \tag{109}$$

with $\mathcal{W}^c_l = \frac{(\ \ i\ \ t)^l}{(l+1)!}\mathrm{ad}^l_{H_0}(H_1)\, V_0$. The circuit given in Fig. 8 can be used to compute the derivative by implementing H_0 and H_1 using the LCU method if the Hamil-

tonians are given by sums of Pauli operators [26, 39].

2. LCU implementation

We first consider only two Pauli operators, i.e., $H_0 = P_0$ and $H_1 = P_1$. This helps us illustrate the structure of the quantum circuit that we generalize later to arbitrary Hamiltonians using standard LCU methods. In this case, the circuits given in Fig. 8 can be implemented using (multi-controlled) single-qubit rotations R_x, R_y and R_z and (multi-qubit) Toffoli gates.

The number of multi-controlled gates needed to implement a gradient approximation circuit of order L scales as $O[L^2 \log(L)]$. For general Hamiltonians H_0 and H_1 generated by K Pauli operators, the overall depth is $O[K^2 L^2 \log(K) \log(L)]$. Note as before that the factor of $\log(L)$ can be further brought down to $\log(\log(L))$ by using $\log(L)$ additional ancillas [64]. We observe further that we do not need all multi-controlled bits for the (multi-controlled) Pauli gates that implement the controlled operations $P_1, P_0, ..., P_0$: the first one, P_1, acts on all entries of the LCU state and does not need controls on the LCU register, but only on the first qubit. The other gates are multi-controlled on $L \quad 1$, $L \quad 2$,..., 1 entries, respectively, so they require only 1, 1, 2, ..., L multi-controlled bits each. This approach reduces the number of gates by a constant factor, even though the asymptotic circuit depth remains the same.

We now turn to the implementation of the LCU register. Each commutator circuit ($l = 0, ..., L$) in Fig. 8 (a) allows for the estimation of the (renormalized) mean value – see Eq. (15) in Ref. [50]:

$$g_l = \left(\frac{1}{2} \right)^l \mathrm{Re}\Big\{ (-i)^l \, \mathrm{tr}\Big\{ \mathrm{ad}^l_{H_1}(H_0) \rho_{V_0} \mathcal{O} \Big\} \Big\}, \tag{110}$$

where $\rho_{V_0} = V_0 \rho V_0^\dagger$ – see Eq. (108). As a consequence, the LCU circuit in Fig. 8 (a) estimates the quantity:

$$\mathcal{G}_L = \sum_{l=0}^{L} w_l(\Delta t) g_l, \tag{111}$$

where $w_l(\Delta t)$ are appropriate Δt-dependent LCU weights that reproduce the coefficients given in Eq. (109). By comparing Eq. (110) with Eq. (109), we see that the state we need to prepare in the LCU register is nothing else than a coherent state with parameter equal to $\sqrt{2\Delta t}$ implemented by the operator W^{LCU} shown in Fig. 8 (a):

$$W^{\mathrm{LCU}}(\Delta t) \, |0\rangle_L = \left| \sqrt{2\Delta t} \right\rangle, \tag{112}$$

$$\left| \sqrt{2\Delta t} \right\rangle = e^{-|\quad t|} \sum_{l=0}^{\infty} \frac{(\sqrt{2\Delta t})^l}{\sqrt{(l)!}} |l\rangle \, . \tag{113}$$

The positive weights are given by $w_l(\Delta t) = \frac{(2 \quad t)^l}{l!} e^{\quad 2| \quad t|}$. We observe that compared to Eqs. (102) and (109), using this state would lead to a $l!$ coefficient rather than $(l+1)!$. The derivation of this gradient encoding scheme is provided in Appendix B 1. A generalization to the case in which the gradient is not computed at $\theta_1 = 0$ and the Hamiltonians are not just n-qubit Pauli operators (for example because there are multiple control Hamiltonians that are kept fixed while performing the derivative with respect to θ_1) can be obtained using the LCU approach to implement the drift Hamiltonian and by encoding coefficients proportional to $(\Delta t)^l$ in the LCU register, see Eq. (112).

3. Expectation value of the truncation error

We want to analyze the convergence behaviour of the gradient approximation in Eq. (106) for a quantum gate of the type given in Eq. (107). In particular, we consider here the gradient of the POTQ test given in Eq. (90). The square truncation error R_L is given by:

$$R_L^2 = \left(\sum_{l=L+1}^{\infty} T_l \right)^2, \tag{114}$$

where T_l is given in Eq. (105). Here we use R_L to describe the square truncation error of the POTQ gradient, while $\mathcal{R}_L$ is defined analogously for the gradient of the infidelity. We can estimate the behavior of the expected value of the truncation error as a function of a random Hamiltonian generator, in order to potentially determine the truncation length L that is needed to achieve a given target precision in the gradient approximation. In the following section, we specifically consider the framework of quantum control to understand the behaviour of the SU(d) gradient from the perspective of the LCU implementation. We employ random n-qubit drift and control Hamiltonians H_0 and H_1 that are sampled from the Gaussian Unitary Ensemble (GUE) [105], respectively. We also employ random target unitaries U, generated with a QR decomposition. The gradient is evaluated at control value equal to zero ($\theta_1 = 0$) to facilitate the simulation, but this method can be applied to arbitrary control problems, since we can always redefine the drift Hamiltonian to contain control Hamiltonians, whose pulse parameters are not varied.

The formal derivation of the exact average of the square truncation error with respect to Hermitian matrices H_0 and H_1 sampled from the GUE is given in Appendix C. The asymptotic behaviour of the squared truncation error is given by:

$$\mathbb{E}[R_L^2] \sim O\left(\frac{\Lambda_1^2}{d^2} \left(\|\boldsymbol{\theta}\| \Lambda_0 \right)^{2(L+1)} \right), \tag{115}$$

where $d = 2^n$, Λ_1 and Λ_0 are the Dyson coefficients of the distributions of H_0 and H_1, and $\boldsymbol{\theta}$ is the parameter vector of the control problem as defined in Eq. (107). The value $\mathbb{E}[R_L^2]$ can also provide us with a bound for the av-

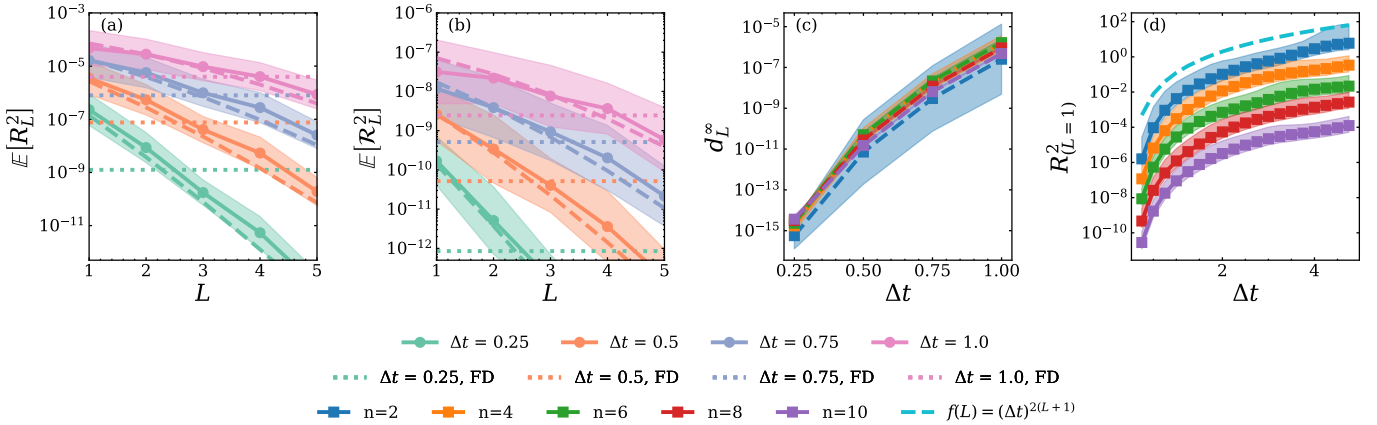


Figure 9. Representation of the convergence of the SU(d) derivative given in Eq. (106), using the parameters defined in Eq. (109) and as such represented using LCU. We use here the squared approximation error R_L^2 – POTQ, Eq. (90) – and $\mathcal{R}_L^2$ – infidelity, Eq. (92) –, so that we can compare it easily with our theoretical predictions. Numerical simulations of the truncation error are shown as straight lines and the theoretical prediction as dashed lines, with different colours representing different t. The values are averaged over 50 random Hamiltonians H_0 and H_1 sampled from the GUE [105] with Dyson coefficient $\Lambda_0 = \Lambda_1 = 1$. (a), (b) show the gradient approximation of the unitary of a 6-qubit Hamiltonian for the POTQ cost function and the fidelity cost function, respectively, and for different values of θ as a function of the order of approximation, n, in the expansion L. Dotted lines represent the symmetric $O(1)$ finite-difference (FD) approach with a shift of $\delta = 0.75$. The $O(1)$ approach is the best choice when sampling classically from quantum circuits due to the scaling of the variance with δ – $O(1/\delta^2)$, see also Ref. [44]. (c) shows the maximum norm of the difference between the true gradient computed with JAX and the gradient approximation as a function of θ – see Eq. (116). (d) shows the precision in the approximation of the first order in the expansion for the POTQ cost function, which for this case corresponds to terms in the LCU that are proportional to H_1 ($L = 0$) and $[H_0, H_1]$ ($L = 1$). We see that for several cost functions, a relatively short circuit already produces a high-quality approximation of the SU(d) derivative.

erage error itself since $|\mathbb{E}[R_L]| \leq \sqrt{\mathbb{E}[R_L^2]}$. We simulate the result by computing the gradient approximation in Eq. (106) with the problem structure given in Eq. (109), i.e., with $\theta_1 = 0$ and $\theta_0 = 1$. We average the approximation error over 50 random Hamiltonians H_0 and H_1 sampled from the GUE with $\Lambda_0 = \Lambda_1 = 1$ (see Ref. [105] and Appendix D) for different numbers of qubits and values of the parameter θ. The results of our simulation of the SU(d) gradient are shown in Fig. 9 for different test values of θ. The gradient used for comparison is obtained with J X [106]. Full lines and dashed lines represent the simulated average of R_L^2 – POTQ gradient – (a) and $\mathcal{R}_L^2$ – infidelity gradient - (b) up to an order L. Dotted lines represent the symmetric finite-difference (FD) approach with a shift $\delta = 0.75$ – see also Ref. [44]. We see that the theoretical predictions match the average value of the squared truncation error within the standard deviation represented by the shaded regions and that the approximate gradient to order L quickly reaches the precision of a symmetric $O(1)$ FD approach, which is commonly used when sampling classically from quantum circuits [44].

By setting an appropriate maximum expansion index L, we can study the convergence of the gradient approximation. The convergence with respect to L is particularly relevant for the LCU implementation, because, if the integer L needed to reach a gradient precision δ is reasonably small, so will be the depth of the corresponding LCU-based quantum circuit.

In Fig. 9 (c) we show the maximum norm of the unitary gradient approximation:

$$d_L^\infty = \left\| \frac{\partial V(\boldsymbol{\theta})}{\partial \theta_1}\Big|_{\boldsymbol{\theta}=(1,0)^T} \quad \mathcal{W}_L^c \right\|_\infty , \tag{116}$$

for the maximum index considered, i.e., for us $L = 14$. as a function of the parmeter Δt. This plot helps us visualize the precision of the unitary gradient itself if compared to the precision of the gradient of the two quantum cost functions given in Fig. 9 (a) and (b). In Fig. 9 (d) we see as an example the change of R_L^2 for $L = 1$ as a function of θ: the behaviour of the function reflects the predicted dependence of $\mathbb{E}[R_L^2]$ from θ, i.e., $O[\theta^{2(L+1)}]$, as shown in Eq. (D14) and in more detail in Eq. (D13).

E. Gradients of quantum dynamics

In the case of pulse-level optimization, the problem usually has a bilinear structure with a drift Hamiltonian H_0 and control Hamiltonians $H_1, ..., H_{K\ 1}$ – see also Eq. (100) – with corresponding time-dependent control fields $u_1^{\phi_1}(t), ..., u_{K\ 1}^{\phi_{K\ 1}}(t)$. W.l.o.g., we can assume that these control fields are each parametrized independently by real vectors $\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_{K\ 1}$. The gradient of the unitary evolution of $H_{\boldsymbol{\phi}}(t) = H_0 + \sum_{k=1}^{K\ 1} u_k^{\phi_k}(t) H_k$, where

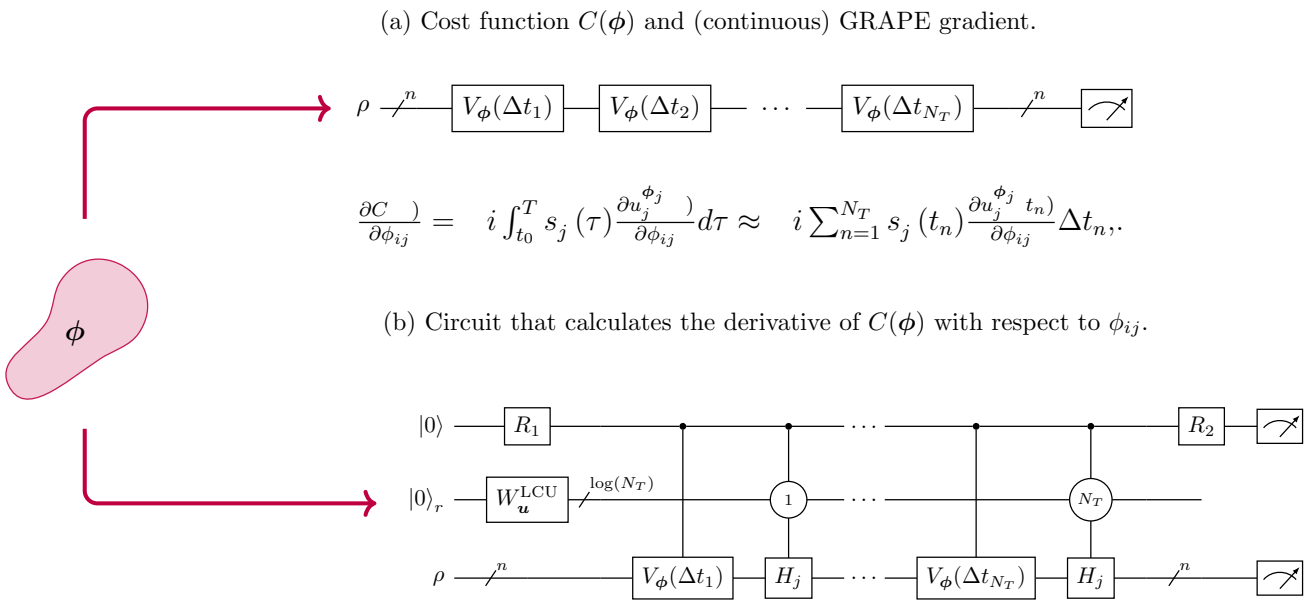


Figure 10. Schematic representation of the LCU-GRAPE circuit for a general quantum cost function optimized with control pulses. The cost function circuit is pictured in (a). Compared to the previous example, i.e., Fig. 6, the circuit uses the unitary evolution $V_{\boldsymbol{\phi}}(t_0,T) = \mathcal{T}\exp\Big\{ i\int_{t_0}^{T} H_{\boldsymbol{\phi}}(\tau)d\tau\Big\}$, where $\mathcal{T}$ is the time-ordering operator and $H(t) = H_0 + \sum_{j=1}^{K} u_j^{\boldsymbol{\phi}_j}(t)H_j$ the Hamiltonian of a bilinear single-control problem parametrized by a vector $\boldsymbol{\phi}$. The LCU-GRAPE circuit shown in (b) is an estimator of the (renormalized) gradient given in (a) and therefore implements, on a quantum circuit, the gradient sampling procedure outlined in Refs. [52, 53]. The operation $W_{\boldsymbol{u}}^{\mathrm{LCU}}$ loads the LCU register that uses $r = \lceil \log(N_T)\rceil$ qubits (we consider again for the sake of simplicity only linear combinations of positive coefficients) with coefficients proportional to $\frac{\partial u_j\ t_i)}{\partial\ \ ij}\ \ t_i$ and can therefore The derivatives with respect to the single time-sliced control pulses can be implemented in different way, either with the SU(d) gradient circuit – see Fig. (8) and the procedure outlined in Section IV D, or with another type of gate gradient estimation method, depending on the structure of the spectrum of the gate element $V_{\boldsymbol{\phi}}(t_i), i = 1, ..., N_T$ itself. We also see that this gradient resembles the structure of a forward derivative, because each time-sliced gate depends on the same set of parameters $\boldsymbol{\phi}$, so the gradient results a sum of each contribution. Overall, this procedure gives a $O(T^2/\epsilon^2)$ and $O(T/\epsilon)$ sampling complexity for non-amplified and amplified estimates, respectively.

$\boldsymbol{\phi} = (\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_{K\ \ 1})^{\mathrm{T}}$, that is of $V_{\boldsymbol{\phi}}(t) = \mathcal{T}e^{\ i\int_{t_0}^{T} H(\tau)d\tau}$, where $\mathcal{T}$ is the time-ordering operator, from the initial time $t = t_0$ to the final time $t = T$ with respect to the parameters, is given by [100]:

$$\frac{\partial V_{\boldsymbol{\phi}}}{\partial \phi_{ij}} = \ i\int_{t=t_0}^{T} V_{\boldsymbol{\phi}}(t_0,\tau)H_j\frac{\partial u_j^{\boldsymbol{\phi}_j}(t)}{\partial \phi_{ij}}V_{\boldsymbol{\phi}}(\tau,T)d\tau. \quad (117)$$

For a control cost function of the type of Eq. (4), i.e.:

$$C(\boldsymbol{\phi}) = \mathrm{tr}\Big\{V_{\boldsymbol{\phi}}(t_0,T)\rho V_{\boldsymbol{\phi}}^{\dagger}(t_0,T)\mathcal{O}\Big\}, \quad (118)$$

the gradient is given by – see Appendix B 2 and Refs. [52, 53, 107]:

$$\frac{\partial C(\boldsymbol{\phi})}{\partial \phi_{ij}} = \ i\int_{t=t_0}^{T} s_j^{\boldsymbol{\phi}}(\tau)\frac{\partial u_j^{\boldsymbol{\phi}_j}(t)}{\partial \phi_{ij}}d\tau, \quad (119)$$

$$s_j^{\boldsymbol{\phi}}(\tau) = \mathrm{tr}\{\rho(T)[H_j(t_0,\tau),\mathcal{O}]\}. \quad (120)$$

Eq. (119) resembles the implementation of the analog LCU algorithm [1]. Full quantum control gradients are usually too challenging to implement for basic qubit optimization on near-term devices. However, quantum control has been suggested as a possible ansatz to optimize variational quantum algorithms [108]. In these implementations, the use of LCU algorithms can be considered beneficial. Usually, the gradient given in Eq. (119) is computed for a discretized time dynamics, i.e., where $T = \sum_{n=1}^{N_T}\Delta t_n$, in which case we have:

$$\frac{\partial C(\boldsymbol{\phi})}{\partial \phi_{ij}} \approx \ i\sum_{n=1}^{N_T} s_j^{\boldsymbol{\phi}}(t_n)\frac{\partial u_j^{\boldsymbol{\phi}_j}(t_n)}{\partial \phi_{ij}}\Delta t_n, \quad (121)$$

which is the discretized version of the GRAPE gradient [100]. We refer to the circuit estimator for this quantity as LCU-GRAPE. Assuming the variance of the control operators is bounded, the sampling complexity of this approach with a classical sampling process scales as $O(T^2/\epsilon^2)$ and therefore as $O(T/\epsilon)$ for an amplified sampling. The circuit given in Fig. 10 (LCU-GRAPE), when

combined with amplitude estimation, potentially allows for quadratic speed up in the evaluation of GRAPE-like gradients. However, the depth of such LCU circuits and the additional ancillas makes them impractical on near-term quantum devices and in the control of experimental qubits for gate synthesis and state preparation [109]. On the other hand, Ref. [52] draws connections between (Ordinary Differential Equation Networks) ODENets [110] and quantum dynamics typical of control systems. The analysis contained therein provides us with training methods for ODENets on quantum systems using Eq. (119) and stochastic parameter-shift rules [45]. As we discussed above, classical estimation of such a derivative is quadratic in time – $O(T^2/\epsilon^2)$ – both in the case of SE and LCU, while amplified LCU-GRAPE can potentially reach $O(T/\epsilon)$. For large parametrized quantum models, this represents a significant speedup. While its relevance is limited for NISQ circuits, it will most probably increase in the next years as logical error rates continue to decrease.

V. SUMMARY AND CONCLUSION

In this work, we thoroughly analyze the sampling complexity of different LCU estimators in different contexts. In the first part, we focused on reviewing the approaches to LCU sampling and estimation compared to SE. The complexity of estimating observables using LCU and SE without any quantum AE routine is the same, i.e., $O(L^2/\epsilon^2)$ [1]. In the case of AE, the sampling complexity of the LCU estimator reduces to $O(L/\epsilon)$, which is considerably faster than the $O(L\sqrt{L}/\epsilon)$ scaling of SE. We also draw connections between classical probability theory, circuit sampling and DQC1 by considering the SA-LCU approach presented in Ref. [1]. In the second part of the work, we focus on the specific application to gradient estimation, and more specifically we discuss how LCU can be used to represent the gradient of arbitrary cost functions that depend on unitary evolution operators, such as the SU(d) gradient introduced in Ref. [44] and GRAPE-like gradients (LCU-GRAPE). We present and discuss in particular the circuits that allow for the estimation of SU(d) gradients and control gradients, and analyze the convergence properties of the gradient approximation both from a numerical and an analytical perspective using concepts from random matrix theory. These results are relevant for advanced implementations of gradient-based optimization on quantum hardware that make use of either non-standard multi-qubit gates or circuit ansätze based on quantum optimal control theory.

VI. CODE AND DATA AVAILABILITY

The code and data used for this work are available at Ref. [111].

ACKNOWLEDGMENTS

This work was supported by AIDAS, The European Joint Virtual Lab, an initiative of Forschungszentrum Jülich (FZJ) and the French Alternative Energies and Atomic Energy Commission (CEA), by the German Federal Ministry of Education and Research (BMBF), project QSolid, Grant No. 13N16149, by the German Research Foundation (DFG) under Germany's Excellence Strategy – Cluster of Excellence Matter and Light for Quantum Computing (ML4Q) EXC 2004/1 – 390534769 and by the Jülich Supercomputing Center (JSC). We acknowledge funding from the Horizon Europe program (HORIZON-CL4-2021-DIGITAL-EMERGING-02-10) via the project 101080085 (QCFD) and by HORIZON-CL4-2022-QUANTUM-01-SGA Project under Grant 101113946 OpenSuperQPlus10. We are grateful to Markus Heinrich, Manuel Guatto, Robert Zeier and Roberto Gargiulo for the stimulating discussions. Simulations were realized in PYTHON using the libraries QISKIT [67], JAX [106], QCLIB [112] and NUMBA [113].

[1] S. Chakraborty, Implementing any Linear Combination of Unitaries on Intermediate-term Quantum Computers, Quantum **8**, 1496 (2024).
[2] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, W.-K. Mok, S. Sim, L.-C. Kwek, and A. Aspuru-Guzik, Noisy intermediate-scale quantum algorithms, Reviews of Modern Physics **94** (2022).
[3] G. Kochenberger, J.-K. Hao, F. Glover, M. Lewis, Z. Lü, H. Wang, and Y. Wang, The unconstrained binary quadratic programming problem: a survey, Journal of Combinatorial Optimization **28**, 58 (2014).
[4] D. Wierichs, J. Izaac, C. Wang, *et al.*, General parameter-shift rules for quantum gradients, Quantum **6**, 677 (2022).
[5] J. Bowles, D. Wierichs, and C.-Y. Park, Backpropagation scaling in parameterised quantum circuits (2024), arXiv:2306.14962.
[6] A. Abbas, R. King, H.-Y. Huang, W. J. Huggins, R. Movassagh, D. Gilboa, and J. R. McClean, On quantum backpropagation, information reuse, and cheating measurement collapse (2023), arXiv:2305.13362.
[7] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2010).
[8] P. Krantz, M. Kjaergaard, F. Yan, *et al.*, A quantum engineer's guide to superconducting qubits, Applied Physics Reviews **6**, 021318 (2019).

[9] A. Blais, A. L. Grimsmo, S. M. Girvin, *et al.*, Circuit quantum electrodynamics, Rev. Mod. Phys. **93**, 025005 (2021).

[10] H. Häffner, C. Roos, and R. Blatt, Quantum computing with trapped ions, Physics Reports **469**, 155 (2008).

[11] M. Saffman, T. G. Walker, and K. Mølmer, Quantum information with rydberg atoms, Rev. Mod. Phys. **82**, 2313 (2010).

[12] M. DeCross, E. Chertkov, M. Kohagen, and M. Foss-Feig, Qubit-reuse compilation with mid-circuit measurement and reset, Phys. Rev. X **13**, 041057 (2023).

[13] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm (2014), arXiv:1411.4028.

[14] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, Variational quantum algorithms, Nature Reviews Physics **3**, 625 (2021).

[15] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nature Communications **5**, 4213 (2014).

[16] H.-Y. Huang, R. Kueng, and J. Preskill, Predicting many properties of a quantum system from very few measurements, Nature Physics **16**, 1050 (2020).

[17] H.-Y. Huang, R. Kueng, and J. Preskill, Information-theoretic bounds on quantum advantage in machine learning, Physical Review Letters **126** (2021).

[18] W. J. Huggins, K. Wan, J. McClean, *et al.*, Nearly optimal quantum algorithm for estimating multiple expectation values, Phys. Rev. Lett. **129**, 240501 (2022).

[19] K. Wada, N. Yamamoto, and N. Yoshioka, Heisenberg-limited adaptive gradient estimation for multiple observables, PRX Quantum **6** (2025).

[20] R. Babbush, D. W. Berry, and H. Neven, Quantum simulation of the sachdev-ye-kitaev model by asymmetric qubitization, Phys. Rev. A **99**, 040301 (2019).

[21] D. Wecker, M. B. Hastings, and M. Troyer, Progress towards practical quantum variational algorithms, Phys. Rev. A **92**, 042303 (2015).

[22] N. C. Rubin, R. Babbush, and J. McClean, Application of fermionic marginal constraints to hybrid quantum algorithms, New Journal of Physics **20**, 053020 (2018).

[23] M. Consiglio, Variational quantum algorithms for many-body systems (2025), arXiv:2502.11985.

[24] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, Quantum machine learning, Nature **549**, 195 (2017).

[25] S. Jerbi, L. J. Fiderer, H. Poulsen Nautrup, J. M. Kübler, H. J. Briegel, and V. Dunjko, Quantum machine learning beyond kernel methods, Nature Communications **14**, 517 (2023).

[26] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Phys. Rev. A **99**, 032331 (2019).

[27] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Circuit-centric quantum classifiers, Physical Review A **101** (2020).

[28] L. Schatzki, A. Arrasmith, P. J. Coles, *et al.*, Entangled datasets for quantum machine learning (2021), arXiv:2109.03400.

[29] F. Preti and J. Z. Bernád, Statistical evaluation and optimization of entanglement purification protocols, Phys. Rev. A **110**, 022619 (2024).

[30] S. G. Schirmer, F. C. Langbein, C. A. Weidner, and E. Jonckheere, Robust control performance for open quantum systems, IEEE Transactions on Automatic Control **67**, 6012 (2022).

[31] F. Preti, T. Calarco, and F. Motzoi, Continuous quantum gate sets and pulse-class meta-optimization, PRX Quantum **3**, 040311 (2022).

[32] A. Cervera-Lierta, J. S. Kottmann, and A. Aspuru-Guzik, Meta-variational quantum eigensolver: Learning energy profiles of parameterized hamiltonians for quantum simulation, PRX Quantum **2**, 020329 (2021).

[33] N. Oshnik, P. Rembold, T. Calarco, *et al.*, Robust magnetometry with single nitrogen-vacancy centers via two-step optimization, Phys. Rev. A **106**, 013107 (2022).

[34] M. Dalgaard, C. A. Weidner, and F. Motzoi, Dynamical uncertainty propagation with noisy quantum parameters, Phys. Rev. Lett. **128**, 150503 (2022).

[35] F. Preti, T. Calarco, J. M. Torres, *et al.*, Optimal two-qubit gates in recurrence protocols of entanglement purification, Phys. Rev. A **106**, 022422 (2022).

[36] J. Li, X. Yang, X. Peng, *et al.*, Hybrid quantum-classical approach to quantum optimal control, Phys. Rev. Lett. **118**, 150503 (2017).

[37] D. Jaksch, P. Givi, A. J. Daley, and T. Rung, Variational quantum algorithms for computational fluid dynamics, AIAA Journal **61**, 1885–1894 (2023).

[38] R. Somma, G. Ortiz, J. E. Gubernatis, *et al.*, Simulating physical phenomena by quantum networks, Phys. Rev. A **65**, 042323 (2002).

[39] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, Quantum Information and Computation **12**, 901 (2012).

[40] A. M. Childs, R. Kothari, and R. D. Somma, Quantum algorithm for systems of linear equations with exponentially improved dependence on precision, SIAM Journal on Computing **46**, 1920 (2017).

[41] A. N. Chowdhury and R. D. Somma, Quantum algorithms for gibbs sampling and hitting-time estimation, Quantum Info. Comput. **17**, 41–64 (2017).

[42] Z. Holmes, N. Coble, A. T. Sornborger, *et al.*, On nonlinear transformations in quantum computation (2023), arXiv:2112.12307.

[43] K. Wada, N. Yamamoto, and N. Yoshioka, Heisenberg-limited adaptive gradient estimation for multiple observables, PRX Quantum **6**, 020308 (2025).

[44] R. Wiersema, D. Lewis, D. Wierichs, *et al.*, Here comes the SU(N): multivariate quantum gates and gradients, Quantum **8**, 1275 (2024).

[45] L. Banchi and G. E. Crooks, Measuring Analytic Gradients of General Quantum Evolution with the Stochastic Parameter Shift Rule, Quantum **5**, 386 (2021).

[46] G. E. Crooks, Gradients of parameterized quantum gates using the parameter-shift rule and gate decomposition (2019), arXiv:1905.13311 [quant-ph].

[47] A. F. Izmaylov, R. A. Lang, and T.-C. Yen, Analytic gradients in variational quantum algorithms: Algebraic extensions of the parameter-shift rule to general unitary transformations, Phys. Rev. A **104**, 062443 (2021).

[48] O. Kyriienko and V. E. Elfving, Generalized quantum circuit differentiation rules, Phys. Rev. A **104**, 052417 (2021).

[49] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309

(2018).
[50] D. Li, D. Dulal, M. Ohorodnikov, H. Wang, and Y. Ding, Efficient quantum gradient and higher-order derivative estimation via generalized hadamard test (2024), arXiv:2408.05406.
[51] A. Gilyén, T. Kiss, and I. Jex, Exponential sensitivity and its cost in quantum physics, Sci. Rep. **6**, 20076 (2016).
[52] J. Leng, Y. Peng, Y.-L. Qiao, *et al.*, Differentiable analog quantum computing for optimization and control, in *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22 (Curran Associates Inc., Red Hook, NY, USA, 2024).
[53] K. Kottmann and N. Killoran, Evaluating analytic gradients of pulse programs on quantum computers (2023), arXiv:2309.16756.
[54] S. Khatri, R. LaRose, A. Poremba, *et al.*, Quantum-assisted quantum compiling, Quantum **3**, 140 (2019).
[55] R. Iten, R. Colbeck, I. Kukuljan, J. Home, and M. Christandl, Quantum circuits for isometries, Physical Review A **93** (2016).
[56] A. J. da Silva and D. K. Park, Linear-depth quantum circuits for multiqubit controlled gates, Physical Review A **106** (2022).
[57] A. Barenco, C. H. Bennett, R. Cleve, *et al.*, Elementary gates for quantum computation, Phys. Rev. A **52**, 3457–3467 (1995).
[58] D. K. Park, I. Sinayskiy, M. Fingerhuth, *et al.*, Parallel quantum trajectories via forking for sampling without redundancy, New Journal of Physics **21**, 083024 (2019).
[59] F. Preti, M. Schilling, S. Jerbi, *et al.*, Hybrid discrete-continuous compilation of trapped-ion quantum circuits with deep reinforcement learning, Quantum **8**, 1343 (2024).
[60] S. Mohamed, M. Rosca, M. Figurnov, *et al.*, Monte carlo gradient estimation in machine learning, Journal of Machine Learning Research **21**, 1 (2020).
[61] A. Sequeira, L. P. Santos, and L. S. Barbosa, Policy gradients using variational quantum circuits, Quantum Machine Intelligence **5**, 18 (2023).
[62] M. Hayashi, *Quantum Information Theory: Mathematical Foundation* (Springer Berlin Heidelberg, 2017).
[63] L. Wasserman, *All of statistics : a concise course in statistical inference* (Springer, New York, 2010).
[64] F. Motzoi, M. P. Kaicher, and F. K. Wilhelm, Linear and logarithmic time compositions of quantum many-body operators, Phys. Rev. Lett. **119**, 160503 (2017).
[65] I. F. Araujo, D. K. Park, F. Petruccione, and A. J. da Silva, A divide-and-conquer algorithm for quantum state preparation, Scientific Reports **11**, 6329 (2021).
[66] P. S. Bullen, *Handbook of Means and Their Inequalities* (Springer Netherlands, 2003).
[67] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross, B. R. Johnson, and J. M. Gambetta, Quantum computing with Qiskit (2024), arXiv:2405.08810.
[68] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, Contemporary Mathematics **305**, 53 (2002).
[69] Y. Suzuki, S. Uno, R. Raymond, *et al.*, Amplitude estimation without phase estimation, Quantum Information Processing **19** (2020).
[70] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, Iterative quantum amplitude estimation, npj Quantum Information **7**, 52 (2021).
[71] S. M. Kay, *Fundamentals of statistical processing, volume I* (Prentice Hall, Philadelphia, PA, 1993).
[72] L. K. Grover, A fast quantum mechanical algorithm for database search, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96 (Association for Computing Machinery, New York, NY, USA, 1996) p. 212–219.
[73] J. Cui, P. J. S. de Brouwer, S. Herbert, P. Intallura, C. Kargi, G. Korpas, A. Krajenbrink, W. Shoosmith, I. Williams, and B. Zheng, Quantum monte carlo integration for simulation-based optimisation (2024), arXiv:2410.03926.
[74] G. Nannicini, Quantum algorithms for optimizers (2025), arXiv:2408.07086.
[75] E. Knill and R. Laflamme, Power of one bit of quantum information, Phys. Rev. Lett. **81**, 5672 (1998).
[76] K. Plekhanov, M. Rosenkranz, M. Fiorentini, *et al.*, Variational quantum amplitude estimation, Quantum **6**, 670 (2022).
[77] P. Shyamsundar, Non-boolean quantum amplitude amplification and quantum mean estimation (2021), arXiv:2102.04975.
[78] N. Stamatopoulos, D. J. Egger, Y. Sun, *et al.*, Option Pricing using Quantum Computers, Quantum **4**, 291 (2020).
[79] M. Abramowitz and I. A. Stegun, Handbook of mathematical functions with formulas, graphs, and mathematical tables (1964).
[80] K. Wada, K. Fukuchi, and N. Yamamoto, Quantum-enhanced mean value estimation via adaptive measurement, Quantum **8**, 1463 (2024).
[81] A. A. Mele, Introduction to Haar Measure Tools in Quantum Information: A Beginner's Tutorial, Quantum **8**, 1340 (2024).
[82] J. Soch *et al.*, Statproofbook/statproofbook.github.io: The book of statistical proofs (2024).
[83] M. Heinrich, M. Kliesch, and I. Roth, Randomized benchmarking with random quantum circuits (2023), arXiv:2212.06181.
[84] H.-Y. Huang, R. Kueng, and J. Preskill, Predicting many properties of a quantum system from very few measurements, Nature Physics **16**, 1050 (2020).
[85] A. W. Harrow, A. Hassidim, and S. Lloyd, Quantum algorithm for linear systems of equations, Phys. Rev. Lett. **103**, 150502 (2009).
[86] S. Jerbi, L. J. Fiderer, H. P. Nautrup, J. M. Kübler, H. J. Briegel, and V. Dunjko, Quantum machine learning beyond kernel methods, Nature Communications **14** (2023).
[87] K. Oshio, Y. Suzuki, K. Wada, K. Hisanaga, S. Uno, and N. Yamamoto, Adaptive measurement strategy for noisy quantum amplitude estimation with variational quantum circuits, Phys. Rev. A **110**, 062423 (2024).
[88] F. Motzoi, J. M. Gambetta, S. T. Merkel, *et al.*, Optimal control methods for rapidly time-varying hamiltonians, Phys. Rev. A **84**, 022307 (2011).
[89] M. Dalgaard, F. Motzoi, J. H. M. Jensen, *et al.*, Hessian-based optimization of constrained quantum control, Phys. Rev. A **102**, 042612 (2020).
[90] T. Caneva, T. Calarco, and S. Montangero, Chopped random-basis quantum optimization, Phys. Rev. A **84**,

022326 (2011).
[91] J. A. Nelder and R. Mead, A Simplex Method for Function Minimization, The Computer Journal **7**, 308 (1965).
[92] M. Ostaszewski, E. Grant, and M. Benedetti, Structure optimization for parameterized quantum circuits, Quantum **5**, 391 (2021).
[93] R. A. Fisher, Iris, UCI Machine Learning Repository (1988).
[94] L. Deng, The mnist database of handwritten digit images for machine learning research, IEEE Signal Processing Magazine **29**, 141 (2012).
[95] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, 1st ed. (Springer, 2007).
[96] Y. H. Wang, On the number of successes in independent trials, Statistica Sinica **3**, 295 (1993).
[97] L. Bittel, J. Watty, and M. Kliesch, Fast gradient estimation for variational quantum algorithms (2022), arXiv:2210.06484.
[98] A. Mari, T. R. Bromley, and N. Killoran, Estimating the gradient and higher-order derivatives on quantum hardware, Phys. Rev. A **103**, 012405 (2021).
[99] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, *et al.*, Automatic differentiation in machine learning: a survey, Journal of Machine Learning Research **18**, 1 (2018).
[100] N. Khaneja, T. Reiss, C. Kehlet, *et al.*, Optimal control of coupled spin dynamics: design of nmr pulse sequences by gradient ascent algorithms, Journal of Magnetic Resonance **172**, 296 (2005).
[101] A. G. Baydin, B. A. Pearlmutter, D. Syme, *et al.*, Gradients without backpropagation (2022), arXiv:2202.08587.
[102] G. Belouze, Optimization without backpropagation (2022), arXiv:2209.06302.
[103] J.-Y. Wu, M. Rossi, H. Kampermann, S. Severini, L. C. Kwek, C. Macchiavello, and D. Bruß, Randomized graph states and their entanglement properties, Physical Review A **89** (2014).
[104] S. Machnes, U. Sander, S. J. Glaser, *et al.*, Comparing, optimizing, and benchmarking quantum-control algorithms in a unifying programming framework, Phys. Rev. A **84**, 022305 (2011).
[105] T. Tao, *Topics in random matrix theory*, Graduate studies in mathematics (American Mathematical Society, Providence, RI, 2012).
[106] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, JAX: composable transformations of Python+NumPy programs (2018).
[107] M. H. Goerz, *Optimizing Robust Quantum Gates in Open Quantum Systems*, Ph.D. thesis, Kassel, Universität Kassel, Fachbereich Mathematik und Naturwissenschaften (2015).
[108] A. B. Magann, C. Arenz, M. D. Grace, *et al.*, From pulses to circuits and back again: A quantum optimal control perspective on variational quantum algorithms, PRX Quantum **2**, 010101 (2021).
[109] J. Kelly, R. Barends, B. Campbell, *et al.*, Optimal quantum control using randomized benchmarking, Phys. Rev. Lett. **112**, 240504 (2014).
[110] R. T. Chen, Y. Rubanova, J. Bettencourt, *et al.*, Neural ordinary differential equations, Advances in neural information processing systems **31** (2018).
[111] F. Preti, `https://github.com/franz3105/GPVQuEst` (2025).
[112] I. F. Araujo, I. C. S. Araújo, L. D. da Silva, C. Blank, and A. J. da Silva, Quantum computing library (2023).
[113] S. K. Lam, A. Pitrou, and S. Seibert, Numba: A llvm-based python jit compiler, in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC* (2015) pp. 1–6.
[114] J. A. Jones, Quantum computing with nmr, Progress in Nuclear Magnetic Resonance Spectroscopy **59**, 91 (2011).
[115] P. W. Shor and S. P. Jordan, Estimating jones polynomials is a complete problem for one clean qubit, Quantum Info. Comput. **8**, 681–714 (2008).
[116] D. Shepherd, Computation with unitaries and one pure qubit (2006), arXiv:quant-ph/0608132.
[117] D. Poulin, R. Blume-Kohout, R. Laflamme, and H. Ollivier, Exponential speedup with a single bit of quantum information: Measuring the average fidelity decay, Phys. Rev. Lett. **92**, 177906 (2004).
[118] S. Khairy, R. Shaydulin, L. Cincio, *et al.*, Learning to optimize variational quantum circuits to solve combinatorial problems, Proceedings of the AAAI Conference on Artificial Intelligence **34**, 2367 (2020).
[119] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subasi, L. Cincio, and P. J. Coles, Variational quantum linear solver, Quantum **7**, 1188 (2023).
[120] T. Morimae, K. Fujii, and J. F. Fitzsimons, Hardness of classically simulating the one-clean-qubit model, Physical Review Letters **112** (2014).
[121] D. Aharonov, V. Jones, and Z. Landau, A polynomial quantum algorithm for approximating the jones polynomial (2006), arXiv:quant-ph/0511096.
[122] J. L. W. V. Jensen, Sur les fonctions convexes et les inégalités entre les valeurs moyennes, Acta Mathematica **30**, 175 (1906).
[123] A. Stam, Some inequalities satisfied by the quantities of information of fisher and shannon, Information and Control **2**, 101 (1959).
[124] S. G. Bobkov, G. P. Chistyakov, and F. Götze, Fisher information and the central limit theorem, Probability Theory and Related Fields **159**, 1 (2014).
[125] J. M. Arrazola, T. R. Bromley, J. Izaac, C. R. Myers, K. Brádler, and N. Killoran, Machine learning method for state preparation and gate synthesis on photonic quantum computers, Quantum Science and Technology **4**, 024004 (2019).
[126] R. Liu, S. V. Romero, I. Oregi, E. Osaba, E. Villar-Rodriguez, and Y. Ban, Digital quantum simulation and circuit learning for the generation of coherent states, Entropy **24** (2022).
[127] M. Potters and J.-P. Bouchaud, *A First Course in Random Matrix Theory: for Physicists, Engineers and Data Scientists* (Cambridge University Press, 2020).
[128] E. P. Wigner, On the distribution of the roots of certain symmetric matrices, Annals of Mathematics **67**, 325 (1958).

Appendix A: Trace estimation with DQC1

1. DQC1

When dealing with NMR quantum computers [114], it is common to have access to n-qubit mixed states. On these and similar quantum systems, having a quantum computer that only uses mixed states and some control qubits would be beneficial, rather than using only pure states that undergo unitary evolutions. It turns out that there exist problems that can be solved exponentially faster on this type of quantum computer compared to a classical computer [115]. However, it has been shown that this computation model is also significantly weaker than standard quantum computation [115, 116]. This model is referred to nowadays as DQC1 [75] (Deterministic Quantum Computation with One Clean Qubit). Several problems involving computing distance measures between unitaries and states using quantum circuits are DQC1-complete or -hard [117–119]. One for all, trace estimation of unitaries is DQC1-complete [116]. The same is true for energy estimation for Hamiltonians with very low (logarithmic) connectivity [116]. It seems that problems involving LCU-based sampling tend to be DQC1-hard [119]. In general, one can construct a DQC1-cost function by implementing a LCU-type circuit with controlled operations [58].

2. Sampling Unitary traces and DQC1 basics

The complexity class DQC1 that allows direct computation with one clean qubit uses a single- or multi-qubit controlled unitary V acting on a maximally mixed state (which corresponds to uniformly sampled random pure states) to perform quantum computation [75]. This model has been shown to be hard to simulate on classical computers if more than three output qubits are considered [120]. Estimating the re-normalized real part of the trace of a unitary V is a complete problem for DQC1 [116]. The circuit that allows this estimation is given in Fig. 11 (a) for mixed-states inputs and (b) for pure states.
The probability of measuring the control qubit for the circuit that uses mixed states – see Fig. 11 (a) – in the zero (+) or one (-) state is given by

$$\bar{p}_{\pm} = \frac{1}{2}\left(1 \pm \frac{1}{d}\,\mathrm{Re}\{\mathrm{Tr}\{V\}\}\right). \tag{A1}$$

The imaginary part can be obtained by inserting an S gate before the final measurement in the circuits given in Fig. (11). We can see that if we want to compute the trace of a unitary, the variance of the estimator for the trace behaves as a Bernoulli variable [121]:

$$\mathrm{Var}(\bar{x}) = \bar{p}_{+}(1 \quad \bar{p}_{+}). \tag{A2}$$

Here, we consider only one of the two outcomes and thus set $\bar{p} = \bar{p}_{+}$. Without using a maximally mixed state $\rho = \frac{}{d}$, computing the trace requires preparing $d = 2^n$ orthonormal basis states $\{|i\rangle\}_{i=1}^{d}$ and estimating their corresponding probability distribution p_i:

$$p_i = \frac{1}{2}(1 + \mathrm{Re}\{\langle i|\, V\, |i\rangle\}). \tag{A3}$$

As the real part of the trace of V is given by $\mathrm{Re}\{\mathrm{Tr}\{V\}\} = \sum_{i=1}^{d} \mathrm{Re}\{\langle i|\, V\, |i\rangle\}$, we can construct an estimator for the trace by creating an estimator $\bar{x}'$ that collects the counts of the d different circuits and sums them together.
The variance of this estimator behaves as a sum of independent Bernoulli variables, each one with variance $p_i(1 \quad p_i)$:

$$\mathrm{Var}(\bar{x}') = \frac{1}{d^2}\sum_{i=1}^{d} p_i(1 \quad p_i), \tag{A4}$$

and is d-times smaller than the variance of the estimator based on the maximally mixed state.

Proof. We prepare the d states $|i\rangle, i = 1, ..., d$ and apply the Hadamard test on each of them using the controlled unitary V. Since the states are prepared on d different Hilbert spaces, the variance of the estimates is simply the sum of the variances, each one equal to $p_i(1 \quad p_i)$ for $i = 1, ..., d$. Afterwards, we use the fact that $p_i \leq \sqrt{p_i}$ for $0 \leq p_i \leq 1$

and write

$$\frac{1}{d^2}\sum_{i=1}^{d}\sum_{j=1}^{d}\left(\mathbb{E}\left[x_i x_j\right] \quad p_i p_j\right) \overset{\text{C.S.}}{\leq} \frac{1}{d^2}\sum_{i=1}^{d}\sum_{j=1}^{d}\sqrt{p_i}\sqrt{p_j}(1 \quad \sqrt{p_i}\sqrt{p_j}). \tag{A5}$$

Then we apply one of the generalized mean inequalities (GM) [66], which follows from the Jensen's inequality [122]:

$$\sum_{i=1}^{d}\sum_{j=1}^{d}\sqrt{p_i}\sqrt{p_j} \overset{\text{GM}}{\leq} \frac{1}{d}\sum_{i=1}^{d} p_i, \tag{A6}$$

and we obtain

$$\frac{1}{d^2}\sum_{i=1}^{d}\sum_{j=1}^{d}\left(\mathbb{E}\left[x_i x_j\right] \quad p_i p_j\right) \leq \frac{1}{d}\sum_{i=1}^{d} p_i\left(1 \quad \frac{1}{d}\sum_{i=1}^{d} p_i\right) = \mathrm{Var}(\bar{x}). \tag{A7}$$

The last term is the variance of the DQC1 estimator. This derivation assumes that the circuits are somewhat correlated. However, if no correlations are present, because the estimators are prepared independently from each other, the term on the left hand side is d times smaller than the full variance with non-zero covariances. Therefore, we have

$$\mathrm{Var}(\bar{x}') \leq \frac{1}{d}\mathrm{Var}(\bar{x}). \tag{A8}$$

□

We see that the variance of the first estimator is at least d times smaller than the one of the DQC1 estimator. However, the DQC1 estimator does not require the preparation of d different states, as long as we have access to a source of randomly distributed pure states in input (or, equivalently, a maximally mixed state). In summary, the query complexity of both estimators with precision ϵ is $O(d^2/\epsilon^2)$ – in the former case, we need to sample the sum of d circuit outputs with precision $O(d/\epsilon^2)$, in the latter we sample from one circuit with precision $O(d^2/\epsilon^2)$.

3. Conditional sampling and averaging

The sampling methods discussed in the previous sections offer some insight in the computational structure of these methods. In particular, we see that an LCU circuit (or the equivalent ancilla-free approach) maps an originally Bernoulli-like estimation problem to an equivalent Bernoulli-like estimation problem. If the original goal is to sum L Bernoulli estimates $p_1, ..., p_L$, whose value lies between 0 and L, the LCU circuit outputs the renormalized version of this estimate, whose value lies between 0 and 1. As a consequence, the corresponding non-renormalized estimate $L\bar{p}$ behaves as a sum of strongly correlated variables with variance $L^2\bar{p}(1 \quad \bar{p})$. Eq. (45) shows that the correlations induced by the LCU circuits are stronger than any classical correlation. In classical probability theory, the sum of L independent Bernoulli-distributed trials follows the Poisson-Binomial distribution [96]. Interestingly, this distribution can be approximated by a Binomial distribution with mean $\bar{p}$. However, the variance of such distribution is always L times larger than the true variance of the Poisson-Binomial distribution. The output of the LCU circuit corresponds exactly to the approximation of the Poisson-Binomial distribution, which instead is the distribution of L independent Hadamard tests. A different question arises when we sum potentially correlated variables, for example if the originally independent Hadamard tests are sampled from a joint distribution. In this case, there may be potential correlations present induced, e.g., by random circuit sampling [83], due to the law of total variance.

Appendix B: SE with near-term amplitude estimation

Before moving to to SE with amplitude amplification, we briefly review the principles of Maximum Likelihood Quantum Amplitude Estimation (MLQAE) [69]. MLQAE starts from the amplified states:

$$|\psi^m\rangle = \mathcal{Q}^m|\psi\rangle = \sin((2m+1)\theta_p)\,|1\rangle\,|\psi_1\rangle + \cos((2m+1)\theta_p)\,|0\rangle\,|\psi_2\rangle\,, \tag{B1}$$

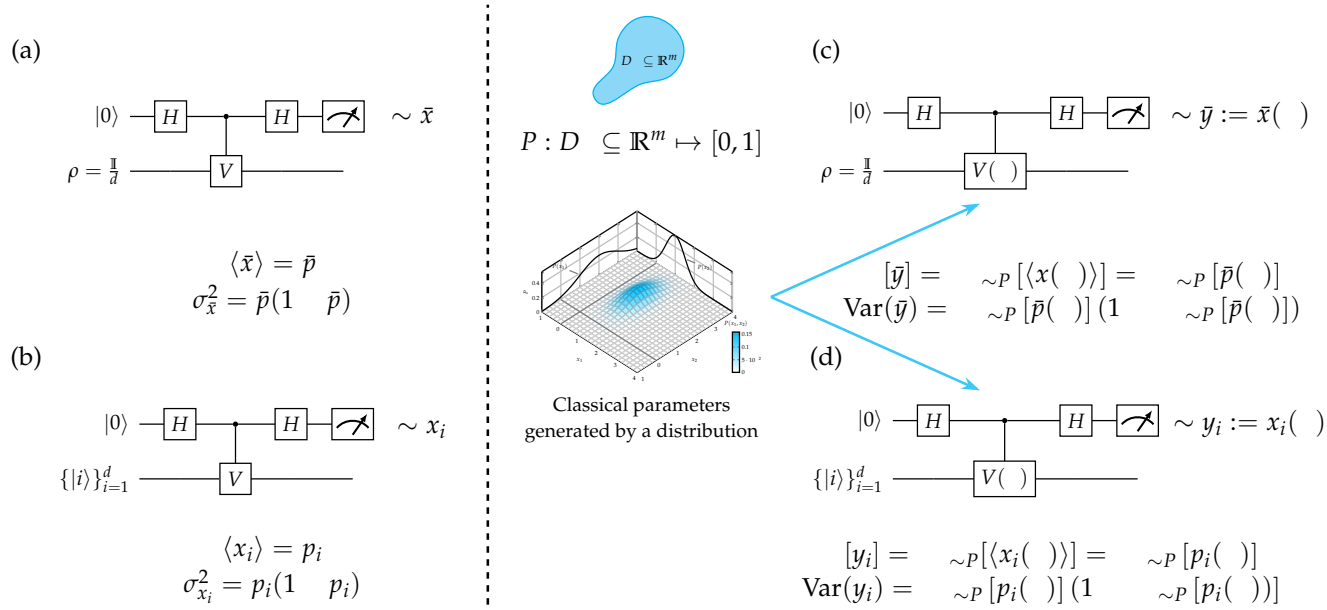


Figure 11. In this figure we consider different types of sampling problems that use the Hadamard-like test circuit normally employed for DQC1 tasks. In (a) and (b) the same circuits are considered with a fixed unitary V, without any external dependance. The values $\bar{x}$ and $x_i, i = 1, ..., d$ represent the Bernoulli counts sampled using the circuits. The circuits (c) and (d) instead depend for some internal degrees of freedom on an external parameter $\boldsymbol{\lambda} \in D \subseteq \mathbb{R}^m$ sampled from a probability distribution $P : D \subseteq \mathbb{R}^m \mapsto [0, 1]$. In this case, $\bar{y}$ and $y_i, i = 1, ..., d$ represent the new Bernoulli counts sampled using both the circuits and the distribution P. Therefore, sampling from these circuits requires averaging over the number of shots sampled from the circuits and the number of samples obtained from the probability distribution P. In (c), the circuit uses a maximally mixed state, whereas in (d) the d basis states $|i\rangle, i = 1, ..., d$ are prepared and the corresponding probability distribution of measuring the control qubit in state 0 or 1 is sampled.

where $\sin(\theta_p)^2 = p$, where p is the probability of measuring $|1\rangle$ and also the parameter to estimate, using the language of estimation theory. The states in Eq. (B1) are obtained after applying m times the Grover operator of AE [68]: $\mathcal{Q} = \mathcal{V} S_0 \mathcal{V}^{\ 1} S_\chi$, as defined in Eq. (61). If we prepare m copies of the initial state $|\psi\rangle$ and perform independent runs of AE each one with one of the operators $\mathcal{Q}, \mathcal{Q}^2, ..., \mathcal{Q}^M$ respectively, the probability distribution of this estimation problem is given by

$$P(\theta_p, \boldsymbol{x}, \boldsymbol{N}) = \prod_{i=1}^{M} \prod_{k=1}^{N_m} \sin^2((2m+1)\theta_p)^{x_{mk}} \cos^2((2m+1)\theta_p)^{1 \quad x_{mk}}, \tag{B2}$$

where the integers N_m are the number of shots from each circuit implementing $\mathcal{Q}, \mathcal{Q}^2, ..., \mathcal{Q}^M$ and x_{mk}, $k = 1, ..., N_m$, $m = 1, ..., M$ represents the data used for parameter estimation, i.e., the binary measurement results of the shot k from the mth circuit where the state is found in $|1\rangle$ (while $1 \quad x_{mk}$ is the outcome corresponding to $|0\rangle$). All data is represented by the vector $\boldsymbol{x} = (x_{11}, ..., x_{MN_M})^{\mathrm{T}}$ and the discrete parameters of the distribution are given by the vector of integers $\boldsymbol{N} = (N_1, ..., N_M)^{\mathrm{T}}$. Ref. [69] shows that if one applies the principle of Maximum Likelihood Estimation on P and if the allocation of the number of shots is chosen optimally, the total number of queries $n_q = \sum_{m=1}^{M} N_m$, which accounts for both the shots and the repeated applications of the operator Q, needed can potentially reach the Heisenberg limit, that is, the Fisher information of the Maximum Likelihood estimator obeys the following bound [69]:

$$F \leq \frac{n_q^2}{p(1 \quad p)}. \tag{B3}$$

Now we consider again the problem of estimating outputprobabilities $p_1, ..., p_L$ from quantum circuits $\mathcal{V}_i, ..., \mathcal{V}_L$ defined as in Eq. (63). Each of these circuits resembles a Hadamard test. We define the random variables corresponding to the *good* outputs of each circuit as $X_1, ..., X_L$, where $\mathbb{E}[X_i] = p_i$, for $i = 1, ..., L$. Let us approach the problem from

the point of view of MLQAE. If we assume that each estimate is obtained through MLQAE and that the estimator reaches the Heisenberg limit, then each estimator $\tilde{p}_1, ..., \tilde{p}_L$ that uses either near-term AE [69] has an error proportional to:

$$\epsilon_i^2 = \frac{1}{F(X_i)} = \frac{\sigma_i^2}{[n_q^{(i)}]^2}, \tag{B4}$$

where $\sigma_i^2 = p_i(1 \quad p_i) - F$ is the Fisher information of the ith random variable X_i associated with p_i – instead of the classical $\epsilon_i^2 = \frac{\sigma_i^2}{n_q^{(i)}}$. However, if these amplitudes are summed classically, the uncertainty propagation obeys again the rules of classical propagation of the uncertainty. This is due to the Stam inequality, which argues that for L random variables $X_1, ..., X_L$ the Fisher information of any given parameter obeys the following bound [123, 124]:

$$\frac{1}{F(X_1 + ... + X_L)} \geq \frac{1}{F(X_1)} + ... + \frac{1}{F(X_L)}. \tag{B5}$$

As a consequence, for the sum of amplified estimates with Heisenberg-like scaling of the classical Fisher information – see Ref. [69] –, we have the following bound for the error:

$$\epsilon^2 \geq \frac{1}{F(X_1 + ... + X_L)} \geq \sum_{i=1}^{L} w_i^2 \frac{\sigma_i^2}{[n_q^{(i)}]^2}. \tag{B6}$$

The total number of queries from all circuits can be minimized with respect to $n_q^{(i)}$, $i = 1, ..., L$ using Lagrange multipliers [22]:

$$\mathcal{L}(n_q^{(i)}, ..., n_q^{(L)}, \lambda) = \sum_{i=1}^{L} n_q^{(i)} + \lambda \left(\sum_{i=1}^{L} \frac{w_i^2 \sigma_i^2}{[n_q^{(i)}]^2} \quad \epsilon^2 \right) \tag{B7}$$

$$\left(\forall 1 \leq i \leq L : \ \frac{\partial \mathcal{L}}{\partial n_q^{(i)}} = 0 \right) \wedge \left(\frac{\partial \mathcal{L}}{\partial \lambda} = 0 \right). \tag{B8}$$

The minimization procedure results in:

$$\forall 1 \leq i \leq L : \ [n_q^{(i)}]^3 = 2\sigma_i w_i^2 \lambda \tag{B9}$$

$$2\lambda = \left(\frac{1}{\epsilon^2} \sum_{i=1}^{L} (w_i^2 \sigma_i)^{\frac{1}{3}} \right)^{\frac{3}{2}}, \tag{B10}$$

and leads to a total number of queries N_q equal to

$$N_q = \sum_{i=1}^{L} n_q^{(i)} = \frac{1}{\epsilon} \sum_{i=1}^{L} \left(\sum_{i=1}^{L} (2\sigma_i w_i^2)^{\frac{1}{3}} \right)^{\frac{1}{2}} \sim O\left(\frac{L\sqrt{L}}{\epsilon} \right). \tag{B11}$$

As expected, we obtain a partial improvement over the classical evaluation even by implementing MLQAE. However, the improvement is worse than a LCU circuit where MLQAE is applied. It seems that the typical LCU entanglement is needed in order to obtain a better improvment. It is unclear whether another MLQAE approach could obtain an asymptotical sampling complexity of $O(L/\epsilon)$ without the use of an LCU circuit. If the estimation is divided on batches of size k, which are encoded in corresponding LCU circuits and subsequently amplified, and whose results are then summed classically, one has a complexity of $O(\frac{L}{} \sqrt{L/k})$, which reduces to both the LCU and SE amplfied case for $k = 1$ and $k = L$, respectively.

a. *Classical shadow tomography for quantum cost functions*

To better contextualize the use of the LCU in estimation, we want to compare it to other popular estimation strategies that emerged in the last years: Tomography based on classical shadows [16, 84]. A classical shadow $\hat{\rho}$ is an estimator of the true density matrix of a quantum experiment ρ. It can be used to estimate the mean value of any

observable with respect to ρ. In facts, for any channel acting on ρ – $d = 2^n$ for n qubits –, we can write [16, 81]:

$$\mathcal{Q}(\rho) = \sum_{l=1}^{d} \mathbb{E}_{U\sim\nu} \left[\mathrm{tr}\{U\rho U^\dagger |l\rangle \langle l|\} U^\dagger |l\rangle \langle l| U \right], \tag{B12}$$

where $U \sim \nu$ denotes random sampling from the Haar measure. Then the estimator $\tilde{\rho}$

$$\tilde{\rho} = \mathcal{Q}^{-1}(U^\dagger |l\rangle \langle l| U) \tag{B13}$$

is an unbiased estimator of ρ [81], which can be used to estimate mean values of arbitrary observables. For an observable with L non-zero non-commuting Pauli coefficients, the sampling complexity is $O[L\log(L)/\epsilon^4]$. The significant reduction in the number of queries needed is traded off with a worse scaling with the precision ϵ. In the presence of AE sampling, as we see from Eq. (B11) – see also [81] –, the variance of the estimator reduces to $O(\sqrt{L}/\epsilon)$, which results in $O[\sqrt{L}\log(L)/\epsilon^3]$. In order to be performed efficiently, classical shadows have two requirements: (a) the inversion operation of the channel needs to be efficient [6] and (b) an efficient method to sample Haar random unitaries $U \sim \nu$ needs to be available. Both the classical shadow approach and the approach that uses the Jordan algorithm speed up the oracular evaluation of multiple observables, but do not reduce the variance. If the estimation of a quantum cost function is carried out using the shadow estimation protocol, the estimation of its gradient can, in theory, also be carried out using the same method [6]. General gradient estimation of quantum cost functions based on LCU, finite-difference or PSR approaches has a linear dependence from the number of variational parameters N, i.e., $O(N)$. This represents a significant bottleneck to the efficient implementation of variational circuits optimization [14] and quantum machine learning algorithms [24, 27]. In this case, we know that estimating the gradient of a quantum cost function with observable $\mathcal{O}$ that evolves with a unitary $V_i = \exp\{-iH_i\theta_i\}$ corresponds to estimating the mean value of the observable $i[\mathcal{O}, H_i]$. For a sequence of independent gates with the same structure, a nested estimation is required in which $i[\mathcal{O}_i, H_i]$, $\mathcal{O}_i = \prod_{j=1}^{i} V_i \rho \left(\prod_{j=1}^{i} V_i\right)^\dagger$ fully determines the gradient component $\frac{\partial}{\partial\theta_i} C(\boldsymbol{\theta_i})$ for $i = 1, ..., N$. Ref. [6] shows that no general backpropagation can be accomplished without access to a quantum memory. However, there seem to be classes of circuits that are not classically simulable and whose gradients can be sampled in sub-linear time with respect to the number of parameters N [5].

1. SU(d) gradient with coherent state

We consider the cost function $C(\boldsymbol{\theta}) = \mathrm{tr}\{V_c(\boldsymbol{\theta})\rho V_c(\boldsymbol{\theta})^\dagger Z_{\mathrm{prod}}\}$. A generalization to arbitrary observables can be achieved by implementing either the SE or the LCU estimator. A generalization to positive and negative linear combinations is given in Section II D. The gradient of the cost function is provided in Eq. (88) for a unitary $V_c(\boldsymbol{\theta}) = e^{-i(H_0\theta_0 + H_1\theta_1)t}$ – see Eq. (107). The derivative with respect to θ_1 evaluated at $\theta_1 = 0$ is given by:

$$\frac{\partial C(\boldsymbol{\theta})}{\partial\theta_1}\Big|_{\boldsymbol{\theta}=(1,0)^T} = \mathrm{tr}\left\{\frac{\partial V_c(\boldsymbol{\theta})}{\partial\theta_i}\rho V_c^\dagger(\boldsymbol{\theta}) Z_{\mathrm{prod}}\right\} = 2\|\boldsymbol{a}\|_1 \sum_{l=0}^{\infty} \mathrm{Re}\left\{\frac{(-2i\Delta t)^l}{(l+1)!}\,\mathrm{tr}\left\{\left(\frac{1}{2}\right)^l \mathrm{ad}^l_{H_0}(H_1)\rho_c(\boldsymbol{\theta}) Z_{\mathrm{prod}}\right\}\right\} \tag{B14}$$

where on the right hand side we used the substitution $\rho_c(\boldsymbol{\theta}) = V_c(\boldsymbol{\theta})\rho V_c^\dagger(\boldsymbol{\theta})$. We note that for a complex value $z \in$:

$$\mathrm{Re}\{i^l z\} = \begin{cases} \mathrm{Im}\{z\} & l \mod 4 = 1 \\ \mathrm{Re}\{z\} & l \mod 4 = 2 \\ \mathrm{Im}\{z\} & l \mod 4 = 3 \\ \mathrm{Re}\{z\} & l \mod 4 = 0, \end{cases} \tag{B15}$$

so we cannot simply move the i^l outside of the real part operation. The adjoint term is computed by the circuit given in Fig. 8 (b) for a given order l. Furthermore, we use a Hadamard test-like circuit. For a renormalized observable Z_{prod} whose mean value lies in $I = (-1, 1)$, we can always use LCU and a Hadamard test [50, 87] to estimate $\frac{1}{2}(1 \pm \mathrm{Re}\{\rho_c(\boldsymbol{\theta})Z_{\mathrm{prod}}\})$ (using the Hadamard gate H) and/or $\frac{1}{2}(1 \pm \mathrm{Im}\{\rho_c(\boldsymbol{\theta})Z_{\mathrm{prod}}\})$ (using the Hadamard gate and the S gate, HS). We introduce a multi-controlled register for the LCU summation that goes from 0 to $L-1$ with

$r = \lceil \log(L) \rceil$ qubits :

$$\left|\sqrt{2\Delta t}_L\right\rangle = \frac{1}{\sqrt{\mathcal{M}_L}} \sum_{l=0}^{L\quad 1} \frac{(\sqrt{2\Delta t})^l}{\sqrt{(l)!}} \left|l\right\rangle, \tag{B16}$$

where $\mathcal{M}_L = \sum_{l=0}^{L\quad 1} \frac{|2\quad t|^l}{l!}$ is the renormalization factor. We then use a Hadamard test circuit with Z_{prod}^{-} as measurement and controlled commutator circuits $l = 0, ..., L\quad 1$ as in Fig. 8 (b), we can estimate the quantity:

$$q_{\pm}^{L} = \frac{1}{2}\left(1 \pm \frac{1}{\mathcal{M}_L} \sum_{l=0}^{L\quad 1} \frac{(2\Delta t)^l}{l!} \text{Re}\Big\{\text{tr}\Big\{(\quad i/2)^l \text{ad}_{H_0}^l(H_1)\rho_c(\boldsymbol{\theta}) Z_{\text{prod}}\Big\}\Big\}\right), \tag{B17}$$

In the limit of L approaching infinity, we have the coherent state:

$$\left|\sqrt{2\Delta t}\right\rangle = e^{\quad |\quad t|} \sum_{l=0}^{\infty} \frac{(\sqrt{2\Delta t})^l}{\sqrt{(l)!}} \left|l\right\rangle, \tag{B18}$$

and

$$q_{\pm}^{\infty} = \frac{1}{2}\left(1 \pm \frac{1}{\mathcal{M}_{\quad t}} \sum_{l=0}^{\infty} \frac{(2\Delta t)^l}{l!} \text{Re}\Big\{\text{tr}\Big\{(\quad i/2)^l \text{ad}_{H_0}^l(H_1)\rho_c(\boldsymbol{\theta}) Z_{\text{prod}}\Big\}\Big\}\right), \tag{B19}$$

where $\mathcal{M}_{\quad t} = \exp\{2|\Delta t|\}$ is the normalization coefficient of the coherent state. The estimate given by the commutator circuit is already renormalized thanks to the l Hadamard gates in Fig. 8. The addition of a number l of S gates provides the $(\quad i)^l$ terms. Eq. (B19) does not match Eq. (B14) yet. In order to match the two equations up to a constant, we change the first controlled operation on the LCU register to be the identity, whereas the operator that corresponds to $\text{ad}_{H_0}^l(H_1)$ is conditioned on the next entry in the register. As a result we have:

$$e_{\pm}^{\infty} = \frac{1}{2}\left[1 \pm \frac{1}{\mathcal{M}_{\quad t}}\left(b(\boldsymbol{\theta}) + \sum_{l=1}^{\infty} \frac{(2\Delta t)^l}{l!}\left(\quad \frac{1}{2}\right)^{l\quad 1} \text{Re}\Big\{i^{l\quad 1}\,\text{tr}\Big\{\text{ad}_{H_0}^{l\quad 1}(H_1)\rho_c(\boldsymbol{\theta}) Z_{\text{prod}}\Big\}\Big\}\right)\right], \tag{B20}$$

and

$$e_{+}^{\infty} \quad e^{\infty} = \frac{1}{\mathcal{M}_{\quad t}}\left(b(\boldsymbol{\theta}) + 2\Delta t \sum_{l=0}^{\infty} \frac{(2\Delta t)^l}{(l+1)!} \text{Re}\Big\{(\quad i/2)^l\,\text{tr}\Big\{\text{ad}_{H_0}^l(H_1)\rho_c(\boldsymbol{\theta}) Z_{\text{prod}}\Big\}\Big\}\right) = \frac{1}{\mathcal{M}_{\quad t}}\left(b(\boldsymbol{\theta}) + \Delta t \frac{\partial}{\partial\theta_1} C(\boldsymbol{\theta})\Big|_{\boldsymbol{\theta}=(1,0)^T}\right), \tag{B21}$$

with the bias $b(\boldsymbol{\theta}) = \text{Re}\{\text{tr}\{\rho_c(\boldsymbol{\theta}) Z_{\text{prod}}\}\}$ that needs to be removed. The variance analysis of the LCU approach shown in Table I implies a $O(\mathcal{M}^2_{\quad t}/\epsilon^2) = O\left(e^{4|\quad t|}/\epsilon^2\right)$ asymptotic sampling complexity. For an approximation index L we have $O(\mathcal{M}_L^2/\epsilon^2)$. As the de-biased estimate is multiplied with Δt, compensating for it will increase the variance by a factor $1/\Delta t^2$. As such, the optimal choice for Δt is $O(1)$. Amplitude estimation reduces the sampling complexities by a quadratic factor. The required coherent states can be constructed efficiently with, e.g., the quantum algorithms developed in [125, 126].

2. Quantum control gradients

In quantum control problems, the unitary dynamics is determined by the Hamiltonian with drift operator H_0 and control operators $H_1, ..., H_{K\quad 1}$:

$$H_{\boldsymbol{\phi}}(t) = H_0 + \sum_{k=1}^{K\quad 1} H_k u_k^{\boldsymbol{\phi}_k}(t). \tag{B22}$$

The time-dependent control pulses $u_1^{\boldsymbol{\phi}_1}(t), ..., u_{K\ 1}^{\boldsymbol{\phi}_{K\ 1}}$ depend on parameters $\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_{K\ 1}$. The solution of the control problem depends on a parameter vector $\boldsymbol{\phi} = (\boldsymbol{\phi}_1, ..., \boldsymbol{\phi}_{K\ 1})^{\mathrm{T}}$. The unitary control dynamics with time-ordering operator $\mathcal{T}$ is given by:

$$V_{\boldsymbol{\phi}}(t_0, T) = \mathcal{T} \exp\left\{ i \int_{t=t_0}^{T} H_{\boldsymbol{\phi}}(\tau) d\tau \right\}. \tag{B23}$$

We want now to consider the case, similar to Refs. [44, 53], in which we aim to perform a variational pulse-level optimization of a quantum cost function using the unitary Eq. (B23). Therefore, we need to estimate the gradient of such quantum cost functions with respect to the control dynamics.

a. *Continuous GRAPE gradient*

We study now the derivative of the cost function $C(\boldsymbol{\phi}) = \mathrm{tr}\Big\{ V_{\boldsymbol{\phi}}(t_0, T) \rho V_{\boldsymbol{\phi}}^{\dagger}(t_0, T) \mathcal{O} \Big\}$ – see Eq. (118) – with respect to the control parameters ϕ_{ij}, where i refers to the ith real parameter value of the vector $\boldsymbol{\phi}_j$ that corresponds to the jth control operator [52, 53]:

$$\frac{\partial C(\boldsymbol{\phi})}{\partial \phi_{ij}} = \mathrm{tr}\left\{ \frac{\partial V_{\boldsymbol{\phi}}}{\partial \phi_{ij}} \rho V_{\boldsymbol{\phi}}^{\dagger} \mathcal{O} \right\} + \mathrm{tr}\left\{ V_{\boldsymbol{\phi}} \rho \frac{\partial V_{\boldsymbol{\phi}}^{\dagger}}{\partial \phi_{ij}} \mathcal{O} \right\} = +i \int_{t_0}^{T} \mathrm{tr}\{ \rho(T) V_{\boldsymbol{\phi}}(t_0, \tau) H_j V_{\boldsymbol{\phi}}(t_0, \tau) \mathcal{O} \} \frac{\partial u_j^{\boldsymbol{\phi}_j}(\tau)}{\partial \phi_{ij}} d\tau = \tag{B24}$$

$$i \int_{t_0}^{T} \mathrm{tr}\{ \rho(T) [H_j(t_0, \tau), \mathcal{O}] \} \frac{\partial u_j^{\boldsymbol{\phi}_j}(\tau)}{\partial \phi_{ij}} d\tau = \quad i \int_{t_0}^{T} s_j^{\boldsymbol{\phi}}(\tau) \frac{\partial u_j^{\boldsymbol{\phi}_j}(\tau)}{\partial \phi_{ij}} d\tau,$$

where $\rho(T) = V_{\boldsymbol{\phi}}(t_0, T) \rho V_{\boldsymbol{\phi}}(T, t_0)$, $H_j(t_0, \tau) = V_{\boldsymbol{\phi}}(t_0, \tau) H_j V_{\boldsymbol{\phi}}(\tau, t_0)$ and $s_j^{\boldsymbol{\phi}}(\tau) = \mathrm{tr}\{ \rho(T) [H_j(t_0, \tau), \mathcal{O}] \}$. Here we used the property of the propagator $V(t_0, T) = V(t_0, \tau) V(\tau, T)$ and the cyclic properties of the trace. We see that the gradient of the quantum cost function with respect to the pulse parameters is equal to the time integral of a different time-dependent quantum cost function multiplied with the derivative of classical time-dependent signals $u_j^{\boldsymbol{\phi}_j}(\tau)$, which can be determined using classical automatic differentiation of the input pulses coming from an amplitude waveform generator.

b. *LCU GRAPE circuit*

Similarly as in the case of the SU(d) gradient, the quantum control gradient consists of a (continuous) sum of estimates of quantum observables. Due to the presence of the integral, this gradient can be evaluated using either the analog or the standard LCU approach [1]. Another possibility is to use the GRAPE approach [100]: the time propagation of the pulse is divided in a sequence of time samples $t_0, t_1, ..., t_{N_T\ 1}$ for a pulse $u(t_0), u(t_1), ..., u(t_{N_T\ 1})$. In this case, the gradient is given by a discrete sum of terms – see Eq. (121) –, which can be estimated using the circuit given in Fig. (10) (b). The length of the circuit grows here as $O[N_T \log(N_T)]$, which is the number of discrete time step considered to represent the continuous integral given in Eq. (119).

As the GRAPE circuits is particularly challenging to implement on basic quantum devices, it cannot be used to perform gate set optimization and compilation on quantum hardware. It can be implemented, however, in contexts where quantum control is used as an ansatz for variational quantum algorithms [108] and on a quantum hardware platform that allows for efficient implementations of multi-qubit operations. In particular, the amplified version of this gradient allows for quadratic speed-up over naive gradient evaluation, due to the presence of a (coherent) quantum summation process instead of a classical one.

Appendix C: Truncation Error in the Gradient Series

By inserting the expansion defined in Eq. (102) into the gradient given in Eq. (90), we obtain:

$$\nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}) = \sum_{l=0}^{\infty} T_l(\boldsymbol{\theta}), \tag{C1}$$

with

$$T_l = \frac{(\ \ 1)}{(l+1)!}\|\boldsymbol{\theta}\|^l \frac{1}{d}\,\mathrm{Re}\,\mathrm{Tr}\big[(\ \ i)^l V \mathrm{ad}_H^l(G) U^\dagger\big], \tag{C2}$$

and substituting $G = \nabla_{\boldsymbol{\theta}} \bar{H}(\boldsymbol{\theta})$. For an Lth order expansion

$$\nabla_{\boldsymbol{\theta}} C_L(\boldsymbol{\theta}) = \sum_{l=0}^{L} T_l(\boldsymbol{\theta}), \tag{C3}$$

the remainder is then given by

$$R_L(\boldsymbol{\theta}) = \sum_{l=L+1}^{\infty} T_l(\boldsymbol{\theta}). \tag{C4}$$

In the following sections we will derive bounds and estimations for the square of the truncation error R_L.

Appendix D: Gradient Convergence with Random Operators

We consider the Hermitian operators A and B, which are sampled from 2 Gaussian Unitary Ensembles (GUE) of operators, parametrized by the Dyson indexes β_A, β_B and operator norm expectation values $\Lambda_A^2 = \mathbb{E}\left[\mathrm{Tr}\ A^2)\right] = \mathbb{E}\left[\sum_{i=1}(\lambda_i^A)^2\right]$, $\Lambda_B^2 = \mathbb{E}\left[\mathrm{Tr}\ B^2)\right] = \mathbb{E}\left[\sum_{i=1}\lambda_i^B\right]$, where λ_i^A, λ_i^B are the eigenvalues of A, B respectively. The GUE is defined by the following Gaussian measure defined on the space of $d \times d$ complex Hermitian matrices [127]:

$$\frac{1}{Z_{\mathrm{GUE}}} e^{\ \frac{d}{2}\,\mathrm{tr}\{H\}^2}, \tag{D1}$$

where the partition function is given by $Z_{\mathrm{GUE}} = 2^{d/2}\ \ \frac{\pi}{d}\big)^{\frac{d^2}{2}}$. In the simulations and derivations that use random Hamiltonians – see, e.g., Fig. 9 – such Hamiltonians are sampled using Eq. (D1). We furthermore consider the unitaries to be $U, V \sim$ Haar on $U(d)$, where we first treat the case of independent U and V to then generalize to the correlated case as required for partially optimized systems. We are interested in deriving statistical properties of the remainder defined in Equation (C4). Using the unitary invariance of the trace, we can express the operators in the diagonal basis of A, so that

$$A = U \underbrace{\mathrm{diag}(\lambda_1, \ldots, \lambda_d)}_{\tilde{A}} U^\dagger, \tag{D2}$$

with the diagonal operator $\tilde{A}$. In this basis B is expressed as

$$\tilde{B} = \sum_{p,q=1}^{d} b_{pq} \left|p\right\rangle \left\langle q\right|. \tag{D3}$$

For the first order we consider the commutator,

$$[\tilde{A}, \tilde{B}] = \sum_{p,q=1}^{d} (\lambda_p \quad \lambda_q) b_{pq} \left|p\right\rangle \left\langle q\right|. \tag{D4}$$

This generalizes to the adjoint operator recursively:

$$\mathrm{ad}_{\tilde{A}}^l(\tilde{B}) = [\tilde{A}, \mathrm{ad}_{\tilde{A}}^{l\ 1}(\tilde{B})] = \sum_{p,q=1}^{d} (\lambda_p \quad \lambda_q)^l b_{pq} \left|p\right\rangle \left\langle q\right|, \tag{D5}$$

which conveniently does not introduce additional sums. Multiplication with the unitary operator V, which we also

represent in the eigenbasis of A as $\tilde{V} = \sum_{p,q=1}^{d} v_{pq} |p\rangle \langle q|$ gives

$$\tilde{V}\mathrm{ad}_{\tilde{A}}^{l}(\tilde{B}) = \sum_{p,q,r=1}^{d} (\lambda_p \quad \lambda_q)^l v_{rp} b_{pq} |r\rangle \langle q| . \tag{D6}$$

The (real) trace of this is then equivalent to the original operators, and reduces to

$$\mathrm{Re}\Big\{\mathrm{Tr}\left(V\mathrm{ad}_{A}^{l}(B)\right)\Big\} = \mathrm{Re}\Big\{\mathrm{Tr}\left(\tilde{V}\mathrm{ad}_{\tilde{A}}^{l}(\tilde{B})\right)\Big\} = \sum_{p,q=1}^{d} (\lambda_p \quad \lambda_q)^l \,\mathrm{Re}\{v_{qp} b_{pq}\}. \tag{D7}$$

Similarly for the complete remainder term, we have

$$R_L = \sum_{l=L+1}^{\infty} T_l = \frac{1}{d} \sum_{p,q=1}^{d} \mathrm{Re}(v_{qp} b_{pq}) \sum_{l=L+1}^{\infty} \frac{(\quad 1)^l \|\boldsymbol{\theta}\|^l (\lambda_p \quad \lambda_q)^l}{(l+1)!}. \tag{D8}$$

The expectation value of the squared remainder follows as

$$\mathbb{E}[R_L^2] = \frac{1}{d^2} \sum_{p,q=1}^{d} \mathbb{E}\left[\mathrm{Re}(v_{qp} b_{qp})^2\right] \sum_{l,m=L+1}^{\infty} \frac{(\quad 1)^{l+m} \|\boldsymbol{\theta}\|^{l+m} \,\mathbb{E}\left[(\lambda_p \quad \lambda_q)^{l+m}\right]}{(m+1)!(l+1)!}, \tag{D9}$$

Due to circular symmetry of the Haar measure unitary, we have $\mathbb{E}[v_{qp}] = \mathbb{E}[v_{qp}^2] = 0$ and $\mathbb{E}[\|v_{qp}\|^2] = \frac{1}{d}$. For B meanwhile we have $\mathbb{E}[b_{qp}] = 0$ and $\mathbb{E}[\|b_{qp}\|^2] = \frac{\Lambda^2}{d}$ (for $p \neq q$). Hence we find $\boxed{\mathbb{E}\left[\mathrm{Re}(v_{qp} b_{qp})^2\right] = \frac{1}{2}\mathbb{E}[\|v_{qp}\|^2]\,\mathbb{E}[\|b_{qp}\|^2] = \frac{\Lambda_B^2}{2d^2}}$, summed over $d(d \quad 1)$ index combinations (p,q) with $p \neq q$.

We will approximate the eigenvalue distribution without correlations, neglecting the level repulsion, using the large d limiting case known as the Wigner semi-circle law [128], so that the eigenvalue distribution is approximated as

$$\rho(\lambda) = \frac{2}{\pi R^2} \sqrt{R^2 \quad \lambda^2} \quad \text{for } |\lambda| \leq R, \tag{D10}$$

with $R = 2\Lambda_A$. Due to the symmetry of the Wigner semi-circle law, the odd moments vanish $\mathbb{E}[\lambda^{2m+1}] = 0$. for the even moments we have

$$\mathbb{E}[\lambda^{2m}] = \frac{1}{m+1} \left(\frac{R_H}{2}\right)^{2m} \binom{2m}{m} = C_n \Lambda_A^{2m}, \tag{D11}$$

with the Catalan numbers $C_n = \frac{1}{n+1} \quad ^{2n}_{n})$. Hence we find using binomial expansion

$$\mathbb{E}[(\lambda_p \quad \lambda_q)^{2m}] = \sum_{l=0}^{2m} \binom{2m}{l} (\quad 1)^l \,\mathbb{E}[\lambda_p^l]\,\mathbb{E}[\lambda_q^{2m \quad l}] = \Lambda_A^{2m} \sum_{l=0}^{m} \binom{2m}{2l} C_l C_{m \quad l}, \tag{D12}$$

with all odd terms vanishing.

We hence find for the remainder in Equation (D9), using the substitution $2k = l + m$

$$\mathbb{E}[R_L^2] = \frac{\Lambda_B^2 d(d \quad 1)}{2d^4} \sum_{k=L+1}^{\infty} (\|\boldsymbol{\theta}\|\Lambda_A)^{2k} \sum_{l=0}^{k} \binom{2k}{2l} C_l C_{k \quad l} \sum_{m=L+1}^{2k \quad L} \frac{1}{(m+1)!(2k \quad m+1)!}. \tag{D13}$$

To leading order (i.e., setting $k = L+1$) the final sum contributes only a single term $1/(L+2)!^2$, so that

$$\mathbb{E}[R_L^2] \approx \frac{\Lambda_B^2 (d \quad 1)}{2d^3} (\|\boldsymbol{\theta}\|\Lambda_A)^{2(L+1)} \sum_{l=0}^{L+1} \frac{^{2(L+1)}_{2l}) C_l C_{L+1 \quad l}}{(L+2)!^2}. \tag{D14}$$

1. Generalizing the result for the in delity

The gradient of the infidelity, which is computed with the Hilbert-Schmidt test, is given by:

$$\nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}) = \frac{2}{d^2} \mathrm{Re}\{\mathrm{tr}\{ V(\boldsymbol{\theta}) U^\dagger) \otimes \nabla_{\boldsymbol{\theta}} V(\boldsymbol{\theta}) U^\dagger) \}\}. \tag{D15}$$

By using the SU(d) derivative:

$$\nabla_{\boldsymbol{\theta}} V(\boldsymbol{\theta}) = \sum_{l=0}^{\infty} \frac{(i)^l}{(l+1)!} \|\boldsymbol{\theta}\|_1^l \mathrm{ad}^l_{\bar{H}(\boldsymbol{\theta})} \bar{\nabla}_{\boldsymbol{\theta}} H(\boldsymbol{\theta})) V(\boldsymbol{\theta}) = \sum_{l=0}^{\infty} \mathcal{W}_l, \tag{D16}$$

so that

$$\nabla_{\boldsymbol{\theta}} C(\boldsymbol{\theta}) = \frac{2}{d} \sum_{l=0}^{\infty} \mathrm{Re}\{\mathrm{tr}\{ V(\boldsymbol{\theta}) U^\dagger) \otimes \mathcal{W}_l U^\dagger) \}\} = \sum_{l=0}^{\infty} \mathcal{T}_l. \tag{D17}$$

We now turn to the squared truncation error of the gradient of the (in)fidelity, which is given in Eq. (92). We denote this truncation error with $\mathcal{R}_L = \sum_{l=L+1}^{\infty} \mathcal{T}_l$. The procedure is the same as for the gradient of the POTQ circuit. In this case, Eq. (D9) becomes:

$$\mathbb{E}[\mathcal{R}_L^2] = \frac{1}{d^4} \sum_{p,q,s=1}^{d} \mathbb{E}\left[\mathrm{Re}(v_{ss}\, v_{qp} b_{qp})^2\right] \sum_{l,m=L+1}^{\infty} \frac{(1)^{l+m} \|\boldsymbol{\theta}\|^{l+m} \mathbb{E}\left[(\lambda_p \quad \lambda_q)^{l+m}\right]}{(m+1)!(l+1)!}, , \tag{D18}$$

where we reuse $[V(\boldsymbol{\theta})U^\dagger]_{p,q} = v_{pq}$. Using column phase invariance, the expectation values expand as

$$\mathbb{E}\left[\mathrm{Re}(v_{ss}\, v_{qp} b_{qp})^2\right] = \frac{1}{2} \mathbb{E}[\|v_{ss}\|^2 \|v_{qp}\|^2 \|b_{qp}\|^2] = \frac{1}{2} \mathbb{E}[\|v_{ss}\|^2 \|v_{qp}\|^2] \mathbb{E}[\|b_{qp}\|^2]. \tag{D19}$$

The v-dependent term contains correlations between the components, due to the column normalization, which can easily be derived via the fourth order moments for Haar unitaries, so that

$$\mathbb{E}[\|v_{ss}\|^2 \|v_{qp}\|^2] = \begin{cases} \frac{1}{d(d+1)} & \text{for} \quad s \in p, q \; q = s \\ \frac{1}{d^2 \quad 1} & \text{for} \quad s \notin p, q \end{cases}. \tag{D20}$$

Hence, with $\mathbb{E}[\|b_{qp}\|^2] = \frac{\Lambda^2}{d}$, we find for the sum

$$\sum_{p,q,s=1}^{d} \mathbb{E}\left[\mathrm{Re}(v_{ss}\, v_{qp} b_{qp})^2\right] = \frac{1}{2} \sum_{p,q=1}^{d} \underbrace{\left(\frac{2}{d(d+1)} + \frac{d \quad 2}{d^2 \quad 1} \right)}_{= \frac{d^2 \quad 2}{d^2 \; d \quad 1)}} \mathbb{E}[\|b_{qp}\|^2] = \frac{d^2 \quad 2}{2d^3} \Lambda_B^2. \tag{D21}$$

The squared truncation error of the fidelity gradient can therefore be approximated asymptotically as:

$$\mathbb{E}[\mathcal{R}_L^2] = \frac{4(d^2 \quad 2)}{d^3(d \quad 1)} \mathbb{E}[R_L^2] \approx \frac{4}{d^2} \mathbb{E}[R_L^2]. \tag{D22}$$

2. Partially optimized U

When using the gradients to optimize V, we can no longer assume random V. In the following section we discuss the changes to our estimation theory introduced by these correlations for the most pertinent case, the minimization of the infidelity between U and $V(\boldsymbol{\theta})$. With a fidelity $F = \frac{1}{d^2} \left|\mathrm{tr}\ V(\boldsymbol{\theta}) U^\dagger)\right|^2 = \frac{1}{d^2} \left|\mathrm{tr}\,(W)\right|^2$ we can decompose $W = \sum_{i=0}^{d^2 \quad 1} w_i P_i$, where P_i are the n qubit normalized Pauli matrices, with $P_0 = I$ and the w_i are the Pauli expansion coefficients, so that $|w_0|^2 = F$ and $\sum_{i=1}^{d^2 \quad 1} |w_i|^2 = 1 \quad F$. We assume that the remaining $d^2 \quad 1$ components, that are not constrained by the fidelity, can still be considered as randomly sampled according to the Haar distribution,

and only constrained to the reduced normalization 1 F. In order to compute the resulting expectation value, let us rewrite the gradient of the infidelity from Eq. (D15) as

$$\boldsymbol{\theta} I(\boldsymbol{\theta}) = \frac{2}{d}\,\mathrm{Re}\Big[\underbrace{\frac{1}{d}\,\mathrm{tr}\;V(\boldsymbol{\theta})U^\dagger\big)^*}_{=w_0^*}\,\mathrm{tr}\quad \boldsymbol{\theta} V(\boldsymbol{\theta})U^\dagger\big)\Big], \tag{D23}$$

where we used the property of the trace $\mathrm{tr}\{A\otimes B\} = \mathrm{tr}\{A\}\,\mathrm{tr}\{B\}$ for complex matrices A, B. We use again the expansion:

$$\nabla_{\boldsymbol{\theta}} V = \sum_{l=0}^{\infty} \frac{(\quad i)^l}{(l+1)!}\|\boldsymbol{\theta}\|^l\,\mathrm{ad}^l_{\bar{H}(\boldsymbol{\theta})}\;\nabla_{\boldsymbol{\theta}}\bar{H}(\boldsymbol{\theta})\big)\,V, \tag{D24}$$

so that

$$\boldsymbol{\theta} I(\boldsymbol{\theta}) = \frac{2}{d}\sum_{l=0}^{\infty}\frac{1}{(l+1)!}\,\mathrm{Re}\Big[w_0^*\,(\quad i)^l\|\boldsymbol{\theta}\|^l\;\mathrm{tr}\Big(\mathrm{ad}^l_{\bar{H}(\boldsymbol{\theta})}\;\nabla_{\boldsymbol{\theta}}\bar{H}(\boldsymbol{\theta})\big)\,W\Big)\Big] = \frac{2}{d}\sum_{l=0}^{\infty}\mathcal{I}_l. \tag{D25}$$

The trace of a product of matrices represented in the (normalized) Pauli expansion picture, i.e., $A = \sum_{i=1}^{d^2\;\;1} a_i P_i$ and $B = \sum_{j=1}^{d^2\;\;1} b_j P_j$, where P_i, P_j are normalized n-qubit Pauli strings, is given by:

$$\mathrm{tr}(AB) = \sum_{i=0}^{d^2\;\;1} a_i b_i. \tag{D26}$$

The adjoint operators are composed of commutators, which are traceless, and hence the first Pauli component of the adjoint vanishes. By separating $W = W_0 + W_r$, so that $W_0 = w_0 P_0$ and $W_r = \sum_{i=1}^{d^2\;\;1} w_i P_i$, we can transform

$$\mathrm{tr}\Big(\mathrm{ad}^l_{\bar{H}(\boldsymbol{\theta})}\;\nabla_{\boldsymbol{\theta}}\bar{H}(\boldsymbol{\theta})\big)\,W\Big) = \mathrm{tr}\Big(\mathrm{ad}^l_{\bar{H}(\boldsymbol{\theta})}\;\nabla_{\boldsymbol{\theta}}\bar{H}(\boldsymbol{\theta})\big)\,W_r\Big) = \sum_{p,q=1}^{d}(\lambda_p \quad \lambda_q)^l v_{qp} b_{pq}, \tag{D27}$$

where we reuse the previous notation, where $B = \quad \boldsymbol{\theta}\bar{H}(\boldsymbol{\theta})$, but with the reduced amplitude $[W_r]_{qp} = v_{qp}$. We know, that $W = w_0 I + i\sqrt{\epsilon}A$, with $|w_0|^2 = F$, $\epsilon = 1 \quad F$ and identifying $W_0 = w_0 I$, $W_r = i\sqrt{\epsilon}A$, where A is unitary. We are then left with $\mathbb{E}[|v_{pq}|^2] = \frac{}{d} = \frac{1\;\;F}{d}$. Hence, for the correlated remainder $\mathcal{R}_l = \sum_{l=L+1}^{\infty}\mathcal{I}_l$, the expected value over random Hamiltonians H_0 and H_1 reads:

$$\begin{aligned}
\mathbb{E}[R_L^2] &= \frac{4F(\boldsymbol{\theta})}{d^2}\sum_{p,q=1}^{d}\mathbb{E}[|v_{qp}|^2]\,\mathbb{E}[|b_{qp}|^2]\sum_{l,m=L+1}^{\infty}\frac{(\quad 1)^{l+m}\|\boldsymbol{\theta}\|^{l+m}\,\mathbb{E}\left[(\lambda_p \quad \lambda_q)^{l+m}\right]}{(m+1)!(l+1)!}\\
&= \frac{4F(\boldsymbol{\theta})}{d^2}d(d \quad 1)\frac{[1 \quad F(\boldsymbol{\theta})]\,\Lambda_B^2}{2d^2}\sum_{l,m=L+1}^{\infty}\frac{(\quad 1)^{l+m}\|\boldsymbol{\theta}\|^{l+m}\,\mathbb{E}\left[(\lambda_p \quad \lambda_q)^{l+m}\right]}{(m+1)!(l+1)!}\\
&= \frac{2(d \quad 1)[F(\boldsymbol{\theta}) \quad F^2(\boldsymbol{\theta})]\Lambda_B^2}{d^3}\sum_{l,m=L+1}^{\infty}\frac{(\quad 1)^{l+m}\|\boldsymbol{\theta}\|^{l+m}\,\mathbb{E}\left[(\lambda_p \quad \lambda_q)^{l+m}\right]}{(m+1)!(l+1)!}.
\end{aligned} \tag{D28}$$