Journal of Physics A:
Mathematical and
Theoretical

PAPER • OPEN ACCESS

A novel approach to reduce derivative costs in variational quantum algorithms

To cite this article: G Minuto *et al* 2025 *J. Phys. A: Math. Theor.* **58** 185301

OPEN ACCESS
IOP Publishing
Journal of Physics A: Mathematical and Theoretical
J. Phys. A: Math. Theor. **58** (2025) 185301 (21pp)
https://doi.org/10.1088/1751-8121/adcf62

A novel approach to reduce derivative costs in variational quantum algorithms

G Minuto[1,2,*], D Melegari[2,3], S Caletti[4] and P Solinas[2,3]

[1] Dept. of Informatics, Bioengineering, Robotics, and Systems Engineering (DIBRIS), Polytechnic School of Genoa University, Genova, Italy
[2] INFN—Sezione di Genova, via Dodecaneso 33, I-16146 Genova, Italy
[3] Dipartimento di Fisica, Università di Genova, via Dodecaneso 33, I-16146 Genova, Italy
[4] Institute for Theoretical Physics, ETH, CH-8093 Zürich, Switzerland

E-mail: giovanni.minuto@uniroma1.it, dmelegari@infn.it, scaletti@phys.ethz.ch and paolo.solinas@unige.it

Received 18 September 2024; revised 14 April 2025
Accepted for publication 22 April 2025
Published 29 April 2025

Abstract

We present a detailed numerical study of an alternative approach, named quantum non-demolition measurement (QNDM) (Solinas *et al* 2023 *Eur. Phys. J. D* **77** 76), to efficiently estimate the gradients or the Hessians of a quantum observable. This is a key step and a resource-demanding task when we want to minimize the cost function associated with a quantum observable. In our detailed analysis, we account for all the resources needed to implement the QNDM approach with a fixed accuracy and compare them to the current state-of-the-art method (Mari *et al* 2021 *Phys. Rev. A* **103** 012405; Schuld *et al* 2019 *Phys. Rev. A* **99** 032331; Cerezo *et al* 2021 *Nat. Rev. Phys.* **3** 625). We find that the QNDM approach is more efficient, i.e. it needs fewer resources, in evaluating the derivatives of a cost function. These advantages are already clear in small dimensional systems and are likely to increase for practical implementations and more realistic situations. A significant outcome of our study is the implementation of the QNDM method in Python, provided in the supplementary material (Caletti and Minuto 2024 https://github.com/ simonecaletti/qndm-gradient). Given that most variational quantum algorithms (VQA) can be formulated within this framework, our results can have significant implications in

[*] Author to whom any correspondence should be addressed.

quantum optimization algorithms and make the QNDM approach a valuable alternative to implement VQA on near-term quantum computers.

Keywords: quantum computing, quantum variational algorithm, quantum chemistry, quantum machine learning, gradient

1. Introduction

Complex optimization problems, such as drug, molecular, and material design, are some of the research fields in which quantum computers are likely to have an impact in the mid-term timescale [1–3]. These problems share a common structure: given a cost function that represents our physical quantity of interest, we would like to find its minimum. The scheme to approach these problems in a quantum computer is a hybrid quantum–classical one, where variational quantum algorithms (VQA) [4] can be naturally implemented. The picture is the following: quantum computers are used to calculate the gradient of the cost function evaluated at a certain point in the parameter space. This can be done with different techniques but is usually done by evaluating the cost function in two points that are close in the parameter space and exploiting the parameter-shift rule to compute the derivative [5]. Then, this information is fed into a classical computer, which calculates how to adjust the parameters of the quantum circuit in order to reduce the cost function. By iterating this procedure, as in variational problems in physics, the quantum circuit should converge toward the configuration that generates one of the states minimizing the cost function (either a local or a global minimum).

The classical optimization procedures are well-known and established [6, 7] and they usually exploit information on the derivatives (gradient) of the cost function to update the parameters. At the quantum level, the way to extract the desired information from a quantum system is still debated and open [8, 9]. Indeed, the derivative cannot be directly associated with a Hermitian operator [10, 11] and, therefore, there is no unique way to measure it. Still, this is a crucial step since it usually causes a bottleneck in computation with quantum computers. The most straightforward and used approach to obtain the gradient of the cost function is to run a quantum circuit and measure a quantum observable at a certain point in the parameter space. Then, we repeat the procedure changing the circuit parameters by a little along each direction, separately [5, 12, 13]. We call this method, the direct measurements (DM) approach and, for every point in the parameter space, it requires two observable measurements to determine each component of the gradient.

In this article, we analyze a novel method to compute the derivatives of the cost function implemented on the quantum circuit. This is called the quantum non-demolition measurement (QNDM) approach and it was first theoretically described in reference [14]. The main idea is to extract the information about the derivative with a single measurement of a quantum observable. This is obtained by coupling a quantum *detector* to the original quantum system so that the information about the value of the derivative of the cost function is stored in the phase of the detector. Finally, the phase is measured to extract this information [14].

In the following, we present a detailed comparison between the two methods. We consider the evaluation of the cost function at a fixed accuracy, and we extract information about the repetitions of the circuit evaluation required. We do that by estimating the statistical error (mean squared error (MSE)) in the two approaches, and we calculate the total resource cost in both cases. This theoretical analysis is followed by full computational simulations, in which we compare the total resources needed in the two approaches to estimate the derivative of the cost function directly from their implementations.

We find that, already for cases with limited complexity, the QNDM approach has a substantial advantage over the DM ones. The resource-saving will increase for more practical and interesting problems of intermediate dimension. We observe that there are other methods that rely on measuring an additional detector, such as the approach proposed in [15]. However, the computational cost of obtaining the derivative with this method is comparable to that of the DM approach.

The paper is structured as follows. In section 2, we recall the basic ideas of VQA and the two approaches to derivative evaluation. Section 3 is devoted to the analysis of the statistical errors and the resource cost. Section 4.2 presents the numerical comparison, while section 6. contains the conclusions. The codes to calculate derivatives with QNDM used within this article are accessible via GitHub [16]. In appendix B we show how to install the QNDM package.

2. Theoretical discussion

2.1. General settings

We briefly recall the standard implementation of Variational quantum algorithms in a quantum computer (for more details see [5, 14, 17]). We consider a system of n qubits initialized in the state $|\psi_0\rangle = |0\rangle_1 \otimes |0\rangle_2 \otimes \ldots \otimes |0\rangle_n \equiv |00\ldots0\rangle$. The generic unitary transformation acting on them is denoted $U(\vec{\theta})$, where $\vec{\theta}$ is a vector in the parameter space. The quantum state obtained by applying the transformation $U(\vec{\theta})$ to the initial state of the circuit is given by $|\psi(\vec{\theta})\rangle = U(\vec{\theta})|\psi_0\rangle$. The $U(\vec{\theta})$ transformation can be implemented as a sequence of *parameterized* unitary transformations $U_j(\theta_j)$ and arbitrary transformations (independents of $\vec{\theta}$) V_j, $j \in [1, m]$, such that

$$U\left(\vec{\theta}\right) = V_m U_m\left(\theta_m\right) \ldots . V_2 U_2\left(\theta_2\right) V_1 U_1\left(\theta_1\right). \tag{1}$$

We consider $U_j(\theta_j) = \exp\{-\mathrm{i}\theta_j \hat{H}_j\}$ and $\hat{H}_j$ is the corresponding generator of the transformation. The quantum circuit to generate this transformation is represented in figure 1(a). Assuming that $\hat{H}_j^2 = \mathbb{1}$, we have that

$$U_j\left(\theta_j\right) = \mathrm{e}^{-\mathrm{i}\hat{H}_j\theta_j/2} = \cos\frac{\theta_j}{2}\mathbb{1} - \mathrm{i}\sin\frac{\theta_j}{2}\hat{H}_j. \tag{2}$$

This accounts for a multitude of relevant cases such as when $\hat{H}_j$ is a tensor product of any multi-qubits Pauli matrices [5]. However, to further simplify the discussion and the simulation, in the following, we restrict our attention to the case (presented in figure 1) in which

$$U_j\left(\vec{\theta}_j\right) = R_1^j\left(\theta_1^j\right)\ldots R_n^j\left(\theta_n^j\right), \tag{3}$$

where each operator $R_i^j(\theta_i^j)$ is a parameterized single qubit gate (see equation (2)) and $\hat{H}_j$ is a single Pauli matrix. Likewise, for the V_j part of the transformation, which is independent of the parameters, we select the following specific structure

$$V_j = C_1\mathrm{NOT}_2 \ldots C_{n-1}\mathrm{NOT}_n, \tag{4}$$

where $C_{i-1}\mathrm{NOT}_i$ is a two-qubit operator with the control on the $i-1$th qubit and the action on the ith qubit [18]. The sequence of $U_j(\vec{\theta}_j)$ and V_j is usually called *layer* and, as shown in figure 1, is a combination a parametrized and entangling unitary transformations.

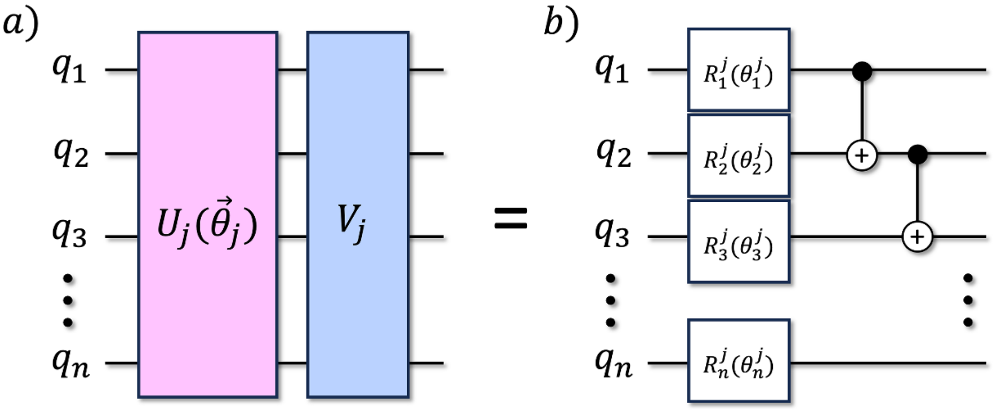

Figure 1. (a) Structure for the jth layer as a sequence of a parameterized layer $U_j(\vec\theta_j)$ and an unparameterized layer V_j for a n qubit system. (b) Example of a layer in terms of single-qubit gates depending on the θ_i^j parameters and two qubits gates $C_{i-1}\mathrm{NOT}_i$.

In optimization problems, the goal is to find the quantum state $|\psi(\vec\theta)\rangle$ which minimizes the expectation value of a quantum observable $\hat M$. To accomplish this task, we define a cost function f that we want to minimize with respect to the parameters $\vec\theta$

$$f\left(\vec\theta\right) = \langle\psi\left(\vec\theta\right)|\hat M|\psi\left(\vec\theta\right)\rangle = \langle 0|U^\dagger\left(\vec\theta\right)\hat M U\left(\vec\theta\right)|0\rangle\,. \tag{5}$$

In many applications [19, 20], the observable $\hat M$ can be written as a weighted sum of Pauli strings $\hat P_i = \prod_j \hat A_i^j$, where $\hat A_j \in [X_j, Y_j, Z_j, I_j]$ is one of the usual Pauli operators acting on the jth qubit. With this notation, we have that

$$\hat M = \sum_{i=1}^{J} h_i \hat P_i \tag{6}$$

where J is the total number of Pauli string composing $\hat M$.

We adopt the so-called hybrid quantum–classical optimization procedure described in the introduction [4, 13]. The minimization algorithm consists of two steps that must be iterated until the minimum of $f(\vec\theta)$ is reached. First, with the help of a quantum computer, we evaluate the gradient of $f(\vec\theta)$ in a given point of the parameter space $\vec\theta$. Then, the information about the gradient is fed into a classical computer which, with standard optimization algorithms [21–23], evaluates in which direction of the $\vec\theta$ space we move to reach the minimum of f. In this paper, we focus on the quantum part of the algorithm, that is the calculation of the gradient of $f(\vec\theta)$ done with a quantum computer and we do not discuss the classical one.

To measure the derivative of $f(\vec\theta)$ along the lth direction in the parameter space $\vec\theta$, we measure $f(\vec\theta + s\hat e_l)$ and $f(\vec\theta - s\hat e_l)$ where $\hat e_l$ is the unit vector along the θ_l direction and s is a shift parameter. Then, we calculate the quantity [5]

$$g_l = \frac{\partial f\left(\vec\theta\right)}{\partial\theta_l} = \frac{f\left(\vec\theta + s\hat e_l\right) - f\left(\vec\theta - s\hat e_l\right)}{2\sin s}\,. \tag{7}$$

Notice that g_l, unlike finite difference methods, is not an approximation but the exact derivative of $f(\vec\theta)$. By repeating this procedure for all directions θ_l we obtain the full gradient of

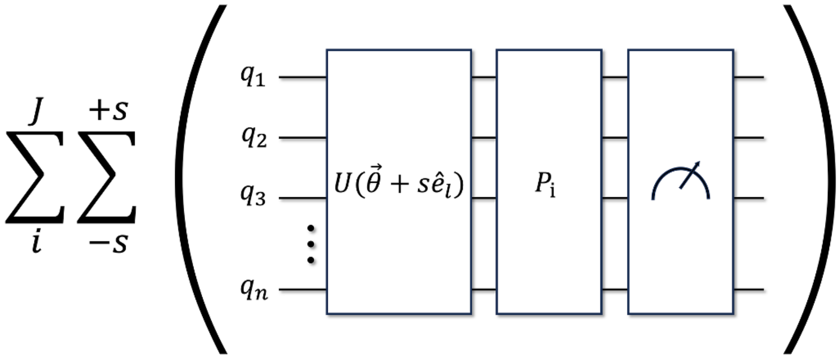

Figure 2. Pictorial representation for calculation of derivative of the function $f(\vec{\theta}) = \langle 0|U^\dagger(\vec{\theta})\hat{M}U(\vec{\theta})|0\rangle$ with $\hat{M} = \sum_{i=1}^J h_i\hat{P}_i$ using the DM protocol, as described in equation (8). The n qubits circuit is evolved under the parametric transformation $U(\vec{\theta}+s\hat{e}_l)$, defined similarly to equation (1), followed by a base rotation P_i, which represents the ith tensor product of Pauli matrices. A global measurement is then performed on the register, yielding a real-valued output. The first summation indicates that the circuit must be executed for each Pauli string ($i = 1,\dots,J$) in the observable $\hat{M}$. The second summation accounts for the execution of the circuit with different parameter shifts $\pm s$, as shown in equation (8).

$f(\vec{\theta})$, which is then fed into the classical optimization algorithm. Setting $s = \pi/2$ yields the parameter shift rule described in [12, 24–26].

2.2. DM

The standard approach to measure the value of the derivatives of the cost function (7) is the DM method [12]. This method consists of measuring separately the average values of the operator $\hat{M}$ in the two points, i.e. $\vec{\theta}\pm s\hat{e}_l$, and then calculating the derivative as in equation (7). To calculate the values of $f(\vec{\theta}+s\hat{e}_l)$, we run the quantum circuit to implement the corresponding $U(\vec{\theta}+s\hat{e}_l)$ transformation, and then we perform a projective measurement *for each Pauli string* $\hat{P}_i$ that appears in the definition (6) of the observable $\hat{M}$ [5, 19]. For every Pauli string $\hat{P}_i$ in $\hat{M}$, we have to iterate these steps until the statistical accuracy is reached. The same procedure is then implemented for $f(\vec{\theta}-s\hat{e}_l)$ and then, the derivative can be calculated as in equation (7).

Following this procedure, a single derivative g_l in the DM approach can be written as

$$g_l = \sum_i \frac{h_i}{2\sin s}\left(\mathrm{Tr}_S\left[\hat{P}_i U^\dagger\left(\vec{\theta}+s\hat{e}_l\right)\rho_s^0 U\left(\vec{\theta}+s\hat{e}_l\right)\right]\right.$$
$$\left. -\mathrm{Tr}_S\left[\hat{P}_i U^\dagger\left(\vec{\theta}-s\hat{e}_l\right)\rho_s^0 U\left(\vec{\theta}-s\hat{e}_l\right)\right]\right), \tag{8}$$

where $\rho_s^0 = |\psi_0\rangle\langle\psi_0|$, $U(\vec{\theta})$ is the operator in equation (1) and Tr_S denotes the trace over all the qubits. An implementation in terms of quantum circuits is shown in figure 2.

2.3. Quantum non-demolition measurement

In this section, we briefly recall the main characteristics of the QNDM method [14]. The key idea is to store the information about the derivative of the cost function in the phase of an ancillary qubit named the *detector*. More specifically, by coupling the original system and the detector twice, it is possible to store the information about both $f(\vec{\theta}+s\hat{e}_l)$ and $f(\vec{\theta}-s\hat{e}_l)$ in the phase of the detector. Then, the detector is measured to extract information about the derivative [14].

The advantage of this approach lies in the fact that the couplings between the system and the detector are sequential and the measure on the detector is performed only at the end of the procedure. Therefore, even if the quantum circuit must be iterated to obtain the desired statistical accuracy, we extract the derivative information by performing just a single measurement on the detector qubit instead of the two needed with the DM approach. The details of the QNDM algorithm can be found in [14]. Here, we summarize only the main steps.

Suppose we are interested in the quantum observable $\hat{M}$ in equation (6). We add an ancillary qubit and implement the system-detector coupling through the operator $U_\pm = \exp\{\pm \mathrm{i}\lambda Z_\mathrm{a}\otimes \hat{M}\}$ where Z_a is a Pauli operator acting on the ancillary detector and λ is the system-detector coupling constant. We notice that if $\hat{M}$ is an operator involving high complexity, the implementation of operator $U_\pm$ is, in general, very resource-consuming. This might seem a critical drawback, but the QNDM approach offers a clear and elegant way to bypass this problem, instead of implementing the operator $U_\pm = \exp\{\pm\mathrm{i}\lambda Z_\mathrm{a}\otimes\sum_{i=1}^{J} h_i\hat{P}_i\}$, we can implement the product of single Pauli string operators $\prod_i^J \exp\{\pm\mathrm{i}\lambda Z_\mathrm{a}\otimes h_i\hat{P}_i\}$ without relying on the Trotterization approximation [27], as demonstrated in [14]. Considering the described quantum circuit, this corresponds to the following transformation (in the full system and detector Hilbert space)

$$U_\mathrm{tot} = \mathrm{e}^{\mathrm{i}\lambda Z_\mathrm{a}\otimes\hat{M}} U\left(\vec{\theta}+s\hat{e}_l\right) U^\dagger\left(\vec{\theta}-s\hat{e}_l\right) \mathrm{e}^{-\mathrm{i}\lambda Z_\mathrm{a}\otimes\hat{M}} U\left(\vec{\theta}-s\hat{e}_l\right). \tag{9}$$

The initial state for the system plus the detector $|\psi_0\rangle\left(|0\rangle_\mathrm{D}+|1\rangle_\mathrm{D}\right)/\sqrt{2}$, where a Hadamard gate is applied to the initial state of the detector. After implementing the transformation (9), we measure the accumulated phase $\exp\{\mathrm{i}\phi(\lambda)\}$ on the detector degrees of freedom. It can be shown [10, 11, 14, 28, 29] that this is a quasi-characteristic function $\mathcal{G}_\lambda$

$$\mathcal{G}_\lambda \equiv \mathrm{e}^{\mathrm{i}\phi(\lambda)} = \frac{{}_\mathrm{D}\langle 0|\rho_\mathrm{D}{}^f|1\rangle_\mathrm{D}}{{}_\mathrm{D}\langle 0|\rho_\mathrm{D}^0|1\rangle_\mathrm{D}}, \tag{10}$$

where $|0\rangle_\mathrm{D}$ and $|1\rangle_\mathrm{D}$ are eigenstates of the detector operator Z_a, while ρ_D^0 and ρ_D^f are the density matrices of the detector before and after the application of U_tot, respectively.

Taking the derivatives of $\mathcal{G}_\lambda$ with respect to λ and evaluating them in $\lambda=0$, we have access to the information of the different moments of the distribution describing the variation of the cost function [10, 11, 14, 28, 29]. Here, we are interested only in the first moment which directly gives us g_l. Following reference [14], it can be shown that the first derivative of $\mathcal{G}_\lambda$ reads

$$\begin{aligned}
-\mathrm{i}\partial_\lambda\mathcal{G}_\lambda\Big|_{\lambda=0} &= 2\mathrm{Tr}_S\left[U^\dagger\left(\vec{\theta}+s\hat{e}_l\right)\hat{M}U\left(\vec{\theta}+s\hat{e}_l\right)\rho_S^0 - U^\dagger\left(\vec{\theta}-s\hat{e}_l\right)\hat{M}U\left(\vec{\theta}-s\hat{e}_l\right)\rho_S^0\right] \\
&= 2\sum_i h_i\mathrm{Tr}_S\left[U^\dagger\left(\vec{\theta}+s\hat{e}_l\right)\hat{P}_iU\left(\vec{\theta}+s\hat{e}_l\right)\rho_S^0\right. \\
&\quad\left.-U^\dagger\left(\vec{\theta}-s\hat{e}_l\right)\hat{P}_iU\left(\vec{\theta}-s\hat{e}_l\right)\rho_S^0\right],
\end{aligned} \tag{11}$$

where Tr_S denotes the trace over the *system* degrees of freedom. By direct comparison, equation (11) is proportional to equation (8),

$$-\mathrm{i}\partial_\lambda\mathcal{G}_\lambda\Big|_{\lambda=0} = 2\sin(s)\,g_l. \tag{12}$$

Thus, both the QNDM and the DM approach can be used to compute the derivative of $f(\vec{\theta})$.

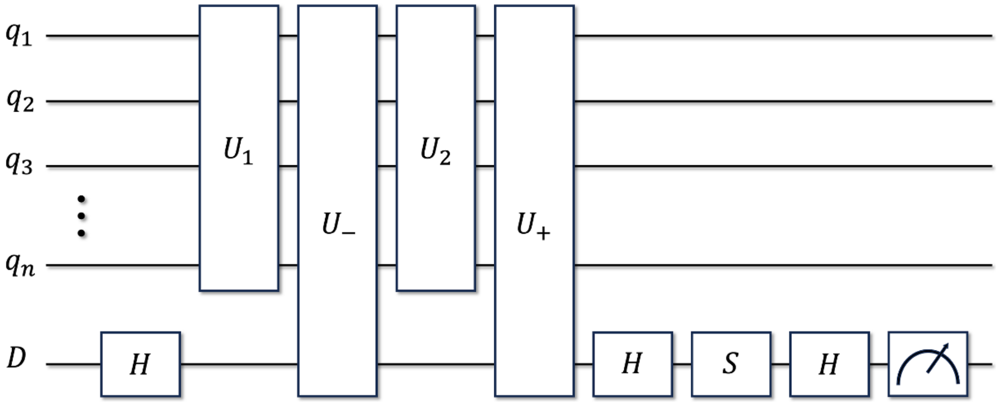

Figure 3. Pictorial representation of the quantum circuit implementing the QNDM protocol. Here, H is the Hadamard gate, S is the phase gate, $U_1 = U(\vec{\theta} - s\hat{e}_l)$, $U_2 = U^\dagger(\vec{\theta} - s\hat{e}_l)U(\vec{\theta} + s\hat{e}_l)$ and $U_\pm = \exp\{\pm i\lambda\hat{Z}_\mathrm{a} \otimes \hat{M}\}$ is the system-detector coupling operator.

We note that it is possible to approximate, at the linear regime, the derivative of the quasi-characteristic function to the argument of the accumulated detector phase $\phi(\lambda)$ [14]. In practice, this approximation holds if the coefficient in the argument of the system-detector coupling operator, $U_\pm = \exp\{\pm i\lambda\hat{Z}_\mathrm{a} \otimes \hat{M}\}$, is small. This condition is satisfied for sufficiently small λ, allowing the derivative to be expressed as

$$-i\partial_\lambda \mathcal{G}_\lambda|_{\lambda=0} = -i\partial_\lambda e^{i\phi(\lambda)}|_{\lambda=0} = \partial_\lambda \phi(\lambda)|_{\lambda=0} \approx \frac{\phi(\lambda) - \phi(0)}{\lambda} = \frac{\phi(\lambda)}{\lambda} \tag{13}$$

where in the last steps we have used the fact that, for symmetry reason, $\phi(0) = 0$ (see, for example, reference [14]).

Then, the phase accumulated by the detector during the interaction with the system might be measured with interferometric techniques. The other logical operations we need are the Hadamard gate H and a phase gate S defined by [18]

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{14}$$

$$S = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\frac{\pi}{2}} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix} \tag{15}$$

In terms of circuit implementation, the interferometric measure is built by applying a Hadamard gate, a phase gate, and another Hadamard to the detector and to measuring the populations of the two possible outcomes $P_{0,1}$ (see figure 3).

It can be shown (see reference [14]) that, for small λ, $\phi(\lambda) = -\arcsin(2P_0 - 1)$. Furthermore, using equation (13) the derivative of the cost function $f(\vec{\theta})$ reads

$$g_l \sim -\frac{\arcsin(2P_0 - 1)}{2\lambda \sin s}, \tag{16}$$

where we have added the term $2\sin s$ to the denominator to directly get the value of the derivatives according to the parameter shift rule.

3. Mean squared error and cost analysis

Understanding how statistical errors affect the estimate of the derivatives and gradient plays a pivotal role in judging the performance and reliability of the different methods presented above. As a measure of the quality of the estimation of the cost function derivative, we consider MSE, i.e. the average of the difference between the estimated values $\hat{g}_l$ and the exact value g_l, squared. We introduce the bias and variance as [30]

$$\mathrm{Bias}\left(\hat{g}_l\right) = \mathbb{E}\left(\hat{g}_l\right) - g_l\,, \tag{17}$$

$$\mathrm{Var}\left(\hat{g}_l\right) = \mathbb{E}\left[\left(\hat{g}_l - \mathbb{E}\left(\hat{g}_l\right)\right)^2\right]\,, \tag{18}$$

where $\mathbb{E}(\cdot)$ is the expectation over the statistical distribution of the measurement outcomes. The bias represents a systematic distortion of a statistical result, and it remains present even in the limit of an infinite number of shots N. This is a consequence of the linear approximation used in equation (13). On the other hand, variance quantifies the degree of dispersion, indicating how far a collection of values deviates from its mean value. Unlike the bias, the variance depends on the shot number N [5]. In terms of these quantities, we define the MSE as

$$\mathrm{MSE}\left(\hat{g}_l\right) = \mathrm{Bias}\left(\hat{g}_l\right)^2 + \mathrm{Var}\left(\hat{g}_l\right)\,. \tag{19}$$

We start our discussion by computing the bias and variance for the QNDM method, following the methodology outlined in reference [5]. They read

$$\mathrm{Bias}_{\mathrm{QNDM}}\left(\hat{g}_l\right) = \left(\frac{\phi\left(\lambda\right)}{\lambda} - \mathrm{i}\partial_\lambda \mathcal{G}_\lambda\right) = \frac{\lambda \partial_\lambda^2 \mathcal{G}_\lambda}{2} + \mathcal{O}\left(\lambda^2\right)\,, \tag{20}$$

$$\mathrm{Var}_{\mathrm{QNDM}}\left(\hat{g}_l\right) = \sigma^2_{\mathrm{QNDM}}\,. \tag{21}$$

We observe that the QNDM approach, as expected, exhibits bias different from zero arising from the linear approximation employed in equation (13) to obtain the derivative of the quasi-characteristic function $\mathcal{G}_\lambda$. Therefore, its value is directly proportional to the second derivative of the function $\mathcal{G}_\lambda$ multiplied by λ. To compute the variance σ^2_{QNDM}, we must incorporate the statistical error $\sigma^2_{P_0}$ associated with the detector population P_0 that we measure. For N_{QNDM} single qubit measurements of the detector, we obtain $\sigma^2_{P_0} = \sigma^2_{\mathrm{D}}/N_{\mathrm{QNDM}}$ where σ_{D} is the single shot variance obtained measuring the detector. Following the standard error propagation formulas [31] for equation (16), we obtain σ^2_{QNDM} from the variance for the detector population, which is given by

$$\sigma^2_{\mathrm{QNDM}} = \frac{\sigma^2_{P_0}}{4\sin^2 s}\frac{1}{\lambda^2\left(1-\left(2P_0-1\right)^2\right)} = \frac{\sigma^2_{\mathrm{D}}}{4N_{\mathrm{QNDM}}\sin^2 s}\frac{1}{\lambda^2\left(1-\left(2P_0-1\right)^2\right)}\,. \tag{22}$$

Thus, the corresponding MSE is

$$\mathrm{MSE}_{\mathrm{QNDM}}\left(\hat{g}_l\right) = \frac{\lambda^2\left(\partial_\lambda^2 \mathcal{G}_\lambda\right)^2}{4} + \frac{\sigma^2_{\mathrm{D}}}{4N_{\mathrm{QNDM}}\sin^2 s}\frac{1}{\lambda^2\left(1-\left(2P_0-1\right)^2\right)}\,. \tag{23}$$

Notably, in $\mathrm{MSE}_{\mathrm{QNDM}}$, the variance term is directly proportional to λ^{-2} while the bias term is proportional to λ^2. Thus, the optimal reduction of the MSE hinges on selecting an appropriate value for λ.

As highlighted in the previous section, the derivative of the quasi-characteristic can be approximately calculated if the coefficient in the argument of the system-detector coupling operator, $U_\pm = \exp\{\pm \mathrm{i}\lambda \hat{Z}_\mathrm{a} \otimes \hat{M}\}$, is small. Consequently, the appropriate choice of λ depends, also, from the coefficients h_i of observable $\hat{M} = \sum_{i=1}^{J} h_i \hat{P}_i$. In the computational analysis, section 4, we present a significant example where a suitable value of λ is selected.

In the DM approach, we need to measure the system qubits twice to obtain $f(\vec{\theta} + s\hat{e}_l)$ and $f(\vec{\theta} - s\hat{e}_l)$. Proceeding as above, we have

$$\mathrm{Bias}_{\mathrm{DM}}\left(\hat{g}_l\right) = 0 \tag{24}$$

$$\mathrm{Var}_{\mathrm{DM}}\left(\hat{g}_l\right) = \frac{\sigma^2_{\mathrm{DM}}\left(\vec{\theta} + s\hat{e}_l\right) + \sigma^2_{DM}\left(\vec{\theta} - s\hat{e}_l\right)}{4\sin^2 s}. \tag{25}$$

The bias contribution for DM vanishes because the parameter shift rule calculates the exact value of derivative [5]. The variance, instead, is composed by the terms $\sigma^2_{\mathrm{DM}}(\vec{\theta} \pm s\hat{e}_l)$. Intuitively, they correspond to the evaluation of $f(\vec{\theta} \pm s\hat{e}_l)$. Following reference [5], we assume that the variance of the measured observable depends weakly on the parameter shift such that $\sigma^2_{\mathrm{DM}}(\vec{\theta} + s\hat{e}_l) + \sigma^2_{\mathrm{DM}}(\vec{\theta} - s\hat{e}_l) \sim 2\sigma^2_{\mathrm{DM}}$ for any value of s. As we have seen in section 2.2, to obtain the derivatives, we have to measure each Pauli string individually in the parameter space point $\vec{\theta} \pm s\hat{e}_l$ (see equation (8) and [5, 13]). Then, the statistical error formula for the estimation of $f(\vec{\theta})$ is given by

$$\sigma^2_{\mathrm{DM}} = \sum_{i=1}^{J} \frac{h_i^2 \sigma_s^2}{N_{\mathrm{DM}}} \tag{26}$$

where h_i are the parameters in $\hat{M}$ in equation (6) and σ_s is the single shot variance obtained measuring the n qubits [5]. In equation (26), as in reference [5, 17], we assume that σ_s is equal for each Pauli string measurement. Following these observations, we conclude that the MSE of DM approach is only due to the statistical noise and it is equal to

$$\mathrm{MSE}_{\mathrm{DM}}\left(\hat{g}_l\right) = \sum_{i=1}^{J} \frac{h_i^2 \sigma_\mathrm{s}^2}{2N_{\mathrm{DM}} \sin^2 s}. \tag{27}$$

A remark must be made regarding the results obtained in equations (23) and (27). As demonstrated in section 4, an appropriate choice of λ could provide a constant advantage in terms of error, giving to QNDM an advantage in terms of efficiency cost.

$$\frac{\mathrm{MSE}_{\mathrm{DM}}\left(\hat{g}_l\right)}{\mathrm{MSE}_{\mathrm{QNDM}}\left(\hat{g}_l\right)} > 1. \tag{28}$$

3.1. Cost analysis and comparison between DM and QNDM approaches

To make a comparison between the two approaches, we use the cost $\mathcal{C}\left(g_l^i\right)$ ($i = \mathrm{QNDM}, \mathrm{DM}$) that gives the scaling in terms of the total number of gates required to compute a derivative in the cost function for the two algorithms represented in figures 2 and 3. In addition, we use the resource ratio $\mathcal{C}(g_l^{\mathrm{DM}}/g_l^{\mathrm{QNDM}}) = \mathcal{C}(g_l^{\mathrm{DM}})/\mathcal{C}(g_l^{\mathrm{QNDM}})$.

The resource costs express the number of gates, both single-qubit and two-qubit gates, used in one of two approaches. They are functions of the number of Pauli strings J, the number of qubits n, and the number of logical operators k used in the unitary transformation $U(\theta)$, equation (1). As in the previous section, N_{DM} and N_{QNDM} represent the number of shots or repetitions for the two approaches.

For DM approach, described in section 2.2, a circuit with n qubits is transformed by a unitary operator $U(\theta - se_l)$, and P_i is measured N_{DM} times. This process is repeated for each of

Table 1. Resource cost function to compute first-order derivatives with the DM and the QNDM approaches. As can be seen, the QNDM has a clear cost advantage. However is import to emphasize that in the QNDM approach we run a more deeper quantum circuit, figure 3, than DM, figure 2.

Method	Shots	Costs
DM	N_{DM}	$\mathcal{C}\left(g_l^{\mathrm{DM}}\right) = 2N_{\mathrm{DM}}J(k+n)$
QNDM	N_{QNDM}	$\mathcal{C}(g_l^{\mathrm{QNDM}}) = N_{\mathrm{QNDM}}(3k+8Jn)$

the J Pauli strings in $\hat{M}$ to calculate the expectation value $\langle M(\theta - se_l)\rangle$. The same procedure is then repeated with the unitary operator $U(\theta + se_l)$ to obtain $\langle M(\theta + se_l)\rangle$. Summing all the contributions we get a computational cost equal to $2N_{\mathrm{DM}}J(k+n)$. For QNDM approach, described in section 2.3, a circuit with $n+1$ qubits is initially transformed by the operator $U_1 = U(\theta - se_l)$, followed by the application of the exponential operator $U_- = \exp\{-\mathrm{i}\lambda Z_{\mathrm{a}} \otimes \hat{M}\}$. Next, the operator $U_2 = U^{\dagger}(\theta - se_l)U(\theta - se_l)$ is applied. Finally, the second exponential operator $U_+ = \exp\{+\mathrm{i}\lambda Z_{\mathrm{a}} \otimes \hat{M}\}$ is applied, and the detector phase is measured. This procedure is repeated N_{QNDM} times. The resulting computational cost is $N_{\mathrm{QNDM}}(3k+8Jn)$. The computed values of the resource costs are summarized in table 1.

In this analysis, we focus on two regimes. The first occurs when $k \gg nJ$, meaning that the number of logical gates required to implement $U(\vec{\theta})$ exceeds the number of Pauli strings J (multiplied by the number of qubits). From table 1 and reference [14], we have that the resource ratio scales as

$$\mathcal{C}\left(\frac{g_l^{\mathrm{DM}}}{g_l^{\mathrm{QNDM}}}\right) = \frac{2J}{3k}\frac{N_{\mathrm{DM}}}{N_{\mathrm{QNDM}}} = \mathcal{O}\left(J\right). \tag{29}$$

This result indicates that the cost of computing a derivative using DM is J times higher than QNDM , leading to a linear speedup in this regime as the number of Pauli strings increases. This regime is particularly relevant for quantum chemistry simulations. For instance, in simulations of moderately complex molecules [32], typical values are $k \approx 10^9 - 10^{10}$ operations for implementing $U(\vec{\theta})$, $n \approx 10^2 - 10^3$ (corresponding to the expected size of near-term quantum processors), and $J > 10^3$ as the number of Pauli strings in the Hamiltonian. Given these parameters, the condition $k \gg nJ$ holds, resulting in a reduction of approximately J in resource usage when employing the QNDM approach. This advantage is expected to grow as simulations scale to larger and more complex molecules.

This regime also holds significant potential for simulations in quantum machine learning since the expressivity, i.e. the ability of a quantum circuit to generate (pure) states that are well representative of the Hilbert space [33] of a quantum circuit is directly linked to the number of parametrized rotation in $U(\vec{\theta})$, i.e. k.

The opposite regime is reached when $k \ll nJ$, i.e. when the $\hat{M}$ operator is composed of many Pauli strings (thus, its averages might be difficult to estimate or measure) and the logical space to be searched is reduced (thus, we need unitary transformation with a limited number of logical gates). In this case, the ratio between the resources employed reads

$$\mathcal{C}\left(\frac{g_l^{\mathrm{DM}}}{g_l^{\mathrm{QNDM}}}\right) = \frac{k}{4n}\frac{N_{\mathrm{DM}}}{N_{\mathrm{QNDM}}} = \mathcal{O}\left(k\right). \tag{30}$$

Whereas the k factor comes from the linear cost advantage of QNDM with respect to DM for this regime (see table 2). This scenario is promising for various applications in VQAs as pointed out in [34].

4. Computational analysis

4.1. Errors analysis

In this section, we present a numerical comparison of error estimates obtained using the two methods. To gain a comprehensive understanding of the errors, we conducted multiple tests across different configurations. We have taken the number of qubits $n = 10$ and the Pauli strings in $\hat{M}$ are randomly selected from the set of all possible Pauli string operators. The coefficients h_i in equation (6) are drawn from a Gaussian distribution centered at 0 with different values of standard deviations. In all simulations, the results remain consistent with those obtained for a standard deviation of 5, which we report here. The analysis is performed as a function of the number of Pauli strings J. As discussed in section 3, to approximate the derivative of the $\mathcal{G}_\lambda$, the value of λ must be taken small enough to justify the linear approximation in equation (13). As a consequence, the value of λ depends (in a complex way) on the coefficients of the observable $\hat{M}$. For this case, we set a value of λ equal to the square root of the inverse of the sum of the absolute values of all coefficients h_i, $\lambda = 1/\sqrt{\sum_i |h_i|}$. For the numerical analysis, we have used the simulator provided by IBM Quantum known as Aer [35]. All the simulations are run to a shot count of 500, a standard value for optimization.

The value of the derivatives g_l depends on many parameters:

(j) the logical gates needed to implement $U(\vec{\theta})$, e.g. see equation (9);
(jj) their total number k;
(jjj) the direction l along which is calculated the derivative;
(jv) the Pauli strings $\hat{P}_i$ that appear in $\hat{M}$;
(v) their number J, as pointed out in equation (6).

For this reason, to ensure statistical significance, we need to average over these parameters. In the following, we use the term 'realization' to denote a random choice of all the *relevant* parameters and possibilities described in the points from (*j*) to (*v*).

In figure 4, we plot the derivative average $\mu(g_l^i(J))$ on the vertical axis, along with the average mean squared error $\mu(\mathrm{MSE}^i(J))$. For each point, we calculate $L = 100$ derivatives $g_l^i(J)$, varying the above parameters (*j*) to (*v*), and calculate their corresponding $\mathrm{MSE}(g_l^i(J))$. Finally, we average all the derivatives and their corresponding MSEs.

$$\mu\left(g_l^i(J)\right) = \frac{1}{L}\sum_{\Omega}^{L=100} g_l^i(J), \quad \mu\left(\mathrm{MSE}^i(J)\right) = \frac{1}{L}\sum_{\Omega}^{L=100} \mathrm{MSE}\left(g_l^i(J)\right), \tag{31}$$

where the sum over Ω is intended over all the indices $(j), (jj), (jjj), (jv), (v)$. The horizontal axis represents the number of Pauli strings J. The DM and QNDM results correspond to the red and blue markers, respectively.

We emphasize that, while the average of the gradient $\mu(g_l^i(J))$ does not carry direct information about the bias or variance of the method, the mean squared error $\mu(\mathrm{MSE}^i(J))$ is the relevant quantity to assess the performance of the estimators. As shown in the figure 4, $\mu(g_l^i(J))$ is close to zero, reflecting the fact that it averages derivatives from random realizations, in contrast, the MSE reveals a consistent difference between the two methods, in favor of QNDM.

Notably, the ratio between the errors of the DM and QNDM approaches remains approximately constant as the number of Pauli strings increases, but the absolute MSE grows with J for both methods. This highlights that the QNDM approach, under fixed shot count, yields increasingly better relative precision compared to the DM method. In practical terms, this means that

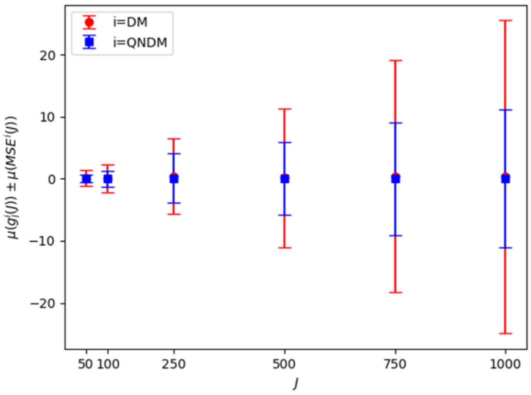

Figure 4. The plot represents the average derivative $\mu(g_l^i(J))$ along with the average mean squared error $\mu(\mathrm{MSE}^i(J))$ [i =DM (red markers) and i =QNDM (blue markers)]. Each point is a function of the number of Pauli strings J. The results correspond to a shot count of 500. The simulated quantum system consists of $n = 10$ qubits. Each point is averaged over $L = 100$ realizations, as discussed in the main text.

to reach the same level of precision as QNDM, the DM method would require a higher number of shots.

4.2. Costs comparison

For practical situations, it is of paramount importance to quantitatively estimate any advantage of the QNDM algorithm over DM and in which conditions this is reached. From the discussion in section 3.1, we expect the QNDM approach to perform better, i.e. it needs fewer resources, in some specific but interesting regimes. See for example equations (29) and (30). Aiming in this direction, we present a full numerical analysis of the resources needed to calculate the derivatives of the cost function. For a meaningful comparison, we fix the number of shots for the QNDM approach is set to $N_{\mathrm{QNDM}} = 500$, while for the DM approach, the number of shots is adjusted for each instance to ensure $MSE_{\mathrm{QNDM}} = MSE_{\mathrm{DM}}$. This means that if the MSE of QNDM for 500 shots is lower than the MSE of DM with 500 shots, the number of shots for DM must be increased (or vice versa) to satisfy the condition $MSE_{\mathrm{QNDM}} = MSE_{\mathrm{DM}}$. As in the previous section, we present our cost analysis for a general example in which all coefficients of the observable $\hat{M}$ are drawn from a Gaussian distribution centered at 0 with a standard deviation of 5. As before, we took a value of λ equal to the square root of the inverse of the sum of the absolute values of all coefficients h_i, $\lambda = 1/\sqrt{\sum_i |h_i|}$.

The results of figure 5 illustrate the expected advantage of QNDM over the DM approach. Panel (a), on the left, shows the resource cost $\mathcal{C}\left(g_l^i\right)$ for the two approaches in the limit $k \gg nJ$. They are plotted as a function of the number of logical operators k needed to implement $U(\vec{\theta})$. The simulations have been done for $n = 10$ qubits and a with $J = 24$ different Pauli strings. Each value in figure 5 is calculated by averaging over different realizations with randomly chosen values of the parameters J, k, and the set of Pauli strings in $\hat{M}$. The average is performed also over the unitary transformations $U(\vec{\theta})$. This is obtained through a series of layers parameterized by $U_j(\vec{\theta}_j)$ and entanglement layers denoted as V_j, as illustrated in equations (3)

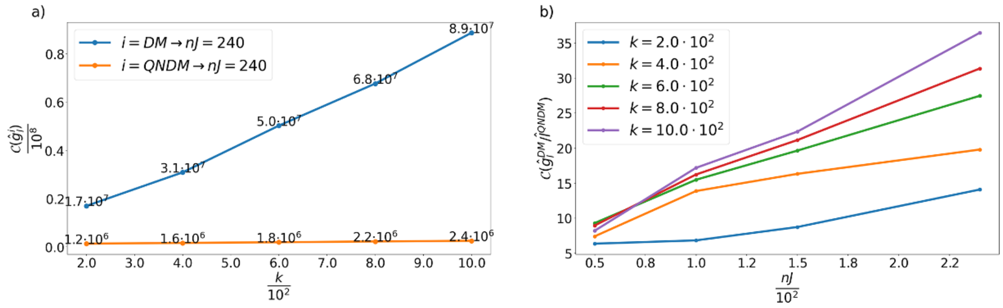

Figure 5. Regime $k \gg nJ$. (a) The averaged number of resources $\mathcal{C}(g_l^i)$ needed to estimate the derivative of the cost function along the direction l; here, $i = \text{QNDM}$ (orange line) and $i = \text{DM}$ (blue line). $\mathcal{C}(g_l^i)$ is plotted as a function of the number of logical operators k for a fixed $nJ = 240$. (b) Ratio between DM and QNDM resources numbers $\left(\mathcal{C}(g_l^{\text{DM}}/g_l^{\text{QNDM}}\right))$ as a function of Pauli string J each colored line corresponds at a fixed k. The quantum circuit is composed of $n = 10$ qubits and the average is taken over $L = 50$ realization as discussed in the main text. The number of shots for the QNDM is $N_{\text{QNDM}} = 500$, for DM, it is calculated for each realization such that the MSE errors are equal for both methods.

and (4). Within the parameterized layer, the constituent single-qubit rotations $R_i^j(\theta_i^j)$, and their corresponding parameters $\vec{\theta}_i^j$, are selected at random.

In panel (a) of figure 5, as expected, QNDM needs fewer resources than DM to evaluate the derivatives. Less obvious is that already for a limited number of logical operators, i.e. $k \sim 10^2$, we achieve a reduction of more than an order of magnitude in the number of logical gates.

In panel (b) of figure 5, on the right, we show the ratio of the resources in terms of the number of Pauli strings J, comparing different possible choices for k. Already for $k \sim 10^2$, the ratio closely follows the theoretical linearity predicted in equation (29). In this regime, the DM method requires more than 35 times the resources of the QNDM approach.

The regime in which $k \ll nJ$ is analyzed in figure 6. Panel (a), on the left, shows the resource cost function $\mathcal{C}(g_l^i)$ $(i = \text{QNDM}, \text{DM})$ obtained to estimate the derivative along the direction l. The values are plotted as a function nJ. The simulations are done for $n = 10$ qubits with $k = 0.5 \cdot 10^3$ fixed. Each value in figure 6 is calculated by performing an average over a collection of different realizations, as discussed above. As delineated in table 1, for DM (the blue line) the number of resources increases linearly in the number of J. Already for $nJ \sim 10^3$, the DM method already requires 10 times more resources than the QNDM one.

In panel (b), on the right, we show the ratio of the resources as a function of the number of logical operators k and for different values of nJ. The expected advantage, equation (30), of the QNDM approach is confirmed in this regime. Already for $nJ \sim 10^3$ the DM approach needs 40 times more resources than the QNDM one.

Notice in all our simulations and both regimes, we have considered relatively small and simple systems to provide a more conservative estimate. We might expect a further increasing advantage in realistic computational problems, as they usually require more logical gates and involve more complex observables $\hat{M}$.

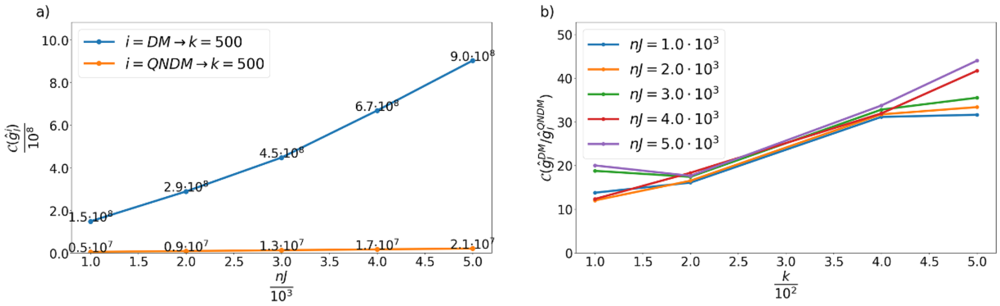

Figure 6. Regime $k \ll nJ$. The averaged number of resources $\mathcal{C}(g_j^i)$ needed to estimate the derivative of the cost function along the direction j ($i =$ QNDM(orange line) and $i =$ DM(blue line)). $\mathcal{C}(g_j^i)$ is plotted as a function of the product nJ for a fixed number of logical operators $k = 0.5 \times 10^3$. (b) Ratio between DM and QNDM resources $\left(\mathcal{C}\left(g_j^{\mathrm{DM}}/g_j^{\mathrm{QNDM}}\right)\right)$ as a function of the number of logical operators k each colored line corresponds at a fixed nJ. The quantum circuit is composed of $n = 10$ qubits and the average is taken over $L = 50$ realization as discussed in the main text. The number of shots for the QNDM is $N_{\mathrm{QNDM}} = 500$, for DM, it is calculated for each realization such that the MSE errors are equal for both methods.

5. Second derivative and Hessian calculation

The QNDM approach can be naturally extended to the calculation of higher derivatives [14]. Having access to the second derivatives of the cost function allows us to evaluate the Hessian matrix. This can be useful for the implementation of second-order optimizers like the Newton optimizer or the Diagonal Newton optimizer [5].

With a straightforward extension of equation (7), it can be shown that the second derivative, requires the evaluation of the cost function f in four points [5, 14]. To implement this in the DM approach, we need four independent measurements of the observables $\hat{M}$. On the other hand, following the QNDM paradigm, we can just couple the system to the detector four times to store in the phase of the latest the same amount of information [14]. Then, analogously to the first-order derivative, we perform a measure on the detector only once. For further details and the numerical simulations about the second derivatives with QNDM method, we address the interested reader to appendix A. In this section, instead, we present the persistence of the advantages observed in first derivatives even when extending to second derivatives.

5.1. Costs analysis for second derivatives

In table 2, we insert the costs for calculating second derivatives. We denote the resource cost function, $\mathcal{C}(g_{w,l}^i)$, for both the approaches ($i =$ QNDM, DM) in terms of the number of gates required to compute the second derivative along the directions w and l. These values are obtained from [5, 13, 14]. As above, the costs depend on the number of Pauli strings J, the number of qubits n, the number of shots or repetitions N_i, and the number of logical operators needed to implement $U(\vec{\theta})$, denoted by k.

From table 2, we observe that the cost of calculating the second derivative with DM is twice the cost for the first derivative with the same method. For QNDM, the variation in cost is due to the coefficient of k (which increases from 3 to 7) and to the coefficient of J, which is twice the

Table 2. Resource cost function to compute second derivatives with the QNDM and the DM approaches.

Method	Shots	Costs
DM	N_{DM}	$\mathcal{C}\left(g_{w,l}^{\mathrm{DM}}\right) = N_{\mathrm{DM}} 4J(k+n)$
QNDM	N_{QNDM}	$\mathcal{C}\left(g_{w,l}^{\mathrm{QNDM}}\right) = N_{\mathrm{QNDM}}(7k+16Jn)$

one for a single derivative. A detailed description of the implementation of both approaches for computing the second derivative is provided in appendix A.

For a fixed value of MSE error and when $k \gg nJ$, the ratio of the resource cost functions reads

$$\mathcal{C}\left(\frac{g_{w,l}^{\mathrm{DM}}}{g_{w,l}^{\mathrm{QNDM}}}\right) = \frac{4J}{7}\frac{N_{\mathrm{DM}}}{N_{\mathrm{QNDM}}} = \mathcal{O}\left(J\right) \tag{32}$$

The opposite regime is $k \ll nJ$. In this case, the resource ratio is given by

$$\mathcal{C}\left(\frac{g_{w,l}^{\mathrm{DM}}}{g_{w,l}^{\mathrm{QNDM}}}\right) = \frac{k}{4n}\frac{N_{\mathrm{DM}}}{N_{\mathrm{QNDM}}} = \mathcal{O}\left(k\right) \tag{33}$$

These results demonstrate that the advantage in terms of the cost function for QNDM is also maintained in the case of the second derivative. As already mentioned, all the simulations used to validate these results are included in appendix A.

6. Conclusions

In this paper, we have presented a detailed study of an alternative approach to evaluating the derivatives of a cost function using a quantum computer, along with its implementation in *Python* publicly available via GitHub [16]. This method called QNDM, was first introduced in reference [14]. In addition to providing a quantitative estimate of the resources needed to run the QNDM protocol, we have conducted an in-depth comparison with the state-of-the-art DM approach [13]. The results obtained are encouraging and pave the way for the first applications of the QNDM algorithm to real-world problems since it allows us to consistently reduce the resources needed for the computation.

This advantage, as highlighted in [14], arises from the use of a quantum detector to store information about the derivative of the cost function, allowing us to run the quantum circuit and perform measurements only once. In contrast, the DM approach requires executing two separate circuits, with each circuit needing to be repeated for the number of Pauli strings. For the second derivative case, this requirement increases to four circuits.

The numerical simulations performed using the Qiskit framework have confirmed the analytical estimates. Even for small systems and simplified quantum circuits, we observe a considerable reduction in the resources needed to run the QNDM approach compared to the DM method. We anticipate that this advantage will further increase for more practical and complex problems, such as the simulation of chemical compounds, molecules, or small quantum systems [36, 37]. These results position the QNDM approach as a valuable alternative for implementing VQAs on the next generation of noisy quantum computers.

Data availability statement

The codes used to obtain the results for the QNDM and DM approaches presented in this article are openly available on GitHub at the following URL: https://github.com/simonecaletti/qndm-gradient.

Acknowledgments

PS, GM, and DM acknowledge financial support from INFN. This work was carried out while GM was enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome in collaboration with the Dept. of Informatics, Bioengineering, Robotics, and Systems Engineering, Polytechnic School of Genova.

SC would like to thank the University of Amsterdam and the Delta-ITP program for their support during the initial part of this work.

Appendix A. Second derivatives

In this section, we present a detailed analysis of the second derivative. The parameter shift rule, equation (5), can be generalized to calculate the second derivative [5].

$$
\begin{aligned}
g_{w,l} = & \frac{\partial^2 f\left(\vec{\theta}\right)}{\partial\theta_l\partial\theta_w} \\
= & \left[f\left(\vec{\theta}+s\left(\hat{e}_l+\hat{e}_w\right)\right)-f\left(\vec{\theta}+s\left(-\hat{e}_l+\hat{e}_w\right)\right)\right. \\
& \left.-f\left(\vec{\theta}+s\left(\hat{e}_l-\hat{e}_w\right)\right)+f\left(\vec{\theta}-s\left(\hat{e}_l+\hat{e}_w\right)\right)\right]\left[2\sin^2 s\right]^{-1}.
\end{aligned} \tag{S1}
$$

We can re-write equation (S2) in terms the density matrix representing the n-qubits initial state, i.e. $\rho_s^0 = |\psi_0\rangle\langle\psi_0|$

$$
\begin{aligned}
\frac{\partial^2 f\left(\vec{\theta}\right)}{\partial\theta_l\partial\theta_i} = & \left[\mathrm{Tr}_S\left[U^\dagger\left(\vec{\theta}+s\left(e_l+e_w\right)\right)\hat{M}U\left(\vec{\theta}+s\left(e_l+e_w\right)\right)\rho_S^0\right.\right. \\
& -U^\dagger\left(\vec{\theta}+s\left(-e_l+e_w\right)\right)\hat{M}U\left(\vec{\theta}+s\left(-e_l+e_w\right)\right)\rho_S^0 \\
& -\ U^\dagger\left(\vec{\theta}+s\left(e_l-e_w\right)\right)\hat{M}U\left(\vec{\theta}+s\left(e_l-e_w\right)\right)\rho_S^0 \\
& \left.\left.+U^\dagger\left(\vec{\theta}-s\left(e_l+e_w\right)\right)\hat{M}U\left(\vec{\theta}-s\left(e_l+e_w\right)\right)\rho_S^0\right]\right]\left[4\sin^2 s\right]^{-1}.
\end{aligned} \tag{S2}
$$

It follows that for DM method to calculate the values of $f(\vec{\theta}+s(\hat{e}_l+\hat{e}_w))$, we run the quantum circuit to implement the corresponding $U(\vec{\theta}+s(\hat{e}_l+\hat{e}_w))$ and then perform a projective measurement for each Pauli string composing the observable $\hat{M}$ [5]. The same procedure is then implemented for $f(\vec{\theta}+s(-\hat{e}_l+\hat{e}_w))$,$f(\vec{\theta}+s(\hat{e}_l-\hat{e}_w))$,$f(\vec{\theta}-s(\hat{e}_l+\hat{e}_w))$, then the derivative can be easily extracted. In figure S1 the DM is represented schematically.

On the other hand, in the QNDM method [14], we estimate the second derivative measuring the circuit shown in figure S2, and we avoid the repetitions for each Pauli string and

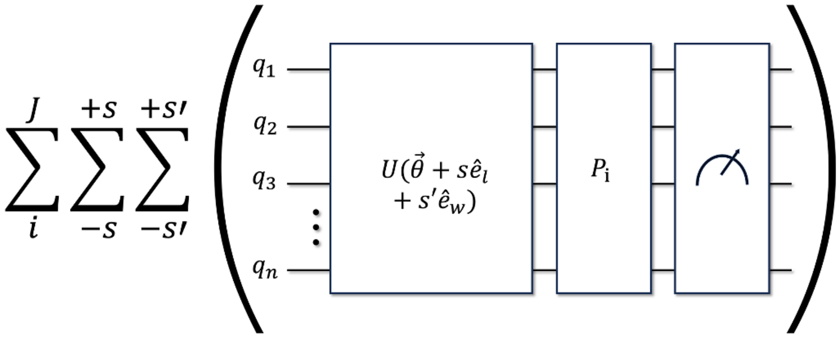
$\sum_{i}^{J}\sum_{-s}^{+s}\sum_{-s'}^{+s'}$
q_1
q_2
q_3
q_n
$U(\vec{\theta}+s\hat{e}_l + s'\hat{e}_w)$
P_i

Figure S1. Quantum circuit of the implementation of the DM protocol. Here $U(\vec{\theta}+s\hat{e}_l+s'\hat{e}_w)$ is equation (1) with two parameters shifted in directions $\hat{e}_l$ and $\hat{e}_w$. The J is equal to the number of Pauli strings and the values $\pm s$ and $\pm s'$ are the shift for the parameter shift rule.

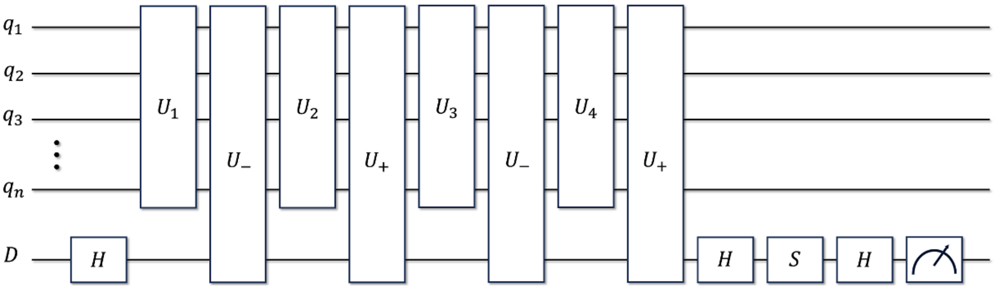

Figure S2. Quantum circuit of the implementation of the QNDM protocol for second derivatives. Here, H is the Hadamard gate, S is the phase gate, $U_1 = U(\vec{\theta}+s(e_l-e_w))$, $U_2 = U^\dagger(\vec{\theta}+s(e_l-e_w))U(\vec{\theta}-s(e_l+e_w))$, $U_3 = U^\dagger(\vec{\theta}-s(e_l+e_w))U(\vec{\theta}+s(-e_l+e_w))$, $U_4 = U^\dagger(\vec{\theta}+s(-e_l+e_w))U(\vec{\theta}+s(e_l+e_w))$ and $U_\pm = \exp\{\pm i\lambda\hat{Z}_a\otimes\hat{M}\}$ is the system-detector coupling operator.

for each term of the parameter shift rule. The unitary transformation corresponding to the full QNDM evolution is

$$U^2_{\text{tot}} = e^{i\lambda Z_a\otimes\hat{M}}U_4 e^{-i\lambda Z_a\otimes\hat{M}}U_3 e^{i\lambda Z_a\otimes\hat{M}}U_2 e^{-i\lambda Z_a\otimes\hat{M}}U_1, \tag{S3}$$

where the U_y operators (with $y=1,..,4$) correspond to the operators working in the $\vec{\theta}$ space and they read

$$\begin{aligned} U\left(\vec{\theta}+s\left(e_l-e_w\right)\right) &= U_1 \\ U^\dagger\left(\vec{\theta}+s\left(e_l-e_w\right)\right)U\left(\vec{\theta}-s\left(e_l+e_w\right)\right) &= U_2 \\ U^\dagger\left(\vec{\theta}-s\left(e_l+e_w\right)\right)U\left(\vec{\theta}+s\left(-e_l+e_w\right)\right) &= U_3 \\ U^\dagger\left(\vec{\theta}+s\left(-e_l+e_w\right)\right)U\left(\vec{\theta}+s\left(e_l+e_w\right)\right) &= U_4. \end{aligned} \tag{S4}$$

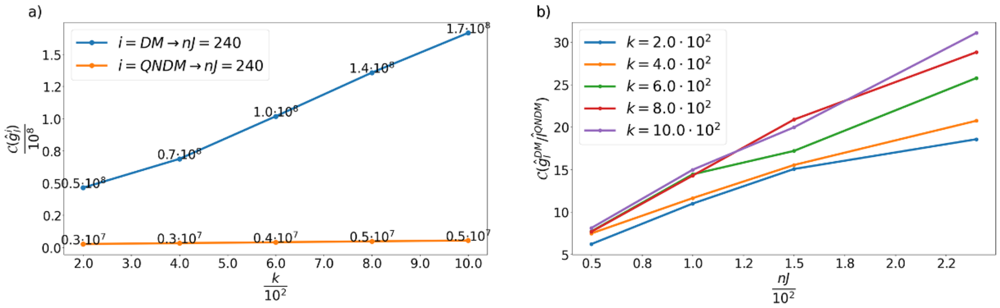

Figure S3. Regime $k \gg nJ$. (a) Number of resources $\mathcal{C}(g^i_{w,l})$ needed to estimate the second derivative of the cost function along the directions l and w; here, $i = \text{QNDM}$ (orange line) and $i = \text{DM}$ (blue line). $\mathcal{C}(g^i_{w,l})$ is plotted in function of the number of logical operators k for a fixed $nJ = 240$. (b) Ratio between DM and QNDM resources numbers $\mathcal{C}(g^{\text{DM}}_{w,l}/g^{\text{QNDM}}_{w,l})$ as a function of Pauli string nJ each colored line corresponds at a fixed k. The quantum circuit is composed of $n = 10$ qubits and the average is taken over $L = 50$ realization as discussed in the main text. The number of shots for the QNDM is $N_{\text{QNDM}} = 500$, for DM, it is calculated for each realization such that the MSE errors are equal for both methods.

A.1. Errors analysis for second derivative

The MSEs for the second derivatives for both methods, respectively, are equal to

$$\text{MSE}_{\text{QNDM}}\left(\hat{g}_{w,l}\right) = \frac{\lambda^2\left(\partial^2_\lambda \mathcal{G}_\lambda\right)^2}{4} + \frac{\sigma_D^2}{16 N_{\text{QNDM}} \sin^4 s} \frac{1}{\lambda^2\left(1-(2P_0-1)^2\right)} \tag{S5}$$

$$\text{MSE}_{\text{DM}}\left(\hat{g}_{w,l}\right) = \sum_i^J = \frac{h_i^2 \sigma_s^2}{4 N_{\text{DM}} \sin^4 s}. \tag{S6}$$

A.2. Cost simulations for second derivative

In section 5.1, we found the asymptotic cost ratios $\mathcal{C}(g^{\text{DM}}_{w,l}/g^{\text{QNDM}}_{w,l})$ for the two regimes: $k \ll nJ$ equation (33) and $k \gg nJ$ equation (32). In figures S4 and S3, we include simulations that illustrate how the theoretical results are also maintained for the second derivatives. For each value presented in the figures, we averaged over $L = 50$ different realizations. Panel (a) in figure S3 shows the numbers of resources, $\mathcal{C}(g^i_{w,l})$, in the function of the logical operators k, when we consider the limit of $k \gg nJ$. The simulations are done for $n = 10$ qubits, fixed $nJ = 240$, and fixed shots $N = 500$. In the second panel (b) we show the ratio of the resources as a function of the number of Pauli strings J and for k different numbers of logical operators in $U(\vec{\theta})$. Panel (a) in figure S4 shows the numbers of resources, $\mathcal{C}(g^i_{w,l})$, in function of nJ. We are working in the limit $k \ll nJ$. The simulations are done for $n = 10$ qubits, a fixed number of logical operators $k = 5 \cdot 10^2$, and fixed shots $N = 500$. In the second panel (b) we show the ratio of the resources as a function of logical operators k for different values of the product nJ. The advantage of the QNDM over the DM approach is confirmed also for the second derivatives.

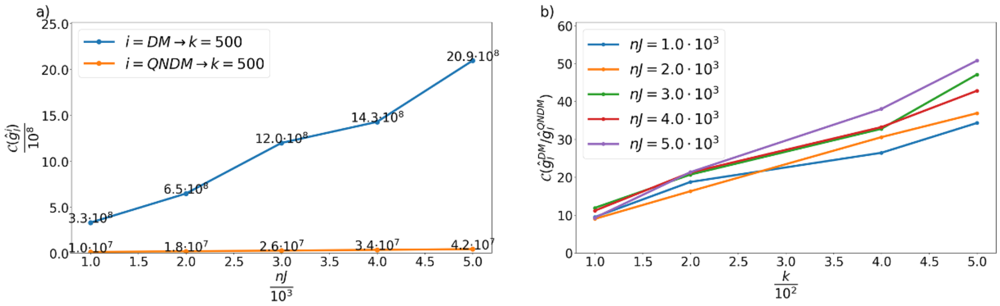

Figure S4. (a) The averaged number of resources $\mathcal{C}(g^i_{w,l})$ needed to estimate the second derivative of the cost function along the directions l and w; here, $i = \text{QNDM}$ (orange line) and $i = \text{DM}$ (blue line). $\mathcal{C}(g^i_{w,l})$ is plotted in function of the number of Pauli strings J for a fixed number of logical operators $k = 5 \cdot 10^3$. (b) Ratio between DM and QNDM resources numbers $\mathcal{C}(g^{\text{DM}}_{w,l}/g^{\text{QNDM}}_{w,l})$ as a function of logical operators k each coloured line corresponds at a fixed nJ. The quantum circuit is composed of $n = 10$ qubits and the average is taken over $L = 50$ realization as discussed in the main text. The number of shots for the QNDM is $N_{\text{QNDM}} = 500$, for DM, it is calculated for each realization such that the MSE errors are equal for both methods.

Appendix B. how to install QNDM

To install the QNDM library just clone the *GitHub* repository (link in [16]) using the command

```
git clone https://github.com/simonecaletti/qndm-gradient.git
```

All the functions for the QNDM algorithm are defined in the *qndm* folder. The *test_scripts* folder instead contains a set of scripts with some simple tests, like QNDM and DM derivatives evaluation and optimization tasks. To have access to the interface with the hamlib library [37] you need to install the *mat2qubit* package [38]. The instructions are contained in the *hdf5-install.sh* script, so just run

```
bash hdf5-install.sh
```

HamLib (for Hamiltonian Library), is freely available online and contains problem sizes ranging from 2 to 1000 qubits. It includes problem instances of the Heisenberg model, Fermi-Hubbard model, Bose–Hubbard model, molecular electronic structure, molecular vibrational structure, MaxCut, Max-k-SAT, Max-k-Cut, QMaxCut, and the traveling salesperson problem [37]. To test the installation create the folder *output_test* and run a test script, for example

```
python3 test_qndm.py
```

Run it from the *qndm-gradient/* folder or add the corresponding path to your PYTHON PATH environment variable. If the installation is working correctly a *QNDM_der.csv* and a *RunCard_Der.txt* file have been created. The first one contains information about the gradient computation using the QNDM algorithm, while the second is an automatically generated runcard containing the details of the run.

ORCID iDs

G Minuto https://orcid.org/0009-0004-8784-1765
P Solinas https://orcid.org/0000-0002-3764-7671

References

[1] Gilyén A, Arunachalam S and Wiebe N 2019 Optimizing quantum optimization algorithms via faster quantum gradient computation *Proc. 2019 Annual ACM-SIAM Symp. on Discrete Algorithms (SODA)* pp 1425–44
[2] Moll N *et al* 2018 *Quantum Sci. Technol.* **3** 030503
[3] Barkoutsos P K, Gkritsis F, Ollitrault P J, Sokolov I O, Woerner S and Tavernelli I 2021 *Chem. Sci.* **12** 4345
[4] McClean J R, Romero J, Babbush R and Aspuru-Guzik A 2016 *New J. Phys.* **18** 023023
[5] Mari A, Bromley T R and Killoran N 2021 *Phys. Rev. A* **103** 012405
[6] Boyd S and Vandenberghe L 2004 *Convex Optimization* (Cambridge University Press)
[7] Adby P 2013 *Introduction to Optimization Methods* (*Chapman and Hall Mathematics Series*) (Springer)
[8] Banchi L and Crooks G E 2021 *Quantum* **5** 386
[9] Wierichs D, Izaac J, Wang C and Lin C Y-Y 2022 *Quantum* **6** 677
[10] Solinas P and Gasparinetti S 2015 *Phys. Rev. E* **92** 042150
[11] Solinas P and Gasparinetti S 2016 *Phys. Rev. A* **94** 052103
[12] Schuld M, Bergholm V, Gogolin C, Izaac J and Killoran N 2019 *Phys. Rev. A* **99** 032331
[13] Cerezo M *et al* 2021 *Nat. Rev. Phys.* **3** 625
[14] Solinas P, Caletti S and Minuto G 2023 *Eur. Phys. J. D* **77** 76
[15] Guerreschi G G and Smelyanskiy M 2017 Practical optimization for hybrid quantum-classical algorithms (arXiv:1701.01450 [quant-ph])
[16] Caletti S and Minuto G 2024 qndm-gradient (available at: https://github.com/simonecaletti/qndm-gradient)
[17] McClean J R, Boixo S, Smelyanskiy V N, Babbush R and Neven H 2018 *Nat. Commun.* **9** 4812
[18] Nielsen M A and Chuang I L 2010 *Quantum Computation and Quantum Information* (Cambridge University Press)
[19] McArdle S, Endo S, Aspuru-Guzik A, Benjamin S C and Yuan X 2020 *Rev. Mod. Phys.* **92** 015003
[20] Tilly J *et al* 2022 *Phys. Rep.* **986** 1
[21] Kingma D P and Ba J 2017 Adam: a method for stochastic optimization (arXiv:1412.6980 [cs.LG])
[22] Kübler J M, Arrasmith A, Cincio L and Coles P J 2020 *Quantum* **4** 263
[23] Sweke R, Wilde F, Meyer J, Schuld M, Faehrmann P K, Meynard-Piganeau B and Eisert J 2020 *Quantum* **4** 314
[24] Li J, Yang X, Peng X and Sun C-P 2017 *Phys. Rev. Lett.* **118** 150503
[25] Mitarai K, Negoro M, Kitagawa M and Fujii K 2018 *Phys. Rev. A* **98** 032309
[26] Mitarai K and Fujii K 2019 *Phys. Rev. Res.* **1** 013006
[27] Hatano N and Suzuki M 2005 Finding exponential product formulas of higher orders *Quantum Annealing and Other Optimization Methods* (Springer) pp 37–68
[28] Solinas P, Amico M and Zanghì N 2021 *Phys. Rev. A* **103** L060202
[29] Solinas P, Amico M and Zanghì N 2022 *Phys. Rev. A* **105** 032606
[30] Cox D R and Hinkley D V 1974 *Theoretical Statistics* (Chapman & Hall)
[31] Taylor J 1997 *An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements* (*Asmsu/Spartans.4.spartans Textbook*) (University Science Books)
[32] Wecker D, Hastings M B and Troyer M 2015 *Phys. Rev. A* **92** 042303
[33] Sim S, Johnson P D and Aspuru-Guzik A 2019 *Adv. Quantum Technol.* **2** 1900070
[34] Flynn B, Gentile A A, Wiebe N, Santagati R and Laing A 2022 *New J. Phys.* **24** 053034
[35] IBM Quantum 2024 Aersimulator (available at: https://docs.quantum.ibm.com/api/qiskit/0.42/qiskit_aer.AerSimulator)
[36] Weber S J, Chantasri a, Dressel J, Jordan a N, Murch K W and Siddiqi I 2014 *Nature* **511** 570

[37] Sawaya N P *et al* 2024 Hamlib: a library of hamiltonians for benchmarking quantum algorithms and hardware (arXiv:2306.13126 [quant-ph])

[38] Sawaya N P 2022 mat2qubit: a lightweight pythonic package for qubit encodings of vibrational, bosonic, graph coloring, routing, scheduling, and general matrix problems (arXiv:2205.09776 [quant-ph])