

Physical implementation of analytic derivatives on a photonic quantum computer

Sebastiano Corli[✉], Giorgio Panichi, Samuele Altiglia[✉], Edoardo Suerra[✉], Simone Cialdi, and Enrico Prati^{✉*}
Università degli Studi di Milano Dipartimento di fisica Aldo Pontremoli, I-20139 Milan, Italy

 (Received 24 February 2026; revised 21 April 2026; accepted 9 July 2026; published 31 July 2026)

A limiting factor to the exploitation of continuous variables in quantum computing, in particular in the field of quantum machine learning, has been the hindrances of computing the derivatives of a model. The constraint occurs when applying the parameter-shift rule to a photonic circuit, and the number of non-Gaussian gates exceeds a logarithmic trend with respect to the length of the circuit. This condition occurs when dealing with most nonlinear functions. Indeed, it hampers subroutines, such as backpropagation, which are paramount for machine learning and, most significantly, the computation of the derivatives of the model itself. To overcome this hurdle, we propose a method to compute derivatives of functions, implemented by continuous variable circuits, directly through photonic quantum circuits. Unlike the parameter-shift rule, this method does not fail when the number of non-Gaussian gates scales more than logarithmically against the depth of the circuit. We exploit the fact that computing the derivative of a wave function is equivalent to applying the momentum operator, which in the position basis acts as its derivative. The action of this operator is physically embodied by photon-added coherent states. We prove that, up to the second order of derivation, the number of operations does not scale with the depth of the circuit, but only with the order of derivation. We showcase numerical examples of derivatives of elementary functions parametrized on simulated photonic quantum circuits. Thanks to this method, we unleash an embodiment of quantum neural networks on photonic quantum computing platforms.

DOI: [10.1103/yrst-tr19](https://doi.org/10.1103/yrst-tr19)

I. INTRODUCTION

The class of continuous-variables (CV) quantum computers leverages the quadratures of the electromagnetic field in order to perform a floating-point calculus [1]. These variables indeed encode into the continuous spectrum of the canonical variables $\hat{q}$ and $\hat{p}$ the quantum information units, referred to as qumodes [2,3]. While linear transformations can be implemented by Gaussian operations, such as displacement or squeezing, nonlinear transformations require non-Gaussian gates, based on phenomena, such as the Kerr effect [2,4], even though measurement-induced techniques, relying on purely Gaussian operations and non-Gaussian source states, are rising [5–11]. In quantum machine learning, nowadays the paramount technique to compute gradients of parametric quantum circuits is implemented through the parameter-shift rule. This method consists of evaluating the quantum circuit shifting the value of a specific parameter, which provides the derivative of the circuit with respect to the

evaluated parameter [12–14]. This approach therefore grants a technique to perform gradient descent over quantum circuits on classical devices, even though bound to gates with two distinct eigenvalues—mainly single-qubit rotations. A focus on this technique is provided in Appendix A. To overcome such a limitation, innovative techniques have been recently introduced, e.g., decomposing the unitary operators in a polynomial expansion [15,16] or rather through the decomposition of the cost function in terms of discrete Fourier transform [12]. For instance, optimization techniques to train Gaussian Boson sampling (GBS) algorithms and compute the gradient over a cost function have been addressed in Ref. [17]. GBS is a paradigm of CV quantum computing, which relies on injecting squeezed light into the circuit, thereafter performing passive linear operations via a mesh of Mach-Zehnder interferometers and eventually counting the number of photons. Even though an exponential quantum advantage arises from switching from the CV frame to the discrete samples in the Fock space, all the operations involved in the circuit are Gaussian. A method for computing the gradients in the discrete variables (DV) frame, i.e., with single-photon sources, on passive interferometers has been addressed in Ref. [18].

Nevertheless, in the frame of CV computing, adding non-Gaussian gates hinders the feasibility of performing gradient descent techniques through the parameter-shift

*Contact author: enrico.prati@unimi.it

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

rule [13]. Specifically, since non-Gaussian operations increase the degree of the polynomial of $\hat{q}$ and $\hat{p}$, the higher the number of non-Gaussian gates, the more inefficient the parameter-shift rule becomes. When approximating a function through a photonic circuit, in the first place the encoding of the input variable has to be performed by a displacement gate, i.e., introducing a coherent state [2]. Employing the parameter-shift rule, calculating the derivatives of such a function becomes unfeasible by growing with the number of non-Gaussian operations, which are still mandatory for universal computation [4,7,19]. Such an issue thus prevents the computation of the derivatives of the primitive function, limiting the knowledge of the physical system it represents. This issue, for instance, poses a main obstacle when implementing quantum physics-informed neural networks (QPINNs), where the computation of first and second derivatives of the target functions is mandatory [20–25]. On qubit-based or hybrid architectures, this problem can be overcome through the parametric shift rule or classical backpropagation [26–29], while when shifting to qumodes, it has not been addressed yet. Even if gradient-free methods—such as COBYLA or Nelder-Mead [12,30–34]—are available for the parameters to be optimized, native methods for evaluating derivatives on photonic platform, with respect to the variables of integration, are essential.

In this work, we introduce a method alternative to the parameter-shift rule to analytically calculate the first and second derivatives of any function implemented on CV photonic quantum hardware. Contrary to the parameter-shift rule, this new method does not fail when the number of non-Gaussian gates exceeds a logarithmic trend, with respect to the length of the circuit. As an interesting fact, computing the derivative of a wave function is equivalent to applying the momentum operator after the displacement gate. Indeed, in first quantization, the same operator acts as the derivative operator over the wavefunction in the position basis. The method combines the output of multiple circuits—retrieved by a homodyne detection—into a single final result. As such, the method proves to be completely parallelizable, once the proper circuits have been assembled. Since most of the differential equations in physics are of the second order, this work limits to the development of a method to compute, for any function parametrized on a photonic CV circuit, the first- and the second-order derivatives. Unlike classical feed-forward neural networks, the complexity of this operation does not scale with the width of the circuit [35–37], but it depends solely on the order of derivation, mainly due to the preparation of the photon-added quantum states. The main strategy to compute directly derivatives on a photonic hardware relies on photon-added coherent states (PACS), generated by the excitation of single or multiple photons of a coherent state [38–41]. Single- and double-photon sources can be achieved through

experimental techniques known as parametric down conversion [42–45].

In Sec. II, we formalize the expression of the first and second derivatives of the target function, represented by the photonic circuit. In Sec. III, we sum up some staple notions about PACS and their properties. In Sec. IV, we build the quantum circuits to compute such derivatives on a photonic quantum computer through the photon-added coherent state. In Sec. V, we showcase some numerical examples to prove the validity of the analytic expression. In Sec. VI, we compare the complexity laws of classical feed-forward neural networks, when computing their derivatives, with respect to the optical quantum circuit.

II. DERIVATIVES OF THE TARGET FUNCTION

In this section, we outline the representation of the first and second derivatives for a given function $f_\theta(x)$ parametrized on a photonic circuit. This function is defined as $f_\theta: D \subseteq \mathbb{R} \rightarrow C \subseteq \mathbb{R}$. We show a complete proof for the first derivative, while we leave it in Appendix B for the second order.

Computing the values of $f_\theta(x)$ corresponds to retrieving the expectation value from a homodyne detection $\text{Tr}_o[\hat{\rho}_o \hat{q}_o]$. We refer to the index o as the output mode, which we want to perform the homodyne measurement on, since other modes can be involved throughout the process. In the Heisenberg picture, the same expression for $f_\theta(x)$ in terms of CV operators can be described as

$$\begin{aligned} f_\theta(x) &= \text{Tr}_o[\hat{\rho}_o \hat{q}_o] = \text{Tr}[\hat{\rho} \hat{q}_o] \\ &= \otimes^n \langle 0 | \hat{D}_i^\dagger(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | 0 \rangle^{\otimes n}, \end{aligned} \quad (1)$$

where $\hat{D}_i$ is the displacement operator, which installs the input variable x in the i th channel, while the $\hat{L}_\theta$ operator gathers all of the successive operations—i.e., the layers of the computation. Take notice, $\hat{L}_\theta$ is an operator acting between $\mathcal{F}^{\otimes n}$ Fock spaces, including the modes of the input and output channels. This operator can include linear multiport interferometers, squeezing and displacement operations or non-Gaussian operations as well. For further details on these layers of operations, see [2,21].

A. First derivative

In this section, we perform the derivative of the parametric function $f_\theta(x)$ defined in Eq. (1), and we assess how the differential operator acts on the Fock space. The circuit implementation of the expression achieved in Eq. (2) is provided in the first theorem in Sec. IV and in Fig. 2.

Theorem 1. Given an operator $\hat{L}_\theta: \mathcal{F}^{\otimes n} \rightarrow \mathcal{F}^{\otimes n}$ depending on a set θ of parameters, $\mathcal{F}$ being the Fock space, given an input mode indexed by $i \in \{1, \dots, n\}$ and an output mode indexed by $o \in \{1, \dots, n\}$, given a parametric

function $f_\theta: \mathbb{R} \rightarrow \mathbb{R}$, the first derivative of $f_\theta(x)$ as from Eq. (1) is

$$\begin{aligned} f'(x) &= -i\sqrt{2}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) \hat{p}_i \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | 0 \rangle^{\otimes n} \\ &\quad + i\sqrt{2}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{p}_i \hat{D}_i(x) | 0 \rangle^{\otimes n} \\ &= 2xf(x) - {}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | 0 \rangle^{\otimes n} + \\ &\quad - {}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) \hat{a}_i \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | 0 \rangle^{\otimes n}. \end{aligned} \quad (2)$$

Proof. The derivative of the $f_\theta(x)$ function means to derive the displacement operator with respect to its parameter x :

$$\begin{aligned} &{}^{\otimes n} \langle 0 | \partial_x [\hat{D}_i^\dagger(x)] \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | 0 \rangle^{\otimes n} \\ &\quad + {}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \partial_x [\hat{D}_i(x)] | 0 \rangle^{\otimes n}. \end{aligned} \quad (3)$$

We know that the displacement operator, for real values x , assumes the following representation:

$$\hat{D}(x) = \exp[x(\hat{a} - \hat{a}^\dagger)] = \exp[i\sqrt{2}x\hat{p}], \quad (4)$$

i.e., the expression of the translations in first quantization (with an additional $\sqrt{2}$ factor). Therefore, the expression for the derivative of $\hat{D}(x)$ reads as

$$\partial_x \hat{D}(x) = [\hat{a} - \hat{a}^\dagger] \hat{D}^\dagger(x) = i\sqrt{2}\hat{p} \hat{D}(x). \quad (5)$$

Inserting the above expression for $\partial_x \hat{D}_i(x)$ into the circuit in scheme (2), we then achieve

$$\begin{aligned} &-i\sqrt{2}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) \hat{p}_i \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | 0 \rangle^{\otimes n} \\ &\quad + i\sqrt{2}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{p}_i \hat{D}_i(x) | 0 \rangle^{\otimes n}, \end{aligned} \quad (6)$$

proving the first equality of the theorem. Take notice, since $\hat{p}$ and $\hat{D}(x)$ commute, the above scheme could be written swapping the two operators. Nevertheless, we now see how applying $\hat{p}$ after the displacement operator allows a feasible implementation of an optical circuit.

We now reshape the above expression in terms of ladder operators, thus the above expression is split into four terms:

$$\begin{aligned} &{}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) [\hat{a}_i^\dagger - \hat{a}_i] \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | 0 \rangle^{\otimes n} \\ &\quad + {}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta [\hat{a}_i - \hat{a}_i^\dagger] \hat{D}_i(x) | 0 \rangle^{\otimes n}. \end{aligned} \quad (7)$$

Because of the property of the coherent states, $\hat{a}_i \hat{D}_i(x) | 0 \rangle = x \hat{D}_i(x) | 0 \rangle$, the same expression can be read as

$$\begin{aligned} &{}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) [x - \hat{a}_i] \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | 0 \rangle^{\otimes n} \\ &\quad + {}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta [x - \hat{a}_i^\dagger] \hat{D}_i(x) | 0 \rangle^{\otimes n}. \end{aligned} \quad (8)$$

Reassembling the four terms properly:

$$\begin{aligned} &2xf_\theta(x) - {}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) \hat{a}_i \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | 0 \rangle^{\otimes n} + \\ &\quad - {}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{a}_i^\dagger \hat{D}_i(x) | 0 \rangle^{\otimes n}, \end{aligned} \quad (9)$$

which ends the proof. $\blacksquare$

Some physical considerations now arise. The term $\hat{a}^\dagger \hat{D}(x) | 0 \rangle$ must be interpreted as a single-photon-added coherent state (SPACS) and is well known and detailed in the literature how to prepare such a state. If instead we apply $i\sqrt{2}\hat{D}(x)\hat{p}|0\rangle = -\hat{D}(x)\hat{a}^\dagger|0\rangle$, such a state would have no physical interpretation.

B. Second derivatives

We now state a theorem about the explicit expression of the second derivative $f''(x)$. The circuitual implementation of the expression achieved in Eq. (10) is provided in the second theorem in Sec. IV and in Fig. 3.

Theorem 2. Given an operator $\hat{L}_\theta: \mathcal{F}^{\otimes n} \rightarrow \mathcal{F}^{\otimes n}$ depending on a set θ of parameters, $\mathcal{F}$ being the Fock space, given an input mode indexed by $i \in \{1, \dots, n\}$ and an output mode indexed by $o \in \{1, \dots, n\}$, given a parametric function $f_\theta: \mathbb{R} \rightarrow \mathbb{R}$, the second derivative of the function $f_\theta(x)$ in Eq. (1) reads

$$\begin{aligned} f''(x) &= 4x^2 f(x) - 2f(x) - 4xf'(x) \\ &\quad + {}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) \hat{a}_i^2 \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | 0 \rangle^{\otimes n} \\ &\quad + {}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) [\hat{a}_i^\dagger]^2 | 0 \rangle^{\otimes n} \\ &\quad + {}^{\otimes n} \langle 0 | \hat{D}_i^\dagger(x) \hat{a}_i \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{a}_i^\dagger \hat{D}_i(x) | 0 \rangle^{\otimes n}. \end{aligned} \quad (10)$$

Proof. See Appendix B.

III. PHYSICAL IMPLEMENTATION OF PACS

Before to describe the circuits able to perform the derivation of the function f , we first describe the physics of the photon-added coherent states, whose role is crucial in the method we propose. PACS can be considered as single-photon excitations of a classical coherent field [38], introducing quantum phenomena in between the characterization of a coherent state and a Fock state. Such states can be represented as

$$|\alpha, m\rangle = k_{\alpha, m} (\hat{a}^\dagger)^m |\alpha\rangle, \quad (11)$$

α represents here the field amplitude of the coherent state, m the population of the Fock state—the number of photons—and $k_{\alpha, m}$ is the proper normalization factor for this new state [41]:

$$k_{\alpha, m} = [m! L_m(-|\alpha|^2)]^{-1/2}, \quad (12)$$

with $L_m(x)$ being the Laguerre polynomials. For the SPACS, this peculiar state assumes the form

$$|\alpha, 1\rangle = k_{\alpha,1} \hat{a}^\dagger |\alpha\rangle = \frac{\hat{a}^\dagger |\alpha\rangle}{\sqrt{1 + |\alpha|^2}}. \quad (13)$$

In fact, we can notice that the norm of the $\hat{a}^\dagger |\alpha\rangle$ state is given by

$$\|\hat{a}^\dagger |\alpha\rangle\|^2 = \langle \alpha | \hat{a} \hat{a}^\dagger | \alpha \rangle = 1 + \langle \alpha | \hat{a}^\dagger \hat{a} | \alpha \rangle = 1 + |\alpha|^2, \quad (14)$$

from which the normalization in Eq. (13). Single photons can be generated by stimulating the spontaneous emission of a single down-converted photon in the mode of a coherent state, injected into an optical parameter amplifier. The Hamiltonian of interaction can be described as [38,46]

$$\hat{H} = i\hbar\chi(\hat{a}_i^\dagger \hat{a}_s^\dagger - \hat{a}_i \hat{a}_s), \quad (15)$$

where s refers to the signal mode, while i to the idle one. The χ parameter, instead, is proportional to the amplitude of the classical pump and the second-order susceptibility of the medium. The evolution of the initial state is described exponentiating the above Hamiltonian, $\exp(-i\hat{H}t/\hbar) = \exp(g(\hat{a}_i^\dagger \hat{a}_s^\dagger - \hat{a}_i \hat{a}_s))$, with $g = \chi t$ being the parameter gain. For $g \ll 1$, which means for very short times, the first-order expansion of the time evolution can be written as

$$\begin{aligned} |\psi(t)\rangle &= \hat{U}(t)|\psi(0)\rangle \\ &= [1 + g(\hat{a}_i^\dagger \hat{a}_s^\dagger - \hat{a}_i \hat{a}_s) + o(g^2)]|\alpha\rangle_s |0\rangle_i \\ &= |\alpha\rangle_s |0\rangle_i + g\hat{a}_s^\dagger |\alpha\rangle_s |1\rangle_i + o(g^2). \end{aligned} \quad (16)$$

With a few calculations, the probability to get the state with an idler signal 1 is $|g|^2(1 + |\alpha|^2)$. If getting up to the $o(g^3)$ order in the Taylor expansion, it is possible to detect a two photons-added coherent state as well [47], with probability $\frac{1}{4}|g|^4(|\alpha|^2 + 1)^2 = \frac{1}{4}|g|^4(|\alpha|^4 + 2|\alpha|^2 + 1)^2$, which is indeed proportional to the expression of the Laguerre polynomial $L_2(-|\alpha|^2)$. By extending the expansion up to the second order, we achieve the two photon-added coherent state (TPACS), which, we are seeing in the next sections, cover a paramount role in the algorithm we propose.

Experimentally, in the standard scheme [38], a weak coherent state $|\alpha\rangle$ is mode-matched with the signal output of a low-gain optical parametric amplifier, while the idler mode is monitored by a single-photon detector, see Fig. 1. A dichroic mirror (DM) is set to tune a total reflectivity for the coherent state $|\alpha\rangle$ and complete transmissivity for the pump source. The outcome results into a superposition of the two sources of light. Take notice, the pump is a coherent state as well, with a wavelength $\lambda_{\text{pump}} = \lambda_\alpha/2$, $|\alpha\rangle$

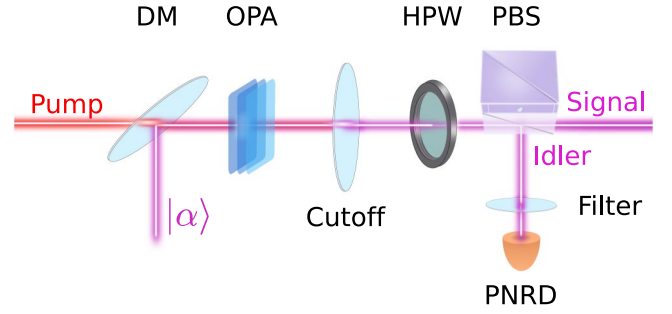


FIG. 1. In the above figure, the apparatus to generate a PACS is described. The signal is the $|\alpha, m\rangle$ state we are employing in Sec. IV to compute the derivatives of a parametric $f_\theta(x)$ function implementable on a quantum optical circuit.

being the coherent state $|\alpha\rangle$ we are evolving into a PACS. When the two beams of light enter the optical parametric amplifier (OPA), the pump source generates couples of photons, namely the signal and the idler, through a non-linear process in the crystal described by the Hamiltonian in Eq. (15) and the process in Eq. (16). Nonetheless, the photons emitted by this process half their wavelength, matching indeed λ_α . In the next step, the pump radiation is filtered by an opportune cutoff. The idle and signal photons are therefore separated by a polarized beam-splitter (PBS). In order to match the splitting induced by the beam splitter, a half wave plate (HWP) rotates the polarization of the incoming beams. Eventually, before the idle photons enter a photon-number-resolver detector (PNRD), accurate spatial and temporal mode matching between the coherent input and the down-converted field is ensured by applying an appropriate spatial and spectral filtering to the idler channel, typically using single-mode fibers and Bragg-grating filters. In the low-gain regime, conditioning on the detection of a photon in the idler projects the signal mode onto a single-photon-added coherent state $\hat{a}^\dagger |\alpha\rangle$ up to normalization. The scheme can be extended to higher-order photon addition by operating at higher pump powers and conditioning on multiphoton detection events in the idler, as originally proposed in [48]. Recent experiments have demonstrated the heralded generation of multiphoton-added coherent states up to three added photons by employing multiplexed on-off detectors acting as effective photon-number-resolving detectors [49] (see Fig. 1). In these implementations, coincidence detection across several detector channels enables selective projection onto $(\hat{a}^\dagger)^m |\alpha\rangle$, while homodyne tomography is used for state characterization. Although true photon-number resolving detectors based on superconducting technologies promise improved efficiency and scalability, present realizations remain probabilistic and strongly limited by mode matching, detection efficiency, and the rapidly decreasing heralding rates for increasing photon number.

IV. CIRCUITAL IMPLEMENTATION OF THE DERIVATIVES

In order to implement $f(x)$, all we need is a circuit endowed with a single mode, or even multiple modes picking up the i th to encode the input. Nevertheless, to compute its derivatives $f'(x)$ and $f''(x)$ an auxiliary mode is required, and moreover, we need PACS as input states for the computation. In this section, we describe how to compute the expressions for the first and second derivatives in Eqs. (2) and (10) via some opportune circuits. We provide the explicit calculus for the first derivative, delegating it to Appendix C for the second derivative. An alternative approach to compute the first derivatives is proposed in Appendix D

A. Circuitual implementation of the first derivative

We now state a theorem, which allows to build a circuitual representation of the result in Eq. (2).

Theorem 3. For a given Fock space $\mathcal{F}^{\otimes n} \otimes \mathcal{F}$, let $\hat{L}_\theta: \mathcal{F}^{\otimes n} \rightarrow \mathcal{F}^{\otimes n}$, let $\hat{B}S_{jk}$ be a 50:50 balanced beam-splitter entangling the channels j and k , let $i \in \{1, \dots, n\}$ be the channel where to encode the input variable and $a = n + 1$ the auxiliary mode. Labeling $|\sqrt{2}x\rangle_i$ the coherent state on the i th mode with average on the $\hat{q}$ quadrature equal to $\sqrt{2}x \in \mathbb{R}$, $|0\rangle_a$, and $|1\rangle_a$ as Fock states with occupation levels, respectively, of 0 and 1 on the auxiliary mode, and $|\sqrt{2}x, 1\rangle_i$ a single-photon-added coherent state on the i th mode, we can describe the signals $g_{io}^{(-)}(x)$, $g_{io}^{(+)}(x)$ from two circuits, respectively, as

$$\begin{aligned} g_{io}^{(-)}(x) &= {}_a\langle 1 | {}_i\langle \sqrt{2}x | \hat{B}S_{ia}^\dagger \langle \mathbf{0} | \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta | \mathbf{0} \rangle \hat{B}S_{ia} | \sqrt{2}x \rangle_i | 1 \rangle_a \end{aligned} \quad (17a)$$

$$\begin{aligned} g_{io}^{(+)}(x) &= {}_a\langle 0 | {}_i\langle \sqrt{2}x, 1 | \hat{B}S_{ia}^\dagger \langle \mathbf{0} | \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta | \mathbf{0} \rangle \hat{B}S_{ia} | \sqrt{2}x, 1 \rangle_i | 0 \rangle_a, \end{aligned} \quad (17b)$$

where $|\mathbf{0}\rangle = |0\rangle^{\otimes n-1}$. Once these two values are retrieved, the first derivatives $f'_\theta(x)$ can be computed as the combination of these signals along with x and $f_\theta(x)$:

$$2xf(x) - \frac{g_{io}^{(+)}(x)k_{x,1}^{-2} - g_{io}^{(-)}(x)}{x}, \quad (18)$$

with $k_{x,1}$ the normalization factor for the PACS described in Eq. (12).

Proof. First, we sort out the action of the balanced beam-splitter over the Fock states and the coherent/photon-added coherent states. These two states can be described, in a compact notation, as

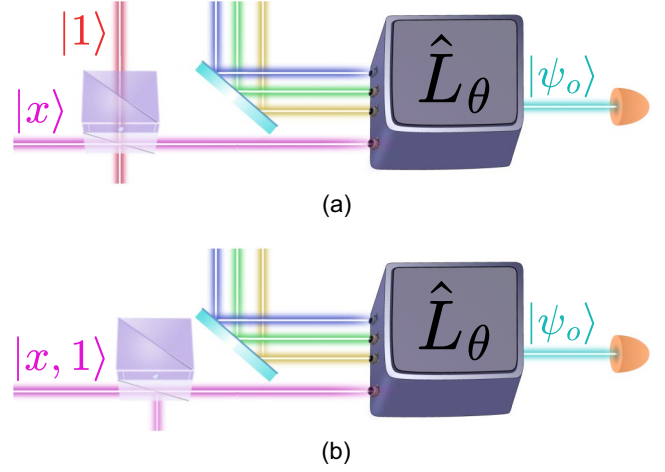


FIG. 2. The input mode (violet) is processed through a balanced beam-splitter; therefore, it enters the circuit, implemented by the $\hat{L}_\theta$ operator, along with all the other modes (blue, yellow, and green). Eventually, on the output mode (cyan), a homodyne detection is performed. (a) The input state $|x\rangle$ is a coherent one, and it is made to interact with a single-photon state $|1\rangle$ into the beam splitter. (b) The input state $|x, 1\rangle$ is a SPACS, and it is made to interact with the vacuum state into the beam splitter.

$$k_{x,1}^{s_j} \hat{a}_j^\dagger \hat{D}_i(\sqrt{2}x) |0\rangle_i |0\rangle_a. \quad (19)$$

Here, if $j = i$, we have the photon-added coherent state for the i th mode along with the vacuum state on the auxiliary channel; otherwise, if $j = a$, we have a single-photon state for a and a coherent state on i . The s_j factor is equal to 0 if $j = a$, since no normalization is required, while $s_j = 1$ for $j = i$ in order to normalize the SPACS in the Heisenberg picture. These two circuits are reported in Fig. 2. Recalling the expression for the displacement operator in Eq. (4), acting with the balanced beam-splitter transforms the above operators into

$$k_{x,1}^{s_j} \frac{\hat{a}_i^\dagger \pm \hat{a}_a^\dagger}{\sqrt{2}} \hat{D}_i(x) \hat{D}_a(x) |0\rangle_i |0\rangle_a = k_{x,1}^{s_j} \frac{\hat{a}_i^\dagger \pm \hat{a}_a^\dagger}{\sqrt{2}} |x\rangle_i |x\rangle_a. \quad (20)$$

The $+$ sign holds if $j = i$, while the $-$ one for the $j = a$ case. For further details about the transformation on the displacement operator, see Eq. (C3). Afterward, we apply the operator $\hat{L}_\theta$ in order to allocate the $f_\theta(x)$ function on the optical circuit, and we thus perform the homodyne measurement over the whole state $|\Psi\rangle = |\mathbf{0}\rangle |x\rangle_a |x\rangle_i$:

$$k_{x,1}^{2s_j} {}_a\langle x | {}_i\langle x | \langle \mathbf{0} | \frac{\hat{a}_i \pm \hat{a}_a}{\sqrt{2}} \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \frac{\hat{a}_i^\dagger \pm \hat{a}_a^\dagger}{\sqrt{2}} | \mathbf{0} \rangle | x \rangle_i | x \rangle_a. \quad (21)$$

Again, $|\mathbf{0}\rangle = |0\rangle^{\otimes n-1}$. Call $g_{io}^{(+)}(x)$ the signal from the circuit with $j = i$, $g_{io}^{(-)}(x)$ the circuit with $j = a$. If we

subtract these signals, we achieve

$$\begin{aligned} & k_{x,1}^{-2} g_{io}^{(+)}(x) - g_{io}^{(-)}(x) \\ &= {}_a \langle x |_i \langle x | \hat{a}_a^\dagger \langle \mathbf{0} | \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{a}_i^\dagger | \mathbf{0} \rangle | x \rangle_i | x \rangle_a \\ &+ {}_a \langle x |_i \langle x | \hat{a}_i^\dagger \langle \mathbf{0} | \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{a}_a^\dagger | \mathbf{0} \rangle | x \rangle_i | x \rangle_a, \end{aligned} \quad (22)$$

but since $\langle x | \hat{a} | x \rangle = \langle x | \hat{a}^\dagger | x \rangle = x$, the ladder operators over the auxiliary mode can be factorized out, achieving

$$\begin{aligned} & k_{x,1}^{-2} g_{io}^{(+)}(x) - g_{io}^{(-)}(x) \\ &= x ({}_i \langle x | \langle \mathbf{0} | \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{a}_i^\dagger | \mathbf{0} \rangle | x \rangle_i \\ &+ {}_i \langle x | \langle \mathbf{0} | \hat{a}_i^\dagger \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta | \mathbf{0} \rangle | x \rangle_i). \end{aligned} \quad (23)$$

Comparing the above result with the expression for $f'(x)$ in Eq. (2), we have proved the initial statement. ■

The algorithm to implement the first derivative of $f_\theta(x)$ is summed up in the pseudocode 1.

B. Circuit for the second derivative

We now state a theorem, which allows to build a circuital representation of the result in Eq. (10).

Theorem 4. For a given Fock space $\mathcal{F}^{\otimes n} \otimes \mathcal{F}$, let $\hat{L}_\theta: \mathcal{F}^{\otimes n} \rightarrow \mathcal{F}^{\otimes n}$, let $\hat{B}S_{jk}$ be a 50:50 balanced beam-splitter entangling the channels j and k , let $i \in \{1, \dots, n\}$ be the channel where to encode the input variable and $a \notin \{1, \dots, n\}$ an auxiliary mode. Labeling $|\sqrt{2}x\rangle_i$ a coherent state on the i th mode with field amplitude $\sqrt{2}x \in \mathbb{R}$, $|0\rangle_a$ and $|1\rangle_a$ as Fock states with occupation levels, respectively, of 0 and 1 on the auxiliary mode a , $|\mathbf{0}\rangle = |0\rangle^{\otimes n-1}$ and $|\sqrt{2}x, 1\rangle_i$ a single-photon-added coherent state on the i th mode, we can describe the signals $g_{io}^{(+)}(x)$, $g_{io}^{(-)}(x)$, $g_{2\gamma}(x)$, and $g_{1\gamma}(x)$ from four circuits as

$$\begin{aligned} & g_{io}^{(+)}(x) \\ &= {}_a \langle 0 |_i \langle \sqrt{2}x, 2 | \hat{B}S_{ia}^\dagger \langle \mathbf{0} | \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta | \mathbf{0} \rangle \hat{B}S_{ia} | \sqrt{2}x, 2 \rangle_i | 0 \rangle_a \end{aligned} \quad (24a)$$

$$\begin{aligned} & g_{io}^{(-)}(x) \\ &= {}_a \langle 2 |_i \langle \sqrt{2}x | \hat{B}S_{ia}^\dagger \langle \mathbf{0} | \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta | \mathbf{0} \rangle \hat{B}S_{ia} | \sqrt{2}x \rangle_i | 2 \rangle_a \end{aligned} \quad (24b)$$

$$g_{2\gamma}(x) = {}_i \langle \sqrt{2}x, 2 | \langle \mathbf{0} | \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta | \mathbf{0} \rangle | \sqrt{2}x, 2 \rangle_i \quad (24c)$$

$$g_{1\gamma}(x) = {}_i \langle \sqrt{2}x, 1 | \langle \mathbf{0} | \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta | \mathbf{0} \rangle | \sqrt{2}x, 1 \rangle_i \quad (24d)$$

then the second derivatives $f''_\theta(x)$ can be described as

$$\begin{aligned} & \left(k_{x,2}^{-2} g_{io}^{(+)}(x) + g_{io}^{(-)}(x) - \frac{k_{x,2}^{-2}}{2} f_\theta(x) - 2g_{1\gamma}(x) k_{x,1}^{-4} \right. \\ & \left. - \frac{k_{x,2}^{-2}}{2} g_{2\gamma}(x) \right) \frac{2}{x^2} + k_{x,1}^{-2} g_{1\gamma}(x) + 4x^2 f(x) \\ & - 2f(x) - 4xf'(x). \end{aligned} \quad (25)$$

Proof. See Appendix C. The four circuits are represented in Fig. 3. The algorithm for differentiating twice the primitive function f_θ is summed up in the pseudocode 2.

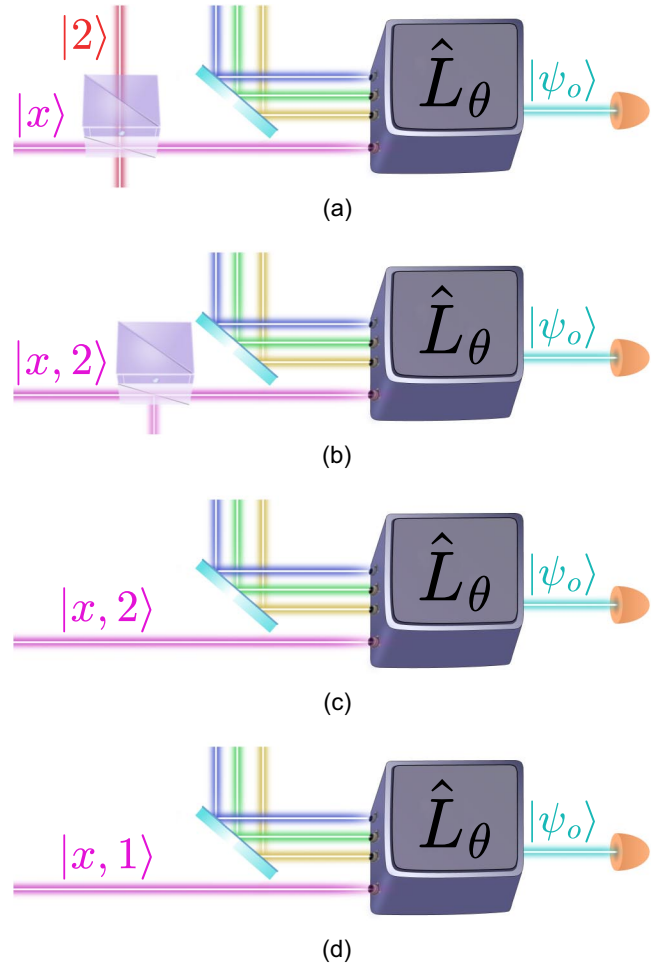


FIG. 3. (a) The input coherent state (violet) is made to interact with a Fock state with occupation 2 via a balanced beam-splitter, then it enters the circuit, implemented by the $\hat{L}_\theta$ logic gate, along with the other computational modes (blue, yellow, and green). Eventually, on the output mode (cyan), a homodyne detection is performed. (b) The coherent state is replaced by a TPACS, and it interacts with the vacuum state in the 50:50 beam-splitter before to enter the circuit. (c) The input state—a TPACS—enters the photonic circuit. (d) The input state—a SPACS—enters the photonic circuit.

TABLE I. Training/evaluation hyperparameters.

Hyperparameter	Value
Run name	constant_lr
Target functions	$\sin(\pi x)$, x^3
LR scheduler	constant
Learning rate	1.0×10^{-2}
Epochs	5000
Collocation points per epoch	50
Input domain	$x \in [-1, 1]$
QNN layers	5
Qumodes	1
Fock cutoff	30
Derivative evaluation points	100
BS derivative scheme	enabled

V. NUMERICAL SIMULATIONS

In this section, we report numerical simulations supporting the derivative-evaluation scheme introduced in Sec. IV. The method we propose is the following. We first train the variational circuit to fit a specific target function $f(x)$ — $\sin(\pi x)$ and x^3 , as from Fig. 4—which

required to train the model with respect to θ , see the loss function in Eq. (28). Thereafter, once the analytical expression of the function is encoded by the variational circuit with sufficient accuracy, i.e., $f_\theta(x) \simeq f(x)$, we can benchmark the results from the method we propose—the expression of $f'_\theta(x)$ and $f''_\theta(x)$ —along with the analytic expression of the derivatives $f'(x)$, $f''(x)$ with respect to the input value x . All simulations are performed with Strawberry Fields, a full-stack Python library for designing and simulating continuous-variable photonic circuits on classical hardware [50]. For the variational training, we employ TensorFlow [51].

A. CV-QNN model and classical emulation

The primitive function $f_\theta(x)$ is represented by a single-mode continuous-variables quantum neural network (CV-QNN) with input encoding $\hat{D}(x)$ followed by a depth- L variational block $\hat{L}(\theta)$ [20–22]. In the numerical experiments reported here, we use $L = 5$ layers and a truncated Fock simulation with cutoff dimension $d_{\text{cut}} = 30$. By repeated experiments, we found that these hyperparameters provide a good tradeoff between a good accuracy

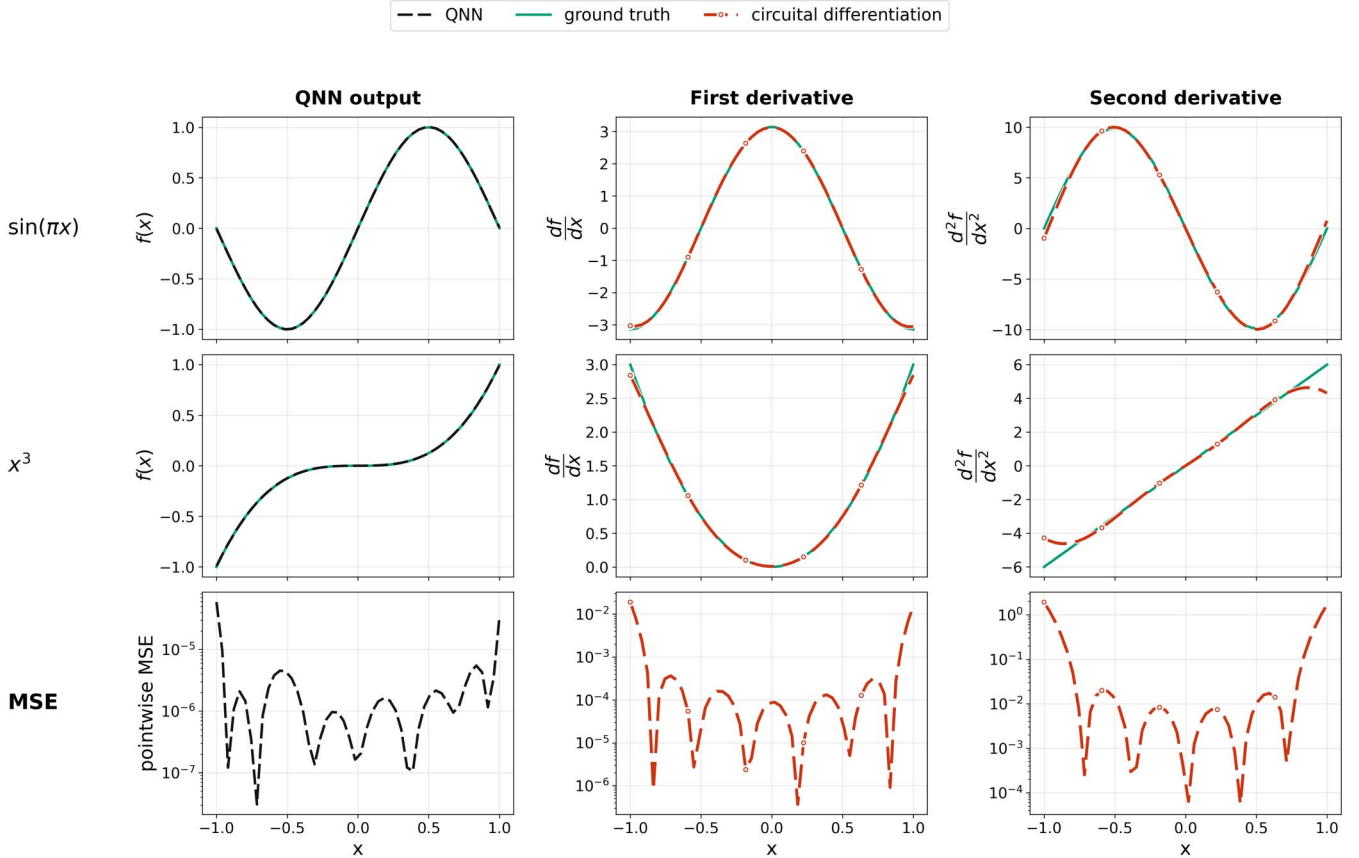


FIG. 4. We showcase two numerical examples, i.e., we first train the variational circuit to fit the functions $\sin(\pi x)$ and x^3 in the $[-1, +1]$ domain. Thereafter, we compute the circuit derivatives for the first and second orders, and we compare the results with the ground truth. In the last row, we exhibit the pointwise MSE loss function (i.e., the MSE computed for each sampled point inside the domain of training) averaged over both the two functions, as reported in Table II.

TABLE II. Mean squared error (MSE) vs ground truth.

Function	MSE[f]	MSE $[\frac{df}{dx}]_{\text{Tape}}$	MSE $[\frac{df}{dx}]_{\text{Circuit}}$	MSE $[\frac{d^2f}{dx^2}]_{\text{Tape}}$	MSE $[\frac{d^2f}{dx^2}]_{\text{Circuit}}$
$\sin(\pi x)$	2.874×10^{-6}	7.383×10^{-4}	7.383×10^{-4}	2.173×10^1	8.831×10^{-2}
x^3	4.140×10^{-6}	1.575×10^{-3}	1.575×10^{-3}	2.289×10^{-1}	2.440×10^{-1}
Mean over functions	3.507×10^{-6}	1.157×10^{-3}	1.156×10^{-3}	1.098×10^1	1.662×10^{-1}

and the computational bottleneck of the classical emulation. Each layer applies the gate sequence

$$\hat{L} = \hat{R}(\phi_1)\hat{S}(r)\hat{R}(\phi_2)\hat{D}(\alpha)\hat{K}(\kappa), \quad (26)$$

i.e., phase rotation, squeezing, phase rotation, displacement, and Kerr nonlinearity. By default, implementing $\hat{L}$ does not require additional modes, see Ref. [2], still the effective number of possible auxiliary qumodes depends on the hardware and on the heuristic process of optimization, see Ref. [21]. The number of auxiliary modes in the circuit, in fact, may depend on the performances of the variational circuit, but does not vary with the computation of the derivatives, where the number of additional modes is fixed to 1. In the numerical simulations we carried on, we set a total number of qumodes $n = 1$ during the training to optimize θ , and we added an additional ancillary mode for the computation of both the first and second derivatives with respect to x once the model has been trained, see Figs. 2 and 3. Referring to Eq. (1), the output of the model is defined as the expectation value of the $\hat{q}$ quadrature in the output mode,

$$f_\theta(x) = \langle \hat{q} \rangle, \quad (27)$$

computed from the simulated state.

B. Training protocol (function approximation)

We train two independent instances of the CV-QNN to approximate the target functions $\sin(\pi x)$ and x^3 over the domain $x \in [-1, 1]$. Training points are generated once per epoch using a one-dimensional Latin-hypercube design with $N = 50$ points, and the loss is the mean squared error

$$\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N (f_\theta(x_i) - f(x_i))^2. \quad (28)$$

Optimization is performed with Adam for 5000 epochs using a full-batch update (batch size $N = 50$) and learning rate $\eta = 10^{-2}$. All trainable parameters are initialized i.i.d. from small normal distributions (standard deviations approximately 10^{-1} for linear-Gaussian gates and approximately 2×10^{-2} for the Kerr strength), and we fix random seeds to ensure reproducibility.

C. Derivative evaluation circuits

After training, derivatives with respect to the *input variable* x are computed by executing a constant number of auxiliary circuits and classically recombining the corresponding homodyne expectation values according to Eqs. (18) and (25). In the simulations, we focus on the *circuitual differentiation* scheme, which reconstructs derivatives from a finite set of homodyne expectation values obtained by running a small number of auxiliary circuits in addition to the primitive one.

D. Circuitual differentiation

After the primitive function $f_\theta(x)$ has been trained, the next step consists of computing its derivatives. The procedure is summed up in pseudocodes 4 and 5, while the representation of the circuits is provided in Figs. 2 and 3.

In this circuitual differentiation scheme, we need to rescale the input displacement as $x \mapsto \sqrt{2}x$ to match the normalization adopted in the analytical derivation, see Eqs. (17) and (24).

E. Numerical stability near $x = 0$

The analytic recombination formulas contain factors $1/x$ and $1/x^2$, which induce removable discontinuities around $x = 0$ in finite-precision arithmetic, see Eqs. (18) and (25). In practice, we handle a small neighborhood of the origin by adopting a numerically stable evaluation rule for $|x| < \varepsilon$ (in the simulations we use $\varepsilon = 10^{-6}$), while using the closed-form recombination elsewhere.

F. Results

Figure 4 reports the learned approximations for $\sin(\pi x)$ and x^3 together with their first and second derivatives. For each target, we show (i) the CV-QNN output $f_\theta(x)$, (ii) the derivative obtained from the circuitual estimators, and (iii) the analytical derivatives of the target functions. The observed agreement confirms that the proposed derivative-evaluation protocol reproduces the correct derivatives of the trained CV-QNN while requiring only a constant number of circuit executions for first and second order.

VI. COMPLEXITY LAWS

In this section, we compare the complexity laws when computing derivatives f'_θ , f''_θ of the model $f_\theta(x)$ on both

classical networks and the circuitual scheme we proposed in the previous sections. We inquire whether a overhead or rather a speedup occur, comparing the proposed method with respect to classical backpropagation. We find that, even though a net quantum advantage cannot be stated, still a quantum utility arise, since the complexity in the number of operations can be reduced by a factor 2 when calculating the first derivatives, and 3 for the second ones. Contrary to the classical backpropagation, computing the first and second derivatives does not scale in terms of complexity.

A. Quantum evaluation of analytic derivatives

All the operations previously described on a quantum optical circuit are highly parallelizable. In the first place, we need a single circuit to evaluate $f_\theta(x)$, two circuits for f'_θ in the first scheme described in Sec. IV A, three in the second scheme reported in Appendix D, while four circuits to compute f''_θ as described in Sec. IV B. All the parameters θ must not vary between f_θ , f'_θ , and f''_θ . The complexity of these operations is summed up in the pseudocode 1 for the first-order derivative, and in pseudocode 2 for the second order.

As we see, in both the pseudocodes 1 and 2, all the operations to prepare the PACS are required before to start computing throughout the layers of the circuit, thus their complexity does not depend on the length of the circuit. Moreover, all the operations before and inside the *for* loop can be parallelized by installing a proper number of circuits to retrieve each outcome (3 circuits for the first derivative, 7 for the second derivative). Overall, to compute both the derivatives $f'_\theta(x)$, $f''_\theta(x)$, and the primitive $f_\theta(x)$, the complexity of the algorithm with respect to the length of the circuits scales as approximately $2nL$, L being

ALGORITHM 1. Quantum circuitual derivative (I).

```

1: # Primitive source state
2: Generate  $|\psi_0\rangle = \hat{D}(x)|0\rangle = |x\rangle$ 
3: # First derivative source states
4: Generate  $|\psi_1\rangle = |x\rangle|1\rangle$ 
5: Generate  $|\psi_2\rangle = |x, 1\rangle|0\rangle$ 
6: # 50:50 beam-splitters
7:  $|\psi_1\rangle = \hat{B}S(\frac{\pi}{4})|\psi_1\rangle$ 
8:  $|\psi_2\rangle = \hat{B}S(\frac{\pi}{4})|\psi_2\rangle$ 
9:  $\hat{L}_\theta = \hat{L}_1 \circ \hat{L}_2 \circ \dots \circ \hat{L}_L$ 
10: for  $l$  in  $\{1, \dots, L\}$  do
11:    $|\psi_0\rangle = \hat{L}_{\theta_l}|\psi_0\rangle$ 
12:    $|\psi_1\rangle = \hat{L}_{\theta_l}|\psi_1\rangle$ 
13:    $|\psi_2\rangle = \hat{L}_{\theta_l}|\psi_2\rangle$ 
14: end for
15:  $f(x) = \langle\psi_0|\hat{q}|\psi_0\rangle$ 
16:  $g_-(x) = \langle\psi_1|\hat{q}|\psi_1\rangle$ 
17:  $g_+(x) = k_{x,1}^{-2}\langle\psi_2|\hat{q}|\psi_2\rangle$ 
18:  $f'(x) = 2xf(x) - [g_+(x) - g_-(x)]/x$ 

```

ALGORITHM 2. Quantum circuitual derivative (II).

```

1: # Primitive source state
2: Generate  $|\psi_0\rangle = \hat{D}(x)|0\rangle = |x\rangle$ 
3: # First derivative source states
4: Generate  $|\psi_1\rangle = |x\rangle|1\rangle$ 
5: Generate  $|\psi_2\rangle = |x, 1\rangle|0\rangle$ 
6: # Second derivative source states
7: Generate  $|\psi_3\rangle = |x, 2\rangle|0\rangle$ 
8: Generate  $|\psi_4\rangle = |x\rangle|2\rangle$ 
9: Generate  $|\psi_5\rangle = |x, 2\rangle$ 
10: Generate  $|\psi_6\rangle = |x, 1\rangle$ 
11: # 50:50 beam-splitters
12: for  $i$  in  $\{1, 2, 3, 4\}$  do
13:    $|\psi_i\rangle = \hat{B}S(\frac{\pi}{4})|\psi_i\rangle$ 
14: end for
15: for  $i$  in  $\{1, \dots, L\}$  do
16:   for  $j$  in  $\{1, 2, 3, 4, 5, 6\}$  do
17:      $|\psi_i\rangle = \hat{L}_{\theta_l}|\psi_i\rangle$ 
18:   end for
19: end for
20:  $f(x) = \langle\psi_0|\hat{q}|\psi_0\rangle$ 
21:  $g_+(x) = \langle\psi_1|\hat{q}|\psi_1\rangle$ 
22:  $g_-(x) = \langle\psi_2|\hat{q}|\psi_2\rangle$ 
23:  $f'(x) = 2xf(x) - [g_+(x) - g_-(x)]$ 
24:  $g_+(x) = \langle\psi_3|\hat{q}|\psi_3\rangle$ 
25:  $g_-(x) = \langle\psi_4|\hat{q}|\psi_4\rangle$ 
26:  $g_{2\gamma}(x) = \langle\psi_5|\hat{q}|\psi_5\rangle$ 
27:  $g_{1\gamma}(x) = \langle\psi_6|\hat{q}|\psi_6\rangle$ 
28:  $f''(x) = [k_{x,2}^{-2}g_+(x) + g_-(x) - \frac{k_{x,2}^{-2}}{2}f_\theta(x) - 2g_{1\gamma}(x)k_{x,1}^{-4} +$   

 $- \frac{k_{x,2}^{-2}}{2}g_{2\gamma}(x)]\frac{2}{x^2} + k_{x,1}^{-2}g_{1\gamma}(x)$ 

```

the number of layers of operations being iterated by the $\hat{L}_\theta$ operator, which can be decomposed as a sequence of $\hat{L}_{\theta_l}$ operators being repeated per each layer, see the pseudocodes 1 and 2. Instead, n accounts the size of the Mach-Zehnder mesh, with n being the number of entries [21]. To implement any algebraic operation, this mesh needs to be repeated twice. Take notice, the size of the circuits, differently from the classical neural networks, does not vary through the layers, and usually n is very close to the number of input modes [20–22]. The sole overhead when preparing the circuits for the derivatives, compared with the f_θ one, consists of the 50: 50 beam-splitter and the preparation of the PACS. The main source of hindrances, indeed, derives from the process of parametric down conversion, which, as stated in Sec. III, still remains intrinsically a probabilistic process.

B. Classical evaluation of analytic derivatives

A classical feed-forward model can be modeled as a composition of multiple activation functions layer after layer, i.e.,

ALGORITHM 3. Classical feed-forward neural network.

```

1:  $y_m = y_m^{(0)} = x_m$  # input
2: for  $i$  in  $\{1, \dots, L\}$  do
3:    $z_m = \sum_k W_{mk}^{(l)} y_m + b_m^{(l)}$ 
4:    $y_m = \sigma(z_m)$ 
5: end for

```

$$f_{\theta}(\mathbf{x}) = \sigma_{\theta_L}^{(L)} \circ \sigma_{\theta_{L-1}}^{(L-1)} \circ \dots \circ \sigma_{\theta_1}^{(1)}(\mathbf{x}), \quad (29)$$

where θ collects all the parameters layer by layer, or in other words, $\theta = \bigcup_{l=1}^L \theta_l$. As we see, the analogy between the classical feed-forward neural network and the quantum parametric circuit is quite straightforward, since both the algorithms scale linearly with the number L of layers. We provide a pseudocode for the computation of the feed-forward neural network in the pseudocode 3, in order to check with ease that the complexity of the algorithm scales with approximately $\bar{n}L$, $\bar{n}$ being the averaged size of the hidden layers. In fact, while the algebraic operations between W_{mk} and y_m involve n steps (n being the entries of the input vector y_n) if the hardware allows complete parallelization, the $\sigma(z_m)$ operation can be accomplished within $O(1)$ if full parallelization, again, is allowed, see Eq. (E2).

Nevertheless, when computing the first derivative of this network w.r.t. its input, the complexity of the algorithm slightly varies. As it can be seen from the pseudocode 4, the process still involves L steps, assessing the complexity of the computation, with respect to the length of the network, up to $O(L)$. Each step requires to compute z_m , y_m , σ'_m , and $\partial_i y_m^{(l)}$, i.e., 4 objects. Consider a perfect parallelizing device to be available, so that all the operations (i.e., row by column multiplication) can be reduced to their minimal complexity. In this frame, computing z_m requires n steps, the complexity for the matrix times vector multiplication. Here, n stands for the size of the input vector y_m . Instead, computing y_m and σ'_m can be highly parallelized, involving a $O(1)$ scaling. Since we are considering a derivative over a single variable x_i , y_{im} can be dealt with as a vector, i being fixed. Therefore, the computation of $\partial_i y_m^{(l)}$ requires again n steps, resulting overall in a approximately $2n$ complexity. Averaging over the number of neurons per layer $\bar{n}$, the overall complexity to compute the first derivative of a feed-forward neural network can be assessed up to approximately $2\bar{n}L$.

We now focus on the computation of the second derivatives. We describe the overall procedure in the pseudocode 5. As we can see, each iteration demands 7 steps of computation, i.e., computing z_m , y_m , σ'_m , and σ''_m , the first derivatives $\partial_i y_m^{(l)}$ and $\partial_j y_m^{(l)}$ and, eventually, the second derivative $\partial_{ij}^2 y_m^{(l)}$. If we consider the case of a second derivative computed in the same variable, i.e.,

ALGORITHM 4. Classical feed-forward derivative (I).

```

1:  $y_m = y_m^{(0)} = x_m$  # input
2:  $y_{im} = \partial_i y_m^{(0)} = \delta_{im}$ 
3: for  $i$  in  $\{1, \dots, L\}$  do
4:    $z_m = \sum_k W_{mk}^{(l)} y_m + b_m^{(l)}$ 
5:    $y_m = \sigma(z_m)$ 
6:    $\sigma'_m = \sigma'(z_m)$ 
7:    $\partial_i y_m^{(l)} = \sigma'_m \sum_k W_{mk} y_{ik}$ 
8:    $y_{im} = \partial_i y_m^{(l)}$ 
9: end for

```

ALGORITHM 5. Classical feed-forward derivative (II).

```

1:  $y_m = y_m^{(0)} = x_m$  # input
2:  $y_{im} = \partial_i y_m^{(0)} = \delta_{im}$ 
3:  $y_{jm} = \partial_j y_m^{(0)} = \delta_{jm}$ 
4:  $y_{ijm} = \partial_{ij}^2 y_m^{(0)} = 0$ 
5: for  $i$  in  $\{1, \dots, L\}$  do
6:    $z_m = \sum_k W_{mk}^{(l)} y_k + b_m^{(l)}$ 
7:    $y_m = \sigma(z_m)$ 
8:    $\sigma'_m = \sigma'(z_m)$ 
9:    $\sigma''_m = \sigma''(z_m)$ 
10:   $\partial_i y_m^{(l)} = \sigma'(z_m) \sum_k W_{mk} y_{ik}$ 
11:   $\partial_j y_m^{(l)} = \sigma'(z_m) \sum_k W_{mk} y_{jk}$ 
12:   $y_{im} = \partial_i y_m^{(l)}$ 
13:   $y_{jm} = \partial_j y_m^{(l)}$ 
14:   $\partial_{ij}^2 y_m^{(l)} = \sigma''_m \sum_{kn} W_{mn}^{(l)} W_{mk}^{(l)} y_{im} y_{jm} + \sigma'_m \sum_k W_{mk}^{(l)} y_{ijm}$ 
15:   $y_{ijm} = \partial_{ij}^2 y_m^{(l)}$ 
16: end for

```

$\partial_{ii}^2 y_m^{(l)}$, the number of operations reduces to 6. Moreover, each matrix-vector multiplication incurs a complexity, which amounts to approximately n (n being the size of the input vector) if optimally parallelized, and $O(n^2)$ for matrix-matrix products (assuming the matrices are both squared). Overall, considering the hypothesis of a derivative over the same variable x_i , the y_{im} , y_{jm} , and y_{ijm} tensors can be reduced to rank 1 (i.e., they can be dealt with as vectors), thus the total amount of operations scales to approximately $3n$ per layer. The cumulative number of operations in the feed-forward network thus amounts to approximately $3\bar{n}L$, $\bar{n}$ being the averaged size of the hidden layers throughout the network itself.

Summing up the results, no quantum supremacy arises, nevertheless a quantum utility can be stated. In fact, while differentiating a classical feed-forward neural network implies raising the complexity of the algorithm by a factor 2 for the first derivatives with respect to the length L and by a factor 3 for the second order, the order of derivation, at least up to f''_{θ} , does not entail higher complexity. Yet, it is

worthy to stress that physical and engineering limitations still pose a hurdle to this implementation, in the first place when generating single photons through the parametric down conversion. Nonetheless, such hurdles do not concern the complexity of the algorithm, but instead the preparation of the input state (the PACS).

VII. CONCLUSIONS

In this work, we propose a method to compute the first and second derivatives for a parametric function implemented on a continuous-variables photonic circuit. We compare this result with the methods developed by Schuld *et al.* in their seminal work [13]. Although, in the CV frame, a pipeline for computing the gradient over the parameters θ is still not available, this work addresses a method to compute the first and second derivatives of the target function $f_\theta(x)$ implemented by the quantum circuit, with respect to the input values x . However, while the task of training the parameters θ can be accomplished with ease via gradient-free methods (such as the aforementioned COBYLA and Nelder-Mead), the computation of the derivatives cannot circumvent differentiation. The method proposed here overcomes the constraint of a logarithmic number of non-Gaussian gates, as established in [13], or rather numerical differentiation, an approach which leads into numerical imprecision and higher computational effort. Observe that, if the circuits were composed solely by Gaussian gates, the method we developed does not provide an advantage with respect to the parameter-shift rule. However, it is well known in the literature that no quantum advantage can be retrieved by purely Gaussian circuits, remarking the relevance of this new approach. This method works even when the number of non-Gaussian gates scales more than logarithmically against the depth of the circuit, a condition which makes the parameter-shift rule unfeasible. The physical phenomenon this new scheme relies on is based on the preparation of the input states as photon-added coherent states. Up to the second order of derivation, the number of operations does not scale with the depth of the circuit, but only with the order of derivation itself. We provide some numerical examples of derivatives of elementary functions, parametrized on simulated photonic quantum circuits, to showcase the correctness of this new method. Thanks to this method, the calculation of first and second derivatives is unleashed on general photonic parametric circuits.

ACKNOWLEDGMENTS

The authors acknowledge support from the Qgraph project funded via Centro Nazionale HPC, Big Data and Quantum Computing (Grant No. D43C22001240001) and the Qxtreme project funded via the Partenariato Esteso FAIR (Grant No. J33C22002830006). G. P. and E. P. gratefully thank Thales Alenia Space Italia for

partially supporting the PNRR Ph.D. Grant No. D.M. 630/2024.

The authors declare that the patent application 102025000034579 “Metodo e sistema di computazione quantistica per il calcolo di derivate, realizzato mediante circuiti fotonici a variabili continue” is pending (date 22/12/2025).

DATA AVAILABILITY

The data that support the findings of this article are not publicly available. The data are available from the authors upon reasonable request.

APPENDIX A: THE PARAMETER-SHIFT RULE

The parameter-shift rule was introduced to compute the derivatives for both the DV quantum gates and the CV ones, provided that a number of at most logarithmically non-Gaussian operations are applied to the circuit. Up to the statement of the authors from Ref. [13], when such a condition holds no more, no approach is actually known. We underline, again, the fact that the work we present is strictly focused on the displacement operators, applied at the beginning of the computation to encode the input variables, allowing in this case to evaluate the derivative of the objective function under any regime and number of non-Gaussian gates. To train and update the parameters θ of the variational circuits, gradient-free methods are already available. In this Appendix, we summarize the approach from Ref. [13] to the parameter-shift rule.

1. Discrete variables

When dealing with discrete variables, the paramount unit of information is the qubit, i.e., a quantum two-levels system. The paramount operators acting on the states of this system are represented by the Pauli gates $\hat{\sigma}_i$. Since $\hat{\sigma}_i^2 = I \ \forall i$, it follows that the sole eigenvalues of any σ_i are given by ± 1 . Furthermore, it also follows that, once $\hat{\sigma}_i$ is exponentiated, due to $\hat{\sigma}_i^2 = I$ we achieve

$$\hat{U}(\theta) = \exp(i\theta\hat{\sigma}_i) = \cos(\theta) + i\hat{\sigma}_i \sin(\theta). \quad (\text{A1})$$

We now define the objective function. In a variational circuit, this function derives from a measurement on a single observable $\hat{O}$ after many layers of parametric gates $\hat{U}(\theta_i)$:

$$f = \langle 0 | \hat{U}^\dagger(\theta_1) \dots \hat{U}^\dagger(\theta_m) \hat{O} \hat{U}(\theta_m) \dots \hat{U}(\theta_1) | 0 \rangle^{\otimes n}. \quad (\text{A2})$$

Here, m represents the total number of parametric gates to be applied, each one endowed with its parameter θ_i , and n the number of qubits. In most cases, all the qubits are

initialized in the $|0\rangle$ state. Suppose we want to derive f with respect to the i th parameter θ_i . In this case, we can write

$$\partial_i f = \partial_i \langle \psi | \hat{U}^\dagger(\theta_i) \hat{O}' \hat{U}(\theta_i) | \psi \rangle \quad (\text{A3})$$

where the new state reads $|\psi\rangle = \prod_{j=1}^{i-1} \hat{U}(\theta_j) |0\rangle^{\otimes n}$ and the observable $\hat{O}' = \prod_{j=m}^{i+1} \hat{U}^\dagger(\theta_j) \hat{O} \prod_{j=i+1}^m \hat{U}(\theta_j)$. The derivative of Eq. (A3) reads

$$\partial_i f = \langle \psi | \hat{U}^\dagger(\theta_i) \hat{O}'(i\hat{\sigma}) \hat{U}(\theta_i) | \psi \rangle + \text{H.c.} \quad (\text{A4})$$

H.c. standing for Hermitian conjugate. After this operation, we definitively reabsorb $\hat{U}(\theta_i)$ into $|\psi\rangle$, making the expectation value

$$\partial_i f = \langle \psi | \hat{O}'(i\hat{\sigma}) | \psi \rangle + \text{H.c.} \quad (\text{A5})$$

Now, we can state the following rule: for any operators $\hat{A}$, $\hat{B}$, $\hat{C}$ we have

$$\begin{aligned} \langle \psi | \hat{A}^\dagger \hat{B} \hat{C} | \psi \rangle + \text{H.c.} &= \frac{1}{2} [\langle \psi | (\hat{A} + \hat{C})^\dagger \hat{B} (\hat{A} + \hat{C}) | \psi \rangle \\ &\quad + \langle \psi | (\hat{A} - \hat{C})^\dagger \hat{B} (\hat{A} - \hat{C}) | \psi \rangle] \end{aligned} \quad (\text{A6})$$

We exploit this result into Eq. (A5), with $\hat{A} = \hat{I}$, $\hat{B} = \hat{O}'$, $\hat{C} = ir^{-1}\hat{\sigma}$, r being ± 1 :

$$\begin{aligned} \partial_i f &= \langle \psi | \hat{O}'(ir^{-1}\hat{\sigma}) | \psi \rangle + \text{H.c.} \\ &= \frac{1}{2} [\langle \psi | (\hat{I} + ir^{-1}\hat{\sigma})^\dagger \hat{O}' (\hat{I} + ir^{-1}\hat{\sigma}) | \psi \rangle \\ &\quad + \langle \psi | (\hat{I} - ir^{-1}\hat{\sigma})^\dagger \hat{O}' (\hat{I} - ir^{-1}\hat{\sigma}) | \psi \rangle] \end{aligned} \quad (\text{A7})$$

Indeed, if we set $\hat{U}(\frac{\pi}{4r})$, we notice that

$$\hat{U}\left(\frac{\pi}{4r}\right) = \cos(\pi/4) + ir^{-1}\sigma \sin(\pi/4) = \frac{\hat{I} + ir^{-1}\hat{\sigma}}{\sqrt{2}} \quad (\text{A8})$$

Summing the contributions from $\hat{U}(\frac{\pi}{4r})$ and $\hat{U}(-\frac{\pi}{4r})$, we achieve the two terms in Eq. (A7), i.e., the derivative of the objective function w.r.t. the parameter θ_i .

2. Continuous variables

In the optical CV frame, information is encoded by the quadratures operators $\hat{q}_i$, $\hat{p}_i$. Limiting the available transformations to Gaussian gates, we can introduce a vectorial operator gathering all the quadratures of the n modes:

$$\hat{\mathbf{C}} = (\hat{I}, \hat{q}_1, \hat{p}_1, \dots, \hat{q}_n, \hat{p}_n). \quad (\text{A9})$$

Indeed, the action of the Gaussian gates leaves the number of quadratures unaltered. Instead, non-Gaussian transformations increase the number of operators describing the system, e.g., $\hat{q}_i \hat{p}_j$, $\hat{q}_i^2$ and so on. We now see how this behavior affects the feasibility of the parameter-shift rule.

In the first place, the action of a unitary operator $\hat{U}(\theta)$ over the quadratures $\hat{\mathbf{C}}_i$ can be formulated as follows:

$$\hat{U}^\dagger(\theta) \hat{\mathbf{C}}_i \hat{U}(\theta) = \sum_j M_{ij}(\theta) \hat{\mathbf{C}}_j \quad (\text{A10})$$

To provide a concrete example, the authors from Ref. [13] sketch the action of the squeezing gate $\hat{S}(r)$. In the phase space, i.e., acting over the quadrature operators instead of the Fock space, its representation reads

$$M(r) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & e^{-r} & 0 \\ 0 & 0 & e^r \end{pmatrix}. \quad (\text{A11})$$

Since the squeezing is a single-mode operator, the vector of quadratures it is acting on is given by $(\hat{I}, \hat{q}, \hat{p})$. We now select an objective function, on which it acts a sequence of gates $\hat{V} \hat{U}(\theta) \hat{W}$. It is mandatory for $\hat{V}$ and $\hat{U}(\theta)$ to be Gaussian. This time, the objective function is computed over a linear combination of quadratures, $\hat{D} = \sum_i b_i \hat{\mathbf{C}}_i$. We thus have

$$\begin{aligned} f &= \langle 0 | \hat{W}^\dagger \hat{U}^\dagger(\theta) \hat{V}^\dagger \hat{D} \hat{V} \hat{U}(\theta) \hat{W} | 0 \rangle \\ &= \sum_{ijk} \langle 0 | \hat{W}^\dagger \hat{\mathbf{C}}_k \hat{W} | 0 \rangle b_i M_{jk}^U(\theta) M_{ij}^V. \end{aligned} \quad (\text{A12})$$

To compute the derivative of f w.r.t. θ , we simply achieve

$$\partial_\theta f = \sum_{ijk} \langle 0 | \hat{W}^\dagger \hat{\mathbf{C}}_k \hat{W} | 0 \rangle b_i \partial_\theta [M_{jk}^U(\theta)] M_{ij}^V. \quad (\text{A13})$$

The number of Gaussian operators is limited: squeezing, displacement, beam-splitters, and phase-shifter are all the Gaussian gates we can implement, and their derivatives in the phase space can be achieved as linear combinations of different circuits as for the DV in Appendix A 1, see Ref. [13].

By introducing non-Gaussian gates, the number of quadratures involved in the $\hat{\mathbf{C}}$ vectorial operator of Eq. (A9) increases exponentially, as well as the size of the Gaussian operators M in the phase space, which have to be adapted to the new quadratures as well. For this reason, the authors state that, if the number of non-Gaussian gates is limited to a logarithmic number w.r.t. the length of the circuit, gradient-free methods are still envisaged to be feasible.

APPENDIX B: PROOF OF THEOREM 2

In this Appendix, we provide an explicit proof of the theorem in Sec. II B. In this section, we adopt the convention $|\mathbf{0}\rangle = |0\rangle^{\otimes n}$ to streamline the notation. If applying the differential operator on $f_\theta(x)$, we achieve the following terms:

$$\begin{aligned} f''_\theta(x) &= \langle \mathbf{0} | \partial_x^2 [\hat{D}_i^\dagger(x)] \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | \mathbf{0} \rangle \\ &+ \langle \mathbf{0} | \hat{D}_i^\dagger(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \partial_x^2 [\hat{D}_i(x)] | \mathbf{0} \rangle \\ &+ 2 \langle \mathbf{0} | \partial_x [\hat{D}_i^\dagger(x)] \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \partial_x [\hat{D}_i(x)] | \mathbf{0} \rangle \end{aligned} \quad (\text{B1})$$

for which the expression for $\partial_x^2 \hat{D}(x)$ is required:

$$\partial_x^2 \hat{D}(x) = -2\hat{p}^2 \hat{D}(x) = [(\hat{a}^\dagger)^2 + \hat{a}^2 - \hat{a}\hat{a}^\dagger - \hat{a}^\dagger\hat{a}] \hat{D}(x). \quad (\text{B2})$$

The first term from Eq. (B1) can thus be written as

$$\langle \mathbf{0} | \hat{D}_i^\dagger(x) [(\hat{a}_i^\dagger)^2 + \hat{a}_i^2 - \hat{a}_i \hat{a}_i^\dagger - \hat{a}_i^\dagger \hat{a}_i] \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | \mathbf{0} \rangle \quad (\text{B3})$$

Nevertheless, some considerations arise. In the first place, $[\hat{a}; \hat{a}^\dagger] = 1$, thus $\hat{a}\hat{a}^\dagger = 1 + \hat{a}^\dagger\hat{a}$. Secondly, $\langle \mathbf{0} | \hat{D}(x) \hat{a}^\dagger = x \langle \mathbf{0} | \hat{D}(x)$, making the above expression to be rewritten as

$$\begin{aligned} &\langle \mathbf{0} | \hat{D}_i^\dagger(x) [x^2 + \hat{a}_i^2 - 2x\hat{a}_i - 1] \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | \mathbf{0} \rangle \\ &= x^2 f(x) + \langle \mathbf{0} | \hat{D}_i^\dagger(x) \hat{a}_i^2 \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | \mathbf{0} \rangle + \\ &- 2x \langle \mathbf{0} | \hat{D}_i^\dagger(x) \hat{a}_i \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | \mathbf{0} \rangle - f(x). \end{aligned} \quad (\text{B4})$$

A similar scheme occurs when evaluating the second term in Eq. (B1), this time with $\hat{a}\hat{D}(x)|0\rangle = x\hat{D}(x)|0\rangle$. Therefore, we set $\hat{a}\hat{a}^\dagger = 1 + \hat{a}^\dagger\hat{a}$.

$$\begin{aligned} &\langle \mathbf{0} | \hat{D}_i^\dagger(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta [x^2 + (\hat{a}_i^\dagger)^2 - 2x\hat{a}_i^\dagger - 1] \hat{D}_i(x) | \mathbf{0} \rangle \\ &= x^2 f(x) + \langle \mathbf{0} | \hat{D}_i^\dagger(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta (\hat{a}_i^\dagger)^2 \hat{D}_i(x) | \mathbf{0} \rangle + \\ &- 2x \langle \mathbf{0} | \hat{D}_i^\dagger(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{a}_i^\dagger \hat{D}_i(x) | \mathbf{0} \rangle - f(x). \end{aligned} \quad (\text{B5})$$

The third term in Eq. (B1) can be rendered as

$$\begin{aligned} &\langle \mathbf{0} | \hat{D}_i^\dagger(x) [-\hat{a}_i + \hat{a}_i^\dagger] \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta [\hat{a}_i - \hat{a}_i^\dagger] \hat{D}_i(x) | \mathbf{0} \rangle \\ &= -x \langle \mathbf{0} | \hat{D}_i^\dagger(x) \hat{a}_i \hat{D}_i(x) | \mathbf{0} \rangle \\ &+ \langle \mathbf{0} | \hat{D}_i^\dagger(x) \hat{a}_i \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{a}_i^\dagger \hat{D}_i(x) | \mathbf{0} \rangle \\ &+ x^2 f(x) - x \langle \mathbf{0} | \hat{D}_i^\dagger(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{a}_i^\dagger \hat{D}_i(x) | \mathbf{0} \rangle. \end{aligned} \quad (\text{B6})$$

Summing all of the terms together, the overall result reads:

$$\begin{aligned} f''(x) &= 4x^2 f(x) + \langle \mathbf{0} | \hat{D}_i^\dagger(x) \hat{a}_i^2 \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta | \mathbf{0} \rangle + \\ &- 4x \langle \mathbf{0} | \hat{a}_i \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{D}_i(x) | \mathbf{0} \rangle + \\ &- 2f(x) + \langle \mathbf{0} | \hat{D}_i(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta (\hat{a}_i^\dagger)^2 \hat{D}_i(x) | \mathbf{0} \rangle + \\ &- 4x \langle \mathbf{0} | \hat{D}_i(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{a}_i^\dagger \hat{D}_i(x) | \mathbf{0} \rangle \\ &+ 2 \langle \mathbf{0} | \hat{D}_i(x) \hat{a}_i \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{a}_i^\dagger \hat{D}_i(x) | \mathbf{0} \rangle. \end{aligned} \quad (\text{B7})$$

Nevertheless, the third and the sixth terms are proportional to the first derivatives. The expression can be thus reassembled into

$$\begin{aligned} f''(x) &= 4x^2 f(x) + \langle \mathbf{0} | \hat{D}_i^\dagger(x) \hat{a}_i^2 \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta | \mathbf{0} \rangle + \\ &- 4x f'(x) - 2f(x) \\ &+ \langle \mathbf{0} | \hat{D}_i(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta (\hat{a}_i^\dagger)^2 \hat{D}_i(x) | \mathbf{0} \rangle \\ &+ 2 \langle \mathbf{0} | \hat{D}_i(x) \hat{a}_i \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{a}_i^\dagger \hat{D}_i(x) | \mathbf{0} \rangle, \end{aligned} \quad (\text{B8})$$

which ends the proof. $\blacksquare$

The terms that need to be computed are the following:

$$\langle \mathbf{0} | \hat{D}_i^\dagger(x) \hat{a}_i^2 \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta | \mathbf{0} \rangle \quad (\text{B9a})$$

$$\langle \mathbf{0} | \hat{D}_i(x) \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta (\hat{a}_i^\dagger)^2 \hat{D}_i(x) | \mathbf{0} \rangle \quad (\text{B9b})$$

$$\langle \mathbf{0} | \hat{D}_i(x) \hat{a}_i \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta \hat{a}_i^\dagger \hat{D}_i(x) | \mathbf{0} \rangle. \quad (\text{B9c})$$

The task to compute the above terms is accomplished in Sec. IV B.

APPENDIX C: PROOF OF THEOREM 4

In this section, we provide an explicit proof of the theorem in section. In order to implement second derivatives, we need two photons instead of a single source, then we apply the usual balanced 50: 50 beam-splitter. Again, as for the first derivatives, the input states read

$$k_{x,2}^{s_j} (\hat{a}_j^\dagger)^2 \hat{D}_i(\sqrt{2}x) |0_i\rangle |0_a\rangle \quad (\text{C1})$$

i being the input channel, and $j = i$ for the $g_{io}^{(+)}(x)$ case, $j = a$ for the $g_{io}^{(-)}(x)$ one. Again, as in the case for the first derivative in Sec. IV A, $k_{x,2}$ is the normalization factor, this time for the TPACS, and $s_j = 0$ if $j = i$, otherwise 0. Depending on the value of the index i , when applying the balanced beam-splitter, we achieve

$$\begin{aligned} &\hat{B} S_{ia} [\hat{a}_j^\dagger]^2 \hat{D}_i(\sqrt{2}x) |0_i\rangle |0_a\rangle \\ &= \hat{B} S_{ia} [\hat{a}_j^\dagger]^2 \hat{D}_i(\sqrt{2}x) \hat{B} S_{ia}^\dagger \hat{B} S_{ia} |0_i\rangle |0_a\rangle. \end{aligned} \quad (\text{C2})$$

Now, considering that $\hat{a}_j \xrightarrow{\hat{B}S} (\hat{a}_i \pm \hat{a}_a)/\sqrt{2}$, we get

$$\begin{aligned}
\hat{B}S_{ia}\hat{D}_i(x\sqrt{2})\hat{B}S_{ia}^\dagger &= \hat{B}S_{ia}\exp((\hat{a}_i - \hat{a}_i^\dagger)x\sqrt{2})\hat{B}S_{ia}^\dagger \\
&= \exp((\hat{a}_i + \hat{a}_a - \hat{a}_i^\dagger + \hat{a}_a^\dagger)x) \\
&= \hat{D}_i(x)\hat{D}_a(x). \tag{C3}
\end{aligned}$$

Thus, the initial state evolves into

$$k_{x,2}^{s_j} \frac{1}{2} [(\hat{a}_i^\dagger)^2 + (\hat{a}_a^\dagger)^2 \pm 2\hat{a}_i^\dagger \hat{a}_a^\dagger] \hat{D}_i(x) \hat{D}_a(x) |0_i\rangle |0_a\rangle. \tag{C4}$$

We thereafter apply the $\hat{L}_\theta$ operator to achieve the final form of the state via two homodyne detections, which we named $g_{io}^{(+)}(x)$ and $g_{io}^{(-)}(x)$:

$$\begin{aligned}
g_{io}^{(\pm)}(x) &= k_{x,2}^{2s_j} \frac{1}{4} {}_a\langle x | {}_i\langle x | \langle 0 | [\hat{a}_i^2 + \hat{a}_a^2 \pm 2\hat{a}_i \hat{a}_a] \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta \\
&\quad \circ [(\hat{a}_i^\dagger)^2 + (\hat{a}_a^\dagger)^2 \pm 2\hat{a}_i^\dagger \hat{a}_a^\dagger] | 0 \rangle | x \rangle_i | x \rangle_a. \tag{C5}
\end{aligned}$$

Henceforth, in this section we adopt the convention $|0\rangle = |0\rangle^{\otimes n-1}$ to streamline the notation. If we sum these two contributions, i.e., we perform $k_{x,2}^{-2} g_{io}^{(+)}(x) + g_{io}^{(-)}(x)$, the overall signal we achieve is

$$\begin{aligned}
&\frac{1}{2} {}_a\langle x | {}_i\langle x | \langle 0 | [\hat{a}_i^2 + \hat{a}_a^2] \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta [(\hat{a}_i^\dagger)^2 + (\hat{a}_a^\dagger)^2] | 0 \rangle | x \rangle_i | x \rangle_a \\
&\quad + 2 {}_a\langle x | {}_i\langle x | \langle 0 | \hat{a}_i \hat{a}_a \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta \hat{a}_i^\dagger \hat{a}_a^\dagger | 0 \rangle | x \rangle_i | x \rangle_a. \tag{C6}
\end{aligned}$$

Considering the terms we need to compute in Equation, and applying the property of the coherent states $\langle x | \hat{a} | x \rangle = x$ and the normalization of the PACS, $\|(\hat{a}^\dagger)^m | x \rangle\| = k_{x,m}^{-2}$, we account for the following terms:

$$\frac{1}{2} {}_i\langle x | \langle 0 | \hat{a}_i^2 \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta (\hat{a}_i^\dagger)^2 | 0 \rangle | x \rangle_i \tag{C7a}$$

$$\frac{x^2}{2} {}_i\langle x | \langle 0 | \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta (\hat{a}_i^\dagger)^2 | 0 \rangle | x \rangle_i \tag{C7b}$$

$$\frac{x^2}{2} {}_i\langle x | \langle 0 | \hat{a}_i^2 \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta | 0 \rangle | x \rangle_i \tag{C7c}$$

$$\begin{aligned}
&\frac{1}{2} {}_i\langle x | \langle 0 | \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta | 0 \rangle | x \rangle_i {}_a\langle x | \hat{a}_a^2 (\hat{a}_a^\dagger)^2 | x \rangle_a \\
&= \frac{1}{2} f_\theta(x) k_{x,2}^{-2} \tag{C7d}
\end{aligned}$$

$$\begin{aligned}
&2 {}_i\langle x | \langle 0 | \hat{a}_i \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta \hat{a}_i^\dagger | 0 \rangle | x \rangle_i {}_a\langle x | \hat{a}_a \hat{a}_a^\dagger | x \rangle_a \\
&= 2 {}_i\langle x | \langle 0 | \hat{a}_i \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta \hat{a}_i^\dagger | 0 \rangle | x \rangle_i k_{x,1}^{-2}. \tag{C7e}
\end{aligned}$$

The $k_{x,1}^{-2}$ and $k_{x,2}^{-2}$ factors derive from the PACS normalization, see Eqs. (12) and (13). Looking at Eq. (10), we understand that the terms we need are those in Eqs. (C7b), (C7c), and (C7e), the last one still without

the $(1+x^2)$ factor. We therefore need to discard the terms in Eqs. (C7a) and (C7d). It is very straightforward to remove this last term, since the value of the primitive function is already available. To remove the first term in Eq. (C7a), it is possible to compute the primitive function but on a TPACS instead of a coherent state. We call this function $g_{2\gamma}(x)$

$$g_{2\gamma}(x) = {}_i\langle x, 2 |^{\otimes n-1} \langle 0 | \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta | 0 \rangle^{\otimes n-1} | x, 2 \rangle_i. \tag{C8}$$

The same strategy can be adopted for the term in Eq. (C7e), this time relying on a SPACS:

$$g_{1\gamma}(x) = {}_i\langle x, 1 |^{\otimes n-1} \langle 0 | \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta | 0 \rangle^{\otimes n-1} | x, 1 \rangle_i. \tag{C9}$$

The unknown terms in Eq. (10) can thus be summarized by computing

$$\begin{aligned}
&\left(k_{x,2}^{-2} g_+(x) + g_-(x) - \frac{k_{x,2}^{-2}}{2} f_\theta(x) - 2g_{1\gamma}(x) k_{x,1}^{-4} + \right. \\
&\quad \left. - \frac{k_{x,2}^{-2}}{2} g_{2\gamma}(x) \right) \frac{2}{x^2} + k_{x,1}^{-2} g_{1\gamma}(x), \tag{C10}
\end{aligned}$$

which ends the proof. $\blacksquare$

APPENDIX D: AN ALTERNATIVE CIRCUIT TO COMPUTE THE FIRST DERIVATIVES

In this section, we propose an alternative approach to the scheme developed in Sec. IV A to compute the first derivatives of a parametric function $f_\theta(x)$. If we consider just one out of the two homodyne signals, see Eq. (21), we are left with the following terms, up to the normalization factor $k_{x,1}^{-2}$ for the $g_{io}^{(\pm)}(x)$ case:

$$\frac{1}{2} {}_i\langle x |^{\otimes n-1} \langle 0 | \hat{a}_i \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta \hat{a}_i^\dagger | 0 \rangle^{\otimes n-1} | x \rangle_i \tag{D1a}$$

$$\pm \frac{x}{2} {}_i\langle x |^{\otimes n-1} \langle 0 | \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta \hat{a}_i^\dagger | 0 \rangle^{\otimes n-1} | x \rangle_i \tag{D1b}$$

$$\pm \frac{x}{2} {}_i\langle x |^{\otimes n-1} \langle 0 | \hat{a}_i \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta | 0 \rangle^{\otimes n-1} | x \rangle_i \tag{D1c}$$

$$\frac{1}{2} {}_i\langle x |^{\otimes n-1} \langle 0 | \hat{L}_\theta^\dagger \hat{Q}_o \hat{L}_\theta | 0 \rangle^{\otimes n-1} | x \rangle_i k_{x,1}^{-2} = \frac{1}{2} f_\theta(x) k_{x,1}^{-2}. \tag{D1d}$$

As usual, we deployed the properties of the coherent states and PACS, i.e., $\langle x | \hat{a} | x \rangle = x$ and $\langle x | \hat{a} \hat{a}^\dagger | x \rangle = k_{x,1}^{-2}$. The second and the third terms are required by Eq. (2); the others instead have to be discarded. We can notice that the first term in Eq. (D1a) is nothing but the computation of f_θ on a SPACS instead of a coherent state and can be achieved from the following circuit output:

ALGORITHM 6. Quantum circuitual derivative (I), alternative scheme.

```

1: # Primitive source state
2: Generate  $|\psi_0\rangle = \hat{D}(x)|0\rangle = |x\rangle$ 
3: # First derivative source states
4: Set  $j_{\pm}$ 
5: if  $j_+$  then
6:   Generate  $|\psi_1\rangle = |x, 1\rangle|0\rangle$ 
7: else if  $j_-$  then
8:   Generate  $|\psi_1\rangle = |x\rangle|1\rangle$ 
9: end if
10: Generate  $|\psi_2\rangle = |x, 1\rangle$ 
11: # 50:50 beam-splitters
12:  $|\psi_1\rangle = \hat{B}S(\frac{\pi}{4})|\psi_1\rangle$ 
13:  $\hat{L}_\theta = \hat{L}_1 \circ \hat{L}_2 \circ \dots \circ \hat{L}_L$ 
14: for  $l$  in  $\{1, \dots, L\}$  do
15:    $|\psi_0\rangle = \hat{L}_{\theta_l}|\psi_0\rangle$ 
16:    $|\psi_1\rangle = \hat{L}_{\theta_l}|\psi_1\rangle$ 
17:    $|\psi_2\rangle = \hat{L}_{\theta_l}|\psi_2\rangle$ 
18: end for
19:  $f(x) = 2^{-1}k_{x,1}^{-2}\langle\psi_0|\hat{q}|\psi_0\rangle$ 
20:  $g_{\pm}(x) = k_{x,1}^{-2j_{\pm}}\langle\psi_1|\hat{q}|\psi_1\rangle$ 
21:  $g_{1\gamma}(x) = k_{x,1}^{-2}\langle\psi_1|\hat{q}|\psi_1\rangle$ 
22:  $f'(x) = \pm 2[g_{\pm}(x) - f(x) - g_{1\gamma}(x)]/x$ 

```

$$g_{1\gamma}(x) = {}_i\langle x, 1 | \otimes^{n-1} \langle 0 | \hat{L}_\theta^\dagger \hat{q}_o \hat{L}_\theta | 0 \rangle \otimes^{n-1} | x, 1 \rangle_i. \quad (\text{D2})$$

The term in Eq. (D1d), instead, is already available due to the knowledge of f_θ . Summing up:

$$\pm 2 \frac{g_{io}^{(\pm)}(x) k_{x,1}^{-2j_{\pm}} - 2^{-1} k_{x,1}^{-2} f_\theta(x) - 2^{-1} k_{x,1}^{-2} g_{1\gamma}(x)}{x} \quad (\text{D3})$$

with $j_+ = 1$ and $j_- = 0$.

APPENDIX E: DERIVATIVES OF CLASSICAL FEED-FORWARD NETWORKS

In this section, we backtrack the process of differentiation of a classical feed-forward neural network. This algebraic procedure is mandatory to yield the pseudocodes 4 and 5. As already stated in Sec. VI B, classical feed-forward model can be modeled as a composition of multiple activation functions layer after layer, i.e.,

$$f_\theta(\mathbf{x}) = \sigma_{\theta_L}^{(L)} \circ \sigma_{\theta_{L-1}}^{(L-1)} \circ \dots \circ \sigma_{\theta_1}^{(1)}(\mathbf{x}), \quad (\text{E1})$$

where $\theta = \bigcup_{i=1}^L \theta_i$, and each layer reads as

$$\begin{aligned} \sigma_{\theta_l}^{(l)}(\mathbf{x}) &= \sigma(W\mathbf{x} + \mathbf{b}) \\ &= \left[\sigma\left(\sum_j w_{1j}x_j + b_1\right), \sigma\left(\sum_j w_{2j}x_j + b_2\right), \dots \right]. \end{aligned} \quad (\text{E2})$$

Here, $w_{ij}, b_i \in \theta_l \ \forall i, j$. To streamline the notation, we sum up the affine transformation into the brackets of σ as $z^{(l)}(x)$:

$$\sigma_{\theta_l}^{(l)}(\mathbf{x}) = \sigma(\mathbf{z}^{(l)}(\mathbf{x})) = [\sigma(z_1^{(l)}(\mathbf{x})), \sigma(z_2^{(l)}(\mathbf{x})), \dots]. \quad (\text{E3})$$

Here, the subscripts i to the z_i variables refer to those in the weight matrix and the bias vector in $\sum_j w_{ij}x_j + b_i$. Since the input from any l th layer is the output from the $l-1$ th one, we now introduce the $\mathbf{y}^{(l-1)} = \sigma(\mathbf{z}^{(l-1)}(\mathbf{y}^{(l-2)}))$ symbol to denote the input variable for $\sigma_{\theta_l}^{(l)}$.

1. First derivatives

If we derive the expression in Eq. (E3) with respect to x_i , the following result holds:

$$\begin{aligned} \frac{\partial y_m^{(L)}}{\partial x_i} &= \frac{\partial}{\partial x_i} \sigma(z_m^{(L)}) = \frac{\partial \sigma(z_m^{(L)})}{\partial z_m^{(L)}} \frac{\partial z_m^{(L)}(\mathbf{y}^{(L-1)})}{\partial x_i} \\ &= \sigma'(z_m^{(L)}) \sum_k W_{mk}^{(L)} \frac{\partial y_k^{(L-1)}}{\partial x_i} = \sum_k S_{mk}^{(L)} \frac{\partial y_k^{(L-1)}}{\partial x_i}. \end{aligned} \quad (\text{E4})$$

As we see, a recursive structure occurs, with $\mathbf{y}^{(1)} = \mathbf{x}$ (i.e., the inputs), so that $S_{ki}^{(0)} = \partial y_k^{(0)} / \partial x_i = \delta_{ik}$. The pseudocode 4 takes into account the first and the fourth members of Eqs. (E4). Moreover, from the same Equation we can retrieve the gradient of the function f_θ :

$$\nabla f_\theta = \prod_{l=L}^1 S^{(l)}, \quad S_{ij}^{(l)} = \sigma'(z_i^{(l)}) W_{ij}^{(l)}. \quad (\text{E5})$$

Each $S^{(l)}$ requires to compute the values of the model function f_θ up to the l th layer, then derive the outputs with respect to the last input variable $z_i^{(l)}$ and multiply them element-wise with the matrix $W_{ij}^{(l)}$.

2. Second derivatives

We now focus on the computation of the second derivatives. We have seen from Eq. (E4) that

$$\frac{\partial y_m^{(l)}}{\partial x_i} = \frac{\partial}{\partial x_i} \sigma(z_m^{(l)}) = \sigma'(z_m^{(l)}) \sum_k W_{mk}^{(l)} \frac{\partial y_k^{(l-1)}}{\partial x_i}. \quad (\text{E6})$$

We now perform a derivative of the second order, which means

$$\begin{aligned} \frac{\partial^2 y_m^{(l)}}{\partial x_i \partial x_j} &= \frac{\partial \sigma'(z_m^{(l)})}{\partial x_i} \sum_k W_{mk}^{(l)} \frac{\partial y_m^{(l-1)}}{\partial x_j} \\ &+ \sigma'(z_m^{(l)}) \sum_k W_{mk}^{(l)} \frac{\partial^2 y_m^{(l-1)}}{\partial x_i \partial x_j}. \end{aligned} \quad (\text{E7})$$

The first term on the second hand of the equation can thus be developed as follows:

$$\begin{aligned} \frac{\partial^2 y_m^{(l)}}{\partial x_i \partial x_j} &= \sigma''(z_m^{(l)}) \sum_{kn} W_{mn}^{(l)} W_{mk}^{(l)} \frac{\partial y_m^{(l-1)}}{\partial x_j} \frac{\partial y_m^{(l-1)}}{\partial x_i} \\ &+ \sigma'(z_m^{(l)}) \sum_k W_{mk}^{(l)} \frac{\partial^2 y_m^{(l-1)}}{\partial x_i \partial x_j}. \end{aligned} \quad (\text{E8})$$

The pseudocode 5 sums up this algebraic procedure.

-
- [1] S. Lloyd and S. L. Braunstein, Quantum computation over continuous variables, *Phys. Rev. Lett.* **82**, 1784 (1999).
 - [2] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, and S. Lloyd, Continuous-variable quantum neural networks, *Phys. Rev. Lett.* **1**, 033063 (2019).
 - [3] S. Corli, L. Moro, D. Dragoni, M. Dispenza, and E. Prati, Quantum machine learning algorithms for anomaly detection: A review, *Future Gener. Comput. Syst.* **166**, 107632 (2025).
 - [4] T. Hillmann, F. Quijandría, G. Johansson, A. Ferraro, S. Gasparinetti, and G. Ferrini, Universal gate set for continuous-variable quantum computation with microwave circuits, *Phys. Rev. Lett.* **125**, 160501 (2020).
 - [5] S. Bangar, L. Sunny, K. Yeter-Aydeniz, and G. Siopsis, Experimentally realizable continuous-variable quantum neural networks, *Phys. Rev. A* **108**, 042414 (2023).
 - [6] S. Takeda and A. Furusawa, Universal quantum computing with measurement-induced continuous-variable gate sequence in a loop-based architecture, *Phys. Rev. Lett.* **119**, 120504 (2017).
 - [7] K. Miyata, H. Ogawa, P. Marek, R. Filip, H. Yonezawa, J.-I. Yoshikawa, and A. Furusawa, Implementation of a quantum cubic gate by an adaptive non-Gaussian measurement, *Phys. Rev. A* **93**, 022301 (2016).
 - [8] E. Suerra, S. Altília, S. Olivares, A. Ferraro, F. Canella, D. Giannotti, G. Galzerano, S. Corli, E. Prati, and S. Cialdi, Generation and detection of squeezed states via a synchronously pumped optical parametric oscillator, *Appl. Phys. B* **132**, 26 (2026).
 - [9] S. Cialdi, E. Suerra, S. Olivares, S. Capra, and M. G. A. Paris, Squeezing phase diffusion, *Phys. Rev. Lett.* **124**, 163601 (2020).
 - [10] S. Cialdi, C. Porto, D. Cipriani, S. Olivares, and M. G. A. Paris, Full quantum state reconstruction of symmetric two-mode squeezed thermal states via spectral homodyne detection and a state-balancing detector, *Phys. Rev. A* **93**, 043805 (2016).
 - [11] S. Corli and E. Prati, Measurement-based quantum compiling via gauge invariance, *Phys. Rev. Appl.* **25**, 044068 (2026).
 - [12] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, General parameter-shift rules for quantum gradients, *Quantum* **6**, 677 (2022).
 - [13] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, *Phys. Rev. A* **99**, 032331 (2019).
 - [14] J. Bowles, D. Wierichs, and C.-Y. Park, Backpropagation scaling in parameterised quantum circuits, *Quantum* **9**, 1873 (2025).
 - [15] A. F. Izmaylov, R. A. Lang, and T.-C. Yen, Analytic gradients in variational quantum algorithms: Algebraic extensions of the parameter-shift rule to general unitary transformations, *Phys. Rev. A* **104**, 062443 (2021).
 - [16] M. Maronese, C. Destri, and E. Prati, Quantum activation functions for quantum neural networks, *Quantum Inf. Process.* **21**, 128 (2022).
 - [17] L. Banchi, N. Quesada, and J. M. Arrazola, Training Gaussian Boson sampling distributions, *Phys. Rev. A* **102**, 012417 (2020).
 - [18] G. de Felice and C. Corlett, Differentiation of linear optical circuits, *arXiv:2401.07997*.
 - [19] F. Hanamura, W. Asavanant, H. Nagayoshi, A. Sakaguchi, R. Ide, K. Fukui, P. van Loock, and A. Furusawa, Implementing arbitrary multimode continuous-variable quantum gates with fixed non-Gaussian states and adaptive linear optics, *Phys. Rev. A* **110**, 022614 (2024).
 - [20] S. Markidis, On physics-informed neural networks for quantum computers, *Front. Appl. Math. Stat.* **8**, 1036711 (2022).
 - [21] G. Panichi, S. Corli, and E. Prati, Quantum physics-informed neural networks for multivariable partial differential equations, *Phys. Rev. Appl.* **25**, 014001 (2026).
 - [22] M. Knudsen and C. B. Mendl, Solving differential equations via continuous-variable quantum computers, *arXiv:2012.12220*.
 - [23] M. Raissi, P. Perdikaris, and G. E. Karniadakis, Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations, *J. Comput. Phys.* **378**, 686 (2019).
 - [24] L. Brevi, A. Mandarino, and E. Prati, Addressing the non-perturbative regime of the quantum anharmonic oscillator by physics-informed neural networks, *New J. Phys.* **26**, 103015 (2024).
 - [25] L. Brevi, A. Mandarino, and E. Prati, A tutorial on the use of physics-informed neural networks to compute the spectrum of quantum systems, *Technologies* **12**, 174 (2024).
 - [26] A. E. Paine, V. E. Elfving, and O. Kyriienko, Physics-informed quantum machine learning: Solving nonlinear differential equations in latent spaces without costly grid evaluations, *arXiv:2308.01827*.
 - [27] A. Sedykh, M. Podapaka, A. Sagingalieva, K. Pinto, M. Pflitsch, and A. Melnikov, Hybrid quantum physics-informed neural networks for simulating computational fluid dynamics in complex shapes, *Mach. Learn.: Sci. Technol.* **5**, 025045 (2024).

- [28] C. Trahan, M. Loveland, and S. Dent, Quantum physics-informed neural networks, *Entropy* **26**, 649 (2024).
- [29] P. Siegl, S. Wassing, D.M. Mieth, S. Langer, and P. Bekemeyer, Solving transport equations on quantum computers—potential and limitations of physics-informed quantum circuits, *CEAS Aeronaut. J.* **16**, 63 (2025).
- [30] M.J. Powell, A direct search optimization method that models the objective and constraint functions by linear interpolation, in *Advances in Optimization and Numerical Analysis* (Springer, New York, 1994), pp. 51–67.
- [31] D. M. Olsson and L. S. Nelson, The Nelder-mead simplex procedure for function minimization, *Technometrics* **17**, 45 (1975).
- [32] A. Oosterwijk, H. A. Dijkstra, and T. van Leeuwen, An adjoint-free method to determine conditional nonlinear optimal perturbations, *Comput. Geosci.* **106**, 190 (2017).
- [33] S. Singer and J. Nelder, Nelder-mead algorithm, *Scholarpedia* **4**, 2928 (2009).
- [34] M. H. Wright, Nelder, mead, and the other simplex method, in *Optimization Stories* (European Mathematical Society-EMS-Publishing House GmbH, Berlin, Germany, 2012), pp. 271–276.
- [35] R. A. Khalek and V. Bertone, On the derivatives of feed-forward neural networks, [arXiv:2005.07039](https://arxiv.org/abs/2005.07039).
- [36] V. I. Avrutskiy, Enhancing function approximation abilities of neural networks by training derivatives, *IEEE Trans. Neural Networks Learn. Syst.* **32**, 916 (2020).
- [37] S. Rodini, Analytical derivatives of neural networks, *Comput. Phys. Commun.* **270**, 108169 (2022).
- [38] A. Zavatta, S. Viciani, and M. Bellini, Single-photon excitation of a coherent state: Catching the elementary step of stimulated light emission, *Phys. Rev. A* **72**, 023820 (2005).
- [39] F. A. Domínguez-Serna, F.J. Mendieta-Jimenez, and F. Rojas, Entangled photon-added coherent states, *Quantum Inf. Process.* **15**, 3121 (2016).
- [40] V. V. Dodonov, M. A. Marchiolli, Y. A. Korennoy, V. I. Man'ko, and Y. A. Moukhin, Dynamical squeezing of photon-added coherent states, *Phys. Rev. A* **58**, 4087 (1998).
- [41] G. S. Agarwal and K. Tara, Nonclassical properties of states generated by the excitations on a coherent state, *Phys. Rev. A* **43**, 492 (1991).
- [42] A. I. Lvovsky, H. Hansen, T. Aichele, O. Benson, J. Mlynek, and S. Schiller, Quantum state reconstruction of the single-photon Fock state, *Phys. Rev. Lett.* **87**, 050402 (2001).
- [43] C. K. Hong and L. Mandel, Experimental realization of a localized one-photon state, *Phys. Rev. Lett.* **56**, 58 (1986).
- [44] P. Grangier, G. Roger, and A. Aspect, Experimental evidence for a photon anticorrelation effect on a beam splitter: A new light on single-photon interferences, *Europhys. Lett.* **1**, 173 (1986).
- [45] A. Ourjoumtsev, R. Tualle-Brouiri, and P. Grangier, Quantum homodyne tomography of a two-photon Fock state, *Phys. Rev. Lett.* **96**, 213601 (2006).
- [46] B. J. Lawrie, P. D. Lett, A. M. Marino, and R. C. Pooser, Quantum sensing with squeezed light, *ACS Photonics* **6**, 1307 (2019).
- [47] Z. Y. Ou, J.-K. Rhee, and L. J. Wang, Observation of four-photon interference with a beam splitter by pulsed parametric down-conversion, *Phys. Rev. Lett.* **83**, 959 (1999).
- [48] D. Kalamidas, C. C. Gerry, and A. Benmoussa, Proposal for generating a two-photon added coherent state via down-conversion with a single crystal, *Phys. Lett. A* **372**, 1937 (2008).
- [49] J. Fadrný, M. Neset, M. Bielak, M. Ježek, J. Břlek, and J. Fiurášek, Experimental preparation of multiphoton-added coherent states of light, *npj Quantum Inf.* **10**, 89 (2024).
- [50] 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).
- [51] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin *et al.*, Tensorflow: Large-scale machine learning on heterogeneous systems (2015).