Evaluating analytic gradients on quantum hardware

Maria Schuld,* Ville Bergholm, Christian Gogolin, Josh Izaac, and Nathan Killoran
Xanadu Inc., 372 Richmond St W, Toronto, Canada M5V 1X6

(Received 9 January 2019; published 21 March 2019)

An important application for near-term quantum computing lies in optimization tasks, with applications ranging from quantum chemistry and drug discovery to machine learning. In many settings, most prominently in so-called parametrized or variational algorithms, the objective function is a result of hybrid quantum-classical processing. To optimize the objective, it is useful to have access to exact gradients of quantum circuits with respect to gate parameters. This paper shows how gradients of expectation values of quantum measurements can be estimated using the same, or almost the same, architecture that executes the original circuit. It generalizes previous results for qubit-based platforms, and proposes recipes for the computation of gradients of continuous-variable circuits. Interestingly, in many important instances it is sufficient to run the original quantum circuit twice while shifting a single gate parameter to obtain the corresponding component of the gradient. More general cases can be solved by conditioning a single gate on an ancilla.

DOI: 10.1103/PhysRevA.99.032331

I. INTRODUCTION

Hybrid optimization algorithms have become a central quantum software design paradigm for current-day quantum technologies, since they outsource parts of the computation to classical computers. Examples of such algorithms are variational quantum eigensolvers [1], quantum approximate optimization [2], variational autoencoders [3], quantum feature embeddings [4,5], variational classifiers [6,7], and quantum compilers [8], but also more general hybrid optimization frameworks [9]. In such applications, the objective or cost function is a combination of both classical and quantum information processing modules, or nodes (see Fig. 1). The quantum nodes execute parametrized quantum circuits, also called *variational circuits*, in which gates have adjustable continuous parameters such as rotations by an angle.

To unlock the potential of gradient-descent-based optimization strategies it is essential to have access to the gradients of quantum computations. While individual quantum measurements produce probabilistic results, the expectation value of a quantum observable—which can be estimated by taking the average over measurement results—is a deterministic quantity that varies smoothly with the gate parameters. It is therefore possible to formally define the gradient of a quantum computation via derivatives of expectations.

The challenge however is to compute such gradients on quantum hardware. As we will lay out below, the derivative of a quantum expectation with respect to a parameter μ used in gate $\mathcal{G}$ involves the "derivative of the gate" $\partial_\mu \mathcal{G}$, which is not necessarily a quantum gate itself. Hence the derivative of an expectation is not a valid quantum expectation. Since in interesting cases the gradient, just as the objective function itself, tends to be classically intractable, we need to express such derivatives as a combination of quantum operations that can be implemented in hardware. Even more, in the case of special-purpose quantum hardware it is desirable that gradients can be evaluated by the same device that is used for the original computation.

This paper derives rules to compute the partial derivatives of quantum expectation values with respect to gate parameters on quantum hardware. A number of results in this direction have been recently proposed in the quantum computing and quantum machine learning literature [6,7,10–12]. References [6,7,10] note that if the derivative $\partial_\mu \mathcal{G}$ as well as the observable whose expectation we are interested in can be decomposed into a sum of unitaries, we can evaluate the derivative of an expectation by measuring an overlap of two quantum states. Mitarai *et al.* [11], leveraging a technique from quantum control, propose an elegant method for gates of the form $\mathcal{G} = e^{-i\mu\sigma}$, where σ is a tensor product of the Pauli operators $\{\sigma_x, \sigma_y, \sigma_z\}$. In this case, the derivative can be computed by what we will call the "parameter shift rule," which requires us to evaluate the original expectation twice, but with one circuit parameter shifted by a fixed value.

In this work, we make several contributions to the literature on quantum gradients. First, we expand the parameter shift rule by noting that it holds for any gate of the form $\mathcal{G} = e^{-i\mu G}$, where the Hermitian generator G has at most two distinct eigenvalues. We mention important examples of this class. Secondly, we show that any other gate can be handled by a method that involves a coherent *linear combination of unitaries* routine [13]. This requires adding a single ancilla qubit and conditioning the gate and its "derivative" on the ancilla while running the circuit. Thirdly, we derive parameter shift rules for Gaussian gates in continuous-variable quantum computing. These rules can be efficiently implemented if all gates following the differentiated gate are Gaussian and the final observable is a low-degree polynomial of the creation and annihilation operators. In fact, the method still works efficiently for some non-Gaussian gates, such as the cubic phase gate, as long as there is at most a logarithmically

*maria@xanadu.ai

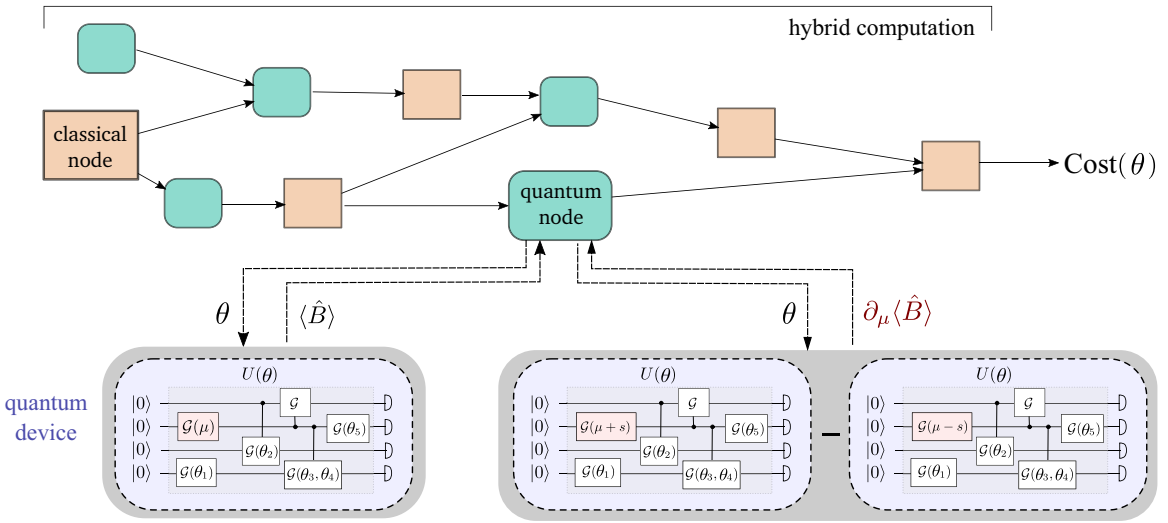

FIG. 1. "Parameter shift rule" in the larger context of hybrid optimization. A quantum node, in which a variational quantum algorithm is executed, can compute derivatives of its outputs with respect to gate parameters by running the original circuit twice, but with a shift in the parameter in question.

large number of these non-Gaussian gates. The results of this paper are implemented in the software framework *PennyLane* [9], which facilitates hybrid quantum-classical optimization across various quantum hardwares and simulator platforms [9].

II. COMPUTING QUANTUM GRADIENTS

Consider a quantum algorithm that is possibly part of a larger hybrid computation, as shown in Fig. 1. The quantum algorithm or circuit consists of a gate sequence $U(\theta)$ that depends on a set θ of m real gate parameters, followed by the measurement of an observable $\hat{B}$ [14]. An example is the Pauli-Z observable $\hat{B} = \sigma_z$, and the result of this single measurement is ± 1 for a qubit found in the state $|0\rangle$ or $|1\rangle$, respectively. The gate sequence $U(\theta)$ usually consists of an ansatz or architecture that is repeated K times, where K is a hyperparameter of the computation.

We refer to the combined procedure of applying the gate sequence $U(\theta)$ and finding the expectation value of the measurement $\hat{B}$ as a *variational circuit*. In the overall hybrid computation one can therefore understand a variational circuit as a function $f : \mathbb{R}^m \to \mathbb{R}^n$, mapping the gate parameters to an expectation,

$$f(\theta) := \langle \hat{B} \rangle = \langle 0|U^\dagger(\theta)\hat{B}U(\theta)|0\rangle. \tag{1}$$

While this abstract definition of a variational circuit is exact, its physical implementation on a quantum device runs the quantum algorithm several times and averages measurement samples to get an *estimate* of $f(\theta)$. If the circuit is executed on a classical simulator, $f(\theta)$ can be computed exactly up to numerical precision.

In the following, we are concerned with the partial derivative $\partial_\mu f(\theta)$ where $\mu \in \theta$ is one of the gate parameters. The partial derivatives with respect to all gate parameters form the gradient ∇f. The differentiation rules we derive consider the expectation value in Eq. (1) and are therefore exact. Just like the variational circuit itself has an "analytic" definition and a "stochastic" implementation, the *evaluation* of these rules with finite runs on noisy hardware return estimates of the gradient [15].

There are three main approaches to evaluate the gradients of a numerical computation, i.e., a computer program that executes a mathematical function $g(x)$, as follows.

(1) *Numerical differentiation*. The gradient is approximated by blackbox evaluations of g, e.g.,

$$\nabla g(x) \approx [g(x + \Delta x/2) - g(x - \Delta x/2)]/\Delta x, \tag{2}$$

where Δx is a small shift.

(2) *Automatic differentiation*. The gradient is efficiently computed through the accumulation of intermediate derivatives corresponding to different subfunctions used to build g, following the chain rule [16].

(3) *Symbolic differentiation*. Using manual calculations or a symbolic computer algebra package, the function ∇g is constructed and evaluated.

Until recently, numerical differentiation (or altogether gradient-free methods) have been the method of choice in the quantum variational circuits literature. However, the high errors of near-term quantum devices can make it unfeasible to use finite difference formulas to approximate the gradient of a circuit. Furthermore, there is a first theoretical study that derives worst-case bounds for the number of times a quantum device has to be queried to converge to a minimum [17], which shows that, under some conditions [18], analytic gradient strategies can take significantly fewer queries than numeric ones.

Several modern numerical programming frameworks, especially in machine learning, successfully employ automatic differentiation [19] instead, a famous example being the ubiquitous backpropagation algorithm for the training of neural networks. Unfortunately, it is not clear how intermediate derivatives could be stored and reused *inside of a quantum computation*, since the intermediate quantum states cannot be measured without impacting the overall computation.

TABLE I. Summary of results. $\mathcal{G}$ refers to the gate with parameter μ for which we compute the partial derivative. $\partial_\mu \mathcal{G}$ refers to the partial derivative of the operator $\mathcal{G}$.

Architecture	Condition	Technique
Qubit	$\mathcal{G}$ generated by a Hermitian operator with two unique eigenvalues	Parameter shift rule
Qubit	No special condition	Derivative gate decomposition + linear combination of unitaries
Continuous-variable	$\mathcal{G}$ Gaussian, followed by at most logarithmically many non-Gaussian operations	Continuous-variable parameter shift rules
Continuous-variable	No special condition	Unknown

To compute gradients of quantum expectation values, we therefore use the following strategy: derive an equation for $\partial_\mu f(\theta),\ \mu \in \theta$, whose constituent parts can be evaluated on a quantum computer and subsequently combined on a classical coprocessor. It turns out that this strategy has a number of favorable properties: it follows similar rules for a range of different circuits, evaluating $\partial_\mu f(\theta)$ can often be done on a circuit architecture that is very similar or even identical to that for evaluating $f(\theta)$, and evaluating $\partial_\mu f(\theta)$ requires the evaluation of only two expectation values.

We emphasize that automatic differentiation techniques such as backpropagation can still be used within a larger overall hybrid computation, but we will not get any efficiency gains for this technique on the intermediate steps of the quantum circuit.

The remainder of the paper will present the recipes for how to evaluate the derivatives of expectation values, first for qubit-based, and then for continuous-variable quantum computing. The results are summarized in Table I.

III. GRADIENTS OF DISCRETE-VARIABLE CIRCUITS

As a first step, the overall unitary $U(\theta)$ of the variational circuit can be decomposed into a sequence of single-parameter gates, which can be differentiated using the product rule. For simplicity, let us assume that the parameter $\mu \in \theta$ only affects a single gate $\mathcal{G}(\mu)$ in the sequence, $U(\theta) = V\mathcal{G}(\mu)W$. The partial derivative $\partial_\mu f$ then looks like

$$\partial_\mu f = \partial_\mu \langle\psi|\mathcal{G}^\dagger \hat{Q} \mathcal{G}|\psi\rangle = \langle\psi|\mathcal{G}^\dagger \hat{Q}(\partial_\mu \mathcal{G})|\psi\rangle + \text{H.c.}, \quad (3)$$

where we have absorbed V into the Hermitian observable $\hat{Q} = V^\dagger \hat{B} V$ and W into the state $|\psi\rangle = W|0\rangle$.

For any two operators B, C we have

$$\langle\psi|B^\dagger \hat{Q} C|\psi\rangle + \text{H.c.} = \tfrac{1}{2}[\langle\psi|(B+C)^\dagger \hat{Q}(B+C)|\psi\rangle - \langle\psi|(B-C)^\dagger \hat{Q}(B-C)|\psi\rangle]. \quad (4)$$

Hence, whenever we can implement $\mathcal{G} \pm \partial_\mu \mathcal{G}$ as part of an overall unitary evolution, we can evaluate Eq. (3) directly. Section III A identifies a class of gates for which $\mathcal{G} \pm \partial_\mu \mathcal{G}$ is already unitary, while Sec. III B shows that an ancilla can help to evaluate the terms in Eq. (3) with minimal overhead and guaranteed success.

A. Parameter-shift rule for gates with generators with two distinct eigenvalues

Consider a gate $\mathcal{G}(\mu) = e^{-i\mu G}$ generated by a Hermitian operator G. Its derivative is given by

$$\partial_\mu \mathcal{G} = -iG\, e^{-i\mu G}. \quad (5)$$

Substituting into Eq. (3), we get

$$\partial_\mu f = \langle\psi'|\hat{Q}\,(-iG)|\psi'\rangle + \text{H.c.}, \quad (6)$$

where $|\psi'\rangle = \mathcal{G}|\psi\rangle$. If G has just two distinct eigenvalues (which can be repeated) [20] we can, without loss of generality, shift the eigenvalues to $\pm r$, as the global phase is unobservable. Note that any single qubit gate is of this form. Using Eq. (4) for $B = \mathbb{1}$ and $C = -ir^{-1}G$ we can write

$$\partial_\mu f = \frac{r}{2}[\langle\psi'|(\mathbb{1} - ir^{-1}G)^\dagger \hat{Q}(\mathbb{1} - ir^{-1}G)|\psi'\rangle - \langle\psi'|(\mathbb{1} + ir^{-1}G)^\dagger \hat{Q}(\mathbb{1} + ir^{-1}G)|\psi'\rangle]. \quad (7)$$

We now show that for gates with eigenvalues $\pm r$ there exist values for μ for which $\mathcal{G}(\mu)$ becomes equal to $\frac{1}{\sqrt{2}}(\mathbb{1} \pm ir^{-1}G)$.

Theorem 1. If the Hermitian generator G of the unitary operator $\mathcal{G}(\mu) = \mathrm{e}^{-i\mu G}$ has at most two unique eigenvalues $\pm r$, the following identity holds:

$$\mathcal{G}\left(\frac{\pi}{4r}\right) = \frac{1}{\sqrt{2}}(\mathbb{1} - ir^{-1}G). \quad (8)$$

Proof. The fact that G has the spectrum $\{\pm r\}$ implies $G^2 = r^2\mathbb{1}$. Therefore, the sine and cosine parts of the Taylor series of $\mathcal{G}(\mu)$ take the following simple form:

$$\mathcal{G}(\mu) = \exp(-i\mu G) = \sum_{k=0}^{\infty} \frac{(-i\mu)^k G^k}{k!} \quad (9)$$

$$= \sum_{k=0}^{\infty} \frac{(-i\mu)^{2k} G^{2k}}{(2k)!} + \sum_{k=0}^{\infty} \frac{(-i\mu)^{2k+1} G^{2k+1}}{(2k+1)!} \quad (10)$$

$$= \mathbb{1} \sum_{k=0}^{\infty} \frac{(-1)^k (r\mu)^{2k}}{(2k)!} - ir^{-1}G \sum_{k=0}^{\infty} \frac{(-1)^k (r\mu)^{2k+1}}{(2k+1)!} \quad (11)$$

$$= \mathbb{1}\cos(r\mu) - ir^{-1}G \sin(r\mu). \quad (12)$$

Hence we get $\mathcal{G}(\frac{\pi}{4r}) = \frac{1}{\sqrt{2}}(\mathbb{1} - ir^{-1}G)$. ■

We conclude that in this case $\partial_\mu f$ can be estimated using two additional evaluations of the quantum device; for these evaluations, we place either the gate $\mathcal{G}(\frac{\pi}{4r})$ or the gate $\mathcal{G}(-\frac{\pi}{4r})$ in the original circuit next to the gate we are differentiating.

Since for unitarily generated one-parameter gates $\mathcal{G}(a)\mathcal{G}(b) = \mathcal{G}(a+b)$, this is equivalent to shifting the gate parameter, and we get the "parameter shift rule" with the shift $s = \frac{\pi}{4r}$:

$$\partial_\mu f = r[\langle\psi|\mathcal{G}^\dagger(\mu+s)\hat{Q}\mathcal{G}(\mu+s)|\psi\rangle - \langle\psi|\mathcal{G}^\dagger(\mu-s) \times \hat{Q}\mathcal{G}(\mu-s)|\psi\rangle] \quad (13)$$

$$= r[f(\mu+s) - f(\mu-s)]. \quad (14)$$

If the parameter μ appears in more than a single gate in the circuit, the derivative is obtained using the product rule by shifting the parameter in each gate separately and summing the results. It is interesting to note that Eq. (14) looks similar to the finite difference rule in Eq. (2), but uses a macroscopic shift and is in fact exact.

The parameter shift rule applies to a number of special cases. As remarked in Mitarai *et al.* [11], if G is a one-qubit rotation generator in $\frac{1}{2}\{\sigma_x, \sigma_y, \sigma_z\}$ then $r = 1/2$ and $s = \frac{\pi}{2}$. If $G = r\vec{n}\cdot\boldsymbol{\sigma}$ is a linear combination of Pauli operators with the three-dimensional normal vector $\vec{n}$, it still has two unique eigenvalues and Eq. (8) can also be derived from what is known as the generalized Euler rule.

Also gates from a "hardware-efficient" variational circuit ansatz may fall within the scope of the parameter shift rule. For example, according to the documentation of Google's *Cirq* programming language [21], their Xmon qubits naturally implement the three gates

$$\mathrm{ExpW}(\mu,\delta) = \exp\{-i\mu[\cos(\delta)\sigma_x + \sin(\delta)\sigma_y]\},$$
$$\mathrm{ExpZ}(\mu) = \exp(-i\mu\sigma_z),$$
$$\mathrm{Exp11}(\mu) = \exp(-i\mu|11\rangle\langle 11|),$$

which all have generators with at most two eigenvalues.

Pauli-based multiqubit gates however in general do not fall in this category. A hardware-efficient example here is the microwave-controlled transmon gate for superconducting architectures [22,23],

$$\mathcal{G}(\mu) = \exp\{\mu[\sigma_x\otimes\mathbb{1} - b(\sigma_z\otimes\sigma_x) + c(\mathbb{1}\otimes\sigma_x)]\},$$

which has four eigenvalues. In these cases, other strategies have to be found to compute exact gradients of variational circuits.

B. Differentiation of general gates via linear combination of unitaries

In case the parameter-shift differentiation strategy does not apply, we may always evaluate Eq. (3) by introducing an ancilla qubit. Since for finite-dimensional systems $\partial_\mu\mathcal{G}$ can be expressed as a complex square matrix, we can always decompose it into a linear combination of unitary matrices A_1 and A_2,

$$\partial_\mu\mathcal{G} = \frac{\alpha}{2}[(A_1 + A_1^\dagger) + i(A_2 + A_2^\dagger)] \quad (15)$$

with real α [24]. A_1 and A_2 in turn can be implemented as quantum circuits. To be more general, for example, when another decomposition suits the hardware better, we can write

$$\partial_\mu\mathcal{G} = \sum_{k=1}^{K}\alpha_k A_k, \quad (16)$$

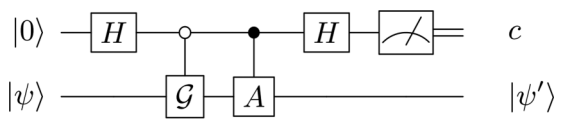

FIG. 2. Quantum circuit illustrating the "linear combination of unitaries" technique [13]. Between interfering Hadamards, two unitary circuits or gates A and $\mathcal{G}$ are applied conditioned on an ancilla. Depending on the state of the ancilla qubit, the effect is equivalent to applying a sum or difference of A and $\mathcal{G}$.

for real α_k and unitary A_k. The derivative becomes

$$\partial_\mu f = \sum_{k=1}^{K}\alpha_k(\langle\psi|\mathcal{G}^\dagger\hat{Q}A_k|\psi\rangle + \mathrm{H.c.}). \quad (17)$$

With Eq. (4) we may compute the value of each term in the sum using a coherent linear combination of the unitaries $\mathcal{G}$ and $A_k = A$, implemented by the quantum circuit in Fig. 2 (here and in the following we drop the subscript k for readability).

First, we append an ancilla in state $|0\rangle$ and apply a Hadamard gate to it to obtain the bipartite state

$$\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)\otimes|\psi\rangle. \quad (18)$$

Next, we apply $\mathcal{G}$ conditioned on the ancilla being in state zero, and A conditioned on the ancilla being in state 1 (remember that both $\mathcal{G}$ and A are unitary). This results in the state

$$\frac{1}{\sqrt{2}}(|0\rangle\mathcal{G}|\psi\rangle + |1\rangle A|\psi\rangle). \quad (19)$$

Applying a second Hadamard on the ancilla we can prepare the final state

$$\tfrac{1}{2}[|0\rangle(\mathcal{G}+A)|\psi\rangle + |1\rangle(\mathcal{G}-A)|\psi\rangle]. \quad (20)$$

A measurement of the ancilla selects one of the two branches and results in either the state $|\psi_0'\rangle = \frac{1}{2\sqrt{p_0}}(\mathcal{G}+A)|\psi\rangle$ with probability

$$p_0 = \tfrac{1}{4}\langle\psi|(\mathcal{G}+A)^\dagger(\mathcal{G}+A)|\psi\rangle, \quad (21)$$

or the state $|\psi_1'\rangle = \frac{1}{2\sqrt{p_1}}(\mathcal{G}-A)|\psi\rangle$ with probability

$$p_1 = \tfrac{1}{4}\langle\psi|(\mathcal{G}-A)^\dagger(\mathcal{G}-A)|\psi\rangle. \quad (22)$$

We then measure the observable $\hat{Q}$ for the final state $|\psi_i'\rangle$, $i = 0, 1$. Repeating this process several times allows us to estimate p_0, p_1 and the expected values of $\hat{Q}$ conditioned on the value of the ancilla,

$$\tilde{E}_0 = \langle\psi_0'|\hat{Q}|\psi_0'\rangle = \frac{1}{4p_0}\langle\psi|(\mathcal{G}+A)^\dagger\hat{Q}(\mathcal{G}+A)|\psi\rangle \quad (23)$$

and

$$\tilde{E}_1 = \langle\psi_1'|\hat{Q}|\psi_1'\rangle = \frac{1}{4p_1}\langle\psi|(\mathcal{G}-A)^\dagger\hat{Q}(\mathcal{G}-A)|\psi\rangle. \quad (24)$$

Comparing with Eq. (4), we find that we can compute the desired left-hand side and thus the individual terms in Eq. (17) from these quantities, since

$$\langle\psi|\mathcal{G}^\dagger\hat{Q}A|\psi\rangle + \mathrm{H.c.} = 2(p_0\tilde{E}_0 - p_1\tilde{E}_1). \quad (25)$$

Note that the measurement on the ancilla is not a typical conditional measurement with limited success probability: either result contributes to the final estimate.

Overall, this approach requires that we can apply the gate $\mathcal{G}$, as well the unitaries A_k from the derivative decomposition in Eq. (16), controlled by an ancilla. Altogether, we need to estimate $2K$ expectation values and $2K$ probabilities, and with Eq. (15) K can always be chosen as 2. The decomposition of $\partial_\mu \mathcal{G}$ into a linear combination of unitaries A_k needs to be found, but this is easy for few qubit gates and has to be done only once.

Note that the idea of decomposing gates into "*classical* linear combinations of unitaries" has been brought forward in Ref. [6], where $\hat{Q}$ had the special form of a σ_z observable, which allowed the authors to evaluate expectations via overlaps of quantum states. Here we added the well-known strategy of *coherent* linear combinations of unitaries [13] to generalize the idea to any observable.

IV. GRADIENTS OF CONTINUOUS-VARIABLE CIRCUITS

We now turn to continuous-variable (CV) quantum computing architectures. Continuous-variable systems [25] differ from discrete systems in that the generators of the gates typically have infinitely many unique eigenvalues, or even a continuum of them. Despite this, we can still find a version of the parameter-shift differentiation recipe which works for Gaussian gates in CV variational circuits if the gate is only followed by Gaussian operations, and if the observable is a low-degree polynomial in the quadratures. The derivation is based on the fact that in this case the effect of a Gaussian gate, albeit commonly represented by an infinite-dimensional matrix in the Schrödinger picture, can be captured by a finite-dimensional matrix in the Heisenberg picture.

As in Sec. III, the task is to compute $\partial_\mu f$. In the Heisenberg picture, instead of evolving the state forward in time with the gates in the circuit, the final observable is evolved "backwards" in time with the adjoint gates. We consider observables $\hat{B}$ that are polynomials of the quadrature operators $\hat{x}_i$, $\hat{p}_i$ (such as $\hat{x}_1\hat{p}_1\hat{x}_2$ or $\hat{x}_1^4\hat{p}_2^3 + 2\hat{x}_1$). By linearity, it is sufficient to understand differentiation of the individual monomials.

For an n-mode system, we introduce the infinite-dimensional vector of quadrature monomials,

$$\hat{C} := (\mathbb{1}, \hat{x}_1, \hat{p}_1, \hat{x}_2, \hat{p}_2, \ldots, \hat{x}_n, \hat{p}_n, \hat{x}_1^2, \hat{x}_1\hat{p}_1, \ldots), \tag{26}$$

sorted by their degree, in terms of which we will expand the observables.

A. CV gates in the Heisenberg picture

Let us consider the Heisenberg-picture action $\mathcal{G}^\dagger \hat{C}_j \mathcal{G}$ of a gate $\mathcal{G}$ on a monomial $\hat{C}_j \in \hat{C}$. This conjugation acts as a linear transformation $\Omega^{\mathcal{G}}$ on $\hat{C}$, i.e.,

$$\Omega^{\mathcal{G}}[\hat{C}_j] := \mathcal{G}^\dagger \hat{C}_j \mathcal{G} = \sum_i M_{ij}^{\mathcal{G}} \hat{C}_i, \tag{27}$$

where $M_{ij}^{\mathcal{G}} = M_{ij}^{\mathcal{G}}(\mu)$ are the elements of a real matrix $M^{\mathcal{G}}$ that depends on the gate parameter. Subsequent conjugations correspond to multiplying the matrices together:

$$\Omega^U[\Omega^V[\hat{C}_k]] = \Omega^U[V^\dagger \hat{C}_k V] = \sum_{ij} M_{ij}^U M_{jk}^V \hat{C}_i. \tag{28}$$

Suppose now that the gate $\mathcal{G}$ is Gaussian. Conjugation by a Gaussian gate does not increase the degree of a polynomial. This means that $\mathcal{G}$ will map the subspace of the zeroth- and first-degree monomials spanned by $\hat{D} := (\mathbb{1}, \hat{x}_1, \hat{p}_1, \hat{x}_2, \hat{p}_2, \ldots, \hat{x}_n, \hat{p}_n)$ into itself,

$$\Omega^{\mathcal{G}}[\hat{D}_j] = \sum_{i=0}^{2n} M_{ij}^{\mathcal{G}} \hat{D}_i. \tag{29}$$

For observables that are higher-degree polynomials of the quadratures, we can use the fact that $\Omega^{\mathcal{G}}$ is a unitary conjugation and that the higher-degree monomials can be expressed as products of the lower-degree ones in $\hat{D}$:

$$\Omega^{\mathcal{G}}[\hat{D}_i\hat{D}_j] = \mathcal{G}^\dagger \hat{D}_i \hat{D}_j \mathcal{G}, \tag{30}$$

$$= \mathcal{G}^\dagger \hat{D}_i \mathcal{G}\mathcal{G}^\dagger \hat{D}_j \mathcal{G}, \tag{31}$$

$$= \Omega^{\mathcal{G}}[\hat{D}_i]\Omega^{\mathcal{G}}[\hat{D}_j]. \tag{32}$$

Hence we may represent any n-mode Gaussian gate $\mathcal{G}$ as a $(2n+1) \times (2n+1)$ matrix in the Heisenberg picture.

We can now compute the derivatives $\partial_\mu f$ using the derivatives of the matrix $M^{\mathcal{G}}(\mu)$. It turns out that, like the derivatives of the finite-dimensional gates in Sec. III, $\partial_\mu M^{\mathcal{G}}$ can be often decomposed into a finite linear combination of matrices from the same class as $M^{\mathcal{G}}$. In fact, the derivatives of all gates from a universal Gaussian gate set can be decomposed to just two terms, so derivative computations in this setting have the same complexity as in the qubit case. We summarize the derivatives of important Gaussian gates in Table II.

As an example, we consider the single-mode squeezing gate with zero phase $S(r, \phi = 0)$, which is represented by

$$M^S(r) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{-r} & 0 \\ 0 & 0 & e^r \end{pmatrix}. \tag{33}$$

Its derivative is given by

$$\partial_r M^S(r) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & -e^{-r} & 0 \\ 0 & 0 & e^r \end{pmatrix}. \tag{34}$$

The derivative itself is not a Heisenberg representation of a squeezing gate, but we can decompose it into a linear combination of such representations, namely

$$\partial_r M^S(r) = \frac{1}{2\sinh(s)}[M^S(r+s) - M^S(r-s)], \tag{35}$$

where s is a fixed but arbitrary nonzero real number. Hence

$$\partial_r[S(r)^\dagger \hat{B}_j S(r)] = \frac{1}{2\sinh(s)}[S(r+s)^\dagger \hat{B}_j S(r+s) - S(r-s)^\dagger \hat{B}_j S(r-s)] \tag{36}$$

for $j \in \{0, 1, 2\}$.

TABLE II. Parameter shift rules for the partial derivatives of important Gaussian gates. Every Gaussian gate can be decomposed into this universal gate set. We use the gate definitions laid out in the Strawberry Fields documentation [26] with $\hbar = 2$. All parameters are real valued. Single-mode gates have been expanded using the set $(\mathbb{1}, \hat{x}, \hat{p})$, whereas the two-mode beam splitter has been expanded using the set $(\mathbb{1}, \hat{x}_a, \hat{p}_a, \hat{x}_b, \hat{p}_b)$. More derivative rules can be found in the PennyLane [9] documentation.

Gate $\mathcal{G}$	Heisenberg representation $M^{\mathcal{G}}$	Partial derivatives of $M^{\mathcal{G}}$
Phase rotation $R(\phi)$	$M^R(\phi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{pmatrix}$	$\partial_\phi M^R(\phi) = \frac{1}{2}[M^R(\phi + \frac{\pi}{2}) - M^R(\phi - \frac{\pi}{2})]$
Displacement $D(r, \phi)$	$M^D(r, \phi) = \begin{pmatrix} 1 & 0 & 0 \\ 2r\cos\phi & 1 & 0 \\ 2r\sin\phi & 0 & 1 \end{pmatrix}$	$\partial_r M^D(r, \phi) = \frac{1}{2s}[M^D(r + s, \phi) - M^D(r - s, \phi)],\ s \in \mathbb{R}$ $\partial_\phi M^D(r, \phi) = \frac{1}{2}[M^D(r, \phi + \frac{\pi}{2}) - M^D(r, \phi - \frac{\pi}{2})]$
Squeezing $S(r)$[a]	$M^S(r) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{-r} & 0 \\ 0 & 0 & e^{r} \end{pmatrix}$	$\partial_r M^S(r) = \frac{1}{2\sinh(s)}[M^S(r + s) - M^S(r - s)],\ s \in \mathbb{R}$
Beam splitter $B(\theta, \phi)$	$M^B(\theta, \phi) = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \cos\theta & 0 & -\alpha & -\beta \\ 0 & 0 & \cos\theta & \beta & -\alpha \\ 0 & \alpha & -\beta & \cos\theta & 0 \\ 0 & \beta & \alpha & 0 & \cos\theta \end{pmatrix}$	$\partial_\theta M^B(\theta, \phi) = \frac{1}{2}[M^B(\theta + \frac{\pi}{2}, \phi) - M^B(\theta - \frac{\pi}{2}, \phi)]$ $\partial_\phi M^B(\theta, \phi) = \frac{1}{2}[M^B(\theta, \phi + \frac{\pi}{2}) - M^B(\theta, \phi - \frac{\pi}{2})]$ $\alpha = \cos\phi\ \sin\theta,\ \beta = \sin\phi\ \sin\theta$

[a]A more general version of the squeezing gate $\tilde{S}(r, \phi)$ also contains a parameter ϕ which defines the angle of the squeezing, and $S(r) = \tilde{S}(r, 0)$. This two-parameter gate can be broken down into a product of single-parameter gates: $\tilde{S}(r, \phi) = R(\frac{\phi}{2})S(r)R(-\frac{\phi}{2})$.

B. Differentiating CV circuits

Again we split the gate sequence into three pieces, $U(\theta) = V\mathcal{G}(\mu)W$. For simplicity, let us at first assume that our observable is a first-degree polynomial in the quadrature operators, and thus can be expanded as $\hat{B} = \sum_i b_i \hat{D}_i$. As shown in the previous section, for Gaussian gates the Heisenberg-picture matrix M is block diagonal, and maps from the space spanned by $\hat{D}$ onto itself. Thus, if $\mathcal{G}$ is Gaussian and V consists of Gaussian gates only, we may write

$$f(\theta) = \langle 0|U^\dagger(\theta)\hat{B}U(\theta)|0\rangle, \tag{37}$$

$$= \sum_{ijk} \langle 0|W^\dagger \hat{D}_k W|0\rangle M^{\mathcal{G}}_{kj}(\mu) M^V_{ji} b_i, \tag{38}$$

where $|0\rangle$ denotes the vacuum state. Now the derivative is simply

$$\partial_\mu f(\theta) = \sum_{ijk} \langle 0|W^\dagger \hat{B}_k W|0\rangle (\partial_\mu M^{\mathcal{G}})_{kj} M^V_{ji} b_i. \tag{39}$$

If $\partial_\mu M^{\mathcal{G}}$ can be expressed as a linear combination $\sum_i \gamma_i M^{\mathcal{G}}(\mu + s_i)$ with $\gamma_i, s_i \in \mathbb{R}$, by linearity we may express $\partial_\mu f$ using the same linear combination, $\partial_\mu f = \sum_i \gamma_i f(\mu + s_i)$. This is the parameter shift rule for CV quantum computing.

What about the subcircuit W that appears before the gate that we differentiate? For the purposes of differentiating the gate $\mathcal{G}$, this subcircuit can be arbitrary, since the above differentiation recipe does not depend on the properties of the matrix M^W. The above recipe works as long as no non-Gaussian gates are between $\mathcal{G}$ and the observable $\hat{B}$.

With observables $\hat{B}$ that are higher-degree polynomials of the quadratures, we can use the property in Eq. (30) to compute the derivative using the product rule:

$$\partial_\mu(\Omega^{\mathcal{G}}[\hat{B}_i\hat{B}_j]) = \partial_\mu(\Omega^{\mathcal{G}}[\hat{B}_i]\Omega^{\mathcal{G}}[\hat{B}_j]) = \partial_\mu\Omega^{\mathcal{G}}[\hat{B}_i]\ \Omega^{\mathcal{G}}[\hat{B}_j] + \Omega^{\mathcal{G}}[\hat{B}_i]\ \partial_\mu\Omega^{\mathcal{G}}[\hat{B}_j]. \tag{40}$$

C. Non-Gaussian transformations

For the above decomposition strategy to work efficiently, the subcircuit V must be Gaussian. In the case that V is non-Gaussian, it will generally increase the degree of the final observable, i.e., $V^\dagger\hat{B}V$ will be higher degree than $\hat{B}$. For example, the cubic phase gate $V(\gamma) = e^{i\gamma\hat{x}^3}$ carries out the transformations

$$V^\dagger(\gamma)\hat{x}V(\gamma) = \hat{x}, \tag{41}$$

$$V^\dagger(\gamma)\hat{p}V(\gamma) = \hat{p} + \gamma x^2. \tag{42}$$

In this case, we will have to consider a higher-dimensional subspace (tracking both the linear and the quadratic terms). If the subcircuit V contains multiple non-Gaussian gates, each one can raise the degree of the observable. Thus the matrices considered in the Heisenberg representation can become large depending on both the quantity and the character of non-Gaussian gates in the subcircuit V. Finding analytic derivative decompositions of circuits containing non-Gaussian gates is more challenging, but not strictly ruled out by complexity arguments. Specifically, in the case where there are only logarithmically few non-Gaussian gates, and each of those gates only raises the degree of quadrature polynomials by a bounded amount, there is still the possibility to efficiently decompose a gradient of an expectation value into a polynomial number of component expectation values.

V. CONCLUSION

We present several hardware-compatible strategies to evaluate the derivatives of quantum expectation values from the output of variational quantum circuits. In many cases of qubit-based quantum computing the derivatives can be computed with a simple parameter shift rule, using the variational architecture of the original quantum circuit. In all other cases it is possible to do the same by using an ancilla and a decomposition of the "derivative of a gate." For continuous-variable architectures we show that, as long as the parameter we differentiate with respect to feeds into a Gaussian gate that is only followed by Gaussian operations, a close relative to the parameter shift rule can be applied. We leave the case of non-Gaussian circuits as an open direction for future research.

[1] 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, Nat. Commun. **5**, 4213 (2014).

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

[3] J. Romero, J. P. Olson, and A. Aspuru-Guzik, Quantum autoencoders for efficient compression of quantum data, Quantum Sci. Technol. **2**, 045001 (2017).

[4] M. Schuld and N. Killoran, Quantum Machine Learning in Feature Hilbert Spaces, Phys. Rev. Lett. **122**, 040504 (2019).

[5] V. Havlicek, A. D. Córcoles, K. Temme, A. W. Harrow, J. M. Chow, and J. M. Gambetta, Supervised learning with quantum enhanced feature spaces, Nature **567**, 209 (2019).

[6] M. Schuld, A. Bocharov, K. Svore, and N. Wiebe, Circuit-centric quantum classifiers (2018), arXiv:1804.00633.

[7] E. Farhi and H. Neven, Classification with quantum neural networks on near term processors (2018), arXiv:1802.06002.

[8] S. Khatri, R. LaRose, A. Poremba, L. Cincio, A. T. Sornborger, and P. J. Coles, Quantum assisted quantum compiling (2018), arXiv:1807.00800.

[9] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, and N. Killoran, PennyLane: Automatic differentiation of hybrid quantum-classical computations (2018), arXiv:1811.04968.

[10] G. G. Guerreschi and M. Smelyanskiy, Practical optimization for hybrid quantum-classical algorithms (2017), arXiv:1701.01450.

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

[12] J.-G. Liu and L. Wang, Differentiable learning of quantum circuit born machine, Phys. Rev. A **98**, 062324 (2018).

[13] A. M. Childs and N. Wiebe, Hamiltonian simulation using linear combinations of unitary operations, Quantum Inf. Comput. **12**, 0901 (2012).

[14] The output of the circuit may consist of the measurements of n mutually commuting scalar observables; however, without loss of generality, they can always be combined into a vector-valued observable with n components.

[15] It is an open question whether such estimates have favorable properties similar to approximations of gradients in stochastic gradient descent.

[16] D. Maclaurin, D. Duvenaud, and R. P. Adams, Autograd: Effortless gradients in numpy, in *ICML 2015 AutoML Workshop* (2015).

[17] A. Harrow and J. Napp, Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms (2019), arXiv:1901.05374.

[18] The analysis makes a number of carefully described assumptions, for example that optimization takes place in a vicinity of the locally convex landscape, and in a black-box setting where the measurement operators are tensor products of Pauli operators.

[19] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, Automatic differentiation in machine learning: A survey, J. Mach. Learn. Res. **18**, 1 (2018).

[20] The rather elegant special case for generators G that are tensor products of Pauli matrices has been presented in Mitarai *et al.* [11]. Here we consider the slightly more general case.

[21] Google Inc., Cirq, https://cirq.readthedocs.io/en/latest/ (2018).

[22] The time-dependent prefactors are summarized as the gate parameter μ, while $b = \frac{J}{\Delta_{12}}$ represents the quotient of the interaction strength J and detuning Δ_{12} between the qubits, and $c = m_{12}$ is a cross-talk factor.

[23] 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.*, Simple All-Microwave Entangling Gate for Fixed-Frequency Superconducting Qubits, Phys. Rev. Lett. **107**, 080502 (2011).

[24] If α contains a renormalization so that $|\mathcal{G}| \leqslant 1$, and $\mathcal{G} = \mathcal{G}_{\rm re} + i\mathcal{G}_{\rm im}$ we can set $A_1 = \mathcal{G}_{\rm re} + i\sqrt{\mathbb{1} - \mathcal{G}_{\rm re}^2}$ and $A_2 = \mathcal{G}_{\rm im} + i\sqrt{\mathbb{1} - \mathcal{G}_{\rm im}^2}$.

[25] C. Weedbrook, S. Pirandola, R. García-Patrón, N. J. Cerf, T. C. Ralph, J. H. Shapiro, and S. Lloyd, Gaussian quantum information, Rev. Mod. Phys. **84**, 621 (2012).

[26] N. Killoran, J. Izaac, N. Quesada, V. Bergholm, M. Amy, and C. Weedbrook, Strawberry Fields: A software platform for photonic quantum computing, Quantum **3**, 129 (2019).