EPJ.org

EPJ Quantum Technology
a SpringerOpen Journal

Benchmarking machine learning algorithms for adaptive quantum phase estimation with noisy intermediate-scale quantum sensors

Nelson Filipe Costa[1], Yasser Omar[1,2], Aidar Sultanov[3*] and Gheorghe Sorin Paraoanu[3]

*Correspondence:
aidar.sultanov@aalto.fi
[3]QTF Centre of Excellence,
Department of Applied Physics,
Aalto University School of Science,
Helsinki, Finland
Full list of author information is
available at the end of the article

Abstract

Quantum phase estimation is a paradigmatic problem in quantum sensing and metrology. Here we show that adaptive methods based on classical machine learning algorithms can be used to enhance the precision of quantum phase estimation when noisy non-entangled qubits are used as sensors. We employ the Differential Evolution (DE) and Particle Swarm Optimization (PSO) algorithms to this task and we identify the optimal feedback policies which minimize the Holevo variance. We benchmark these schemes with respect to scenarios that include Gaussian and Random Telegraph fluctuations as well as reduced Ramsey-fringe visibility due to decoherence. We discuss their robustness against noise in connection with real experimental setups such as Mach–Zehnder interferometry with optical photons and Ramsey interferometry in trapped ions, superconducting qubits and nitrogen-vacancy (NV) centers in diamond.

Keywords: Machine learning; Quantum phase estimation; Qubit

1 Introduction

Precision measurements play a fundamental role in physics, as they constitute a key ingredient of many state-of-the-art applications and experiments testing the limits of scientific theories. But the accuracy to which such measurements can be performed is itself governed by the laws of physics—and, ultimately, by those of quantum mechanics [1–4].

A generic measurement protocol for estimating the value of an unknown parameter consists of preparing a probe in a desired initial state, allowing it to interact with the physical system whose state depends on the parameter and, finally, obtaining a measurement result that encapsulates the information about it. This process, however, is often affected by systematic and statistical errors. While the source of the former may stem from imperfect calibration of the measurement instruments, the origin of the latter can either be accidental, due to insufficient control of the measurement chain, or fundamental, deriving from the nature of the physical measurement [5, 6]. Fortunately, statistical errors, regardless of their origin, can be minimized by repeating the process and averaging the resulting outcomes, as a consequence of the central limit theorem [7]. This theorem states that,

Springer

given a large number N of independent measurement results, their average will converge to a Gaussian distribution with standard deviation scaling as $1/\sqrt{N}$. In metrology, this is referred to as the Standard Quantum Limit (SQL). For many practical applications that involve noisy and error-prone systems, it is essential to devise protocols that yield precision levels close to the SQL. However, it is known that this limit is not fundamental—the ultimate limit set by the laws of quantum physics is the Heisenberg Limit (HL) [8–11].

Recently, several systems have achieved sufficient levels of technological maturity to allow the experimental exploration of these limits. Due to the non-idealities present in these systems, they are typically referred to as NISQ (noisy intermediate-scale quantum). Largely, the motivation for this development has been rooted in quantum information processing, where efficient and precise measurements and state manipulation is required [12], and indeed significant progress towards the implementation of quantum gates and algorithms has been made in optical setups [13, 14], NV centers in diamond [15, 16], trapped ions [17, 18] and superconducting circuits [19, 20]. Since it employs similar control and measurement techniques as quantum computing, the exploration of quantum-enhancing techniques has grown as a separate field, usually referred to as quantum metrology [21, 22], with applications such as ultrasensitive force detection [23], adaptive environment sensing [24], near-surface electric field measurements [25], sensing of weak signals [26] and even detection of gravitational waves [27].

The paradigmatic protocol in quantum metrology is quantum phase estimation. The quantum phase is a parameter that cannot be directly measured and yet, it contains information about other quantities of interest, such as electric or magnetic fields. The traditional quantum metrology approach to phase estimation has been through the use of highly entangled states such as NOON states [1, 3, 28], as well as other specially optimized states [29–31]. However, highly entangled states tend to be very sensitive to noise: for NOON states even the loss of a single qubit (photon) results in a separable state. Thus, the implementation of this method is challenging with respect to real-life applications. Fortunately, it has been later realized that precise quantum phase estimation does not necessarily require entanglement. In optics, it was demonstrated that adaptive homodyne phase measurements yield uncertainties close to the quantum uncertainty of coherent states [32, 33]. It was also shown how to adaptively estimate the angle of a half waveplate that characterizes the linear polarization of photons [34, 35] as well as how to perform phase-shift estimation with polarization-encoded qubits [36]. Furthermore, the inverse quantum Fourier transform, which is the last step in Shor's factorization algorithm and which typically requires gate entangling, can be carried out using local measurements and feedback [37]. This approach has been used to break the SQL in experiments with photons [38], superconducting qubits [39] and NV centers in diamond [40].

The incorporation of machine learning techniques in estimation protocols is a natural step forward. Seminal theoretical work on employing reinforcement learning algorithms has demonstrated the potential of these methods for reaching sensitivities below the SQL when used in conjunction with entanglement [41–47]. Recently, some of these methods have been tested experimentally in an optics setup with their estimation precision being limited by the SQL [48]. This only upholds the potential of applying machine learning for the optimization of parameter estimation under resources and noise level constraints, but the foundations of these methods are still poorly understood. In mathematical statistics, the limits of machine learning algorithms are an active area of investigation—for example

deriving bounds on the number of samples needed to reach a certain accuracy [49]. However, the issue at stake is that typically all the mathematical results, including the SQL, are derived from the so-called central limit $N \gg 1$ and for ideal noiseless systems. Yet the relevant regime for the present status of quantum technologies in the NISQ era is that of moderate N and significant noise. In this context no general unbiased estimator has been found yet [28].

The objective of this work is to employ two machine learning algorithms, the DE and the PSO algorithm, in the design of adaptive estimation schemes capable of driving the measurement precision close to the SQL. We numerically show that the SQL is still a valid bound even in the regime of not too large N. We also demonstrate that machine learning algorithms can be used even in the presence of various experimental noises, consequently providing robust policies with better performance than non-adaptive protocols. For practical devices such as magnetometers based on single qubits, these methods can be directly applied. More precisely, we observe that in order to increase the precision of these devices, one straightforward option is to increase the sensitivity to the parameter to be estimated by, for example, using higher-energy states or by increasing the response using a different biasing point. However, both of these strategies result in an increased coupling to noise sources and, therefore, a compromise needs to be reached in order to achieve the maximum performance. This will be further detailed in the paper when analyzing the experimental realizations.

Both the DE and the PSO algorithm can also be employed as subroutines in order to enhance other quantum algorithms. For example, algorithms that use variable sensing times, or multipass techniques, are in principle able to breaks the SQL and reach the HL. Instead of averaging at every sensing time, which is the typical approach used one can further increase the sensitivity by using our technique. Beyond quantum metrology, machine learning protocols could become useful in other quantum-information paradigmatic problems that involve phase estimation, such as factorization, sampling, and computation of molecular spectra [50]. In particular, our calculations are relevant for NISQ quantum technologies, where the number of qubits is limited and subject to errors and noise. Overall, by benchmarking these two important protocols for the task of quantum phase estimation, we hope that machine learning algorithms will be more prominently employed in applications such as optical interferometry and magnetometry, where the increase of precision is essential and where the aim is set on reaching the HL.

The paper is organized in the following sections. Section 2 describes the general concept of adaptive quantum phase estimation in Ramsey interferometry, discussing the updating procedure as well as the relevant sources of noise. Section 3 presents the two machine learning algorithms, the DE and PSO algorithm. Section 4 presents our main results, where we show how the two algorithms allow us to approach the SQL. In this section we also provide an analysis on the effects of Gaussian noise, Random Telegraph noise (RTN) and quantum decoherence on the performance of the algorithms. Section 5 discusses the implementation of our protocol on several experimental platforms, namely Mach–Zehnder optical interferometers, superconducting qubits, trapped ions and defects in diamond. Finally, we conclude our results in Sect. 6.

Figure 1 Adaptive quantum phase estimation scheme. A qubit m is injected into the quantum circuit either in the quantum state $|0\rangle$ or $|1\rangle$ and its measured outcome $\zeta_m \in \{0,1\}$ is used together with some policy phase instruction $x_m \in [0, 2\pi[$ to prepare the circuit for the next round of measurements. The quantum circuit consists of two Hadamard gates H, a phase shift gate $U_{\phi,\theta}$, a detector D, and a processing unit with phase update instructions for the controllable phase shifter
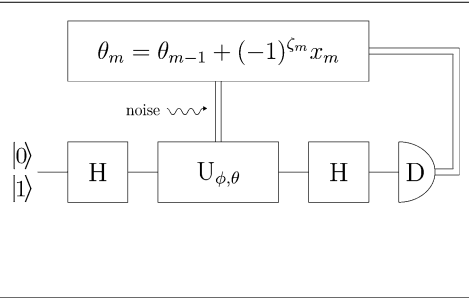
2 Adaptive quantum phase estimation scheme

To perform the estimation of an unknown phase ϕ, we use a generic adaptive scheme that is able to adjust the value of a control phase θ to match the value of ϕ, based on the results of previous measurements in a Ramsey interferometric sequence.

The schematic representation of the adaptive quantum phase estimation scheme displayed in Fig. 1 consists of a quantum circuit made of a Hadamard gate H, a phase shift gate $U_{\phi,\theta}$, another Hadamard gate H and a detector augmented with a processing unit to calculate the value of θ for the next round of measurements. Using standard notations in quantum information, the Hadamard and phase shift gates can be respectively defined as

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \quad \text{and} \quad U_{\phi,\theta} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i(\phi-\theta)} \end{bmatrix}. \tag{1}$$

Under this adaptive quantum estimation scheme, an ensemble of N qubits is injected sequentially into the circuit in randomly chosen quantum states, either in state $|0\rangle$ or state $|1\rangle$, and their outcome is measured to prepare the value of the controllable phase shifter for the next incoming qubit. The input state of the quantum circuit takes the form $|0\rangle_1 \otimes |1\rangle_2 \otimes |1\rangle_3 \otimes \cdots \otimes |0\rangle_N$, which is manifestly separable. After the last Nth qubit is injected and its outcome measured, the final phase value of θ is considered to be the estimated value of ϕ. The initial states of the qubits can be represented in Dirac notation as

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \text{and} \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The idea is to have at the end of the process an estimated controllable phase value θ as close as possible to the value of the unknown phase ϕ. The state of each qubit after the second Hadamard gate is

$$|\psi_\pm\rangle = \frac{1}{2}\big[1 \pm e^{i(\phi-\theta)}\big]|0\rangle + \frac{1}{2}\big[1 \mp e^{i(\phi-\theta)}\big]|1\rangle,$$

where the upper sign corresponds to a qubit whose initial state was $|0\rangle$ and the lower sign to one whose initial state was $|1\rangle$. This yields measurement results $\zeta \in \{0,1\}$ with probabilities

$$\mathcal{P}_+(\zeta = 0|\phi,\theta) = \mathcal{P}_-(\zeta = 1|\phi,\theta) = \cos^2 \frac{\phi - \theta}{2}, \tag{2}$$

and

$$\mathcal{P}_+(\zeta = 1|\phi,\theta) = \mathcal{P}_-(\zeta = 0|\phi,\theta) = \sin^2\frac{\phi - \theta}{2}. \tag{3}$$

If the phase difference $\phi - \theta$ is zero, then θ approximates very well the unknown phase ϕ and a qubit prepared in state $|0\rangle$ will also exit in state $|0\rangle$. Similarly, the same logic can be applied for a qubit injected in the quantum state $|1\rangle$.

In order to estimate the phase ϕ at every step m of the algorithm the controllable phase θ_m has to be updated based on the previous result ζ_{m-1} of the measurement. Here we adopt a standard updating rule

$$\theta_m = \theta_{m-1} + (-1)^{\zeta_m} x_m, \tag{4}$$

which has already been employed successfully in similar setups [44–46], and where the initial value of θ can be fixed to $\theta_0 = 0$ without any loss of generality.

Note that the term x_m in Eq. (4) represents the update policy which will be determined by the machine learning algorithms. This update rule can be viewed as a decision tree where for each qubit m, the controllable phase shifter must update its value θ_m either by adding or by subtracting x_m depending on the measured outcome state of the previous qubit $\zeta_m \in \{0,1\}$. Consequently, for N qubits the number of possible θ_m values increases exponentially as $2^{(N+1)} - 1$. This is a Markovian feedback, since the new controllable phase θ_m depends only on the latest measured outcome state ζ_m.

To evaluate the performance of a policy, we use the Holevo variance [29–31], defined as

$$V_{\mathrm{H}} = S^{-2} - 1 = \left|\langle e^{i(\phi-\theta)}\rangle\right|^{-2} - 1, \tag{5}$$

where $\langle e^{i(\phi-\theta)}\rangle$ represents the average value of $e^{i(\phi-\theta)}$ for different phase values ϕ and their respective estimates θ considered in the learning process of the machine learning algorithms. Here we make the notation abbreviation $\theta = \theta_N$, since the Holevo variance of a policy can only be calculated after the last qubit N is injected into the circuit and its outcome measured. The quantity $S = \langle e^{i(\phi-\theta)}\rangle \in [0,1]$ is called the sharpness of the phase distribution, where the value $S = 1$ corresponds to a perfect estimation of ϕ. For periodically bounded variables such as the phase, the Holevo variance is a direct measure of the standard deviation by $(\Delta\phi)^2 = V_{\mathrm{H}}$. Therefore we have $V_{\mathrm{H}} \sim 1/N$ for the SQL. It is also important for the model of the designed adaptive quantum phase estimation scheme to include the possibility of errors and imperfections that occur in a real experimental situation. This provides an important test to the robustness of the algorithms to relatively general sources of noise which can be encountered on most experimental platforms.

The first source of noises to be considered are the noises in the application of the controlled unitary phase shift transformation $U_{\phi-\theta}$, namely the Gaussian and the Random Telegraph noise. Note that the Random Telegraph noise is particularly relevant for experiments involving solid-state qubits.

In the scenario of Gaussian noise, the noise follows a normal distribution parametrized by the standard deviation σ. Letting $\theta^{(\mathrm{GSN})}$ represent the actual value of the controllable phase shifter subjected to the noise, the Gaussian noise distribution can be defined as:

$$p\big(\theta_m^{\mathrm{GSN}}\big) = \frac{1}{\sqrt{2\pi}\,\sigma}\exp\left[-\frac{1}{2\sigma^2}\big(\theta_m^{\mathrm{GSN}} - \theta_m\big)^2\right]. \tag{6}$$

In the scenario of Random Telegraph noise, the noise follows a discrete distribution where at each time step the controllable phase shifter value can be randomly offset by a fixed valued λ with a probability η. Letting $\theta^{(\text{RTN})}$ represent the value of the controllable phase shifter subject to this source of noise, the Random Telegraph noise distribution can be described as:

$$p\left(\theta_m^{\text{RTN}}\right) = \begin{cases} 1 - \eta, & \theta_m^{\text{RTN}} = \theta_m, \\ \eta, & \theta_m^{\text{RTN}} = \theta_m + \lambda. \end{cases} \tag{7}$$

The other important type of noise to be considered affects the qubit itself. This modifies the unitary evolution generated by a Hermitian Hamiltonian $\mathcal{H}$ to a non-unitary evolution given by the Lindblad master equation

$$\dot{\rho} = -\frac{i}{\hbar}[\mathcal{H}, \rho] + \Gamma_1 \mathcal{D}[\sigma_-](\rho) + \frac{\Gamma_\varphi}{2} \mathcal{D}[\sigma_z](\rho), \tag{8}$$

where Γ_1 is the relaxation rate, Γ_φ is the pure dephasing rate and the dissipation superoperator $\mathcal{D}$ acting on the density matrix ρ is defined by $\mathcal{D}[A](\rho) = A\rho A^\dagger - \frac{1}{2}\{A^\dagger A, \rho\}$. It is also useful to introduce the standard notations T_1 and T_2 for the relaxation and decoherence time, $T_1 = 1/\Gamma_1$ and $T_2 = 1/(\Gamma_1/2 + \Gamma_\varphi)$.

To implement the phase gate $U_{\phi-\theta_m}$ from Eq. (1) the Hamiltonian must be of σ_z type, with a component for the unknown phase ϕ and another one for the control θ_m. Typically, for Ramsey experiments with a phase accumulation time τ, we have $\mathcal{H}_m = \frac{\hbar}{2}(\phi/\tau - \theta_m/\tau)\sigma_z$ at the step m, with $U_{\phi-\theta_m} = \exp[-i\mathcal{H}_m\tau/\hbar]$, up to a global phase factor. The solution of Eq. (8) is a 2×2 matrix with elements $\rho_{00}(\tau) = 1 - \exp(-\tau/T_1)\rho_{11}(0)$, $\rho_{01}(\tau) = \exp(-i\phi + i\theta_m - \tau/T_2)\rho_{01}(0)$, $\rho_{10}(\tau) = \exp(i\phi - i\theta_m - \tau/T_2)\rho_{10}(0)$ and $\rho_{11}(\tau) = \exp(-\tau/T_1)\rho_{11}(0)$. If the state at $\tau = 0$ is prepared by the action of the Hadamard gate from either $|0\rangle$ or $|1\rangle$, corresponding respectively to the $+$ and $-$ signs below, and at the final time τ we apply again a Hadamard gate, we obtain that at every step m of the algorithm the probabilities are modified as

$$P_\pm(\zeta_m|\phi, \theta_m) = \frac{1}{2}\left[1 \pm (-1)^{\zeta_m} \nu \cos(\phi - \theta_m)\right], \tag{9}$$

where $\nu = \exp(-\tau/T_2)$ is called interference visibility. One can check that for maximum visibility, $\nu = 1$, we recover Eqs. (2) and (3). Further considerations can be found in Appendix A.1.

3 Machine learning algorithms

The problem of quantum phase estimation relies on a sequential and cumulative set of measurements to drive the estimation process, thus making it an ideal problem for reinforcement learning algorithms. In this work, we considered the Differential Evolution (DE) [51, 52] and the Particle Swarm Optimization (PSO) [53–55], among other reinforcement learning algorithms, as they are the most commonly employed for similar tasks in literature [41–47].

These algorithms employ a direct search method to the exploration of the search space generated by all the possible policy configurations. Direct search methods use a greedy criterion to drive their exploration. Such methods guarantee fairly fast convergence times,

even though such fast convergence times often come at the expense of becoming trapped in a local minimum. This comes as result of the greedy criterion promoting decisions that lead to shorter and more immediate term rewards usually in detriment of fully exploring the total search space. To avoid this scenario, it is important to perform a thorough study on the controllable parameters of each of the mentioned algorithms before applying them to the quantum phase estimation task. To evaluate the performance of an algorithm there are two fundamental criteria: its ability to converge within the imposed number of iterations to a solution and its ability to converge to a valid solution.

3.1 Differential evolution

The implementation of the DE algorithm to the current problem starts with a set of P populations, each representing a candidate solution for the adaptive scheme update policy. Each of these populations is a vector of size N, with each entry representing a new phase instruction to prepare the controllable phase shifter for each of the qubits being injected into the system. The DE algorithm at each iteration will employ its direct search method to P N-dimensional vectors $x_{i,j}^G$, where $i \in \{1, 2, \ldots, P\}$ and $j \in \{1, 2, \ldots, N\}$ represent each entry of the candidate solutions vectors and G represents the generation of the population vectors.

Each of these vectors is initialized with random values for each entry in the interval $x_{i,j}^0 \in [0, 2\pi]$. Afterwards, at each iteration, the DE algorithm generates possible new candidate solution vectors for the next generation by adding the weighted difference between four population vectors to a fifth vector. This process is referred to as mutation:
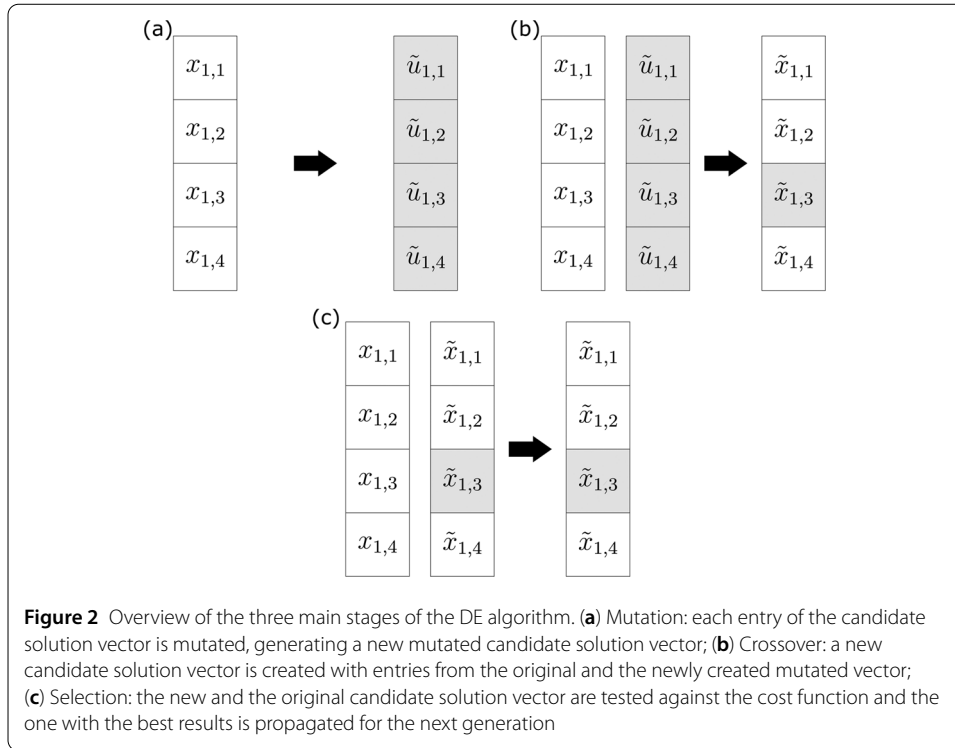
$$\tilde{u}_{i,j}^{G+1} = x_{r_1,j}^G + F \cdot \left(x_{r_2,j}^G + x_{r_3,j}^G - x_{r_4,j}^G - x_{r_5,j}^G \right).$$

Here F represents a constant value in the interval $[0, 1]$ which controls the amplification of the difference between the considered populations. Hence, F will be referred to as the amplification parameter. Note as well that all indexes $\{r_1, \ldots, r_5\}$ are randomly chosen integer values always different between themselves. At this point, the entries of the newly mutated vectors $\tilde{u}_{i,j}^{G+1}$ are randomly mixed with the originally corresponding vectors to increase their diversity. This process is referred to as crossover:

$$\tilde{x}_{i,j}^{G+1} = \begin{cases} x_{i,j}^G, & \text{if } R_1 > C \text{ and } j \neq R_2, \\ \tilde{u}_{i,j}^{G+1}, & \text{if } R_1 \leq C \text{ or } j = R_2. \end{cases}$$

This process is controlled by the crossover parameter C which can take any value in the interval $[0, 1]$. Hence, a crossover only occurs if the random value R_1, which is generated for each population member at each iteration, is below or equal to the chosen crossover parameter. The value of R_2 is an integer randomly chosen for each population at each iteration of the evaluation process to ensure that at least one entry from the newly mutated vectors $\tilde{u}_{i,j}^{G+1}$ is passed to the trial vector for the next generation $\tilde{x}_{i,j}^{G+1}$.

Finally, the new trial vectors are compared against the population vectors of the previous generation to see which perform best against the cost function of the problem and, as a result, become a member of the next generation of populations. This process is referred

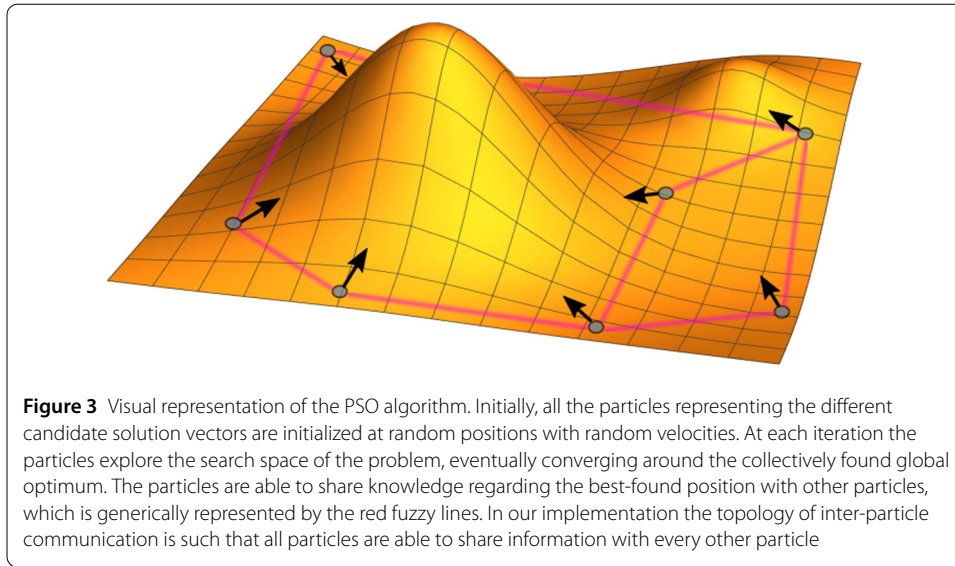**Figure 2** Overview of the three main stages of the DE algorithm. (**a**) Mutation: each entry of the candidate solution vector is mutated, generating a new mutated candidate solution vector; (**b**) Crossover: a new candidate solution vector is created with entries from the original and the newly created mutated vector; (**c**) Selection: the new and the original candidate solution vector are tested against the cost function and the one with the best results is propagated for the next generation

to as selection:

$$
x_{i,j}^{G+1} = \begin{cases} x_{i,j}^{G}, & \text{if } f(x_{i,j}^{G}) < f(\tilde{x}_{i,j}^{G+1}), \\ \tilde{x}_{i,j}^{G+1}, & \text{if } f(x_{i,j}^{G}) \geq f(\tilde{x}_{i,j}^{G+1}). \end{cases}
$$

Here $f(\cdot)$ represents the cost function associated to the quantum phase estimation which is defined by the Holevo variance in Eq. (5). Therefore, if the new trial vectors $\tilde{x}_{i,j}^{G+1}$ minimize this cost function when compared to the previous generation vectors $x_{i,j}^{G}$, then they become part of the next generation of populations. Otherwise, the previous generation populations survive for the next generation. This entire process illustrates the adapted DE algorithm implemented in this work and is schematically represented in Fig. 2.

3.2 Particle swarm optimization

The implementation of the PSO algorithm starts with a set of P particles, each representing an individual candidate solution to the update policy of the adaptive scheme. Each particle can move in the N-dimensional search space associated with the N different phase instructions of the controllable phase shifter for each of the input qubits. Therefore, each particle will be represented by a position vector $x_{i,j}^{G}$ and velocity vector $v_{i,j}^{G}$, where $i \in \{1, 2, \ldots, P\}$ and $j \in \{1, 2, \ldots, N\}$ represent each entry and G the generation of the vectors. Note that the position vector $x_{i,j}^{G}$ corresponds to a candidate solution vector, while the velocity vector $v_{i,j}^{G}$ represents the change in direction of the corresponding position vector in the search space.

All entries of these vectors are initialized with random values in the interval $[0, 2\pi]$. At each iteration, each particle evaluates its current position according to the cost function of the search problem, defined by the Holevo variance in Eq. (5), and compares its value with

Figure 3 Visual representation of the PSO algorithm. Initially, all the particles representing the different candidate solution vectors are initialized at random positions with random velocities. At each iteration the particles explore the search space of the problem, eventually converging around the collectively found global optimum. The particles are able to share knowledge regarding the best-found position with other particles, which is generically represented by the red fuzzy lines. In our implementation the topology of inter-particle communication is such that all particles are able to share information with every other particle

the positions previously visited by itself and with the positions previously visited by the entire collective of particles. If the current position is better than its own previously visited positions, the particle stores it in a variable $pbest_i$, where $i \in [1, \dots, P]$ is the identifier of that particle. If, in addition, the current position is better than all of the other previously visited positions by the entire collective ensemble, the same position is stored in a variable *gbest* shared among all other particles. This process is illustrated in Fig. 3.

Both of these variables will determine the entire exploration of the search space by the P particle candidate solutions. After each iteration, each particle will use this collective knowledge to adjust its displacement for the next turn according to

$$x_{i,j}^{G+1} = x_{i,j}^{G} + w \cdot v_{i,j}^{G+1}$$

and

$$v_{i,j}^{G+1} = v_{i,j}^{G} + \alpha \cdot R_a \cdot (pbest_{i,j} - x_{i,j}) + \beta \cdot R_b \cdot (gbest_j - x_{i,j}).$$

Here the parameter α controls the desirability of each particle to move towards its best found position, while the parameter β controls the desirability of each particle to move towards the best found solution by the entire collective. Both R_a and R_b are uniformly distributed random values in the interval $[0, 1]$. In addition, the parameter w works as a damping weight controlling the change of direction imposed by the new velocity vector at the current position of each particle.

To steer the direction of each particle to a converging candidate policy solution and to avoid overstepping the found minima, an additional parameter $v_{\max}$ is imposed to the algorithm which determines the maximum value that each entry in the velocity vector may take. As the algorithm advances in its search for the optimal policy solution, this parameter will decay with the number of iterations, consequently reducing the step size of each particle and forcing them to converge to a solution. This final adjustment completes the description of the adapted PSO algorithm implemented in this work.

3.3 Parameters analysis

It is important to understand the behaviour and performance of these two algorithms under the different possible configurations of their controllable parameters. To verify the convergence of the solutions, it is important to remember that at each iteration of both algorithms there are P different candidate solutions, each one represented by a given population of N phase value policies. As the algorithm iterates, the candidate solutions should move closer to each other until converging to a final solution. Thus, one way of inferring this convergence value is by calculating the deviation from the average value of each population for each entry of their candidate solution and then averaging these values. To do so, the convergence value L is defined as

$$L = \sum_{j}^{N} \frac{1}{N} \left(\sum_{i}^{P} \frac{\bar{x}_j - x_{i,j}}{P} \right), \tag{10}$$

where $\bar{x}_j$ corresponds to the average value of entry j over all the candidate solutions and $x_{i,j}$ corresponds to the entry j of the candidate solution vector i. Therefore, lower values of L occur when all the candidate solutions are relatively close to each other and the algorithm has converged to a solution. On the other hand, larger values of L indicate that the algorithm was not able to converge to a solution at the given iteration.

The algorithms should also converge to a valid solution. It is not enough that the algorithms converge to a solution, if it is not the correct one. Letting K represent the total number of different phase values of ϕ considered in the learning task imposed to the machine learning algorithms, the performance of a policy can be evaluated by the Holevo variance in Eq. (5). This equation, however, is computationally expensive in its current form. So instead, we can approximate it numerically [29, 42, 46] to reduce computational time. A more efficient evaluation of the Holevo variance can be described as

$$V_H = S^{-2} - 1 = \left| \frac{1}{K} \sum_{k=1}^{K} e^{i[\phi^{(k)} - \theta_N^{(k)}]} \right|^{-2} - 1, \tag{11}$$

where values for $\phi^{(k)} - \theta_N^{(k)}$ close to zero signify lower values of imprecision (sharpness $S \approx 1$) and therefore better performance.

Additionally, the performance V_H of each candidate policy vector is evaluated $M = 5$ separate times and the results averaged in order to smoothen small fluctuations. Repeating the simulation multiple times allows for a more accurate representation of the performance of each policy and thus more consistent results. The number of training instances K can be any arbitrarily large enough number given that it does not hinder the computational time. A large number of training instances K also ensures a faithful representation of ϕ in the interval $[0, 2\pi[$, since they are sampled uniformly and randomly from that same interval. A reasonable choice that satisfies these criteria is $K = 10N^2$, a number sufficiently large to guarantee the convergence of the algorithms [46]. Overall, the time complexity of the algorithms scales with $\mathcal{O}(P \cdot G \cdot N \cdot K \cdot M) \sim \mathcal{O}(N^3)$ which is polynomial in time.

At this point it is possible to completely evaluate the performance of each algorithm according to the different possible configurations of each of their controllable parameters and choose those that achieve better results. While Eq. (10) provides a measurement for

Table 1 Optimal parameters for the DE and PSO algorithms obtained through the analysis described in Appendices A.2 and A.3

Algorithm	F	C	β	α	w	v_{max}
Differential Evolution	0.7	0.8	–	–	–	–
Particle Swarm Optimization	–	–	0.8	0.8	0.8	0.2

the convergence of the algorithms, Eq. (11) shows the precision to which they are able to estimate the value of the unknown phase ϕ.

A thorough study on the performance of both algorithms under the different possible controllable parameter configurations can be found in Appendix A.2 and Appendix A.3. The optimal parameter configuration obtained for each algorithm is summarized in Table 1.

Along with a fixed configuration for each algorithm, it is also important to ensure that all the other variable parameters remain the same in order to draw comparable results. Thus, it is important that for the same number of input qubits N being injected into the system, the population number of candidate solution vectors P and the number of training instances K remain the same under all different scenarios. As previously mentioned, the number of training instances was set to $K = 10N^2$, while the number of populations was defined as $P = 20 + 2 \cdot \text{int}(N/10) - 1$, where $\text{int}(\cdot)$ represents the integer part of the division. Note that for an increasing number of qubits being injected into the system, the population size of candidate solution vectors P and the number of training instances K must also increase to accommodate the increasing complexity of the problem search space.

Ideally, both algorithms would be allowed to run until all the different candidate solution vectors would have converged to a successful policy vector. However, due to time constraints the number of iterations for which both algorithms were allowed to run for each number N of input qubits was set to $G = 100$, regardless of having reached convergence or not. Thus, both algorithms would stop either when they had converged to a solution, or when they reached iteration $G = 100$, and the final policy vector would be the average of all the different candidate solution vectors at that point.

4 Results

In order to provide a benchmark for the two machine learning algorithms discussed above, we first introduce a non-adaptive protocol that can also be run in the presence of noise. This protocol has the advantage of simplicity and we find that for moderate noise values it yields results that are better or comparable with machine learning algorithms. On the other hand, for increased noise values the machine learning protocols show better results.

We start by discussing the ideal configuration where any source of noise or quantum decoherence is neglected, then we consider the Gaussian and Random Telegraph noise configurations and, finally, the visibility loss due to decoherence. These different sources of imperfection were applied to the quantum phase estimation configuration independently. The number of varying qubits used under all the different scenarios was set in the interval $N \in [5, 25]$ and all the remaining parameters were left the same across all the different scenarios. For $N > 20$ we found that the algorithms were already taking more than five days to arrive at a successful policy vector, which ultimately made any larger value of N computationally impracticable under a reasonable amount of time.

The break in performance in non-ideal systems is clearer for larger values of noise, since increasing the value of noise in the estimation process consequently leads to an increased

complexity of the search space for both algorithms. Thus, for larger values of N this added level of complexity becomes even more evident in the scattering of the obtained results. However, it is important to stress that when both algorithms had enough time to converge to a successful policy, they were able to perform with precision values close to the SQL, thus attesting their resilience to noise in the quantum phase estimation scheme.
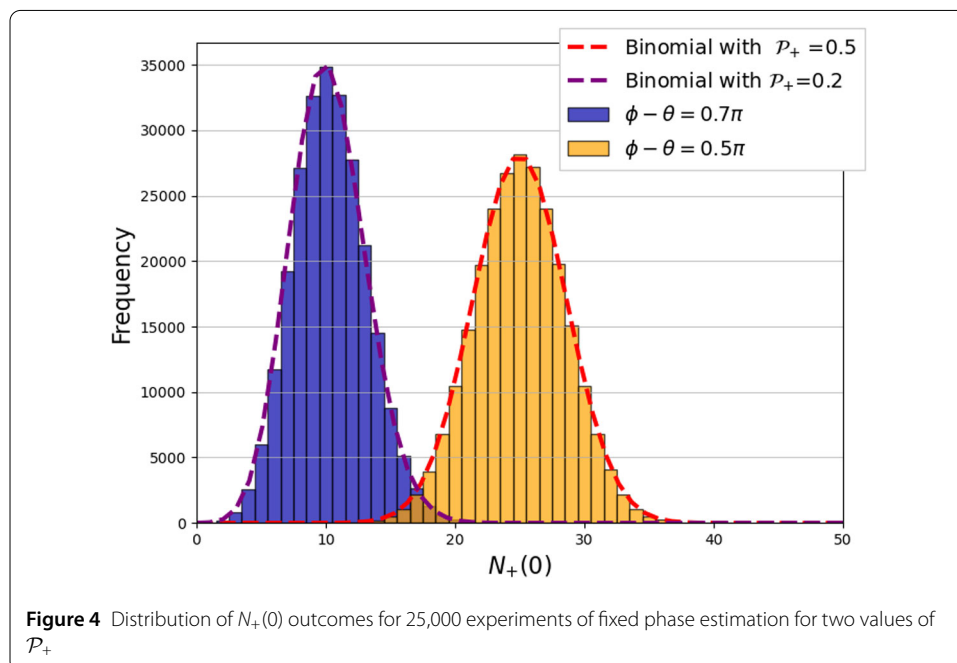
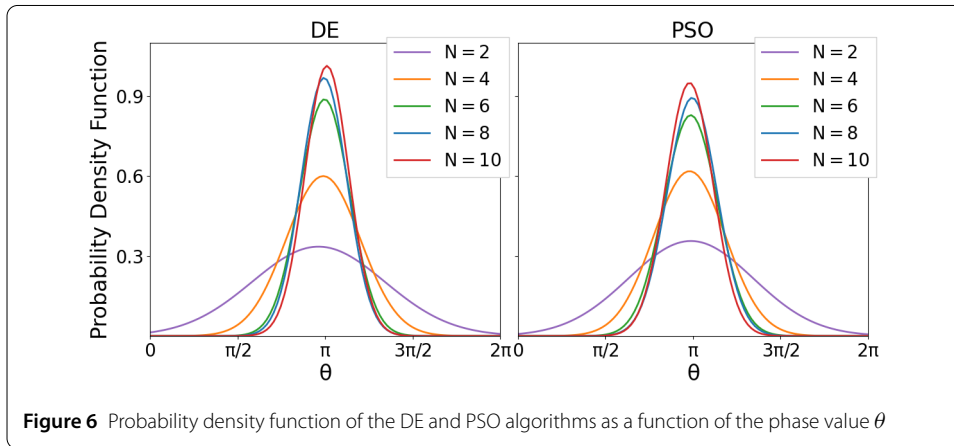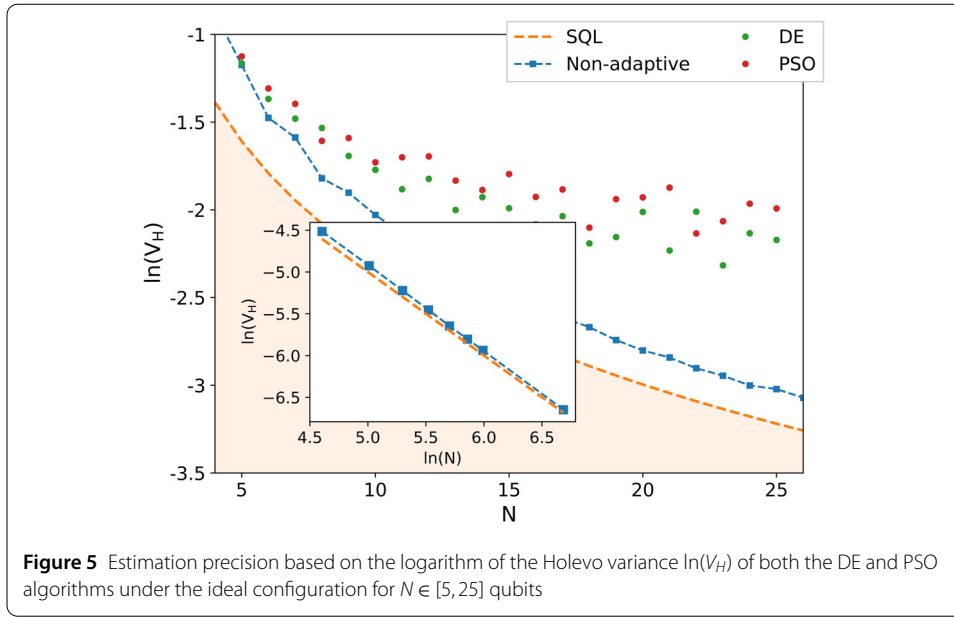### 4.1 A non-adaptive protocol and the standard quantum limit benchmark

The SQL is defined strictly speaking in the limit of large N. Since we do not work in this asymptotic regime, it is important to devise a non-adaptive protocol that reproduces the SQL for $N \gg 1$, yet it yields results also at $N \approx 5 - 25$.

This protocol can be outlined as follows. We consider the random phase that should be estimated ϕ and a fixed control phase θ. Based on these values, we calculate the probability $\mathcal{P}_\pm(0|\phi,\theta)$, see Eqs. (2), (3) or Eq. (9) with $\nu = 1$. Then, for each N we generate N uniformly random numbers in the interval $[0,1]$. If the random number is less than $\mathcal{P}_\pm(0|\phi,\theta)$ we add it into the 0 bin, otherwise we add it to 1. Next, we count the number of elements in the 0 stack, $N_\pm(0)$. Finally, we find an estimation for the phase ϕ^{est}, as $\phi^{\mathrm{est}} = \theta + \arccos(\pm 2N_\pm(0)/N \mp 1)$, and we calculate the Holevo variance with the exponent in Eq. (11) as $\exp[i(\phi - \phi^{\mathrm{est}})]$.

The number of elements $N_+(0)$ and $N_+(1) = N - N_+(0)$ follows the binomial distribution as shown in Fig. 4. This distribution is obtained from two constant phase estimation by 50 measurements, repeated 250,000 times.

This procedure is repeated $K = 10N^2$ times for each phase ϕ uniformly distributed in the interval $[0, 2\pi]$. The resulting Holevo variance is represented by the blue squares in Fig. 5. We have verified numerically that the non-adaptive method reproduces asymptotically the SQL. For example, even at $N = 100$ the difference between simulated V_H and $1/N$ is 10 % and reaches 4 % at $N = 800$. Also, as it is seen from the inset at Fig. 5, the slope for

Figure 4 Distribution of $N_+(0)$ outcomes for 25,000 experiments of fixed phase estimation for two values of $\mathcal{P}_+$

Figure 5 Estimation precision based on the logarithm of the Holevo variance $\ln(V_H)$ of both the DE and PSO algorithms under the ideal configuration for $N \in [5, 25]$ qubits

Figure 6 Probability density function of the DE and PSO algorithms as a function of the phase value θ

this non-adaptive protocol is equal to -1.0265, while for the ideal SQL this slope should be -1.

The same procedure is used to simulate the variance in the presence of different noises, as it is shown below.

4.2 Adaptive protocols in the ideal noiseless configuration

Before analysing the evolution of the performance obtained by the DE and PSO algorithms under a increasing number of qubits N, it is important to show that this increase in N does indeed lead to better estimation values. To do so, the two algorithms were first allowed to converge to a given value of the unknown phase ϕ for different values of N. This experiment was repeated 1000 times for each value of N and, at each time, the value of the estimated phase θ was recorded. Finally, these 1000 different results of θ for each value of N were fitted to a probability density function (PDF) and centered all around the same value $\theta = \pi$ to better compare the results. The results for each algorithm are displayed side by side in Fig. 6.

Considering the results obtained in Fig. 6 it is indeed possible to see that by increasing the number of qubits N at the input level the estimation process leads to values of θ converging further towards the given value of the unknown phase ϕ. It is also possible to see that as N goes higher, this increase in precision starts to be less evident as the algorithms start running into convergence issues. Given that the complexity of the search space of each algorithm scales polynomial in N and that due to time constraints the algorithms were not allowed to run for more than $G = 100$, a decrease in performance under this time limitation for large values of N is expected.

Having arrived at this conclusion, the final results obtained under the ideal scenario, where there is no source of noise or quantum decoherence, are shown in Fig. 5. Observing the results it is possible to see that both algorithms were able to follow the SQL for the ideal-case scenario. It is also possible to reinforce the previous found conclusion that an increase in N leads to better results. In fact, for $N \in [5, 15]$ the scaling of the Holevo variance is well approximated by a power law $V_H \sim N^{-\alpha}$, where $\alpha_{\mathrm{DE}} = 0.75$ for the DE algorithm and $\alpha_{\mathrm{PSO}} = 0.64$ for the PSO algorithm, while the corresponding value for the SQL is $\alpha_{\mathrm{SQL}} = 1$. This also shows that the DE performs slightly better than PSO at reaching low variances, which is consistent with results obtained in other contexts, such as estimating the bias in the quantum walk algorithm [43]. This scaling in precision close to the SQL for $N \in [5, 15]$ is consistent with what has been numerically observed in other algorithms [31].

The fact that the machine learning algorithms do not reach the SQL is also consistent with the known results that the adaptive measurements cannot saturate the Cramér-Rao inequality even for optimal measurements [56, 57]. It is also noticeable that for larger values of N the performance starts to break due to the limiting number of iterations $G = 100$ imposed for the convergence of the algorithms. This is not a malfunction of the algorithms, but a direct consequence of the restricted time available. As the complexity of the search space of the algorithms increases, so does the number of generations required to converge to a successful policy.
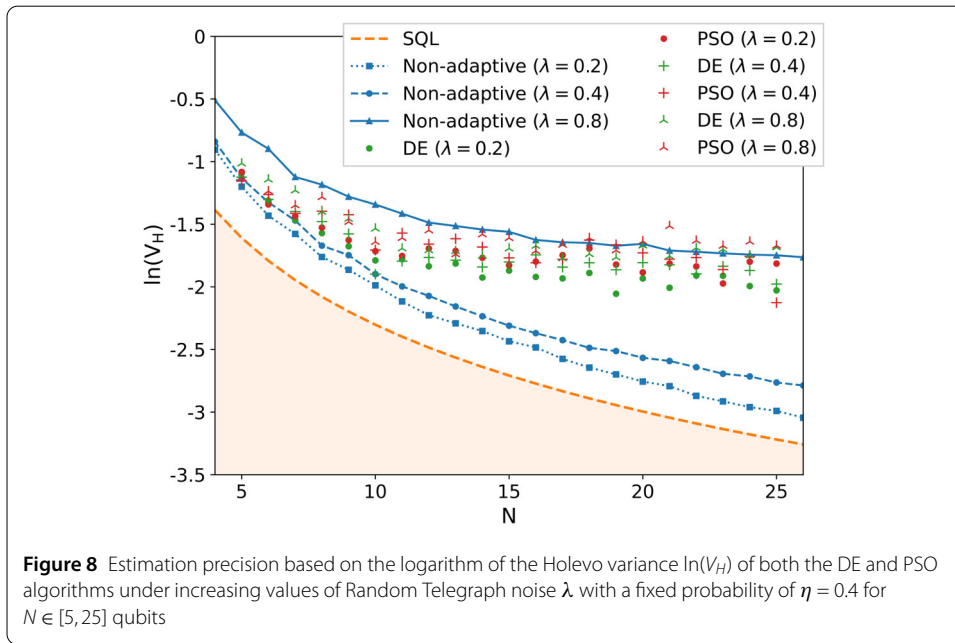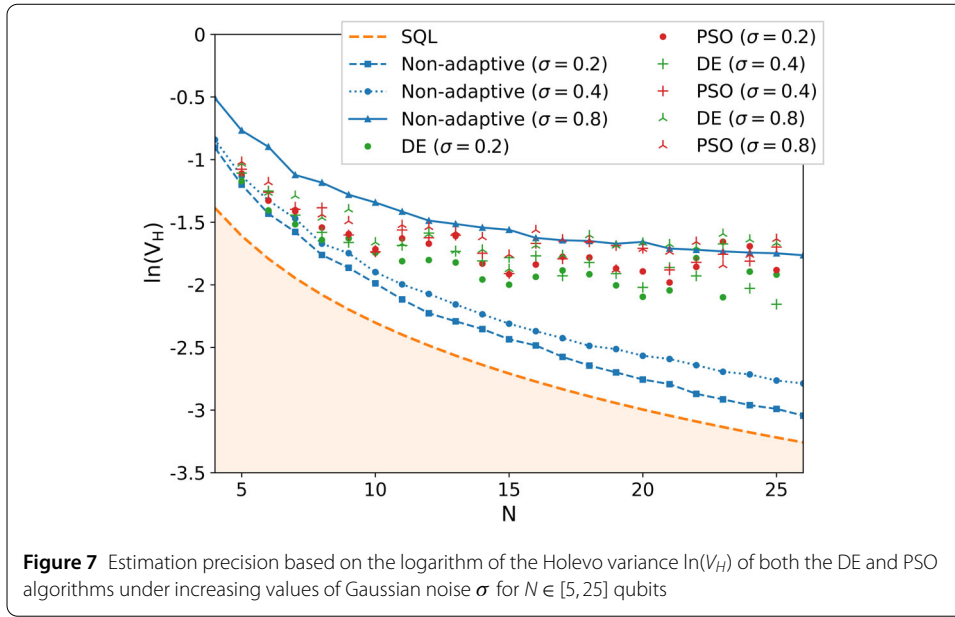
4.3 Configurations with noise

Considering the results obtained under the ideal configuration, it is important to study the resilience of the algorithms to different sorts of imperfections that can be found in an experimental quantum phase estimation scheme. First, the performance of the algorithms was evaluated in the presence of Gaussian noise, followed by Random Telegraph noise and finally in the presence of quantum decoherence.

4.3.1 Gaussian noise

In this scenario the algorithms were tested for increasing amounts of Gaussian noise $\sigma = \{0.2, 0.4, 0.8\}$ when dealing with the controllable phase shifter θ according to Eq. (6). The results obtained under these conditions are presented in Fig. 7.

It is possible to see that as the noise fluctuations increase the precision of both adaptive and non-adaptive algorithms starts to diminish. This is, nevertheless, expected for any estimation process being conducted under increasing values of noise. The break in performance is also perceptible for larger values of N. However, we see from Fig. 7 that the adaptive algorithms are more sensitive to increasing values of Gaussian noise and that for $\sigma = 0.8$ the policies obtained from machine learning become clearly superior to the non-adaptive protocol.

Figure 7 Estimation precision based on the logarithm of the Holevo variance $\ln(V_H)$ of both the DE and PSO algorithms under increasing values of Gaussian noise σ for $N \in [5, 25]$ qubits

Figure 8 Estimation precision based on the logarithm of the Holevo variance $\ln(V_H)$ of both the DE and PSO algorithms under increasing values of Random Telegraph noise λ with a fixed probability of $\eta = 0.4$ for $N \in [5, 25]$ qubits

4.3.2 Random telegraph noise

Considering now the scenario where the estimation scheme was subject to a Random Telegraph noise following Eq. (7), the algorithms were tested against increasing values of $\lambda = \{0.2, 0.4, 0.8\}$ while keeping the probability of switching to the erroneous phase value fixed at $\eta = 0.4$. The results obtained by the algorithms under these configurations are displayed in Fig. 8.

Similarly to the results obtained under the Gaussian noise, the results obtained in Fig. 8 show that both adaptive and non-adaptive algorithms partly follow the SQL curve even in the presence of Random Telegraph noise. The poorer performance for larger values of noise, as well as the break in performance for larger values of N, is also evident under

this scenario for the same reason as before. We can also see that for larger values of λ the machine learning adaptive algorithms are more robust than the non-adaptive algorithm.
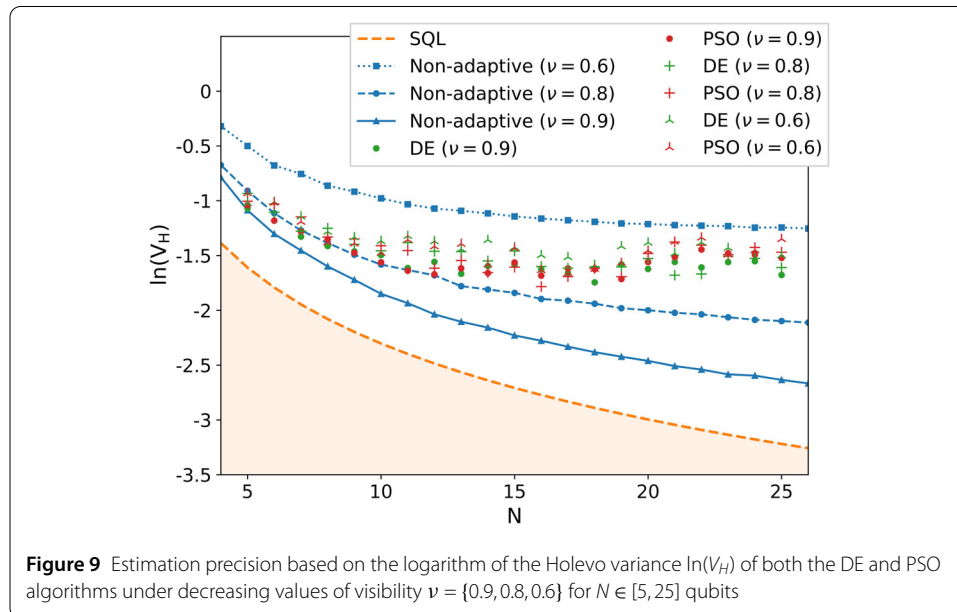
4.3.3 Quantum decoherence

In this last scenario, we study the impact of quantum decoherence in the performance of both algorithms. This is a key source of noise in all non-optical qubits (*e.g.* trapped ions, superconducting qubits, NV centers). The algorithms were tested against decreasing values of visibility $\nu = \{0.9, 0.8, 0.6\}$ according to Eq. (9). Note that unlike the ideal scenario, the reduced visibility also impacts the SQL, see Eq. (18) in Appendix A.1. Also note that the value $\nu = e^{-0.5} \approx 0.6$ appears in certain tasks of non-adaptive parameter estimation as the visibility corresponding to an optimal phase accumulation time τ of half the decoherence time [21]. The results obtained under this configuration are shown in Fig. 9.

Considering the results obtained in Fig. 9, it is evident that reduced values of visibility have a significant impact on the performance of the algorithms. This is an expected behaviour, since it is known that quantum enhancement for systems operated at around the decoherence rate is asymptotically limited to only a constant factor improvement over the SQL [58]. This can be confirmed by an analysis of the Fisher information:

$$\mathcal{F}_{\phi,\theta_m} = \overline{\left[\partial_\phi \ln P_\pm(\zeta_m|\phi,\theta_m)\right]^2} = \sum_{\zeta_m=0,1} \frac{[\partial_\phi P_\pm(\zeta_m|\phi,\theta_m)]^2}{P_\pm(\zeta_m|\phi,\theta_m)} = \frac{\nu^2 \sin^2(\phi-\theta_m)}{1-\nu^2 \cos^2(\phi-\theta_m)}. \quad (12)$$

The Fisher information quantifies the knowledge gained at each measurements. It can also be extended from projective to positive operator-valued measurements, see [59], and it is used to define the SQL (see Appendix A.1). Indeed, from Eq. (12) it can be observed that ν reduces the information extracted after each measurement from its maximum value $\mathcal{F}_{\phi,\theta_m} = 1$ at $\nu = 1$.

Similarly here we observe that relatively large values of decoherence (visibility around 0.6) result in the suppression of precision obtained by the non-adaptive protocol, while the adaptive DE and PSO are less sensitive.

Figure 9 Estimation precision based on the logarithm of the Holevo variance $\ln(V_H)$ of both the DE and PSO algorithms under decreasing values of visibility $\nu = \{0.9, 0.8, 0.6\}$ for $N \in [5, 25]$ qubits

5 Experimental implementations

Summing up the theoretical results obtained so far, we can see that even for noisy systems it is possible to reduce the Holevo variance to values in a band of typically $[-1.2, -1.7]$ (standard deviations between 0.5 and 0.4) with a moderate numbers of $N \in [10, 25]$ qubits. This protocol can be implemented on several experimental platforms, using either optical or microwave frequencies. We describe here the main experimental platforms and we show how the problem of optimizing the precision in the presence of noise can be addressed using the machine learning framework.

The most straightforward implementation is optical Mach–Zehnder interferometry. In this case, the operator $U_{\theta - \phi}$ can be realized by placing a phase shifter ϕ in one branch of the interferometer and a variable-phase shifter θ in the other branch. The latter can be realized physically as a half-wave plate placed on a rotation stage controlled by a computer [60]. The states $|0\rangle$ and $|1\rangle$ correspond to a single photon in one branch or the other of the interferometer (dual-rail encoding). Our results can be compared also with those obtained from running the more powerful Kitaev algorithm, which for the same number of resources (from 10 to 25) results in standard deviations ranging from 0.35 to 0.16 [60], only marginally better than the data reported here. Also, our results are consistent with the theoretical limits reported in [61] for the case of non-entangled initial states ($V_H = 0.5609$, $\ln V_H = -0.25$). We obtain approximately a factor of 2 improvement in the standard deviation. For implementations employing optical interferometry, the visibility is typically very close to 1, and the main noise sources in the feedback loop concern changes in the refractive index of the beam-splitters and variable phase shifter, as well as variations in the optical paths typically caused by temperature.

More recently, an experiment with optical photons has tested the PSO algorithm with sequential non-entangled photons [48]. The single photons were obtained non-deterministically, by generating entangled pairs through spontaneus parametric down-conversion in a 2 mm long beta-barium borate (BBO) crystal and heralding over detection events in another detector. The unknown and the control phases in the two arms of the Mach–Zehnder interferometer were realized with liquid crystal devices, where the relative phase between the horizontal and vertical polarization can be changed depending on the applied electric field. The results with the PSO algorithm obtained agree with our analysis: PSO is quite efficient at reaching values approaching the SQL, especially when the number of resources (number of photons N) is limited. Similarly to our findings, for N exceeding approximately 15 photons the Holevo variance tends to saturate. The robustness with respect to Gaussian phase noise and dephasing noise was also demonstrated by artificially adding errors to the feedback phase.

In the case of qubits based on discrete levels (solid-state and ions/atoms), Mach–Zehnder interferometry corresponds to Ramsey interference [62, 63]. To understand how the phase information is embedded in this system, consider the following generic qubit Hamiltonian driven by a microwave field [64],

$$\mathcal{H} = \frac{\hbar}{2}\omega_{01}\sigma_z + \hbar\Omega\cos\omega t\,\sigma_x. \tag{13}$$

In a frame defined by the unitary $\exp(i\omega t\sigma_z)$ (a clockwise rotation around the z-axis), using $\exp(i\omega t\sigma_z)\sigma_x\exp(-i\omega t\sigma_z) = \sigma_x\cos\omega t - \sigma_y\sin\omega t$ and by further applying the rotating-wave

approximation we get the effective Hamiltonian

$$\mathcal{H} = \frac{\hbar}{2}(\omega_{01} - \omega)\sigma_z + \frac{\hbar\Omega}{2}\sigma_x. \tag{14}$$

A non-zero Rabi frequency $\Omega \neq 0$ can be thus used to create the two Hadamard gate, while the time in-between is spend for the actual sensing. Indeed, if $\Omega = 0$ for a sensing time τ, the resulting unitary becomes

$$U_{\phi,\theta} = \begin{bmatrix} 1 & 0 \\ 0 & e^{i(\omega_{01}-\omega)\tau} \end{bmatrix}, \tag{15}$$

which is exactly Eq. (1) up to an overall irrelevant phase and the identification $\phi = \omega_{01}\tau$, $\theta = \omega\tau$.

Consider now a concrete problem, that of evaluating the magnetic field using a superconducting qubit, a trapped ion, or a nitrogen-vacancy (NV) center in diamond. In these cases, for typical experiments using Ramsey interferometry, the probability is given by Eq. (9), where the dependence of the visibility v on the sensing time τ can be sometimes different from a simple exponential decay $v(\tau) = \exp[-\tau/T_2]$, valid for the case of Gaussian noise (see the Lindblad equation used in Sect. 2). For example, another dependence is $v(\tau) = \exp[-(\tau/T_2)^2]$ if the noise experienced by the qubit is 1/f, see e.g. [40, 63, 64]. Since τ is bounded by T_2, in these setups one might attempt to increase the precision by increasing the sensitivity of ω_{01} to the magnetic field. However, this means increasing the coupling to the magnetic field, which at the same time this will increase the exposure of the qubit to noise. Thus, a tradeoff must be reached between these two competing effects. Our results demonstrate that the increase in noise can be mitigated successfully by the use of machine learning strategies.

In the case of a superconducting qubit in the symmetric transmon design as used in recent magnetometry experiments [63, 65, 66], the information about magnetic field is embedded in the energy level separation as

$$\omega_{01}(B) = \frac{1}{\hbar}\left(\sqrt{8E_C E_{J\Sigma} \cos\left|\pi\frac{BS}{\Phi_0}\right|} - E_C \right), \tag{16}$$

where B is the magnetic field piercing the SQUID area S of the transmon and modulating the total Josephson energy $E_{J\Sigma}$. In order to evaluate B, we can keep τ fixed and adjust the frequency ω (generated by an external signal generator) at every step. In the case of superconducting qubits, with relatively standard values $T_2 = 10\,\mu$s and $\tau = 1\,\mu$s we obtain $v = 0.9$ (one of the values used in Sect. 4) if the noise is Gaussian and $v = 0.99$ if it is 1/f noise.

In order to increase the precision of determining the magnetic field, we can use a higher excited state [65, 67]—for example, the second excited state $|2\rangle$, and considering the harmonic approximation for the transmon, the relevant accumulated phase will be $\approx 2\omega_{01}\tau$; or we can bias the transmon to a magnetic field value where the slope $d\omega_{01}/dB$ is larger. Both situations result in an increase in the noise. In the first case, this is due to the fact that higher energy levels have higher couplings to the electromagnetic environment. This causes an increase in T_2, cause due to both an increase in the T_1 time and an increase in

the pure dephasing time. For example, if the second excited state is used, the decay rate of the $1 - 2$ transition is twice that of the $0 - 1$ transition, which for T_1-limited qubits results in a significant reduction of the interferometric visibility. In the second situation, due to biasing far from the sweet point. In the latter case, let us restrict ourselves for simplicity to the interval $BS \in [-\Phi_0/2, \Phi_0/2]$. We then have

$$\frac{d\omega_{01}}{dB} = -\frac{\pi S \omega_{01}}{\Phi_0} \tan\left(\pi \frac{BS}{\Phi_0}\right).$$

This slope is infinite for $BS = \pm\Phi_0/2$, apparently allowing us to measure B with an infinite precision. However, the displacement of the bias point from the sweet point is accompanied by a significant increase in noise, since the qubit is no longer protected against linear-oder fluctuations. This results again in a visibility ν below unity.

The noises in the control phase θ_m are typically not caused by electronics in the feedback look, but they result from uncontrollable frequency shifts that are poorly understood and controlled. Experimentally, these shifts are of the order of a few MHz. If τ is of the order of microseconds, then the resulting values of λ and σ are well below those considered in this work. Therefore, this type of noise will not affect the performance of our protocols when run on a superconducting qubit.

In the case of trapped ions, sensitive magnetometry using the well-known ^{171}Yb$^+$ ion has been demonstrated [68]. This uses four hyperfine states, $|F = 0, m_F = 0\rangle$, $|F = 1, m_F = 1\rangle$, $|F = 1, m_F = -1\rangle$, and $|F = 1, m_F = 0\rangle$ belonging to the $^2S_{1/2}$ manifold. The latter three states are degenerate and they are separated by the hyperfine splitting $\omega_{\text{hf}}/(2\pi) = 12.642$ GHz from the first state. The degeneracy of these three states can be lifted by the application of a magnetic field. In the first order in magnetic field, the state $|F = 1, m_F = 0\rangle$ remains unmodified but $|F = 1, m_F = \pm 1\rangle$ acquires frequency shifts of $\pm(g_e\mu_B/2\hbar)B$, where $g_e \approx 2$ is the g-factor of the electron and μ_B is the Bohr magnetron. Thus, for magnetic field detection one could in principle use the state $|F = 0, m_F = 0\rangle$ and either of the magnetic-sensitive states $|F = 1, m_F = \pm 1\rangle$ and drive resonant Ramsey $\pi/2$ microwave pulses at around 12 GHz with τ time separation. Then the information about magnetic field is obtained from the phase $\phi = (\omega_{\text{hf}} \pm g_e\mu_B B/2\hbar)\tau$. These states would be exposed not only to the magnetic field that we would like to sense, but also to magnetic field noise, making our results for the noisy case relevant. Further improvements may be achieved by the use of a continuous dynamical decoupling technique, where one could identify the $|F = 1, m_F = 0\rangle \equiv |0\rangle$ and the dark state $\frac{1}{\sqrt{2}}(|F = 1, m_F = -1\rangle + |F = 1, m_F = 1\rangle) \equiv |1\rangle$ as a dressed-qubit states, with a T_2 time exceeding one second [69], three orders of magnitude more than the bare atomic states which is a few miliseconds.

Similar ideas can be applied to NV centers in diamond. These defects have very long decay times, of the order of tens of miliseconds, and total decoherence times are of the order of microseconds and can be extended to hundreds of microseconds by the use of dynamical decoupling pulses [70]. Single NV centers have a triplet $S = 1$ ground-state structure, with the states $m_S = 0$ and $m_S = \pm 1$ separated by the so-called zero-field splitting $D = 2.87$ GHz. By applying a magnetic field along the crystal axis, the levels $m_S = \pm 1$ can be further split by the Zeeman effect [71]; the resulting energy level difference between these levels is $2g_e\mu_B B/\hbar = 2\gamma_e B$ where μ_B is the Bohr magnetron and g_e is the electronic g-factor, $g_e \approx 2$. The gyromagnetic ratio γ_e is defined as $\gamma_e = g_e\mu_B/\hbar$. Because the triplet states can be selectively addressed with microwaves [71] we can immediately identify for

example $|0\rangle = |m_S = 0\rangle$ and $|1\rangle = |m_S = 1\rangle$ as the basis corresponding to our formalism, and we obtain $\omega_{01} = 2\pi D + \gamma_e B$. The encoding of the magnetic field in frequency and subsequently in the phase is linear, $\Phi = (2\pi D + \gamma_e B)\tau$.

To summarize, each of these experimental systems involved relatively well characterized imperfections, and an evaluation of our protocols for any specific setup can be realized by inspecting the general results obtained for various types of noise in Sect. 4. We find that the precision can be optimized by using machine learning protocols to mitigate the noise resulting from the increase in sensitivity. We also note that this tradeoff is expected to occur also if machine learning algorithms are included as subroutines in protocols that use highly entangled states to increase the precision, since highly entangled states are also very susceptible to noise.

6 Conclusion

The objective of this work was to study the performance of machine learning algorithms, namely the DE and PSO algorithm, in a quantum phase estimation scheme and determine their ability to come close to the SQL without resorting to multi-particle entanglement. To this end, both algorithms were tested against different configuration scenarios and were able to follow the SQL curve up to a given value of input qubits N. Under the constraint of $G = 100$ it was possible to notice that the algorithms start to loose performance for larger values of N. This becomes even more relevant in the scenario with quantum decoherence. However, it is important to reiterate that this is not a deficiency of the algorithms, but a direct consequence of the time and computational resources available.

These limitations can be overcome in future works by optimizing the code and making it more time efficient in order to allow both algorithms a larger number of iterations before converging to a valid solution. An immediate improvement would be to fully vectorize the code which would make it significantly faster. Another direct improvement would be to parallelize the code so that it could leverage the power of stronger computers with higher number of cores. This would allow for the different independent simulation threads of both algorithms to be conducted in parallel, making the estimation process even more time efficient. One can also explore recurrent neural networks for the quantum phase estimation task, as they are particularly well suited for regression-based prediction models on time series events. It would be interesting to see how architectures such as the Long Short-Term Memory (LSTM), the Gated-Recurrent Unit (GRU) and the Temporal Convolutional Networks (TCN) would compare against our reinforcement learning algorithms.

Overall, we have shown that machine learning algorithms can provide noise-robust policies and we benchmarked their performances for various types of noise in close connection to experimental realizations. Above a certain critical value of noise, we found that machine learning based protocols show better results than non-adaptive methods. This opens a door to future developments that may push even further the precision of quantum phase estimation with classical machine learning algorithms without resorting to preparation and measurement of highly entangled states.

Appendix
A.1 Theoretical background

Probabilities   Here we present analytical results related to the calculation of the Holevo variance. Given a qubit fed into the adaptive scheme with the adjustable phase set to θ_m

and the unknown phase ϕ at step m, the probability of obtaining the result ζ_m is given by Eq. (9)

$$P_\pm(\zeta_m|\phi,\theta_m) = \frac{1}{2} \pm \frac{\nu}{2}(-1)^{\zeta_m}\cos(\phi - \theta_m), \tag{17}$$

where the + and − signs correspond to the initial qubit in the state $|0\rangle$ respectively $|1\rangle$, and ν is the visibility, $\nu \in [0,1]$.

To simulate numerically $P_\pm(\zeta_m|\phi,\theta_m)$ we draw a random number from the interval $[0,1]$. If this number is smaller than $P_\pm^{(\nu)}(0|\phi,\theta_m)$, then the result ζ_m of the measurement is recorded as 0; if it is larger, it is recorded as 1. With these notations, we can understand easily the origin of the SQL. Indeed, the information about phase is contained in the probabilities Eq. (17). However, the evaluation of this probabilities is affected by measurement projection noise.

Standard quantum limit   Suppose now that we try a simple non-adaptive strategy where we just repeat the Ramsey experiment for N times with an initial states $|0\rangle$ and the same control phase θ, selecting the results with $\zeta = 1$. The precision $\delta\phi$ that we can achieve is given by $\delta P_+(1|\phi,\theta) = (1/2)\nu\sin(\theta - \phi)\delta\phi$. To achieve a small $\delta\phi$ it is advantageous to measure at phase parameters where the sine in this expression is maximal, that is, at around $P_+(1|\phi,\theta) = 1/2$. At the same time, the measurement results have a binomial distribution, with $[\Delta P_+(1|\phi,\theta)]^2 = (1/N)P_+^{(\nu)}(1|\phi,\theta)[1 - P_+(1|\phi,\theta)]$, which results in $\Delta P_+(1|\phi,\theta) = 1/(2\sqrt{N})$ around the region of maximal sensitivity $P_+(1|\phi,\theta) = 1/2$. In order to obtain a signal-to-noise ratio of at least 1, the uncertainty noise should be at most the same as the signal, $\delta P_+(1|\phi,\theta) \geq \Delta P_+(1|\phi,\theta)$, which results in

$$\delta\phi \geq (\Delta\phi)_{\text{SQL}} = \frac{1}{\nu\sqrt{N}}. \tag{18}$$

Thus, the precision is limited by the SQL. As expected, a small visibility $\nu \neq 1$ results in an increased $(\Delta\phi)_{\text{SQL}}$, corresponding to a loss in precision.

We can make this argument more rigorous by using the Cramér-Rao bound,

$$\delta\phi \geq \frac{1}{\sqrt{N\mathcal{F}_{\phi,\theta}}} = \frac{1}{\sqrt{N}}\frac{\sqrt{1 - \nu^2\cos^2(\phi - \theta)}}{\nu|\sin(\phi - \theta)|}. \tag{19}$$

The maximum value of $\mathcal{F}_{\phi,\theta}$ is obtained for $\phi - \theta = \pi/2, 3\pi/2$ and equals ν^2. We then obtain the relation Eq. (18). An alternative derivation consists of calculating the fluctuations of the $\Sigma_z = \sum_{i=1}^{N}\sigma_z$ operator, namely $(\Delta\Sigma_z)^2 = \langle\Sigma_z^2\rangle - \langle\Sigma_z\rangle^2 = N(\langle\sigma_z^2\rangle - \langle\sigma_z\rangle^2)$, reflecting the fact that the fluctuations of the σ_z's are uncorrelated. The average $\langle\sigma_z\rangle = P_+(0|\phi,\theta) - P_+(1|\phi,\theta)$, and using the error propagation rule

$$\Delta\phi = \frac{\Delta\sigma_z^N}{|\partial_\phi\langle\sigma_z^N\rangle|}, \tag{20}$$

we get the same result as in Eq. (19).

Adaptive measurements   In an adaptive measurement, the control phase is adjusted depending on the previous measurement values. Here, in order to simplify the notation, we will not write anymore the $\pm$ subscript for probabilities. Then, the probability of a sequence $\vec{\zeta}_m \equiv \{\zeta_1, \zeta_2, \dots, \zeta_m\}$ of results given the phase adjustments $\vec{\theta}_m \equiv \{\theta_1, \theta_2, \dots, \theta_m\}$ is obtained as

$$P(\vec{\zeta}_m|\phi,\theta) = P(\zeta_m|\phi,\theta_m)P(\zeta_{m-1}|\phi,\theta_{m-1})\cdots P(\zeta_1|\phi,\theta_1). \tag{21}$$

From the Bayes' theorem, we have at the m'th measurement

$$P(\phi|\vec{\zeta}_m,\vec{\theta}_m) = \frac{1}{P(\vec{\zeta}_m)}P(\vec{\zeta}_m|\phi,\vec{\theta}_m)P(\phi). \tag{22}$$

The prior $P(\phi)$ is taken uniformly distributed $P(\phi) = 1/(2\pi)$ and the marginalization $P(\vec{\zeta}_m) = \int_0^{2\pi} d\phi P(\vec{\zeta}_m|\phi,\vec{\theta}_m)P(\phi)$.

To further understand how the Holevo variance changes at every step of the algorithm, we present a recursive calculation procedure based on the Fourier transform. We start by examining the sharpness appearing in the Holevo variance for a given depth m,

$$S = \left| \int_0^{2\pi} d\phi e^{i\phi} P(\phi|\vec{\zeta}_m,\vec{\theta}_m) \right|. \tag{23}$$

From Eq. (22) and Eq. (17) we notice that the probability density is a harmonic function of $\phi, 2\phi, 3\phi, \dots$ therefore we can use the Fourier transform

$$P([k]|\vec{\zeta}_m,\vec{\theta}_m) = \frac{1}{2\pi} \int_0^{2\pi} d\phi e^{-ik\phi} P(\phi|\vec{\zeta}_m,\vec{\theta}_m), \tag{24}$$

where for clarity we use $[k]$ as the index of the k'th Fourier component and with inverse given by

$$P(\phi|\vec{u}_m,\vec{\theta}_m) = \sum_{k=-\infty}^{\infty} e^{ik\phi} P([k]|\vec{u}_m,\vec{\theta}_m). \tag{25}$$

From Eq. (23) we get

$$S = \left| 2\pi P([k=-1]|\vec{\zeta}_m,\vec{\theta}_m) \right|; \tag{26}$$

in other words the Holevo variance can be seen as the $k=-1$ Fourier coefficient of the probability distribution. Moreover, the expectation value of ϕ is obtained from the average of $e^{i\phi}$

$$\langle\phi\rangle = \arg\langle e^{i\phi}\rangle = 2\pi \arg P([k=-1]|\vec{\zeta}_m,\vec{\theta}_m). \tag{27}$$

The Fourier coefficients can be calculated recursively. To see this, we apply the Bayes rule Eq. (22)

$$P(\phi|\vec{\zeta}_m,\vec{\theta}_m) = \frac{1}{P(\vec{\zeta}_m)}P(\zeta_m|\phi,\vec{\theta}_m)P(\vec{\zeta}_{m-1}|\phi,\vec{\theta}_m)P(\phi)$$

$$= P(\phi|\vec{\zeta}_{m-1},\vec{\theta}_{m-1})P(\vec{\zeta}_m|\phi,\vec{\theta}_m)\frac{P(\vec{\zeta}_{m-1})}{P(\vec{\zeta}_m)},$$

where the last fraction is a normalization factor. Thus, if we consider non-normalized probabilities p, we have the Bayes relation $p(\phi|\vec{\zeta}_m,\vec{\theta}_m) = p(\phi|\vec{\zeta}_{m-1},\vec{\theta}_{m-1})p(\vec{\zeta}_m|\phi,\vec{\theta}_m)$, from which we find the Fourier recurrence relation

$$p\big([k]|\vec{\zeta}_m,\vec{\theta}_m\big) = \frac{1}{2}p\big([k]|\vec{\zeta}_{m-1},\vec{\theta}_{m-1}\big)$$
$$+ \frac{1}{4}(-1)^{\zeta_m}\nu\big[e^{-i\theta_m}p\big([k-1]|\vec{\zeta}_{m-1},\vec{\theta}_{m-1}\big)$$
$$+ e^{i\theta_m}p\big([k+1]|\vec{\zeta}_{m-1},\vec{\theta}_{m-1}\big)\big]. \tag{28}$$

To recover the correct normalization we simply impose $\int_0^{2\pi} d\phi P(\phi|\vec{\zeta}_m,\vec{\theta}_m) = 2\pi P([k = 0]|\vec{\zeta}_m,\vec{\theta}_m) = 1$, therefore we identify the normalization factor as $2\pi p([k = 0]|\vec{\zeta}_m)$, with $P([k]|\vec{\zeta}_m,\vec{\theta}_m) = p([k]|\vec{\zeta}_m,\vec{\theta}_m)/(2\pi p([k = 0]|\vec{\zeta}_m,\vec{\theta}_m))$.

However, even with the use of these analytical results the evaluation of probabilities would require significant computational resources especially at large N. The reason is that, as it is clear from Eq. (23), the evaluation needs to be done for all 2^N vectors ζ_N, requiring exponentially more resources as N increases.

A.2 Differential evolution

To study the convergence of the DE algorithm, two parameter control tests were done varying the controllable parameters of interest, F and C, while keeping all the other values constant. The tests were conducted for $N = 10$ qubits, $P = 20$ populations, $G = 50$ generations and $K = 1000$ training instances. All sources of noise were neglected during the test, since they are not needed to study the overall performance of the different controllable parameters configurations and would only slow down the learning process.

The first parameter control test concerned the ability of the algorithm to converge to a solution, whether it was a valid one or not, with different possible parameter configurations. Referring to Eq. (10) to evaluate the convergence of the algorithm the results obtained are displayed in Table 2.

Inspecting the results in Table 2 it is possible to find a pattern where only the configurations where the amplification constant F is strictly smaller than the crossover constant C lead to the convergence of the algorithm within the given number of generations. In other words: $F < C$ guarantees the convergence of the DE algorithm. This newfound rule can be

Table 2 Convergence values $L(F, C)$ for the DE algorithm. Values in bold represent configurations where convergence was achieved. Convergence was considered only for values of $L \leq 0.1256$, which corresponds to a maximum dispersion of approximately 2% of the entire 2π search space for each entry of the different candidate solution vector

F	C					
	0	0.2	0.4	0.6	0.8	1
0	1.0166	**0.0696**	**0.0261**	**0.0224**	**0.0019**	**0.0010**
0.2	0.9334	0.3726	**0.0326**	**0.0116**	**0.0204**	**0.0094**
0.4	1.0897	0.7491	0.7231	**0.0235**	**0.0202**	**0.0269**
0.6	1.1647	0.8364	0.7764	0.5226	**0.0251**	**0.0193**
0.8	1.2183	1.1356	0.9333	1.0901	1.0133	**0.0691**
1	1.3101	1.2460	1.2433	1.3081	1.1359	1.2798

Table 3 Performance results based on the logarithm of the Holevo variance $\ln(V_H(F,C))$ for the DE algorithm. Values in bold represent the configurations which have achieved convergence in Table 2. Lower variance values are obtained by solutions that lead to more precise estimations of the unknown parameter that is being measured

F	C					
	0	0.2	0.4	0.6	0.8	1
0	−0.3735	**−1.2012**	**−0.7845**	**−0.7758**	**−0.7083**	**−0.4196**
0.2	5.2111	−1.6266	**−1.3737**	**−1.4401**	**−1.0244**	**−1.0948**
0.4	−0.3897	−0.2398	0.6181	**−1.5613**	**−1.2976**	**−1.5611**
0.6	0.3131	−1.1896	−1.1589	−1.3696	**−1.8321**	**−1.5453**
0.8	0.6326	−1.0395	0.3694	−0.6645	0.8763	**−1.6784**
1	−0.1786	1.2460	2.5564	0.6131	4.2364	0.0864

understood from a physical interpretation of the process. Larger values for the crossover parameter C encourage the different candidate solutions to experience more configurations, thus covering a larger area of the entire search space. This increased exploration of the search space, leads to a more exhaustive pursuit of the best policy solution. However, by increasing the amplification constant F, the step of this search process is also increased. This may lead the algorithm to overlook possibly desirable candidate solutions, which ultimately may not allow the algorithm to converge. Hence, smaller values of F for larger values of C ensure a more careful examination of the search space at each iteration of the algorithm.

The second parameter control test was concerned with the ability of the algorithm to converge to a valid a solution based on Eq. (5). Recalling that lower values of the Holevo variance correspond to candidate solutions with a higher degree of accuracy and keeping exactly the same configuration setup as before, the results obtained are shown in Table 3.

It is possible to see in Table 3 that the parameter configurations that previously lead the algorithm to converge to a solution also lead the algorithm to converge to a valid solution if and only if $F > 0$. Hence, the previously found rule can be made more specific: $F < C$ $\forall F > 0$ guarantees the convergence of the DE algorithm to a valid solution. Note, however, that the algorithm also obtained good performance results even with configurations that did not follow the previously found rule. These are misleading results and can be quickly disregarded, since the algorithm is told to stop regardless of having converged to a solution or not. Therefore, in these situations the resulting averaged solution among all the populations may haphazardly coincide with a valid one. These situations must, nevertheless, be regarded as fortuitously events and not be considered as possible valid solutions as the algorithm was not indeed able to converge to any solution.

Having arrived at the conclusion that F must be strictly smaller than C while still being bigger than zero for the DE algorithm to work as desired, it is important to find the exact parameter configuration that guarantees the most promising results. A more thorough control test regarding the performance of the algorithm for each configuration following this rule was conducted three times and its results averaged and presented in Fig. 10. Note that this time, the step between the different parameter configurations was reduced to avoid overlooking possible optimal parameter configurations.

Looking at the results in Fig. 10 it is possible to notice that the best results are obtained among the configurations where C is only slightly bigger than F. It is also possible to notice that these results seem to be stronger for values of F in the interval $[0.3, 0.7]$. Ensuring that the step size at each iteration of the algorithm is not too big as to overstep a possible

Figure 10 Visual representation of the performance values obtained for the DE algorithm based on the logarithm of the Holevo variance $\ln(V_H(F,C))$ with the parameter configurations following the rule $F < C$ $\forall F > 0$. Better configurations lead to lower values of $\ln(V_H(F,C))$
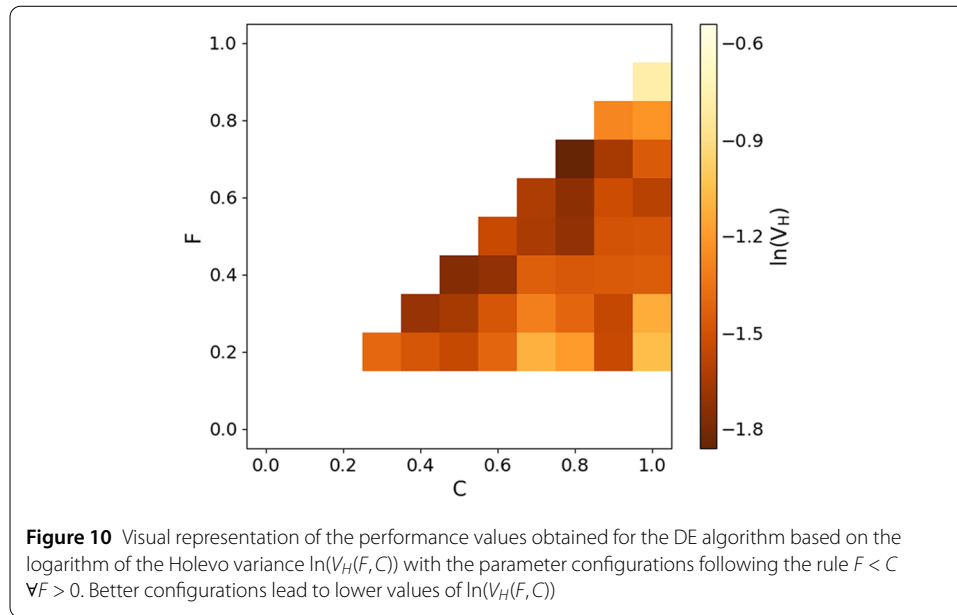
Table 4 Convergence values $L(\alpha, \beta)$ for the PSO algorithm. Convergence was considered only for values of $L \leq 0.1256$, which corresponds to a maximum dispersion of approximately 2% of the entire 2π search space for each entry of the different candidate solution vector

α	β					
	0	0.2	0.4	0.6	0.8	1
0	1.3231	1.3292	1.3139	1.3341	1.4045	1.3105
0.2	**0.1239**	**0.1072**	0.2193	0.2379	0.4736	0.4580
0.4	**0.1014**	**0.0998**	**0.0974**	0.1742	0.2328	0.4280
0.6	**0.0771**	**0.0788**	**0.0808**	**0.0902**	0.1296	0.2410
0.8	**0.0821**	**0.0821**	0.1072	**0.0734**	**0.0806**	0.1589
1	**0.0835**	**0.0976**	0.1008	**0.0887**	**0.0708**	**0.0933**

valid solution, it should not be too small either to avoid getting stuck in a local minimum. This explains why these values of F lead to better results. Among these parameter configurations, the ones where the value of C is in the interval $[0.4, 0.8]$ and is only slightly bigger than F lead to the overall best results. Hence, the optimal parameter configuration considered for the DE algorithm is: $F = 0.7$ and $C = 0.8$.

A.3 Particle swarm optimization

The PSO algorithm has four controllable parameters of interest. Two of these parameters, α and β, are mostly concerned with the ability of the algorithm to converge to a valid solution, while the other two, w and $v_{\max}$, regulate the speed at which the parameter will converge to a solution.

As before, the first step is to study the overall ability of the algorithm to converge to a solution with the different possible parameter configurations. Keeping in mind that only the parameters α and β have a direct impact on the convergence process, all the remaining parameters were set to a constant value and the results obtained are presented in Table 4.

Considering the results obtained, it is possible to see that as long as $\beta > 0$ the algorithm is able to converge to a solution, even if it may take more than $G = 50$ iterations. This can be intuitively understood by remembering that the β parameter defines the desirability of

Table 5 Performance results based on the logarithm of the Holevo variance $\ln(V_H(\alpha, \beta))$ for the PSO algorithm. Values in bold represent the configurations which have achieved convergence in Table 4. Lower variance values are obtained by solutions that lead to more precise estimations of the unknown parameter that is being measured

α	β					
	0	0.2	0.4	0.6	0.8	1
0	1.3082	2.3717	0.7076	1.5727	1.7540	6.0563
0.2	**−1.5475**	**−1.3651**	−1.6623	−1.4530	−0.8383	−1.0281
0.4	**−1.4016**	**−1.5598**	**−1.5295**	−1.6831	−1.6005	1.2126
0.6	**−1.6229**	**−1.5110**	**−1.9324**	**−1.3650**	−1.6605	−1.5408
0.8	**−1.7227**	**−1.4539**	**−1.5615**	**−1.5828**	**−1.8361**	−1.2132
1	**−1.6869**	**−1.3290**	**−1.3956**	**−1.6726**	**−1.5909**	**−1.4826**

each individual to follow the best found solution of the entire group until that point. Thus, as long as this value is bigger than zero, each individual will feel the urge to move towards the best globally found solution. In other words: $\beta > 0$ guarantees the convergence of the PSO algorithm.

Additionally, it is also possible to notice that for values of β equal or larger than α the algorithm is always able to converge to a solution within the stipulated number of iterations. This leads to a second conclusion: $\beta \geq \alpha \ \forall \beta > 0$ guarantees the convergence of the PSO algorithm within the given number of iterations. However, for β values much larger than α the algorithm converges within very few iterations as it gets stuck in local minima. These configurations privilege moving towards the already found best global solution in detriment of exploring other new possible solutions.
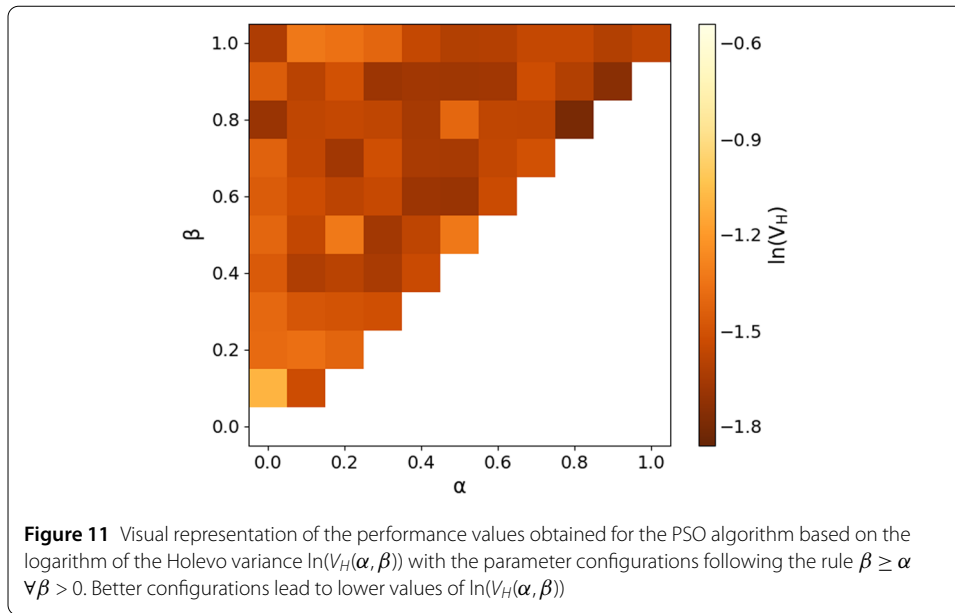
Keeping the same configurations as before, it is necessary to study which of these parameter configurations lead to valid solutions. Referring to Eq. (5) to evaluate their performance, the results obtained are shown in Table 5.

Considering the previously found rule and the results obtained in Table 5 is possible to see that for $\beta \geq \alpha$ while keeping $\beta > 0$ not only guarantees the convergence of the algorithm, but also guarantees the convergence of the algorithm to a valid solution. Therefore, it is possible to come to the conclusion that: $\beta \geq \alpha \ \forall \beta > 0$ guarantees the convergence of the PSO algorithm within the given number of iterations to a valid solution.

Having arrived to this conclusion it is still important to determine which parameter configuration leads to the best results. As before, a more thorough control test on the performance of the algorithm under these conditions was conducted three times and its results averaged and displayed in Fig. 11. Notice that, once again, there is a smaller increment between each parameter step in order to avoid overlooking any possibly better parameter configuration.

Unlike with the DE algorithm, the results obtained for the PSO algorithm are less contrasting. Even though configurations where the value of α and β are similar to each other deliver slightly better results, it appears that having $\beta \geq \alpha \ \forall \beta > 0$ is a condition strong enough to arrive at optimal candidate solutions. This robustness of the algorithm against different possible parameter configurations would have allowed for a multiple number of pair of configurations to be chosen. Therefore, the first two optimal controllable parameters chosen for the PSO algorithm are $\alpha = 0.8$ and $\beta = 0.8$ as they appear to stand out even if only slightly.

To study how the remaining two controllable parameters, the update weight w and the maximum velocity $v_{\max}$, might impact the performance of the algorithm a similar study

Figure 11 Visual representation of the performance values obtained for the PSO algorithm based on the logarithm of the Holevo variance $\ln(V_H(\alpha, \beta))$ with the parameter configurations following the rule $\beta \geq \alpha$ $\forall \beta > 0$. Better configurations lead to lower values of $\ln(V_H(\alpha, \beta))$

was also made for them. Keeping all of the other parameters constant and already using the optimal configuration $\alpha = 0.8$ and $\beta = 0.8$, the performance of the algorithm was studied for the different possible configurations of w and v_{max}. However, this time all the different configurations of these two parameters have lead the algorithm to converge to a solution and, most importantly, to a valid one. This comes as no surprise since these parameters, from a physical viewpoint of their behaviour on the search space, were not expected to have any influence on the ability of the algorithm to converge to a solution. Their impact was mostly expected on converging times.

Having arrived to an optimal configuration for the parameters α and β, the algorithm is quite robust to any configuration of the parameters w and v_{max} and is able to converge to a solution in the available number of iterations. It is nevertheless important to bare in mind that these parameter analysis was made for $N = 10$. For larger values of N, the convergence speed of the algorithms becomes more relevant as the search space scales polynomially with N. Unfortunately, such a parameter evaluation would also have been too computationally expensive and time consuming. Keeping this in mind, the optimal configuration considered for the PSO algorithm is: $\alpha = 0.8$, $\beta = 0.8$, $w = 0.8$ and $v_{max} = 0.2$.

Abbreviations
SQL, Standard Quantum Limit; HL, Heisenberg Limit; DE, Differential Evolution; PSO, Particle Swarm Optimisation; NISQ, Noisy Intermediate-Scale Quantum; RTN, Random Telegraph Noise; LSTM, Long Short-Term Memory; GRU, Gated-Recurrent Unit; TCN, Temporal Convolutional Networks.

Availability of data and materials

The code used in this paper is available at GitHub, or from the authors upon reasonable requests, for anyone interested in simulating these scenarios either for experimental purposes or simply for gaining theoretical insights.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

Conceptualization: GSP and YO. Code development for machine learning: NC. Code development for non-adaptive protocols: AS. Analytical results and connection to experiments: GSP. Writing of the paper: NC and GSP, with additional contributions from YO and AS. All authors have read and approved the final manuscript.

Author details

[1]Instituto Superior Técnico, Universidade de Lisboa, Lisbon, Portugal. [2]Instituto de Telecomunicações, Physics of Information and Quantum Technologies Group, Lisbon, Portugal. [3]QTF Centre of Excellence, Department of Applied Physics, Aalto University School of Science, Helsinki, Finland.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. Giovannetti V, Lloyd S, Maccone L. Quantum-enhanced measurements: beating the standard quantum limit. Science. 2004;306(5700):1330–6.
2. Giovannetti V, Lloyd S, Maccone L. Quantum metrology. Phys Rev Lett. 2006;96(1):010401.
3. Giovannetti V, Lloyd S, Maccone L. Advances in quantum metrology. Nat Photonics. 2011;5(4):222.
4. Tóth G, Apellaniz I. Quantum metrology from a quantum information science perspective. J Phys A, Math Theor. 2014;47(42):424006.
5. Helstrom CW. Quantum detection and estimation theory. J Stat Phys. 1969;1(2):231–52.
6. Holevo AS. Probabilistic and statistical aspects of quantum theory. vol. 1. Berlin: Springer; 2011.
7. Cramér H. Mathematical methods of statistics (PMS-9). vol. 9. Princeton: Princeton University Press; 2016.
8. Braginskiĭ VB, Vorontsov YI. Quantum-mechanical limitations in macroscopic experiments and modern experimental technique. Sov Phys Usp. 1975;17(5):644–50. https://doi.org/10.1070/pu1975v017n05abeh004362.
9. Braginsky VB, Vorontsov YI, Thorne KS. Quantum nondemolition measurements. Science. 1980;209(4456):547–57.
10. Braginsky VB, Braginsky VB, Khalili FY. Quantum measurement. Cambridge: Cambridge University Press; 1995.
11. Ozawa M. Realization of measurement and the standard quantum limit. In: Squeezed and nonclassical light. Berlin: Springer; 1989. p. 263–86.
12. Buluta I, Ashhab S, Nori F. Natural and artificial atoms for quantum computation. Rep Prog Phys. 2011;74(10):104401. https://doi.org/10.1088/0034-4885/74/10/104401.
13. Slussarenko S, Pryde GJ. Photonic quantum information processing: a concise review. Appl Phys Rev. 2019;6(4):041303. https://doi.org/10.1063/1.5115814.
14. Flamini F, Spagnolo N, Sciarrino F. Photonic quantum information processing: a review. Rep Prog Phys. 2018;82(1):016001. https://doi.org/10.1088/1361-6633/aad5b2.
15. Wrachtrup J, Jelezko F. Processing quantum information in diamond. J Phys Condens Matter. 2006;18(21):807–24. https://doi.org/10.1088/0953-8984/18/21/s08.
16. Prawer S, Aharonovich I. Quantum information processing with diamond: principles and applications. 1st ed. Woodhead Publishing, Limited; 2018. https://doi.org/10.5555/3312180.
17. Lange W. In: Meyers RA, editor. Quantum computing with trapped ions. New York: Springer; 2012. p. 2406–36. https://doi.org/10.1007/978-1-4614-1800-9_149.
18. Bruzewicz CD, Chiaverini J, McConnell R, Sage JM. Trapped-ion quantum computing: progress and challenges. Appl Phys Rev. 2019;6(2):021314. https://doi.org/10.1063/1.5088164.
19. Paraoanu GS. Recent progress in quantum simulation using superconducting circuits. J Low Temp Phys. 2014;175(5):633–54. https://doi.org/10.1007/s10909-014-1175-8.
20. You JQ, Nori F. Atomic physics and quantum optics using superconducting circuits. Nature. 2011;474:589. https://doi.org/10.1038/nature10122.
21. Degen CL, Reinhard F, Cappellaro P. Quantum sensing. Rev Mod Phys. 2017;89(3):035002. https://doi.org/10.1103/revmodphys.89.035002.
22. Paris MG. Quantum estimation for quantum technology. Int J Quantum Inf. 2009;7(supp01):125–37.
23. Biercuk MJ, Uys H, Britton JW, VanDevender AP, Bollinger JJ. Ultrasensitive detection of force and displacement using trapped ions. Nat Nanotechnol. 2010;5(9):646–50. https://doi.org/10.1038/nnano.2010.165.
24. Scerri E, Gauger EM, Bonato C. Extending qubit coherence by adaptive quantum environment learning. New J Phys. 2020;22(3):035002. https://doi.org/10.1088/1367-2630/ab7bf3.
25. Brownnutt M, Kumph M, Rabl P, Blatt R. Ion-trap measurements of electric-field noise near surfaces. Rev Mod Phys. 2015;87(4):1419–82. https://doi.org/10.1103/revmodphys.87.1419.
26. Motazedifard A, Dalafi A, Naderi MH. Ultra-precision quantum sensing and measurement based on nonlinear hybrid optomechanical systems containing ultracold atoms or atomic-BE. 2020. arXiv:2011.01336.
27. Abadie J, Abbott B, Abbott R, Abbott T, Abernathy M, Adams C, Adhikari R, Affeldt C, Allen B, Allen G et al. A gravitational wave observatory operating beyond the quantum shot-noise limit. Nat Phys. 2011;7(12):962.
28. Pezzè L, Smerzi A, Oberthaler MK, Schmied R, Treutlein P. Quantum metrology with nonclassical states of atomic ensembles. Rev Mod Phys. 2018;90:035005. https://doi.org/10.1103/RevModPhys.90.035005.

29. Berry D, Wiseman H. Optimal states and almost optimal adaptive measurements for quantum interferometry. Phys Rev Lett. 2000;85(24):5098.
30. Berry DW, Wiseman H, Breslin J. Optimal input states and feedback for interferometric phase estimation. Phys Rev A. 2001;63(5):053804.
31. Berry DW, Higgins BL, Bartlett SD, Mitchell MW, Pryde GJ, Wiseman HM. How to perform the most accurate possible phase measurements. Phys Rev A. 2009;80(5):052114.
32. Wiseman HM. Adaptive phase measurements of optical modes: going beyond the marginal q distribution. Phys Rev Lett. 1995;75:4587–90. https://doi.org/10.1103/PhysRevLett.75.4587.
33. Armen MA, Au JK, Stockton JK, Doherty AC, Mabuchi H. Adaptive homodyne measurement of optical phase. Phys Rev Lett. 2002;89:133602. https://doi.org/10.1103/PhysRevLett.89.133602.
34. Fujiwara A. Strong consistency and asymptotic efficiency for adaptive quantum estimation problems. J Phys A, Math Gen. 2006;39(40):12489–504. https://doi.org/10.1088/0305-4470/39/40/014.
35. Okamoto R, Iefuji M, Oyama S, Yamagata K, Imai H, Fujiwara A, Takeuchi S. Experimental demonstration of adaptive quantum state estimation. Phys Rev Lett. 2012;109:130404. https://doi.org/10.1103/PhysRevLett.109.130404.
36. Brivio D, Cialdi S, Vezzoli S, Gebrehiwot BT, Genoni MG, Olivares S, Paris MGA. Experimental estimation of one-parameter qubit gates in the presence of phase diffusion. Phys Rev A. 2010;81:012305. https://doi.org/10.1103/PhysRevA.81.012305.
37. Griffiths RB, Niu C-S. Semiclassical Fourier transform for quantum computation. Phys Rev Lett. 1996;76:3228–31. https://doi.org/10.1103/PhysRevLett.76.3228.
38. Higgins BL, Berry DW, Bartlett SD, Wiseman HM, Pryde GJ. Entanglement-free Heisenberg-limited phase estimation. Nature. 2007;450(7168):393–6. https://doi.org/10.1038/nature06257.
39. Danilin S, Lebedev A, Vepsäläinen A, Lesovik G, Blatter G, Paraoanu G. Quantum-enhanced magnetometry by phase estimation algorithms with a single artificial atom. npj Quantum Inf. 2018;4(1):29.
40. Bonato C, Blok MS, Dinani HT, Berry DW, Markham ML, Twitchen DJ, Hanson R. Optimized quantum sensing with a single electron spin using real-time adaptive measurements. Nat Nanotechnol. 2016;11:247–52. https://doi.org/10.1038/nnano.2015.261.
41. Hentschel A, Sanders BC. Machine learning for precise quantum measurement. Phys Rev Lett. 2010;104:063603. https://doi.org/10.1103/PhysRevLett.104.063603.
42. Hentschel A, Sanders BC. Efficient algorithm for optimizing adaptive quantum metrology processes. Phys Rev Lett. 2011;107:233601. https://doi.org/10.1103/PhysRevLett.107.233601.
43. Lovett NB, Crosnier C, Perarnau-Llobet M, Sanders BC. Differential evolution for many-particle adaptive quantum metrology. Phys Rev Lett. 2013;110(1):220501.
44. Palittpongarnpim P, Wittek P, Sanders BC. Single-shot adaptive measurement for quantum-enhanced metrology. In: Quantum communications and quantum imaging XIV. vol. 9980. International Society for Optics and Photonics; 2016. p. 99800.
45. Palittapongarnpim P, Wittek P, Sanders BC. Controlling adaptive quantum phase estimation with scalable reinforcement learning. In: 24th European symposium on artificial neural networks. Bruges, April 27–29, 2016. 2016. p. 327–32.
46. Palittapongarnpim P, Wittek P, Zahedinejad E, Vedaie S, Sanders BC. Learning in quantum control: high-dimensional global optimization for noisy quantum dynamics. Neurocomputing. 2017;268:116–26.
47. Palittapongarnpim P, Sanders BC. Robustness of adaptive quantum-enhanced phase estimation. 2018. arXiv preprint. arXiv:1809.05525.
48. Lumino A, Polino E, Rab AS, Milani G, Spagnolo N, Wiebe N, Sciarrino F. Experimental phase estimation enhanced by machine learning. Phys Rev Appl. 2018;10(1):044033.
49. Ciliberto C, Rocchetto A, Rudi A, Wossnig L. Statistical limits of supervised quantum learning. Phys Rev A. 2020;102:042414. https://doi.org/10.1103/PhysRevA.102.042414.
50. Nielsen MA, Chuang I. Quantum computation and quantum information. Cambridge: Cambridge University Press; 2000. https://doi.org/10.1007/978-1-4614-1800-9_149.
51. Storn R, Price K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. J Glob Optim. 1997;11(4):341–59.
52. Price K, Storn RM, Lampinen JA. Differential evolution: a practical approach to global optimization. Berlin: Springer; 2006.
53. Kennedy J. Particle swarm optimization. In: Encyclopedia of machine learning. Berlin: Springer; 2011. p. 760–6.
54. Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In: Micro machine and human science, 1995. MHS'95. Proceedings of the sixth international symposium on. IEEE; 1995. p. 39–43.
55. Shi Y, Eberhart R. A modified particle swarm optimizer. In: Evolutionary computation proceedings, 1998. IEEE world congress on computational intelligence. The 1998 IEEE international conference on. IEEE; 1998. p. 69–73.
56. Chapeau-Blondeau F. Optimizing qubit phase estimation. Phys Rev A. 2016;94:022334. https://doi.org/10.1103/PhysRevA.94.022334.
57. Rodríguez-García MA, Castillo IP, Barberis-Blostein P. Efficient qubit phase estimation using adaptive measurements. 2020. arXiv preprint. arXiv:2012.11088.
58. Sekatski P, Skotiniotis M, Kołodyński J, Dür W. Quantum metrology with full and fast quantum control. Quantum. 2017;1:27.
59. Paraoanu GS. Generalized partial measurements. Europhys Lett. 2011;93(6):64002. https://doi.org/10.1209/0295-5075/93/64002.
60. Higgins BL, Berry DW, Bartlett SD, Wiseman HM, Pryde GJ. Adaptive single-shot phase measurements: the full quantum theory. Nature. 2007;450:393–6.
61. Daryanoosh S, Slussarenko S, Berry DW, Wiseman HM, Pryde GJ. Experimental optical phase measurement approaching the exact Heisenberg limit. Nat Commun. 2018;9:4606.
62. Paraoanu GS. Interaction-free measurements with superconducting qubits. Phys Rev Lett. 2006;97:180406. https://doi.org/10.1103/PhysRevLett.97.180406.

63. Danilin S, Lebedev AV, Vepsäläinen A, Lesovik GB, Blatter G, Paraoanu GS. Quantum-enhanced magnetometry by phase estimation algorithms with a single artificial atom. npj Quantum Inf. 2018;4(1):29. https://doi.org/10.1038/s41534-018-0078-y.
64. Silveri MP, Tuorila JA, Thuneberg EV, Paraoanu GS. Quantum systems under frequency modulation. Rep Prog Phys. 2017;80:056002. https://doi.org/10.1088/1361-6633/aa5170.
65. Shlyakhov AR, Zemlyanov VV, Suslov MV, Lebedev AV, Paraoanu GS, Lesovik GB, Blatter G. Quantum metrology with a transmon qutrit. Phys Rev A. 2018;97(2):022115. https://doi.org/10.1103/physreva.97.022115.
66. Danilin WMS. Quantum sensing with superconducting circuits. 2021. arXiv:2103.11022.
67. Perelshtein MR, Kirsanov NS, Zemlyanov VV, Lebedev AV, Blatter G, Vinokur VM, Lesovik GB. Linear ascending metrological algorithm. Phys Rev Res. 2021;3:013257. https://doi.org/10.1103/PhysRevResearch.3.013257.
68. Baumgart I, Cai J-M, Retzker A, Plenio MB, Wunderlich C. Ultrasensitive magnetometer using a single atom. Phys Rev Lett. 2016;116:240801. https://doi.org/10.1103/PhysRevLett.116.240801.
69. Timoney N, Baumgart I, Johanning M, Varón AF, Plenio MB, Retzker A, Wunderlich Ch. Quantum gates and memory using microwave-dressed states. Nature. 2011;476:185–8. https://doi.org/10.1038/nature10319.
70. Taylor JM, Cappellaro P, Childress L, Jiang L, Budker D, Hemmer PR, Yacoby A, Walsworth R, Lukin MD. High-sensitivity diamond magnetometer with nanoscale resolution. Nat Phys. 2008;4:810–6.
71. Barry JF, Schloss JM, Bauch E, Turner MJ, Hart CA, Pham LM, Walsworth RL. Sensitivity optimization for NV-diamond magnetometry. Rev Mod Phys. 2020;92:015004. https://doi.org/10.1103/RevModPhys.92.015004.