Measuring Analytic Gradients of General Quantum Evolution with the Stochastic Parameter Shift Rule

Leonardo Banchi[1,2] and Gavin E. Crooks[3,4]

[1]*Department of Physics and Astronomy, University of Florence, via G. Sansone 1, I-50019 Sesto Fiorentino (FI), Italy*
[2]*INFN Sezione di Firenze, via G. Sansone 1, I-50019 Sesto Fiorentino (FI), Italy*
[3]*X, the moonshot factory (x.company), Mountain View, CA, USA*
[4]*Berkeley Institute for Theoretical Science, Berkeley, CA, USA*

Hybrid quantum-classical optimization algorithms represent one of the most promising application for near-term quantum computers. In these algorithms the goal is to optimize an observable quantity with respect to some classical parameters, using feedback from measurements performed on the quantum device. Here we study the problem of estimating the gradient of the function to be optimized directly from quantum measurements, generalizing and simplifying some approaches present in the literature, such as the so-called parameter-shift rule. We derive a mathematically exact formula that provides a stochastic algorithm for estimating the gradient of any multi-qubit parametric quantum evolution, without the introduction of ancillary qubits or the use of Hamiltonian simulation techniques. Our algorithm continues to work, although with some approximations, even when all the available quantum gates are noisy, for instance due to the coupling between the quantum device and an unknown environment.

I. INTRODUCTION

In the near-term [1] quantum computers will be too noisy and the number of operations, or *depth* of the circuit, will still be too low to reliably implement conventional quantum algorithms that require full quantum error correction[2]. Therefore, alternative algorithms, better suited for exploiting these devices have been proposed, such as the variational quantum eigensolver [3, 4], the quantum approximate optimization algorithm [5], quantum autoencoders [6], quantum simulation [7], and quantum classifiers for machine learning [8–10]. Because of these applications, several companies involved in the development of quantum computers have released software for the manipulation of parametric quantum states [11–15].

Hybrid quantum-classical optimization algorithms, such as the ones mentioned above, try to overcome the limitations of current quantum computers by pairing them with a classical device. In these hybrid strategies, the "hard" part of the algorithm, which typically involves the manipulation of objects living in a high-dimensional Hilbert space, is done by a quantum computer, which is reset after each measurement. The classical routine then iteratively reprograms the quantum computer in such a way that either the output of quantum measurements or the prepared quantum state have the desired property. These iterative schemes allow the use of shorter-depth circuits that can be implemented within the decoherence time of the device. Typically, the manipulation of the quantum state is performed with parametric quantum gates and the role of the classical routine is to update those parameters either via gradient descent or gradient ascent. Evaluating the gradient of a quantum circuit is as hard as the evaluation of the circuit itself, and therefore it is important to use the quantum computer for estimating it. Several algorithms have been proposed for such purpose, either based on a generalization of the Hadamard test [16, 17] or on the so-called parameter shift rule [18–20], which have a similar complexity. Nonetheless, both algorithms can only be applied when the parametric gates can be written as $e^{i\theta_t \hat{X}_t}$, for parameters θ_t, and where the operators $\hat{X}_t$ have certain special properties. In the general case one has to resort to Hamiltonian simulation techniques [21] that increase the complexity of the algorithm.

Here we show that the parameter-shift rule can be generalized to any multi-qubit quantum evolution, without the need to introduce any ancillary system or Hamiltonian simulation techniques. Our generalization is based on a stochastic strategy that is exact in the limit of many repetitions of the quantum measurement. We analyse the number of repetitions needed to achieve a certain precision by studying the variance of our estimation procedure, and numerically observe that it is comparable to that of the standard parameter shift rule. In near-term computers, unitary gates are an approximation to a more complex, noisy evolution that couples the qubit registers to an unknown environment. We show that our estimation procedure can be applied even when the coupling between system and environment cannot be completely suppressed, and when the gates depend on the parameters in a complex way.

Our paper is organized as follows: in Sec. II we set up the problem and the notation; in Sec. III we discuss the main ideas and introduce analytical formulae and algorithms for estimating the gradient in the general case; in Sec. IV we study applications in quantum control and for optimizing noisy gates; Conclusions are drawn in Sec. V. Explicit pseudo-codes for our algorithms are given in Appendix A. The stochastic variance of our algorithms is studied in Appendix B.

II. BACKGROUND AND NOTATION

We focus on parametric quantum states $|\psi(\boldsymbol{\theta})\rangle$ that depend parametrically on P classical real parameters $\{\theta_p\}$ with $p = 1, \dots, P$. These states are obtained by applying a unitary $\hat{U}(\boldsymbol{\theta})$ onto a $\boldsymbol{\theta}$-independent reference state $|\psi_0\rangle$

$$|\psi(\boldsymbol{\theta})\rangle = \hat{U}(\boldsymbol{\theta})\,|\psi_0\rangle \ . \tag{1}$$

We study the optimization (either maximization or minimization) of the expected value of an observable $\hat{C}$, taken with respect to $|\psi(\boldsymbol{\theta})\rangle$

$$C(\boldsymbol{\theta}) = \langle\psi_0|\,\hat{U}(\boldsymbol{\theta})^\dagger \hat{C}\hat{U}(\boldsymbol{\theta})\,|\psi_0\rangle\ . \tag{2}$$

Several problems can be mapped to the above optimization, such as variational diagonalization and quantum simulation [3, 5, 22], where $\hat{C}$ is the Hamiltonian of a many-body system and the task is to variationally approximate its ground state; and quantum state synthesis, where $\hat{C} = |\psi_{\text{target}}\rangle\langle\psi_{\text{target}}|$, or some machine-learning classifiers [8, 16]. Even quantum control problems [23] or the simulation of gates with time-independent Hamiltonians [24, 25] can be written in the form (2). Indeed, consider the task of finding a good approximation of a certain target unitary gate $\hat{G}$ with a parametric unitary $\hat{U}(\boldsymbol{\theta})$. We may define $|\psi(\boldsymbol{\theta})\rangle = \hat{\mathbb{1}} \otimes \hat{U}(\boldsymbol{\theta})\,|\Phi\rangle$, where $|\Phi\rangle = \sum_{i=1}^{d} |i,i\rangle / \sqrt{d}$ and d is the dimension of the Hilbert space, and similarly $|\psi_{\text{target}}\rangle = \hat{\mathbb{1}} \otimes \hat{G}\,|\Phi\rangle$. Then, from (2) with $\hat{C} = |\psi_{\text{target}}\rangle\langle\psi_{\text{target}}|$, we find

$$C(\boldsymbol{\theta}) = \left(\frac{|\operatorname{Tr}\hat{G}^\dagger\hat{U}(\boldsymbol{\theta})|}{d}\right)^2\ , \tag{3}$$

which is the function normally maximized in quantum control problems [23, 24].

Any unitary operator can be expressed as a matrix exponential $\hat{U}(\boldsymbol{\theta}) = e^{i\hat{X}(\boldsymbol{\theta})}$, where $\hat{X}(\boldsymbol{\theta})$ is a Hermitian operator. When the unitary $\hat{U}(\boldsymbol{\theta})$ is a composition of T simpler gates $\hat{U}_t(\boldsymbol{\theta})$, then we write

$$\hat{U}(\boldsymbol{\theta}) = \prod_{t=1}^{T} U_t(\boldsymbol{\theta}), \qquad \hat{U}_t(\boldsymbol{\theta}) = e^{i\hat{X}_t(\boldsymbol{\theta})}\ , \tag{4}$$

where the products are ordered as $\prod_{t=1}^{T}\hat{U}_t := \hat{U}_T\cdots\hat{U}_1$. The products of Pauli matrices $\hat{\sigma}_{\boldsymbol{\nu}} = \hat{\sigma}_{\nu_1}\otimes\cdots\otimes\hat{\sigma}_{\nu_N}$ form a basis for the space of N-qubit Hermitian operators [26], where $\boldsymbol{\nu} = (\nu_1,\ldots,\nu_N)$ is a multi index, ν_j is either $\{0,x,y,z\}$ and $\hat{\sigma}_0 := \hat{\mathbb{1}}$, $\hat{\sigma}_x$, $\hat{\sigma}_y$, $\hat{\sigma}_z$ are the Pauli matrices. As such, we may expand the operators $\hat{X}_t(\boldsymbol{\theta})$ onto this basis and write

$$\hat{X}_t(\boldsymbol{\theta}) = \sum_{\boldsymbol{\nu}} x_{t,\boldsymbol{\nu}}(\boldsymbol{\theta})\,\hat{\sigma}_{\boldsymbol{\nu}}\ , \tag{5}$$

with coefficients $x_{t,\boldsymbol{\nu}}(\boldsymbol{\theta}) = \operatorname{Tr}\big[\hat{X}_t(\boldsymbol{\theta})\hat{\sigma}_{\boldsymbol{\nu}}\big]/2^N$. It is common to restrict attention to gates that only have a single element in the expansion (5), i.e. $x_{t,\boldsymbol{\mu}}(\boldsymbol{\theta}) = \theta_t\delta_{\boldsymbol{\mu},\boldsymbol{\nu}(t)}$ and

$$\hat{U}_t^{\text{simple}} = \exp(i\theta_t\,\hat{\sigma}_{\boldsymbol{\nu}(t)})\ , \tag{6}$$

where $\boldsymbol{\nu}(t)$ specifies the kind of parametric gate applied at time t. Moreover, most often we consider gates that act on either one- or two-qubit, so at most two Pauli matrices in the product $\hat{\sigma}_{\nu_1}\otimes\cdots\otimes\hat{\sigma}_{\nu_N}$ are different from the identity.

On the other hand, in this paper we do not restrict ourselves to the case (6) and consider the more general parametrization (4) with (5). Via the Leibniz rule, we may write

$$\frac{\partial C(\boldsymbol{\theta})}{\partial\theta_p} = \sum_{t,\boldsymbol{\nu}}\frac{\partial C}{x_{t,\boldsymbol{\nu}}}\,\frac{\partial x_{t,\boldsymbol{\nu}}(\boldsymbol{\theta})}{\partial\theta_p}\ . \tag{7}$$

When the parametrization is such that all gates can be expressed as in Eq. (6), different approaches have been proposed to evaluate the gradient via a carefully designed quantum circuit and classical post-processing, for instance using the Hadamard test [17] or the parameter shift rule [18, 19]. In [20] the parameter shift rule was generalized to some particular cases where there are more than one term in the expansion (5). However, finding the gradient in the general case was still an open question. In the next section we show that by mixing standard operator derivative techniques [27] with Monte Carlo strategies, we can define a procedure to measure gradients of any $C(\boldsymbol{\theta})$, as in Eq. (2), with near-term quantum hardware.

Thanks to the Leibniz rule (7), we may fix the value of t and $\boldsymbol{\nu}$ and study the derivative of C with respect to $x_{t,\boldsymbol{\nu}}$. By repeating the analysis for each possible values of t and $\boldsymbol{\nu}$, from Eq. (7) we may obtain the derivatives with respect to the parameters θ_p, and hence the gradient. Therefore, we fix t and $\boldsymbol{\nu}$ and, to simplify the notation, we drop the dependence on t and $\boldsymbol{\nu}$ to write

$$x := x_{t,\boldsymbol{\nu}}, \qquad \hat{V} := \sigma_{\boldsymbol{\nu}}, \qquad \hat{H} := \sum_{\boldsymbol{\mu}\neq\boldsymbol{\nu}} x_{t,\boldsymbol{\mu}}\hat{\sigma}_{\boldsymbol{\mu}}\ . \tag{8}$$

With a similar spirit, we also define

$$|\phi\rangle := \prod_{s=1}^{t-1}\hat{U}_s\,|\psi_0\rangle\ , \qquad \hat{A} := \hat{U}_{t+}^\dagger\hat{C}\hat{U}_{t+}\ , \tag{9}$$

where $\hat{U}_{t+} = \prod_{s=t+1}^{T}\hat{U}_s$. Thanks to the above simplified notation, we may write the function C in (2) as a function of $x \equiv x_{t,\boldsymbol{\nu}}$ for *fixed* t and $\boldsymbol{\nu}$

$$C(x) = \langle\phi|\,e^{-i(\hat{H}+x\hat{V})}\,\hat{A}\,e^{i(\hat{H}+x\hat{V})}\,|\phi\rangle\ , \tag{10}$$

all the other terms in (10) do not explicitly depend on $x \equiv x_{t,\boldsymbol{\nu}}$. In other words, Eq. (10) is equivalent to Eq. (2), where we have separated the terms that depend on $x \equiv x_{t,\boldsymbol{\nu}}$ for fixed t and $\boldsymbol{\nu}$ from the others.

III. STOCHASTIC PARAMETER SHIFT RULE

Without loss of generality, we fix t and $\boldsymbol{\nu}$ as described in the previous section, and study the derivative of $C(x)$ defined in (10). The derivative with respect to the parameters θ_p can be obtained from (7) by repeating the analysis for all t and $\boldsymbol{\nu}$. We remark that in Eq. (10) the state $|\phi\rangle$ and the operators $\hat{H}$, $\hat{A}$, $\hat{V}$ explicitly depend on t, $\boldsymbol{\nu}$, and on the other values $x_{t',\boldsymbol{\nu}'}$ with either $t' \neq t$ or $\boldsymbol{\nu}' \neq \boldsymbol{\nu}$, but we omit this dependence to simplify the notation. Full algorithms are shown in Appendix A.

The main tool behind our analysis is the following operator identity [27]

$$\frac{\partial e^{\mathcal{Z}}}{\partial x} = \int_0^1 ds\, e^{s\mathcal{Z}}\frac{\partial\mathcal{Z}}{\partial x}e^{(1-s)\mathcal{Z}}\ , \tag{11}$$

which is valid for any operator $\mathcal{Z}$. We may rewrite Eq. (10) as

$$C(x) = \operatorname{Tr}\Big(\hat{A}e^{\mathcal{Z}}[\hat{\rho}]\Big)\ , \tag{12}$$

Algorithm 1 Parameter Shift Rule

1: initialize the computer in the state $|\phi\rangle$, following the preparation routine (9);
2: apply the gate $e^{i(x+\frac{\pi}{4u})\hat{V}}$;
3: measure the observable $\hat{A}$ from (9) and call the result r_+.
4: Repeat steps 1 to 3, but on point 2 apply $e^{i(x-\frac{\pi}{4u})\hat{V}}$ rather than $e^{i(x+\frac{\pi}{4u})\hat{V}}$;
5: measure $\hat{A}$ and call the result r_-.
6: the sample $g_{t,\nu} = u(r_+ - r_-)$ is such that $\partial C/\partial x_{t,\nu} = \mathbb{E}[g_{t,\nu}]$.

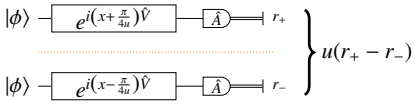

FIG. 1. Parameter Shift Rule [18, 19, 29], only applicable to parametric gates as in Eq. (6) or, more generally, to parametrizations $e^{ix\hat{V}}$ where $\hat{V}$ has two distinct eigenvalues $\pm u$. When $\hat{V}$ is a product of Pauli matrices as in (6), $u = 1$. In the algorithm we consider the derivative $\partial_x C(x)$ of Eq. (10), when $\hat{H} = 0$.

for $\hat{\rho} = |\phi\rangle\langle\phi|$ and for a superoperator $\mathcal{Z}[\hat{\rho}] := [i(\hat{H} + x\hat{V}), \hat{\rho}]$, where $[\hat{A}, \hat{B}] = \hat{A}\hat{B} - \hat{B}\hat{A}$. Eq. (12) then follows from Baker-CampbellHausdorff identity $e^{[\hat{X},\cdot]}\hat{Y} = e^{\hat{X}}\hat{Y}e^{-\hat{X}}$ [28]. We also introduce the superoperator

$$\mathcal{V} := \frac{\partial \mathcal{Z}}{\partial x} = i[\hat{V}, \cdot]. \tag{13}$$

Now we focus on the exponential $e^{\lambda\mathcal{V}}$ with $\mathcal{V}$ defined in (13). From series expansion, since $\hat{V}$ is a tensor product of Pauli matrices (8) and, as such, $\hat{V}^2 = \hat{\mathbb{1}}$, it is simple to show that

$$e^{\lambda\mathcal{V}}[\hat{\rho}] = \hat{\rho} + \sin^2(\lambda)(\hat{V}\hat{\rho}\hat{V} - \hat{\rho}) + \frac{i}{2}\sin(2\lambda)[\hat{V}, \hat{\rho}]\,, \tag{14}$$

from which we get

$$\mathcal{V}e^{\lambda\mathcal{V}}[\hat{\rho}] \equiv \frac{\partial e^{\lambda\mathcal{V}}}{\partial\lambda}[\hat{\rho}] = e^{(\lambda+\pi/4)\mathcal{V}}[\hat{\rho}] - e^{(\lambda-\pi/4)\mathcal{V}}[\hat{\rho}]\,. \tag{15}$$

When $\hat{H} \equiv 0$, it is $\mathcal{Z} = x\mathcal{V}$ and we may use the above equation with $\lambda = x$ to take derivatives in (12). As a result, we get $\partial_x C(x) = C(x + \pi/4) - C(x - \pi/4)$, which is the so-called parameter shift rule, described in Fig. 1, often used for training quantum circuits [18, 19, 29]. Note that, with the formalism of the previous section, $\hat{H} = 0$ corresponds to the use of the simper parametric unitaries of Eq. (6). A more general version of the parameter shift rule can be obtained when the operator $\hat{V}$ has only two distinct eigenvalues [18, 19]. Indeed, we note that the only property we used in (15) is $\hat{V}^2 = \hat{\mathbb{1}}$, which is true for any product of Pauli matrices. If $\hat{V}$ has only two possible eigenvalues $c \pm u$, then we may write $\hat{V} = u\hat{V}' + c\hat{\mathbb{1}}$ where $\hat{V}'^2 = \hat{\mathbb{1}}$ and the dependence on c disappears in (13). Therefore, it is straightforward to generalize the above derivation and find $\partial_x C(x) = u\left[C\left(t + \frac{\pi}{4u}\right) - C\left(t - \frac{\pi}{4u}\right)\right]$. The resulting algorithm is described in Fig. 1. Although the parameter shift rule can be made slightly more general, for instance by replacing the operator $\hat{\sigma}_{\nu(t)}$ in (6) with another operator that has,

Algorithm 2 Stochastic Parameter Shift Rule

1: Sample s from the uniform distribution in [0,1];
2: initialize the computer in the state $|\phi\rangle$, following the preparation routine (9);
3: apply the gate $e^{i(1-s)(\hat{H}+x\hat{V})}$, namely where parameters $x_{t,\mu}$ for fixed t and all possible values of $\boldsymbol{\mu}$ have been rescaled by a factor $(1 - s)$;
4: apply the gate $e^{i\pi\hat{V}/4} \equiv e^{i\pi\hat{\sigma}_\nu/4}$;
5: apply the gate $e^{is(\hat{H}+x\hat{V})}$, where parameters $x_{t,\nu}$ for fixed t and all possible values of $\boldsymbol{\nu}$ have been rescaled by a factor s;
6: measure the observable $\hat{A}$ from (9) and call the result r_+.
7: Repeat steps 2 to 5, but on point 4 apply $e^{i\pi\hat{\sigma}_\nu/4}$ rather than $e^{-i\pi\hat{\sigma}_\nu/4}$;
8: measure $\hat{A}$ and call the result r_-.
9: the sample $g_{t,\nu} = r_+ - r_-$ is such that $\partial C/\partial x_{t,\nu} = \mathbb{E}[g_{t,\nu}]$.

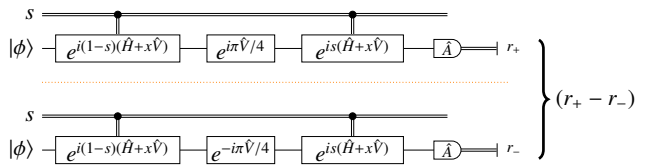

FIG. 2. Stochastic Parameter Shift Rule, valid for any operator $\hat{H}$. In the picture, the values of the classical parameter s are the same.

like $\hat{\sigma}_{\nu(t)}$, only two possible degenerate eigenvalues, it cannot be applied in the general case where $\hat{H} \neq 0$. Nonetheless, we show that the parameter shift rule can be generalized by combining Eq. (15) with Eq. (11). Indeed, evaluating (15) for $\lambda = 0$ we get

$$\frac{\partial\mathcal{Z}[\hat{\rho}]}{\partial x} = i[\hat{V}, \hat{\rho}] = e^{i\pi\hat{V}/4}\hat{\rho}e^{-i\hat{V}\pi/4} - e^{-i\pi\hat{V}/4}\hat{\rho}e^{i\hat{V}\pi/4}\,. \tag{16}$$

From the above equation, calling

$$C_\pm(x, s) = \langle\phi|\, U_\pm(x, s)^\dagger\, \hat{A}\, U_\pm(x, s)\,|\phi\rangle\,, \tag{17}$$

$$U_\pm(x, s) = e^{is(\hat{H}+x\hat{V})}e^{\pm i\frac{\pi}{4}\hat{V}}e^{i(1-s)(\hat{H}+x\hat{V})}\,, \tag{18}$$

we get from (11) and (16)

$$\partial_x C(x) = \int_0^1 [C_+(x, s) - C_-(x, s)]\, ds\,. \tag{19}$$

Eqs.(17)-(19) represent the central result of this paper. Thanks to those formulae, we introduce the Stochastic Parameter Shift Rule, shown in Fig 2. The derivation of the Stochastic Parameter Shift Rule from Eq.(19) is straightforward. Indeed, let $\hat{A} = \sum_m a_m |m\rangle\langle m|$ be the eigenvalue decomposition of $\hat{A}$. Then, because of the Born rule, the outcomes $r_\pm$ are one of the possible values a_m with probability

$$p_\pm(m|s) = |\langle m|\, U_\pm(x, s)\,|\phi\rangle|^2\,. \tag{20}$$

Taking the expectation value with respect to the measurement outcomes and with respect to the uniform probability over s we get from Eq. (19)

$$\mathbb{E}[r_+ - r_-] = \partial_x C(x)\,, \tag{21}$$

which concludes the proof. For a single measurement, both the Parameter Shift Rule of Fig. 1 and the Stochastic Parameter Shift Rule of Fig. 2 provide a random difference between two eigenvalues of $\hat{A}$. Only in the limit over many repetitions of those algorithms does the average over the outcomes converge to the exact value of the gradient. Due to the Chebyshev inequality, the number of repetitions to achieve a certain precision depends on the variance of the random outcomes. In Appendix B we study the variance of the gradient estimator obtained with the Stochastic Parameter Shift Rule and show that it is comparable with that of the standard Parameter Shift Rule.

A. Stochastic optimization

In the previous section we have introduced an algorithm (Fig. 2) to use a quantum computer to sample from a random variable whose average is equal to the gradient of a certain circuit. We say that the output of the Stochastic Parameter Shift Rule provides an *unbiased estimator* of the gradient, in the sense of Eq. (21).

We now focus on the original problem, namely a parametric unitary (4) with many parameters as in (5). We can use the algorithm of Fig. 2 to sample $g_{t,\boldsymbol{\nu}}$ with the property $\partial C/\partial x_{t,\boldsymbol{\nu}} = \mathbb{E}[g_{t,\boldsymbol{\nu}}]$. By repeating the procedure many times and with all possible values of t and $\boldsymbol{\nu}$, due the linearity of the Leibniz rule (7), we may write

$$\frac{\partial C(\boldsymbol{\theta})}{\partial \theta_p} = \mathbb{E}\left[\sum_{t,\boldsymbol{\nu}} g_{t,\boldsymbol{\nu}} \, \frac{\partial x_{t,\boldsymbol{\nu}}(\boldsymbol{\theta})}{\partial \theta_p}\right], \tag{22}$$

where the expectation value $\mathbb{E}$ has the same meaning as in Eq. (21). The full algorithm is shown in Appendix A, algorithm 4. The problem with this approach is that we have to repeat algorithm 2 many times, each time resetting the quantum machine, to get a single sample.

We now introduce a simpler unbiased estimator of the gradient that requires significantly fewer operations to get a single sample. A similar technique has been developed in [17, 29] for parametrizations as in Eq. (6), which was dubbed *doubly stochastic* gradient descend. Here we generalize that approach to general quantum evolution, as in Eq. (4). We start by defining a probability distribution from the "weights" $\partial_p x_{t,\boldsymbol{\nu}}(\boldsymbol{\theta})$, where $\partial_p \equiv \frac{\partial}{\partial \theta_p}$, as

$$q_p(t,\boldsymbol{\nu}) = \frac{1}{\mathcal{N}}\left|\frac{\partial x_{t,\boldsymbol{\nu}}(\boldsymbol{\theta})}{\partial \theta_p}\right|, \qquad \sum_{t,\boldsymbol{\nu}} q_p(t,\boldsymbol{\nu}) = 1\,, \tag{23}$$

with $\mathcal{N} = \sum_{t,\boldsymbol{\nu}} |\partial_p x_{t,\boldsymbol{\nu}}(\boldsymbol{\theta})|$. Setting $n_{p,t,\boldsymbol{\nu}} = \mathcal{N}\mathrm{sign}\left(\partial_p x_{t,\boldsymbol{\nu}}(\boldsymbol{\theta})\right)$ we may then write Eq. (22) as

$$\frac{\partial C(\boldsymbol{\theta})}{\partial \theta_p} = \mathbb{E}_{(t,\boldsymbol{\nu})\sim q_p}\left[n_{p,t,\boldsymbol{\nu}}\, \mathbb{E}\left(g_{t,\boldsymbol{\nu}}\right)\right], \tag{24}$$

where $\mathbb{E}_{(t,\boldsymbol{\nu})\sim q}$ means that, at each iteration, t, and $\boldsymbol{\nu}$ are sampled from the distribution (23). When the functional dependence on the parameters is known, all quantities $q_p(t,\boldsymbol{\nu})$ and $n_{p,t,\boldsymbol{\nu}}$ can be easily computed at each iteration without having to deal with exponentially large spaces. The above equation (24) allows us to define a simple "doubly stochastic" gradient estimator via the following rule

1: sample t and $\boldsymbol{\nu}$ from the distribution (23);
2: use Algorithm 2 to get an estimate $g_{t,\boldsymbol{\nu}}$;
3: the sample $r_{p,t,\boldsymbol{\nu}} = g_{t,\boldsymbol{\nu}} n_{p,t,\boldsymbol{\nu}}$ is such that $\partial C/\partial\theta_p = \mathbb{E}[r_{p,t,\boldsymbol{\nu}}]$.

The full algorithm is shown in Appendix A, algorithm 5. Based on the above equation, in Appendix A we also define an algorithm that can provide an unbiased sample with a single initialization of the quantum device, algorithm 6.

To summarize the results of this section, we can use either (22) or (24) to estimate the gradient of an expectation value (2) with a quantum computer. Once we have an estimate of the gradient, we can optimize $C(\boldsymbol{\theta})$ using stochastic gradient descent (or ascent) algorithms [30], such as Adam [31]. These algorithms are classical, in the sense that, given certain parameters $\boldsymbol{\theta}$ and an estimate of the gradient $\boldsymbol{g}$, the parameters are updated as $\boldsymbol{\theta} \to \boldsymbol{\theta} \pm \eta \boldsymbol{g}$ for a suitably small *learning rate* η. Therefore, we can use a *hybrid* quantum-classical approach to optimize $C(\boldsymbol{\theta})$ where the hard calculations, namely the estimation of the gradients, are delegated to a quantum computer, while the update of the parameters is performed classically.

B. Quantum gates with unavoidable drift

Depending on the hardware, the application of the gates $e^{\pm i\pi\hat{V}/4}$ in Algorithm 2 might be problematic. Let us consider a quantum computer that can only apply the parametric gates

$$\hat{U}(t,b) = e^{it(\hat{H}_0 + b\hat{H}_1)}\,, \tag{25}$$

where $\hat{H}_0$ is some *drift* Hamiltonian that cannot be completely switched off, aside from the trivial case $t = 0$. Such "simple" device is still capable of universal quantum computation, provided that the operators $\hat{H}_0$ and $\hat{H}_1$ are multi-qubit operators that generate the full Lie algebra [32]. Here though, for simplicity, we consider the case where both $\hat{H}_0$ and $\hat{H}_1$ are tensor products of Pauli operators, as introduced in Sec. II. The parameters in the above gate are $\boldsymbol{\theta} = (t, b)$. Using the notation of Eq. (5) we may write

$$\hat{U}(t,b) = e^{i(x_0\hat{H}_0 + x_1\hat{H}_1)}\,, \tag{26}$$

where $x_0 = t$ and $x_1 = bt$. Employing the above gate in Eq. (2), from (7) we get

$$\partial_t C = \frac{\partial C}{\partial x_0} + \frac{\partial C}{\partial x_1} b\,, \qquad \partial_b C = \frac{\partial C}{\partial x_1} t\,. \tag{27}$$

An estimator of $\frac{\partial C}{\partial x_j}$ for $j = 0, 1$ can be obtained with Algorithm 2, where $\hat{V}$ is, respectively, either $\hat{H}_0$ or $\hat{H}_1$. Step 3 in the algorithm corresponds to $\hat{U}((1-s)t, b)$ and Step 5 corresponds to $\hat{U}(st, b)$, so both operations can be easily implemented directly in the device. Step 4 corresponds to the gate $U(\pi/4, 0)$ when estimating $\frac{\partial C}{\partial x_0}$, which is again easy to implement. However, Step 4 for estimating $\frac{\partial C}{\partial x_1}$ corresponds to the gate $e^{i\pi\hat{H}_1/4}$

Algorithm 3 Approximate Stochastic Parameter Shift Rule

1: Sample s from the uniform distribution in [0,1];
2: **for** $m = \{+, -\}$ **do**
3: initialize the computer in the state $|\phi\rangle$;
4: apply the gate $e^{i(1-s)(\hat{H}+x\hat{V})}$;
5: apply the gate $e^{i\epsilon[\hat{H}\pm\pi/(4\epsilon)\hat{V}]}$ where the sign depends on m;
6: apply the gate $e^{is(\hat{H}+x\hat{V})}$;
7: measure the observable $\hat{A}$ and call the result r_m.
8: **end for**
9: An estimate $g_{t,v}$ of $\partial C/\partial x_{t,v}$ is given by $g_{t,v} = r_+ - r_-$.

FIG. 3. Approximate Stochastic Parameter Shift Rule. A compact notation has been used, as this algorithm is identical to the one in Fig. 2, except for the use of the imperfect gates $e^{i\epsilon[\hat{H}\pm\pi/(4\epsilon)\hat{V}]}$ in lieu of $e^{\pm\pi/4\hat{V}}$.

that does not belong to the set of gates (25) and, with our assumptions, cannot be implemented by the device. However, we may substitute that gate with an approximation

$$U\left(\epsilon, \frac{\pi}{4\epsilon}\right) = e^{i(\epsilon\hat{H}_0 \pm \frac{\pi}{4}\hat{H}_1)} = e^{\pm i\frac{\pi}{4}\hat{H}_1} + O(\epsilon)\,. \tag{28}$$

The error coming from the drift term can be small $O(\epsilon)$ if it is possible to set b to a high value $O(\epsilon^{-1})$. With the above gate, in Fig. 3 we define the approximate Stochastic Parameter Shift Rule. The approximate gate introduces a bias in the gradient estimator, but since such bias can be made small, convergence can still be expected [33].

As a relevant example, we study the cross-resonance gate [19, 20]

$$\hat{U}_{\rm CR}(t, b, c) = \exp\left[it\left(\hat{\sigma}_x\otimes\hat{\mathbb{1}} - b\,\hat{\sigma}_z\otimes\hat{\sigma}_x + c\,\hat{\mathbb{1}}\otimes\hat{\sigma}_x\right)\right]\,, \tag{29}$$

a natural gate for certain microwave-controlled transmon superconducting qubit architectures [34]. The results are shown in Fig. 4 for different values of t, c and b, where we show that Algorithms 2 and 3 are basically indistinguishable from each other, and very close to the approximated value obtained numerically, without any randomness, using a finite difference approximation. All numerical results are obtained by analytically computing the probabilities (20) and then simulating the quantum measurement via Monte Carlo sampling. The finite difference approximation is obtained as $\partial_x C(x) \approx (2\varepsilon)^{-1}[C(x+\varepsilon) - C(x-\varepsilon)]$. Note that, although this approximation works fine for numerical approximations using a classical computer, it is not useful for calculating gradients on quantum hardware. Indeed, if we use a quantum device for estimating $C(x \pm \varepsilon)$, then the estimator of ∂C_x has a variance $\approx \varepsilon^2$ which is very high when ε is small.

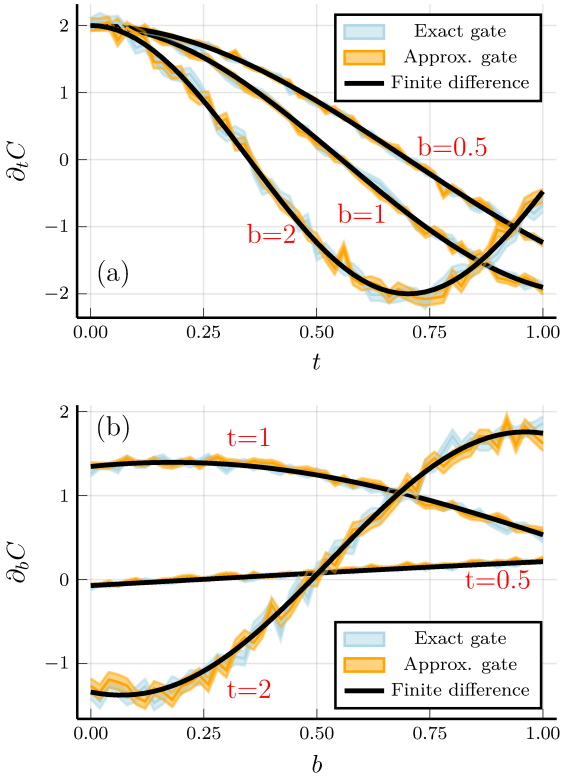

FIG. 4. Gradient of Eq. (2) when using the cross-resonance gate (29). In (a) we study $\partial_t C$ for $c = 0$ and fixed values of $b = \{0.5, 1, 2\}$ with $\hat{C} = \hat{\sigma}_y \otimes \hat{\mathbb{1}}$. In (b) we study $\partial_b C$ for $c = \sqrt{2}$ and fixed values of $t = \{0.5, 1, 2\}$ with $\hat{C} = \hat{\sigma}_y \otimes \hat{\sigma}_y$. We compare the finite difference approximation, with the estimations from Algorithm 2 (Exact gate) or 3 (Approx. gate). Approximated gates are with $\epsilon = 10^{-2}$. Data for the stochastic algorithms are obtained from (22) with 1000 samples. Coloured regions represent the area $m \pm \sigma$ where m is the estimated mean and σ the standard error of the mean.

IV. APPLICATIONS

A. Quantum control with drift

The control of a quantum system is obtained by modulating the interactions via time-dependent pulses. Calling $\lambda_j(t)$ the external pulses and $\hat{V}_j$ the associated operators, the evolution is described by the following time-dependent Hamiltonian

$$\hat{H}(t) = \hat{H}_0 + \sum_{j=1}^{M} \lambda_j(t)\hat{V}_j\,, \tag{30}$$

where M is the number of pulses and $\hat{H}_0$ is the *drift Hamiltonian* that describes the time-evolution of the system when no-pulses are applied. Here we consider $M = 1$ as the generalization is straightforward, and set $\lambda_1 \equiv \lambda$ and $\hat{V}_1 \equiv \hat{V}$. By discretizing the control time T into $N_T = T/\Delta T$ steps of width Δt we get

$$\hat{U}(T) \approx \prod_{p=1}^{N_T} e^{-i\Delta T(\hat{H}_0 + \lambda(p\Delta T)\hat{V})} \tag{31}$$

with error $\approx N_T \Delta T^2$. Pulse design corresponds to the optimization of the parameters $\theta_p := \lambda(p\Delta T)$ to achieve a desired target evolution [23], for which we can apply the procedure of the section III. An alternative is to expand the pulse

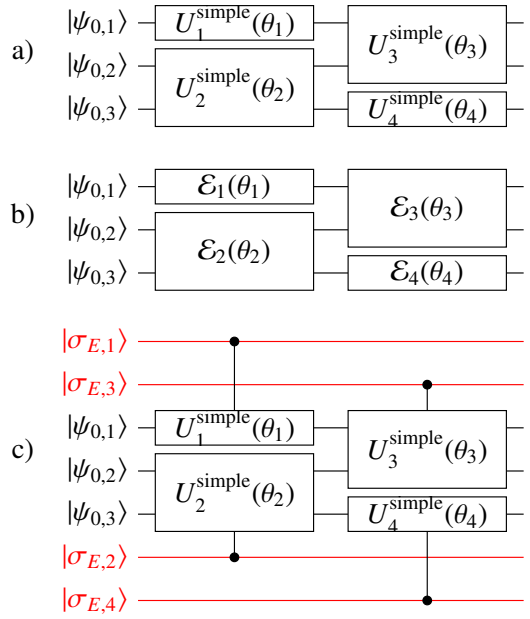

FIG. 5. a) An example parametric quantum circuit with parametric gates as in Eq. (6). b) A noisy version of a), where unitary gates are replaced by non-unitary channels. c) A representation (32) of the noisy gates in b), where each noisy operation is represented as a unitary gate between the qubits and an independent environment (in red).

in the Fourier basis $\lambda(t) = \sum_m a_m \cos(\omega_k t + \phi_k)$ for some frequencies ω_k and tunable amplitudes a_m and phases ϕ_k [35]. Therefore, we may use (7) together with the procedure of Sec. III to estimate the gradient with respect to the parameters $\{\theta_p\} = \{a_m, \phi_m\}$.

B. Parametric circuits with noisy quantum gates

One of the main strengths of our Algorithm 3, and its generalizations in Appendix A, is its ability to work, under reasonable assumptions, even when parametric gates are not perfectly implemented by the device. This is the case in currently available and near-term quantum computers [1].

As a relevant example, consider the quantum circuit of Fig. 5, built from simple parametric gates as in Eq. (6). When the quantum computer can apply the exact gates, then the standard parameter shift rule can be employed. However, quantum devices are always in contact with their surrounding environment, so an exact application of the gate is impossible (without full quantum error correction). More precisely, due to the action of the environment the gate is not unitary but, under some reasonable approximations, can be described by a completely positive map [36, 37]. A completely positive map can always be written as a unitary evolution on the register and its environment. For simplicity let us consider a perfect gate as in (6) with fixed t. Physically, the perfect gate (6) means that a control Hamiltonian $\hat{H}_t^{(R)} := -\hat{\sigma}_{\nu(t)}$ is switched on for a time θ_t, where the index (R) reminds us that the Hamiltonian acts on the registers R only. In realistic implementations the register is coupled with its own environment. If we call $H_t^{(RE)}$ the coupling Hamiltonian between register (R) and environment (E), then we may write the non-unitary gate (see also Fig. 5c) as

$$\mathcal{E}_t(\theta)[\rho] = \text{Tr}_E\left[e^{-i\tau[H_t^{(RE)}+\theta H_t^{(R)}]}\rho_R \otimes \sigma_E e^{i\tau[H_t^{(RE)}+\theta H_t^{(R)}]}\right], \quad (32)$$

where σ_E is the state of the environment, τ is the control time, and θ is the relative strength between $H_t^{(R)}$ and $H_t^{(RE)}$. In Eq. (32) there are three main approximations: i) we neglect any initial quantum correlation between register and environment, so that the non-unitary evolution can be modeled as a completely positive map [36], which in turn implies (32); ii) we assume that the (unknown) initial state of the environment does not depend on θ and τ; iii) we assume that it is possible to tune both τ and, to some extent, the relative strength θ. Under these three conditions, it is possible to use Algorithm 3 and its generalizations of Appendix A to compute the gradient with respect to θ. Indeed, without loss of generality, we may consider $H_t^{(R)}$ as a product of Pauli matrices acting on the register R. When this is not the case we may employ the Leibniz rule (7). All operations in Algorithm 3 are possible, with the substitution $\hat{H}_0 = \hat{H}_t^{(RE)}$ and $\hat{V} = \hat{H}_t^{(R)}$. The rescaled gates correspond to reducing the control time τ by either a factor $(1 - s)$ or s, while the application of the approximate gate (28) can be obtained by making θ large. Note that in a *good* quantum computer, the factor θ should always be large, as the coupling between register and environment should be small. Therefore, derivatives with respect to θ can be obtained using the same operations available in the device.

On the other hand, derivatives with respect to τ are, in general, not possible. We may always expand the coupling Hamiltonian in the Pauli basis via (5) and use the Leibniz rule (7), but in order to obtain the derivative with our Algorithm (3), we have to approximate a highly tuned gate of type $e^{i\pi/4\hat{\sigma}^{(RE)}}$, which couples the system and environment. We believe that for reasonable models of environment, this is not generally possible.

In summary, when the noisy evolution can be written as in Eq. (32), under the approximations defined above, derivatives with respect to θ can be obtained with the same operations available in the machine, while the further parameter τ should only be used to implement the rescaling and not as an optimization parameter.

C. Quantum Natural Gradient

The quantum natural gradient has been proposed in [38, 39] as a way to better describe the geometry of parametric quantum states, enabling faster convergence towards local optima. With the quantum natural gradient the update rule becomes $\boldsymbol{\theta} \to \boldsymbol{\theta} \pm \eta \tilde{F}^{-1}\boldsymbol{g}$, where $\boldsymbol{g}$ is the gradient and $\tilde{F}$ the metric tensor. The role of the metric tensor for noisy parametric quantum evolution has been studied first in [40], where it was shown that it provides a method to investigate the convergence time of standard stochastic gradient descend. When us-

ing the simple parametric gates of (6), the elements of this tensor can be measured efficiently [22, 38]. Moreover, recently the quantum natural gradient has been extended to arbitrary noisy quantum states [39]. In particular, for slightly mixed states it is

$$\tilde{F}_{p,p'} \approx \kappa F_{p,p'}, \qquad F_{p,p'} := \mathrm{Tr}\left[\frac{\partial\hat{\rho}}{\partial\theta_p}\frac{\partial\hat{\rho}}{\partial\theta_q}\right], \tag{33}$$

where $\kappa = 1$ for pure states and $\hat{\rho}$ is the state after the parametric unitaries that, for either noiseless or noisy gates, we can write as $\hat{\rho}(\boldsymbol{\theta}) = \mathcal{E}_T(\boldsymbol{\theta}) \circ \cdots \circ \mathcal{E}_1(\boldsymbol{\theta})[\hat{\rho}_0]$ with $\hat{\rho}_0 = |\psi_0\rangle\langle\psi_0|$. We focus on $F_{p,p'}$ as the parameter κ can be absorbed into the learning rate. The approximation in (33) is valid when the state has a high purity [39] , as it is expected in *good* NISQ computers. We may measure the matrix in Eq. (33) using a combination of the Stochastic Parameter Shift Rule and the SWAP test. The latter is based on the simple observation that, for any $\hat{X}$ and $\hat{Y}$, it is $\mathrm{Tr}\big[\hat{X}\hat{Y}\big] = \mathrm{Tr}\big[\hat{S}(\hat{X}\otimes\hat{Y})\big]$, where $\hat{S}$ is the swap operator [22]. Using the SWAP test and Eq. (7) we get

$$F_{p,p'} = \sum_{t,t',\boldsymbol{\nu},\boldsymbol{\nu}'} \mathrm{Tr}\left[\hat{S}\left(\frac{\partial\hat{\rho}}{\partial x_{t,\boldsymbol{\nu}}}\otimes\frac{\partial\hat{\rho}}{\partial x_{t',\boldsymbol{\nu}'}}\right)\right]\frac{\partial x_{t,\boldsymbol{\nu}}}{\partial\theta_p}\frac{\partial x_{t',\boldsymbol{\nu}'}}{\partial\theta'_p}\,. \tag{34}$$

Then, thanks to our analysis from section III, we may write

$$\begin{aligned} F_{(t,\boldsymbol{\nu}),(t',\boldsymbol{\nu}')} &:= \mathrm{Tr}\left[\hat{S}\left(\frac{\partial\hat{\rho}}{\partial x_{t,\boldsymbol{\nu}}}\otimes\frac{\partial\hat{\rho}}{\partial x_{t',\boldsymbol{\nu}'}}\right)\right] = \qquad (35)\\ &= \sum_{\alpha=\pm,\alpha'=\pm}\alpha\alpha'\int_0^1 ds\int_0^1 ds'\,\mathrm{Tr}\big[\hat{S}\,(\hat{\rho}_{t,\boldsymbol{\nu},s,\alpha}\otimes\hat{\rho}_{t',\boldsymbol{\nu}',s',\alpha'})\big]\,, \end{aligned}$$

where $\hat{\rho}_{t,\boldsymbol{\nu},s,\pm}$ is the state in which the gate $\hat{U}_t$ has been substituted by the gate $U_\pm(x_{t,\boldsymbol{\nu}}, s)$ from Eq. (18), or its noisy implementation as in Sec. IV B. Therefore, an estimator of the matrix elements of the Fisher information matrix can be obtained by sampling two real numbers s and s' from the uniform distribution, and then measuring the overlaps of all quantum states $\rho_{t,\boldsymbol{\nu},s,\alpha}$ and $\rho_{t',\boldsymbol{\nu}',s',\alpha'}$ via the swap test. Note that for noiseless gates the overlaps in (35) can be simplified in some cases. For instance, when $t' = t$ all the gates in the product (4) with larger t disappears from the overlap. It was found in [38] that a good approximation to the natural gradient can be obtained by using only the diagonal elements of F. Motivated by this, we study what happens when we fix t and $\boldsymbol{\nu}$ and call $x \equiv x_{t,\boldsymbol{\nu}}$ as in Sec. III. With the notation of Eqs. (9) and (10), using (11) we may write

$$\begin{aligned} F_{(t,\boldsymbol{\nu}),(t,\boldsymbol{\nu})} &= \mathrm{Tr}\left[\frac{\partial\hat{\rho}}{\partial x}\frac{\partial\hat{\rho}}{\partial x}\right] = \mathrm{Tr}\left(\frac{\partial e^{\mathcal{Z}}}{\partial x}[\hat{\rho}_0]\frac{\partial e^{\mathcal{Z}}}{\partial x}[\hat{\rho}_0]\right) = \\ &= \int_0^1 ds\int_0^1 ds'\,\mathrm{Tr}\Big(i[\hat{V}(s),\hat{\rho}_0]i[\hat{V}(s'),\hat{\rho}_0]\Big) = \\ &= 2(F_2 - |F_1|^2)\,, \end{aligned} \tag{36}$$

where $\hat{\rho}_0 = |\phi\rangle\langle\phi|$, $\hat{V}(s) = e^{is(\hat{H}+x\hat{V})}\hat{V}e^{i(1-s)(\hat{H}+x\hat{V})}$ and we have defined

$$F_2 = \int_0^1 ds\int_0^1 ds'\,\langle\phi|\,\hat{V}(s)\hat{V}(s')\,|\phi\rangle\,, \tag{37}$$

$$F_1 = \int_0^1 ds\,\langle\phi|\,\hat{V}(s)\,|\phi\rangle\,. \tag{38}$$

Since $\hat{V}$ is a product of Pauli matrices $\hat{V}(s)$ is a unitary operator, so both F_2 and F_1 can be measured by first sampling s and s' from the uniform distribution, and then measuring the expectation value using the Hadamard test [41].

V. CONCLUSIONS

We have studied the optimization of a *cost function* defined by taking a quantum measurement on a parametric quantum state, obtained by applying on a fixed reference state a controlled evolution with tunable classical parameters. We have found explicit analytical formulae for the derivatives of the cost function with respect to those classical parameters. Our formulae can be applied to any multi-qubit evolution and generalize the so-called parameter shift rule [18, 19] to the general case, without any restriction on the spectrum of the operator, and without the use of ancillary qubits or Hamiltonian simulation techniques [21].

Based on those exact formulae, we have devised both exact and approximate algorithms for estimating the derivatives of the cost via carefully designed quantum circuits. The exact algorithm works when exact applications of the gates are possible, whereas the approximate algorithm is designed to tackle spurious interactions in the system that cannot be completely removed. As such, our algorithm can also be applied, though with some approximations, when the gates implemented by the quantum device are noisy, as it is the case in near-term quantum devices [1].

The main application of our study is to optimize parametric quantum evolution for quantum optimization [3] and machine-learning problems [10].

ACKNOWLEDGMENTS

The authors acknowledge Xanadu Inc. for hosting the QHACK'19 event (`https://qhack.ai/`) where this study was initiated. L.B. acknowledges support by the program "Rita Levi Montalcini" for young researchers. X, formerly known as Google[x], is part of the Alphabet family of companies, which includes Google, Verily, Waymo, and others (`www.x.company`).

Appendix A: Explicit algorithms

In this appendix we discuss more explicitly all the steps to define unbiased estimators of $\partial C/\partial\theta_p$ that can be measured with the Stochastic Parameter Shift Rule. The full version of Algorithms 2 and 3 is the following:

Algorithm 4 Stochastic Parameter Shift Rule, Eq. (22)

Sample s from the uniform distribution in $[0, 1]$;
set $g_p = 0$;
for $t = 1, \dots, T$ **do**
  for all $\boldsymbol{\nu}$ such that $\partial_{\theta_p} x_{t,\boldsymbol{\nu}}(\boldsymbol{\theta}) \neq 0$ **do**
    for $m = \{+, -\}$ **do**
      initialize the computer in the state $|\psi_0\rangle$;
      sequentially apply the gates $\hat{U}_{t'}$ for $t' = 0 \dots, t-1$ to prepare the state $|\phi\rangle$ in Eq. (9);
      apply the gate $\hat{U}_t^{1-s} \equiv e^{i(1-s)\hat{X}_t(\boldsymbol{\theta})}$, by rescaling all parameters;
      if gates $e^{\pm i\frac{\pi}{4}\hat{\sigma}_{t,\boldsymbol{\nu}}}$ are available **then**
        apply the gate $e^{mi\frac{\pi}{4}\hat{\sigma}_{t,\boldsymbol{\nu}}}$;
      else
        apply the best approximation of $e^{mi\frac{\pi}{4}\hat{\sigma}_{t,\boldsymbol{\nu}}}$, for instance using (28);
      end if
      apply the gate $\hat{U}_t^s \equiv e^{is\hat{X}_t(\boldsymbol{\theta})}$;
      sequentially apply the gates $\hat{U}_{t'}$ for $t' = t+1, \dots, T$;
      measure the observable $\hat{C}$ and call the result $r_{m,t,\boldsymbol{\nu}}$;
    end for
    set $g_{t,\boldsymbol{\nu}} = r_{+,t,\boldsymbol{\nu}} - r_{-,t,\boldsymbol{\nu}}$;
    update $g_p \to g_p + g_{t,\boldsymbol{\nu}} \partial_{\theta_p} x_{t,\boldsymbol{\nu}}(\boldsymbol{\theta})$
  end for
end for
the sample g_p is such that $\partial C/\partial\theta_p = \mathbb{E}[g_p]$.

By repeating the analysis of Sec. III A we find that $\partial C/\partial\theta_p = \mathbb{E}[g_p]$, so by repeating Algorithm 4 many times we may estimate the derivative $\partial C/\partial\theta_p$ with the desired precision. A simple counting argument shows that the number of operations to obtain a single outcome is $O(2TN_p)$ where N_p is the number of non-zero $\partial_{\theta_p} x_{t,\boldsymbol{\nu}}(\boldsymbol{\theta})$. Note that Step 1: in Algorithm 4 can be moved to any other point point before Step 8. By linearity, the average is always the same, although each iteration might have a different value of s. We can reduce the number of operations to get a single estimate with the following algorithm:

Algorithm 5 Doubly Stochastic Parameter Shift Rule, Eq. (24)

Sample s from the uniform distribution in $[0, 1]$;
calculate the probability distribution $q_p(t, \boldsymbol{\nu})$ defined in Eq. (23) and set $n_{p,t,\boldsymbol{\nu}}$ as described in Sec. III A;
sample $(t, \boldsymbol{\nu})$ from q_p;
for $m = \{+, -\}$ **do**
  initialize the computer in the state $|\psi_0\rangle$;
  sequentially apply the gates $\hat{U}_{t'}$ for $t' = 0 \dots, t-1$ to prepare the state $|\phi\rangle$ in Eq. (9);
  apply the gate $\hat{U}_t^{1-s} \equiv e^{i(1-s)\hat{X}_t(\boldsymbol{\theta})}$, by rescaling all parameters;
  if gates $e^{\pm i\frac{\pi}{4}\hat{\sigma}_{t,\boldsymbol{\nu}}}$ are available **then**
    apply the gate $e^{mi\frac{\pi}{4}\hat{\sigma}_{t,\boldsymbol{\nu}}}$;
  else
    apply the best approximation of $e^{mi\frac{\pi}{4}\hat{\sigma}_{t,\boldsymbol{\nu}}}$, for instance using (28);
  end if
  apply the gate $\hat{U}_t^s \equiv e^{is\hat{X}_t(\boldsymbol{\theta})}$;
  sequentially apply the gates $\hat{U}_{t'}$ for $t' = t+1, \dots, T$;
  measure the observable $\hat{C}$ and call the result $r_{m,t,\boldsymbol{\nu}}$;
end for
the sample $g_p = (r_{+,t,\boldsymbol{\nu}} - r_{-,t,\boldsymbol{\nu}}) n_{p,t,\boldsymbol{\nu}}$ is such that $\partial C/\partial\theta_p = \mathbb{E}[g_p]$.

In Algorithm (5) the quantum computer is still reset twice to have a single estimate. Below we define an algorithm where the computer is initialized only once

Algorithm 6 Single-measurement sample of $\partial C/\partial\theta_p$

Sample s from the uniform distribution in $[0, 1]$;
calculate the probability distribution $q_p(t, \boldsymbol{\nu})$ defined in Eq. (23) and set $n_{p,t,\boldsymbol{\nu}}$ as described in Sec. III A;
sample $(t, \boldsymbol{\nu})$ from q_p;
initialize the computer in the state $|\psi_0\rangle$;
sequentially apply the gates $\hat{U}_{t'}$ for $t' = 0 \dots, t-1$ to prepare the state $|\phi\rangle$ in Eq. (9);
apply the gate $\hat{U}_t^{1-s} \equiv e^{i(1-s)\hat{X}_t(\boldsymbol{\theta})}$, by rescaling all parameters;
sample $m \in \{+1, -1\}$ by tossing a fair coin;
if gates $e^{\pm i\frac{\pi}{4}\hat{\sigma}_{t,\boldsymbol{\nu}}}$ are available **then**
  apply the gate $e^{mi\frac{\pi}{4}\hat{\sigma}_{t,\boldsymbol{\nu}}}$;
else
  apply the best approximation of $e^{mi\frac{\pi}{4}\hat{\sigma}_{t,\boldsymbol{\nu}}}$, e.g. using (28);
end if
apply the gate $\hat{U}_t^s \equiv e^{is\hat{X}_t(\boldsymbol{\theta})}$;
sequentially apply the gates $\hat{U}_{t'}$ for $t' = t+1, \dots, T$;
measure the observable $\hat{C}$ and call the result r;
the sample $g_p = 2mrn_{p,t,\boldsymbol{\nu}}$ is such that $\partial C/\partial\theta_p = \mathbb{E}[g_p]$.

The above algorithm corresponds to rewriting Eq. (19) as

$$\partial_x C(x) = \sum_{m=\pm} m p_m \int_0^1 2C_m(x, s)\, ds\,. \tag{A1}$$

with probabilities $p_\pm = \frac{1}{2}$. Putting explicitly the dependence on t and $\boldsymbol{\nu}$ we get from (24) and from the notation (8), (9)

$$\frac{\partial C(\boldsymbol{\theta})}{\partial\theta_p} = \sum_{t,\boldsymbol{\nu},m} m n_{p,t,\boldsymbol{\nu}} p_m q_p(t, \boldsymbol{\nu}) \int_0^1 2 \left\langle \psi_{t,\boldsymbol{\nu},s,m} \right| \hat{C} \left| \psi_{t,\boldsymbol{\nu},s,m} \right\rangle ds, \tag{A2}$$

where $m = \pm$, $t = 1, \dots, T$, and

$$\left|\psi_{t,\boldsymbol{\nu},s,m}\right\rangle := \prod_{t'=t+1}^{T} \hat{U}_{t'}\, \hat{U}_t^{1-s} e^{mi\pi/4\hat{\sigma}_{\boldsymbol{\nu}}} \hat{U}_t^s \prod_{t'=1}^{t-1} \hat{U}_{t'}\, |\psi_0\rangle\,. \tag{A3}$$

Appendix B: Variance of gradient estimators

By comparing the standard Parameter Shift Rule (Fig. 1) and Stochastic Parameter Shift Rule (Fig. 2) we see that the latter has an extra source of randomness due to the sampling over the classical parameter s. The stochastic outcomes of these two algorithms have the same mean, namely the gradient of the cost function, so in the limit of infinitely-many repetitions of the experiment these algorithms provide the same result. However, the variance of the estimators obtained with the two algorithms might be different. Assuming independent identically distributed samples, the variance quantifies the expected error when a *finite* number of measurements is performed, so it is important to study whether the extra stochasticity of the Stochastic Parameter Shift Rule increases the variance of the gradient estimators.

We first study the gradient of $C(t, b)$ obtained with $\hat{C} = \hat{\sigma}_y \otimes \hat{\sigma}_y$ and the cross-resonance gate (29), as in Fig. 4, but

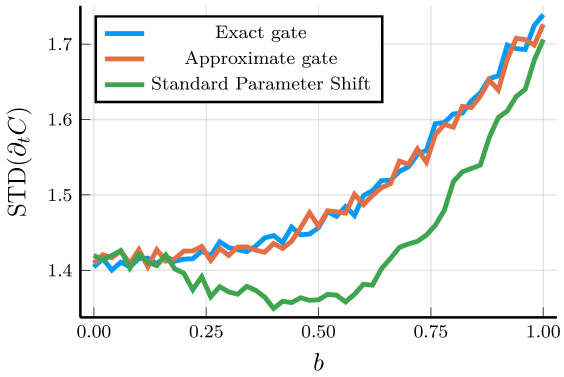

FIG. 6. Empirical standard deviation of the gradient estimator of $\partial_t C(t, b)$, with the same notation of Fig. 4(b), for $c = 0$ and different values of b. The Standard Parameter Shift Rule corresponds to Algorithm 1. For each point, the STD is estimated using 10^4 samples.

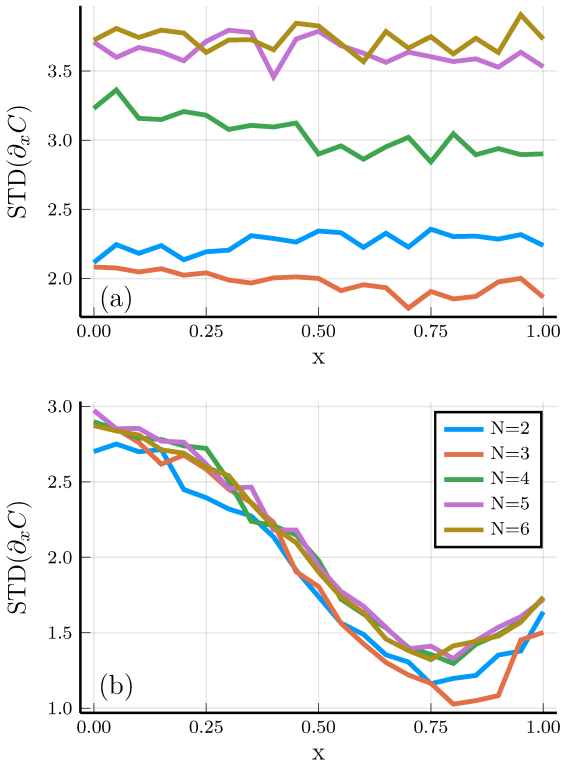

FIG. 7. (a) Empirical standard deviation (STD) of the estimator from Algorithm 2, using $\hat{H} = \hat{H}_a$ and $\hat{V} = \hat{V}_a$ from Eqs. (B1), for different numbers of qubits N. (b) STD of the gradient estimator obtained via the (standard) Parameter Shift Rule, for a related problem with $\hat{H} = 0$, shown in Eq. (B2). In both (a) and (b) the plots are shown for different values of the parameter x, as in Eq. (10), while the STD is estimated via 1000 samples.

with $c = 0$. When $c = 0$ the operator in the exponential has two possible eigenvalues $u = \pm\sqrt{1 + b^2}$, so for computing the derivative $\partial_t C$ we can also apply the standard parameter shift rule, and compare the variance of the resulting estimator with

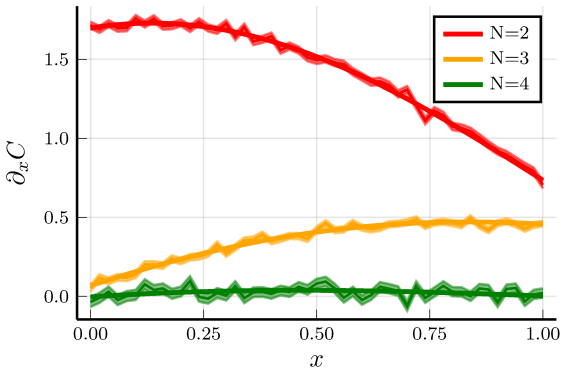

FIG. 8. Finite difference approximation (solid lines), versus estimated gradient via Algorithm 2, with error bars as in Fig. 4. We focus on $C(x)$ with the definitions (10) and model (B1), for different numbers of qubits N.

that obtained from the Stochastic Parameter Shift Rule. Note that, unlike our Algorithm 2, the simpler parameter shift rule cannot be applied to estimate $\partial_b C$.

In Fig. 6 we compare the standard deviation of Algorithms 1, 2, 3. We note that, although Algorithms 2, 3 have extra sampling steps, the resulting variance is comparable with that of Algorithm 1.

We now study how the standard deviation might scale as a function of the number of qubits. In Fig. 7(a) we focus on the Stochastic Parameter Shift Rule, with the following choice of states and operators in (10)

$$\hat{H}_a = \sum_{j=1}^{N} \left[\hat{\sigma}_x^{(j)} \hat{\sigma}_x^{(j+1)} + \frac{\hat{\sigma}_x^{(j)}}{3} + \frac{\hat{\sigma}_z^{(j)}}{2} \right], \qquad \hat{V}_a = \sigma_z^{(1)}, \quad \text{(B1a)}$$

$$\hat{A}_a = \sum_{j=1}^{N} \hat{\sigma}_z^{(j)}, \qquad |\phi_a\rangle = |0\rangle^{\otimes N}, \quad \text{(B1b)}$$

where $\hat{\sigma}_x^{(j)}$ means that the operator $\hat{\sigma}_x$ is applied to the $j$the qubit and $\hat{\sigma}_x^{(N+1)} \equiv \hat{\sigma}_x^{(1)}$. In Eqs. (B1) we have chosen for $\hat{H}_a$ a many-body Hamiltonian with complex entangling dynamics [42]. The empirical mean is shown in Fig. 8 for $N = 2, 3, 4$. Larger values of N are not shown, as they are similar to the case $N = 4$. Since $\hat{H}_a \neq 0$ we cannot apply the standard parameter shift rule of Fig. 1. Therefore, to compare the algorithms 1 and 2 we need to introduce another model with $\hat{H} = 0$, namely where all gates depend on the parameters as in Eq. 6. We build such a model using the same operators introduced in Eqs. (B1) and define

$$\hat{H}_b = 0, \qquad \hat{V}_b = \sigma_z^{(1)}, \quad \text{(B2a)}$$

$$\hat{A}_a = \sum_{j=1}^{N} e^{i\hat{H}_a/2} \hat{\sigma}_z^{(j)} e^{i\hat{H}_a/2}, \qquad |\phi_b\rangle = e^{i\hat{H}_a/2} |0\rangle^{\otimes N}. \quad \text{(B2b)}$$

By comparing Fig. 7(a) with Fig. 7(b), we note that the standard deviations of both estimators have the same order of magnitude, that does not seem to increase too much with the number of qubits N, at least for our choice of Hamiltonians. In

Fig. 7(a) we observe a slight non-monotonic increase, while in Fig. 7(b) the results are basically independent on N. We believe that this difference is mostly due to the particular choice of the models, Eqs. (B1) and (B2), that although related are not identical. Therefore, we may conclude that the stochastic parameter shift rule is basically as efficient as the standard parameter shift rule, but it is more general.

[1] J. Preskill, Quantum **2**, 79 (2018).
[2] A. Montanaro, npj Quantum Inf. **2**, 1 (2016).
[3] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. OBrien, Nat. Commun. **5**, 4213 (2014).
[4] T. Jones, S. Endo, S. McArdle, X. Yuan, and S. C. Benjamin, Phys. Rev. A **99**, 062304 (2019).
[5] E. Farhi, J. Goldstone, and S. Gutmann, arXiv:1411.4028 (2014).
[6] J. Romero, J. P. Olson, and A. Aspuru-Guzik, Quant. Sci. Tech. **2**, 045001 (2017).
[7] Y. Li and S. C. Benjamin, Phys. Rev. X **7**, 021050 (2017).
[8] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, Phys. Rev. A **101**, 032308 (2020), arXiv:1804.00633.
[9] M. Benedetti, E. Lloyd, S. Sack, and M. Fiorentini, Quantum Sci. Technol. (2019).
[10] M. Schuld and F. Petruccione, *Supervised learning with quantum computers* (Springer, 2018).
[11] H. A. et al., Qiskit: An open-source framework for quantum computing (2019).
[12] Cirq: A Python framework for creating, editing, and invoking Noisy Intermediate Scale Quantum (NISQ) circuits (2019), `https://github.com/quantumlib/Cirq`.
[13] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, C. Blank, K. McKiernan, and N. Killoran, arXiv:1811.04968 (2018).
[14] R. S. Smith, M. J. Curtis, and W. J. Zeng, arXiv:1608.03355 (2016).
[15] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, M. Y. Niu, R. Halavati, E. Peters, M. Leib, A. Skolik, M. Streif, D. Von Dollen, J. R. McClean, S. Boixo, D. Bacon, A. K. Ho, H. Neven, and M. Mohseni, arXiv:2003.02989 (2020).
[16] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Nat. Commun. **9**, 4812 (2018).
[17] A. Harrow and J. Napp, arXiv:1901.05374 (2019).
[18] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Phys. Rev. A **98**, 032309 (2018).
[19] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Phys. Rev. A **99**, 032331 (2019).
[20] G. E. Crooks, arXiv:1905.13311 (2019).
[21] A. M. Childs and N. Wiebe, arXiv:1202.5822 (2012).
[22] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, Quantum **3**, 191 (2019).
[23] N. Khaneja, T. Reiss, C. Kehlet, T. Schulte-Herbrüggen, and S. J. Glaser, J. Magn. Reson **172**, 296 (2005).
[24] L. Banchi, N. Pancotti, and S. Bose, npj Quantum Inf. **2**, 16019 (2016).
[25] L. Innocenti, L. Banchi, A. Ferraro, S. Bose, and M. Paternostro, New J. Phys. **(in press)** (2020).
[26] M. A. Nielsen, M. R. Dowling, M. Gu, and A. C. Doherty, Science **311**, 1133 (2006).
[27] R. M. Wilcox, J. Math. Phys. **8**, 962 (1967).
[28] W. Miller, *Symmetry groups and their applications* (Academic Press, 1973).
[29] R. Sweke, F. Wilde, J. Meyer, M. Schuld, P. K. Fährmann, B. Meynard-Piganeau, and J. Eisert, arXiv:1910.01155 (2019).
[30] S. Bubeck, Found. Trends Mach. Learn. **8**, 231 (2015).
[31] D. P. Kingma and J. Ba, arXiv:1412.6980 (2014).
[32] S. Lloyd, Science **273**, 1073 (1996).
[33] J. C. Spall, IEEE Trans. Automat. Contr. **37**, 332 (1992).
[34] J. M. Chow, A. Córcoles, J. M. Gambetta, C. Rigetti, B. Johnson, J. A. Smolin, J. Rozen, G. A. Keefe, M. B. Rothwell, M. B. Ketchen, *et al.*, Phys. Rev. Lett. **107**, 080502 (2011).
[35] T. Caneva, T. Calarco, and S. Montangero, Phys. Rev. A **84**, 022326 (2011).
[36] A. Rivas and S. F. Huelga, *Open quantum systems* (Springer, 2012).
[37] H.-P. Breuer and F. Petruccione, *The theory of open quantum systems* (Oxford University Press on Demand, 2002).
[38] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, arXiv:1909.02108 (2019).
[39] B. Koczor and S. C. Benjamin, arXiv:1912.08660 (2019).
[40] L. Gentini, A. Cuccoli, S. Pirandola, P. Verrucchi, and L. Banchi, arXiv:1912.06744 (2019).
[41] K. Mitarai and K. Fujii, Phys. Rev. Res. **1**, 013006 (2019).
[42] H. Kim and D. A. Huse, Phys. Rev. Lett. **111**, 127205 (2013).