

Quantum natural gradient optimizer on noisy platforms: Quantum approximate optimization algorithm as a case study

Federico Dell'Anna ^{1,2}, Rafael Gómez-Lurbe ³, Armando Pérez ³, and Elisa Ercolessi ^{1,2}

¹*Dipartimento di Fisica e Astronomia dell'Università di Bologna, I-40127 Bologna, Italy*

²*INFN, Sezione di Bologna, I-40127 Bologna, Italy*

³*Departament de Física Teòrica and IFIC, Universitat de València-CSIC, 46100 Burjassot (València), Spain*



(Received 3 April 2025; accepted 28 July 2025; published 13 August 2025)

We investigate the performance of the quantum natural gradient (QNG) optimizer in the presence of noise. Specifically, we evaluate the efficacy of QNG within the quantum approximate optimization algorithm for finding the ground state of the transverse field Ising model. Its performance is benchmarked against the vanilla gradient descent optimizer across two prominent quantum computing platforms: Rydberg atoms and superconducting circuits. Our analysis includes simulations under both idealized noise-free conditions and realistic noisy environments based on calibration data from actual devices. Results demonstrate that QNG consistently outperforms vanilla gradient descent, exhibiting faster convergence on average and greater robustness against random initializations of parameters. This robustness is attributed to the distance regularization in parameter space inherent to QNG. Additionally, QNG achieves a higher convergence rate to the solution, effectively avoiding certain local minima. These findings highlight QNG as a promising tool for optimizing variational quantum algorithms in noisy intermediate-scale quantum devices.

DOI: [10.1103/62wx-tvk5](https://doi.org/10.1103/62wx-tvk5)

I. INTRODUCTION

Hybrid quantum-classical variational algorithms [1–3] are essential for exploring the potential of noisy intermediate-scale quantum (NISQ) devices [4]. These algorithms combine classical computation with quantum processing, where classical resources are used to optimize the parameters of quantum states. By employing heuristic techniques, they tackle variational problems, for instance combinatorial optimization tasks, which are both prevalent and critical in many real-world applications [5]. Consequently, these methods have garnered substantial interest from industries looking to leverage quantum computing, as solving such problems remains notoriously difficult for classical algorithms alone.

The quantum approximate optimization algorithm (QAOA) [6] stands out among hybrid variational algorithms for its promise in achieving quantum speedups on NISQ devices, drawing considerable interest [7]. QAOA has been successfully implemented across a variety of experimental platforms, such as Rydberg atom arrays [8,9], superconducting processors [10], and trapped-ion systems [11].

Similarly to other hybrid algorithms, QAOA involves applying a series of parameterized quantum gates to a quantum state with the goal of minimizing the expectation value of an observable, typically the system's Hamiltonian. The classical part of the algorithm optimizes the gate parameters to achieve this minimization, requiring frequent evaluations of the Hamiltonian's expectation value [12]. The interplay between quantum execution and classical optimization demands repeated runs of the quantum circuit, making the process resource intensive. This implies that finding the minimum of the cost function in fewer steps plays a crucial role.

A significant challenge in this approach is the occurrence of barren plateaus: large flat areas of the parameters landscape with exponentially small gradients as the number of qubits and circuit depth increase [13]. This problem can be exacerbated by noise [14] or by employing cost functions dependent on global observables [15].

This study will focus on a paradigmatic problem: finding the ground-state preparation of the transverse field Ising model (TFIM). In particular, we aim to combine the QAOA with the quantum natural gradient (QNG) optimizer and evaluate its performance on simulations of realistic noisy quantum devices. The quantum natural gradient was originally proposed in Ref. [16] to enhance the optimization of variational quantum algorithms. Several studies have since demonstrated the potential advantages of using QNG techniques in these algorithms [17–19].

The main idea is to leverage information about the geometry of the Hilbert space. Specifically, this involves scaling the gradient of the cost function by the inverse of the Fubini-Study metric, which is the Riemannian metric in the space of parameterized quantum systems [20]. This approach ensures that the algorithm follows the correct direction of steepest descent, thereby accelerating its convergence towards the solution.

Building on this foundation, further research extended the quantum natural gradient approach to tackle challenges such as noise and nonunitary evolutions in quantum systems [21], using their own noise models along with some procedure to approximate the Fubini-Study metric.

In light of the existing work, our aim is to benchmark the performance of the QNG optimizer against the vanilla gradient descent optimizer on two major quantum computing

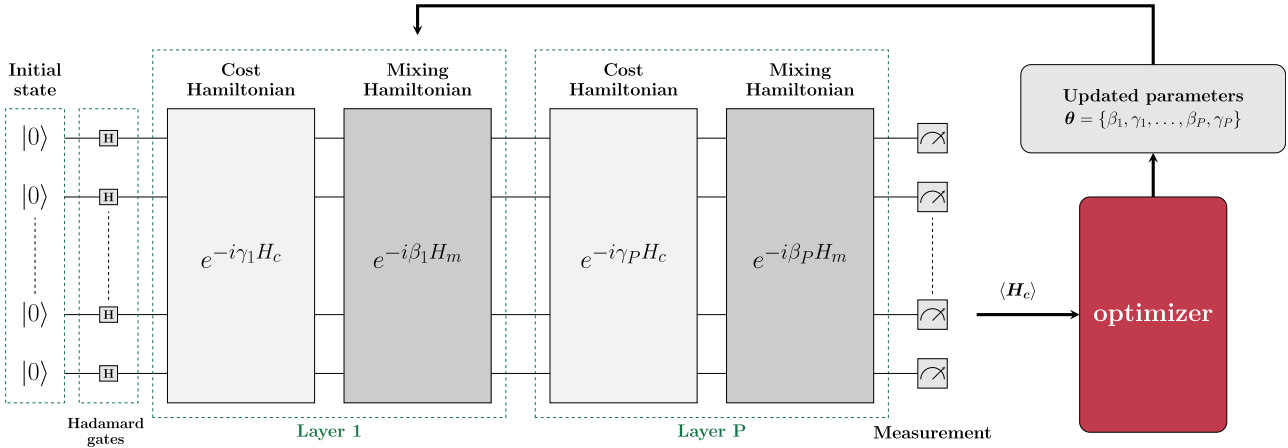


FIG. 1. QAOA scheme: starting from an initial state, Hadamard gates prepare a superposition. Each layer alternates between the application of the cost Hamiltonian $e^{-i\gamma H_c}$ and the mixing Hamiltonian $e^{-i\beta H_m}$. The parameters γ and β are iteratively optimized to minimize the cost function $\langle H_c \rangle$ after the measurement phase.

platforms under realistic noise models: Rydberg atoms and superconducting circuits.

In the current NISQ era, gradient-free optimization techniques are often preferred for QAOA due to their lower measurement overhead and simpler implementation. In this work, we deliberately focus on gradient-based methods—not as a recommendation for near-term optimization but rather as a theoretical and exploratory benchmark to assess the robustness of QNG under realistic noise models.

Indeed, the primary objective of this work is not to demonstrate state-of-the-art optimization performance but rather to investigate how the QNG responds to realistic noise conditions typical of current quantum platforms. For this reason, we adopt the TFIM as a toy model, serving as a well-understood and controllable testbed for benchmarking the behavior of QNG.

This work is organized as follows: In Sec. II, we briefly review the QAOA algorithm, the transverse field Ising model, and the quantum natural gGradient optimizer. Section III introduces the basics of the Rydberg atoms platform, the implementation of QAOA on it, the noise sources considered, and the numerical studies performed. The subsequent Sec. V presents the quantum circuit implementation of QAOA, the realistic noise model used, and the simulations conducted on this platform. Finally, in Sec. VII, we summarize the conclusions drawn from this work, as well as outline future studies and open problems in this research line.

II. QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM AND QUANTUM NATURAL GRADIENT

The QAOA is a hybrid quantum-classical algorithm originally inspired by the adiabatic theorem and designed to solve combinatorial optimization problems. It was first proposed by Edward Farhi and Jeffrey Goldstone in 2014 [6] and has since become one of the most promising quantum algorithms for near-term quantum devices. The algorithm consists of two main components: a classical optimizer and a quantum circuit that prepares a parameterized trial state on a quantum computer. The key idea is to iteratively adjust the parameters

to minimize the expectation value of the cost Hamiltonian, thereby approaching the optimal solution.

A. QAOA circuit

The QAOA circuit, as shown in Fig. 1, involves the following steps:

(1) *Initialization.* The algorithm starts with an initial state: the uniform superposition state $|+\rangle^{\otimes N}$, where N is the number of qubits.

(2) *Parameterized evolution.* The state is then evolved using two types of unitary operators:

(a) Cost Hamiltonian unitary: $U_c(\gamma) = e^{-i\gamma H_c}$

(b) Mixer Hamiltonian unitary: $U_m(\beta) = e^{-i\beta H_m}$

Here H_c is the cost Hamiltonian and H_m is the mixer Hamiltonian. To explore the whole Hilbert space, it is necessary to have $[H_c, H_m] \neq 0$. Then the mixer Hamiltonian is chosen as $H_m = \sum_{i=1}^N \sigma_i^x$, where σ_i^x are Pauli- X operators acting on the i th qubit.

(3) *Layered structure.* The evolution is applied in layers, where each layer consists of a pair of unitaries $U_c(\gamma)$ and $U_m(\beta)$. The depth of the QAOA circuit is determined by the number of layers P , leading to a trial state:

$$|\psi(\vec{\gamma}, \vec{\beta})\rangle = U_m(\beta_P)U_c(\gamma_P) \cdots U_m(\beta_1)U_c(\gamma_1)|+\rangle^{\otimes N}.$$

Here $\vec{\gamma} = (\gamma_1, \gamma_2, \dots, \gamma_P)$ and $\vec{\beta} = (\beta_1, \beta_2, \dots, \beta_P)$ are the parameters to be optimized.

(4) *Measurement and optimization.* The expectation value of the problem Hamiltonian is measured in the trial state:

$$E_c = \langle \psi(\vec{\gamma}, \vec{\beta}) | H_c | \psi(\vec{\gamma}, \vec{\beta}) \rangle. \quad (1)$$

A classical optimizer adjusts the parameters $\vec{\gamma}$ and $\vec{\beta}$ to minimize this expectation value, iterating the process until convergence.

The QAOA merges the advantages of both quantum and classical computing.

B. Application to the transverse field Ising model

In our study, we apply QAOA to the TFIM, a well-known model in quantum mechanics and statistical physics. The

Hamiltonian for the one-dimensional (1D) TFIM is given by:

$$H = -J \sum_{i=1}^N \sigma_i^z \sigma_{i+1}^z - h \sum_{i=1}^N \sigma_i^x, \quad (2)$$

where σ_i^z and σ_i^x are Pauli-Z and Pauli-X operators acting on the i th qubit, J is the interaction strength between neighboring spins, and h is the transverse magnetic field strength. By optimizing the parameters of the QAOA circuit, we aim at finding the ground state of this Hamiltonian. From now on, we work setting $J = 1$, $h = 0.5$ and with periodic boundary conditions.

As previously mentioned, the QAOA follows a sequence of well-defined steps while allowing flexibility in the choice of Hamiltonians used to evolve the system. In particular, for the TFIM, we modify the standard structure of QAOA by defining the unitary operators governing the evolution as $U_{zz}(\gamma) = e^{-i\gamma H_{zz}}$, where $H_{zz} = -J \sum_{i=1}^N \sigma_i^z \sigma_{i+1}^z$, for the first evolution of the state. As the mixing Hamiltonian, we use the standard form $U_m(\beta) = e^{-i\beta H_m}$, where $H_m = -h \sum_{i=1}^N \sigma_i^x$. For our numerical simulations, we set $J = 1$ and $h = 0.5$. However, as the cost function, we measure the expectation value of the full Hamiltonian given in Eq. (2).

This modification is motivated by the fact that the conventional QAOA structure typically employs a mixing Hamiltonian that has the same form as the second term in the problem Hamiltonian Eq. (2). By adjusting the standard procedure, we eliminate this redundancy.

By implementing the algorithm using the QuTiP library [22], we verified that the ground state is reached at a circuit depth of $P = \lfloor \frac{N}{2} \rfloor$ by testing different values of P using the quantum natural gradient as optimizer. This preliminary check will also be performed in the implementations based on Rydberg atoms and IBM superconducting qubits, using the Pulser and PennyLane backends, respectively.

This result implies that circuits with such a depth are sufficiently expressive to capture the ground state while preserving the symmetries of the Hamiltonian. Hence, we adopt this parametrization for the algorithm's evolution, subject to validation during implementation on the selected platform, which inevitably introduces certain approximations, both in the presence and absence of noise.

The energy value obtained at the end of the optimization procedure E_c can be compared with the exact one E_0 , allowing us to compute the accuracy:

$$\delta E := \frac{E_c - E_0}{|E_0|}. \quad (3)$$

The exact energy of the TFIM with $J = 1$ can be calculated analytically [17,23]:

$$E_0 = -E_1 - 2 \sum_{q=1}^r \frac{1}{1 + h^2 + 2h \cos \alpha_q}, \quad (4)$$

where $r = \lfloor N/2 \rfloor$ and

$$\begin{cases} \alpha_q := \frac{(2q-1)\pi}{N}, & E_1 := 0; & \text{for } N \text{ even} \\ \alpha_q := \frac{2q\pi}{N}, & E_1 := 1 + h; & \text{for } N \text{ odd.} \end{cases} \quad (5)$$

Our implementation of QAOA on real quantum platforms, such as Rydberg atoms and superconducting circuits, aims to demonstrate the algorithm's robustness and effectiveness, even in the presence of various sources of noise.

C. Quantum natural gradient

Vanilla gradient descent, or simply gradient descent, is an optimization algorithm used to minimize a function by iteratively moving in the direction of the steepest descent, defined as the negative gradient. It is widely used in machine learning and statistics for optimizing models by adjusting their parameters [24].

Here is a brief overview of the method: We start with an initial guess for the parameters, denoted as $\theta = (\theta_1, \theta_2, \dots, \theta_n)$. Next, we calculate the gradient of the loss function $C(\theta)$ and update the parameters by moving in the opposite direction of the gradient. The update rule for each parameter θ_i is given by:

$$\theta_{i+1} = \theta_i - \eta \nabla C(\theta_i), \quad (6)$$

where η is the learning rate, a small positive number that controls the step size. These steps are repeated until convergence that is typically defined by a condition such as:

$$\|C(\theta_{t+1}) - C(\theta_t)\| < \epsilon_{\text{stop}}, \quad (7)$$

where ϵ is a small threshold indicating that the changes in the parameters are sufficiently small.

The update rule for gradient descent can be derived from the following optimization problem [16]:

$$\theta_{t+1} = \arg \min_{\theta \in \mathbb{R}^d} \left[\langle \theta - \theta_t, \nabla C(\theta_t) \rangle + \frac{1}{2\eta} \|\theta - \theta_t\|_2^2 \right]. \quad (8)$$

Examining this expression naturally raises the following question: We have assumed that the geometry of the parameter space is Euclidean, but what happens if this assumption does not hold and the space instead has a different Riemannian structure? This question was posed by the authors in Ref. [25], leading to the introduction of the natural gradient algorithm.

In this algorithm, the parameter update rule is modified by multiplying the gradient by the inverse of the Fisher information matrix, $I^{-1}(\theta)$. The Fisher information matrix serves as the Riemannian metric that captures the geometry of the statistical model, guiding the gradient in the true direction of steepest descent. By accounting for the local curvature of the probability distribution space, the update rule becomes more natural and can potentially lead to faster optimization in the parameter space [26–29].

In direct analogy to the natural gradient algorithm, the authors in Ref. [16] proposed extending this approach to the quantum setting. In this context, the metric tensor is defined on the manifold of parameterized quantum states. For pure states, this metric is known as the Fubini-Study metric [30]:

$$g_{ij}(\theta) = \text{Re}[\langle \partial_i \psi(\theta) | \partial_j \psi(\theta) \rangle - \langle \partial_i \psi(\theta) | \psi(\theta) \rangle \langle \psi(\theta) | \partial_j \psi(\theta) \rangle], \quad (9)$$

where $|\psi(\boldsymbol{\theta})\rangle$ is a parameterized quantum state, and we use the shorthand $\partial_i \equiv \frac{\partial}{\partial \theta_i}$.

With this metric, the update rule for the QNG is expressed as:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta g^+(\boldsymbol{\theta}_t) \nabla C(\boldsymbol{\theta}_t), \quad (10)$$

where $g^+(\boldsymbol{\theta}_t)$ denotes the pseudoinverse of the metric tensor, which is particularly useful when handling singular matrices.

We can further specify the update rule for the parameters within the QAOA ansatz:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta g^+(\boldsymbol{\theta}_t) \nabla E(\boldsymbol{\theta}_t), \quad (11)$$

where $\boldsymbol{\theta} = (\gamma_1, \beta_1, \gamma_2, \beta_2, \dots, \gamma_P, \beta_P)$. Here $E(\boldsymbol{\theta}) = \text{Tr}[\rho_{2P}(\boldsymbol{\theta})H_c]$ represents the expectation value of the cost Hamiltonian. Additionally, $\rho_{2P}(\boldsymbol{\theta})$ represents the state obtained at the end of the circuit, i.e., after evolving the system with $2P$ parameters. In general, we assume this state to be mixed.

Intuitively, the Fubini-Study metric measures the sensitivity of the quantum state to changes in the parameters. Specifically, the Fubini-Study matrix will exhibit larger entries when the parameter variations significantly impact the quantum state and smaller entries when the effect is minimal. Consequently, since the update rule scales the gradient by the inverse of the metric, the QNG method effectively regularizes such changes in the parameter space.

Interestingly, this metric is related to the quantum Fisher information matrix (QFIM), which defines the ultimate precision limit for quantum multiparameter estimation through the quantum Cramér-Rao bound [31,32]. Specifically, the QFIM is proportional to the Fubini-Study metric:

$$F_{ij}(\boldsymbol{\theta}) = 4g_{ij}(\boldsymbol{\theta}), \quad (12)$$

where $F(\boldsymbol{\theta})$ is the QFIM and $g(\boldsymbol{\theta})$ represents the Fubini-Study metric.

Calculating F for arbitrary quantum states can be complex. For rank- r density matrices, defined as $\rho(\boldsymbol{\theta}) = \sum_{n=1}^r \lambda_n |\lambda_n\rangle\langle\lambda_n|$, the entries of the QFIM are given by [33]:

$$F_{ab} = \sum_{\lambda_i \in S} \frac{(\partial_a \lambda_i)(\partial_b \lambda_i)}{\lambda_i} + \sum_{\lambda_i \in S} 4\lambda_i \text{Re}(\langle \partial_a \lambda_i | \partial_b \lambda_i \rangle) - \sum_{\lambda_i, \lambda_j \in S} \frac{8\lambda_i \lambda_j}{\lambda_i + \lambda_j} \text{Re}(\langle \partial_a \lambda_i | \lambda_j \rangle \langle \lambda_j | \partial_b \lambda_i \rangle), \quad (13)$$

where S denotes the set of nonzero eigenvalues of ρ .

For pure states, this expression simplifies to the Fubini-Study metric tensor, up to a constant factor:

$$F_{ab} = 4 \text{Re}[\langle \partial_a \psi | \partial_b \psi \rangle - \langle \partial_a \psi | \psi \rangle \langle \psi | \partial_b \psi \rangle]. \quad (14)$$

Particularizing for QAOA, this can be rewritten in the following form:

$$F_{ab} = 4 \text{Re} \left[\langle \psi_{a-1} | H_a \left(\prod_{k=a}^{b-1} e^{i\theta_k H_k} \right) H_b | \psi_{b-1} \rangle - \langle \psi_{a-1} | H_a | \psi_{a-1} \rangle \langle \psi_{b-1} | H_b | \psi_{b-1} \rangle \right]. \quad (15)$$

This expression stems from the unitary parameter embedding defined by QAOA. In this framework, the Hamiltonian $H_a = H_{zz}$ when a is odd and $H_a = H_m$ when a is even, while $|\psi_{a-1}\rangle$ represents the state obtained after evolving the initial state with $a - 1$ parameters.

For diagonal elements, this simplifies further to:

$$F_{aa} = 4 \text{Re}[\langle \psi_{a-1} | H_a^2 | \psi_{a-1} \rangle - \langle \psi_{a-1} | H_a | \psi_{a-1} \rangle^2]. \quad (16)$$

To address mixed states arising from noisy circuits, we approximate the QFIM by the expression:

$$F_{ab} = 4 \text{Re} \left[\text{Tr} \left(\rho_{a-1} H_a \left(\prod_{k=a}^{b-1} e^{i\theta_k H_k} \right) H_b \rho_{b-1} \right) - \text{Tr}(\rho_{a-1} H_a) \cdot \text{Tr}(\rho_{b-1} H_b) \right], \quad (17)$$

and for diagonal elements:

$$F_{aa} = 4 \text{Re}[\text{Tr}(\rho_{a-1} H_a^2) - (\text{Tr}(\rho_{a-1} H_a))^2]. \quad (18)$$

Indeed, under the assumption of mild noise, mixed states can be expressed using the spectral decomposition as:

$$\rho = \lambda_0 |\psi\rangle\langle\psi| + \sum_{m=1}^d \lambda_m |\psi_m\rangle\langle\psi_m|, \quad (19)$$

where the dominant eigenvector $|\psi\rangle$ approximates the ideal quantum state [21,34,35].

Thus, when the values of λ_m are small (low noise), the noisy state is close to the pure state, and Eqs. (17) and (18) provide a good approximation for the QFIM.

D. Computation cost of QNG for QAOA

The cost of computing the diagonal approximation of the QFIM for QAOA, as defined in Eq. (16), depends on the number of expectation values required for each entry. Each expectation value generally requires a separate quantum circuit preparation. However, this cost can be reduced by grouping observables into sets of commuting operators. Observables within the same commuting group can be measured simultaneously using a single quantum circuit, thereby reducing the total number of distinct circuits required. This technique is a standard practice in the quantum computing community to reduce the resource overhead when estimating observables composed of many terms, such as Hamiltonians [36].

In the worst-case scenario, for a Hamiltonian consisting of m noncommuting terms—where neither the terms of the Hamiltonian nor those of its square commute with each other—one must evaluate $\mathcal{O}(m^2)$ expectation values to compute a single diagonal entry of the QFIM. This requires preparing and running $\mathcal{O}(m^2)$ different quantum circuits. To estimate each expectation value with a precision of $\mathcal{O}(\epsilon)$, $\mathcal{O}(1/\epsilon^2)$ measurements are needed. Consequently, the total measurement cost for estimating one diagonal QFIM entry is $\mathcal{O}(m^2/\epsilon^2)$.

If the Hamiltonians involved in QAOA have different numbers of terms, then the cost of estimating the full diagonal QFIM becomes $\mathcal{O}(P(m^2 + m'^2)/\epsilon^2)$, where m and m' are

the numbers of noncommuting terms in the cost and mixing Hamiltonians, respectively, and P is the number of QAOA layers. However, as previously noted, this cost can be significantly reduced by grouping commuting observables and measuring them using the same circuit. This strategy is also employed in Ref. [16] for estimating the block-diagonal approximation of the QFIM.

In our specific case of the TFIM, the cost of computing the diagonal entries of the QFIM is substantially reduced due to the structure of the Hamiltonian. All terms in H_{zz} commute with each other, and the same holds for all terms in H_x . This allows us to compute all required expectation values for each diagonal QFIM entry using only one quantum state preparation per group. As a result, only $2P$ distinct quantum circuit preparations are needed for the diagonal approximation of the QFIM.

In contrast, computing the gradient requires $4mP$ different quantum circuits, where m is the number of terms in both the cost and mixing Hamiltonians.

On the other hand, computing the full QFIM is substantially more costly because it requires estimating off-diagonal terms between different layers. In general, estimating these terms necessitates the use of the Hadamard test, which requires an auxiliary qubit for each term involved in Eq. (15). To compute a single off-diagonal QFIM entry, one must perform $\mathcal{O}(mm')$ different Hadamard tests, each requiring a separate quantum circuit with an auxiliary qubit, where m and m' are the number of terms in the respective Hamiltonians H_a and H_b from Eq. (15). Each such estimate with precision $\mathcal{O}(\epsilon)$ requires $\mathcal{O}(1/\epsilon^2)$ measurements, leading to a total cost of $\mathcal{O}(mm'/\epsilon^2)$ measurements per off-diagonal entry.

Because the QFIM is symmetric, and assuming $2P$ parameters in the QAOA circuit, there are only $P(2P + 1)$ independent entries. Thus, the total cost of estimating the full QFIM in terms of measurements is upper-bounded by $\mathcal{O}(P^2m^2/\epsilon^2)$, assuming $m > m'$. If $m' > m$, then the roles can be swapped accordingly.

Importantly, the QFIM must be estimated at each step of the optimization process. Due to this unfavorable scaling, computing the full QFIM becomes impractical for large problem instances. By contrast, the diagonal approximation provides a more feasible alternative, offering potential improvements at a measurement cost similar to that of gradient estimation.

Nonetheless, gradient-based optimization remains out of reach for current near-term quantum devices. There is ongoing research aimed at reducing the complexity of estimating both gradients [37,38] and the QFIM [39], which may enable these optimization strategies to become viable in practice in the near future.

Additionally, there has been rigorous analysis of the measurement overhead associated with variational quantum algorithms that utilize the QNG for optimization. In Ref. [40], the authors derive upper bounds on the number of measurements required to perform QNG at each optimization step for a given target precision. Notably, they show that, asymptotically, the cost of computing the QFIM becomes negligible compared to that of estimating the gradient, particularly as the number of iterations and qubits increases. In this asymptotic regime, QNG introduces only marginal additional overhead

relative to standard gradient descent, while simultaneously improving convergence speed and overall performance. Furthermore, the authors propose a general strategy for optimally allocating measurement resources between the QFIM and the gradient to minimize total overhead.

III. RYDBERG ATOMS IMPLEMENTATION

A quantum computer based on Rydberg atoms represents an ideal platform for testing the efficiency of our algorithm [41–43]. Analog quantum computing leverages the natural evolution of a quantum system to solve specific problems. Instead of programming discrete quantum gates (digital quantum computing), physical parameters, such as magnetic fields or laser intensities, are directly manipulated to induce a continuous-time dynamic evolution.

In our case, we work with a *register* of Rydberg atoms, which serve as the qubits. Analog quantum computing is realized by evolving this atomic register according to a specific Hamiltonian, while continuously adjusting parameters like detuning or laser intensity. Indeed, by acting on all atoms with the same global pulse, one can evolve the entire system with the following Hamiltonian:

$$H(t) = \sum_i \left(\frac{\Omega(t)}{2} \sigma_i^x - \delta(t) n_i + \sum_{j < i} U_{ij} n_i n_j \right) \quad (20)$$

where $n_i = \frac{1}{2}(1 + \sigma_i^z)$. Here $\Omega(t)$ is the Rabi frequency (which determines the strength of the interaction and it is proportional to the amplitude of the laser field), $\delta(t)$ (which denotes the discrepancy between the qubit resonance and the field frequencies) and U_{ij} denotes the blockade interaction parameter. This latter depends on the distance between qubits i and j : $U_{ij} = C_6/R_{ij}^6$, where C_6 is a constant. By the continuously manipulating $\Omega(t)$ and $\delta(t)$, one can achieve a significant level of control over the system's dynamics.

The evolution operators of the two Hamiltonians H_{zz} and H_m can be easily implemented by fixing the qubit register (thus fixing the distance between individual atoms R_{ij}) and varying $\Omega(t)$ to enter and exit the blockade regime. Inside the blockade regime only the interaction terms $n_i n_{i+1}$ will be relevant, whereas outside this regime σ_x will be the dominant factor. The blockade regime is determined by the Rydberg radius:

$$R_b = \left(\frac{C_6}{\Omega(t)} \right)^{1/6}. \quad (21)$$

In both regimes, we set the detuning value $\delta(t) = U_{i,i+1}$. In fact, expanding the interaction term $U_{i,i+1} n_i n_{i+1}$ reveals that it contains both $\sigma_i^z \sigma_{i+1}^z$ and σ_i^z terms. The latter can be canceled by this choice of detuning. An example of this implementation for $N = 12$ can be seen in Fig. 2.

This work focuses on the simulation of pulse sequences (including noise and imperfections) using the Pulser library [44,45], a Python-based library specifically designed to simulate the dynamics of Rydberg atom systems by numerically solving the full time-dependent Schrödinger equation in the Hilbert space of the system (i.e., full state evolution). These factors are intrinsic to real-world quantum devices and must be accurately replicated in simulations to devise effective

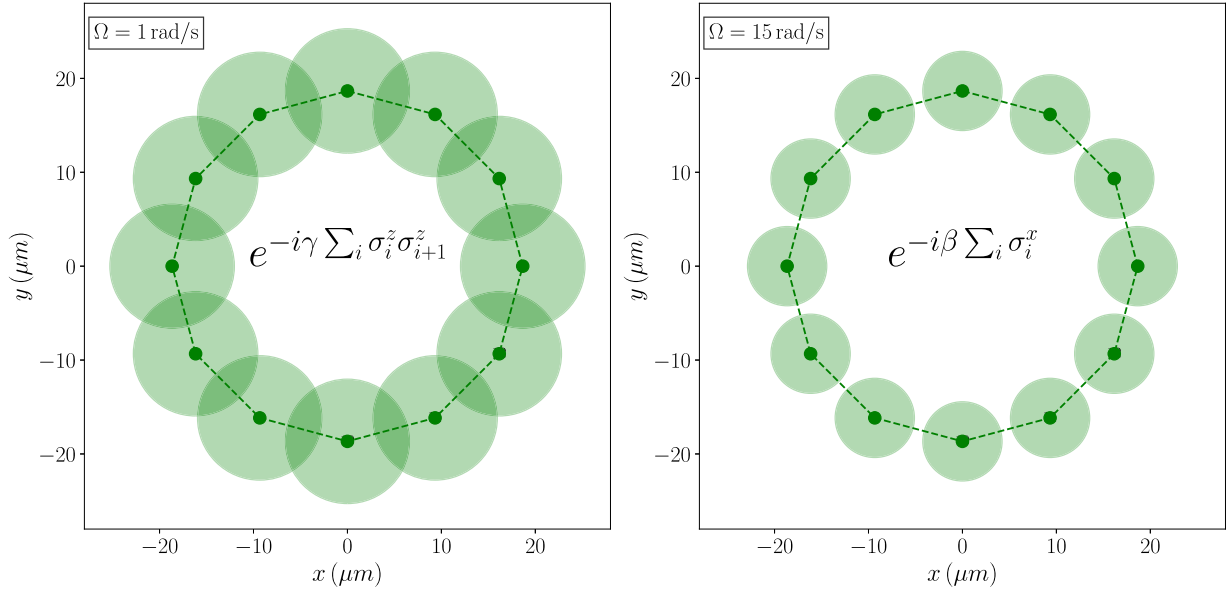


FIG. 2. Effective implementation of the two time-evolution operators of the QAOA ansatz in a Rydberg atom register. In this example with $N = 12$, we demonstrate how these unitaries can be realized by transitioning in and out of the blockade regime, adjusting the value of Ω from 1 rad/s to 15 rad/s.

strategies for addressing them. A comprehensive explanation of these phenomena in Rydberg platforms is provided in Ref. [5]. Here we will briefly outline the aspects considered in this study.

(1) State preparation and measurement (SPAM) errors.

(a) State preparation errors. Initial attempts to prepare the state may occasionally fail to capture all atoms in the ground state. This is modeled using a probability, p , denoting the likelihood of an atom not being in the intended state during preparation.

(b) Measurement errors. Detection inaccuracies contribute to errors. These include falsely identifying an atom in the ground state as excited (“false positives”) and vice versa (“false negatives”).

(2) Laser noises. Laser properties such as frequency and amplitude fluctuations impact their efficacy in addressing atomic level transitions.

(a) Doppler effect. Thermal motion of atoms causes Doppler shifts in the laser frequencies, altering the detuning frequency.

(b) Laser waist. The Gaussian profile of laser amplitude means atoms at the edges of the beam experience slightly lower amplitudes compared to those at the focal point.

(c) Amplitude fluctuations. Laser amplitude varies from pulse to pulse, influencing the consistency of laser operations.

IV. NUMERICAL RESULTS

In this section, we assessed the performance of the QAOA with QNG and vanilla gradient descent by analyzing their effectiveness in preparing the ground state of the TFIM by using Rydberg atoms platform. Our numerical studies included comparisons of convergence rates in reaching the true ground state from multiple random parameter initializations,

both with and without noise, as well as the number of steps required to achieve convergence.

A. Ground-state preparation

As explained in the previous sections, the QAOA algorithm adheres to a specific protocol, which in our case involves initializing the state in $|+\rangle^{\otimes N}$ and evolving it using the Hamiltonians H_{zz} and H_m . By strictly following this procedure, we find numerical evidence that the ground state is attained at a circuit depth $P = \lfloor N/2 \rfloor$, as verified in Sec. II B.

However, as previously mentioned, it is not feasible to evolve precisely using those two Hamiltonians since the interaction term, even though it should be negligible outside the Blockade regime, remains present and might affect the accuracy and the number of layers required to reach the ground state. The same applies within the blockade regime, where $\Omega(t)$, although small, remains present.

We aimed at exploring this aspect by plotting how fidelity varies with the number of layers P for different system sizes, both in the presence and absence of noise (see Fig. 3). We opted to present the fidelity trends because, in some cases, final states with nearly identical energies may still exhibit low fidelity values. As our primary goal is state preparation, our focus is on how closely the achieved state overlaps with the true ground state. The fidelity between a mixed state and a pure state is defined as $F = \langle \psi_g | \rho | \psi_g \rangle$, where ψ_g is the ground-state vector determined through exact diagonalization. According to the simulation results, it is clear that in the noiseless scenario, the ground state is achieved with an accuracy $\delta E_{\text{opt}} < 10^{-9}$ when $P = \lfloor N/2 \rfloor + 1$, where δE_{opt} represents the best accuracy obtained after optimization. This accuracy is calculated as the difference between the energy value obtained by the algorithm at convergence and the true value [see Eq. (3)].

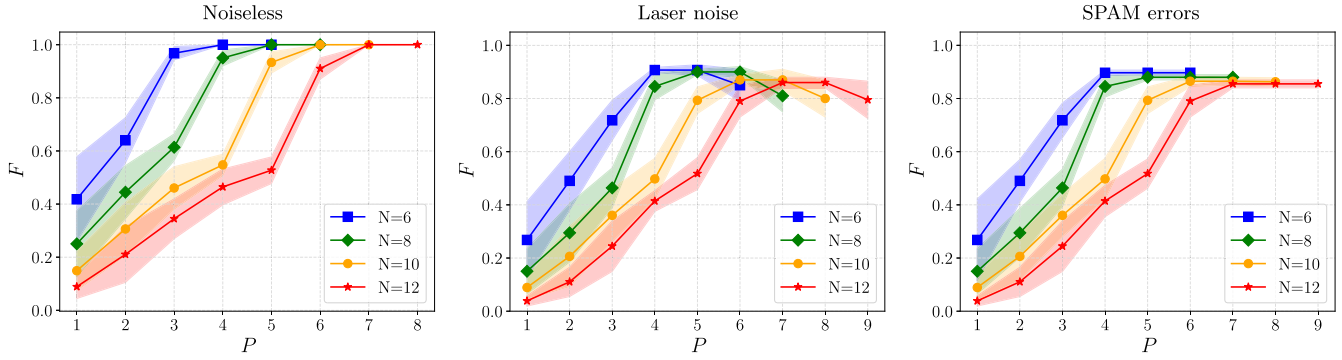


FIG. 3. Growth of fidelity as a function of depth P for various system sizes ($N = [6, 8, 10, 12]$) and under three different scenarios: noiseless, laser noise, and SPAM errors. We used the quantum natural gradient as the optimizer. It is noteworthy that, in each case, the highest value of F is achieved when $P = \lfloor N/2 \rfloor + 1$.

When two different sources of noise are introduced, we find that the optimal fidelity is consistently reached at $P = \lfloor N/2 \rfloor + 1$, with accuracy ranging from 10^{-7} to 10^{-4} , as N changes. Additionally, the plots include fidelity values for two extra layers to evaluate whether this might help the algorithm in achieving the desired ground state. However, the results indicate that adding further layers does not provide any advantage: In the case of laser noise, the fidelity value stabilizes at the next layer after $P = \lfloor N/2 \rfloor + 1$ (though with increased variance), followed by a subsequent decrease. For spam errors, both the fidelity and its variance remain constant for $P \geq \lfloor N/2 \rfloor + 1$. This can be easily understood as adding parameters does not increase noise, unlike the situation with laser noise evolution.

B. Comparison of QNG against vanilla gradient descent

We evaluated the performance of QNG and standard gradient descent in finding the true ground state of 1D TFIM by testing both methods across various system sizes, setting $h = 0.5$ and $J = 1$. The simulations were conducted under three different conditions: noiseless, with laser noise, and with SPAM errors. The maximum number of iterations allowed for the optimization was set to 5000 while the convergence criterion to stop the optimization procedure, according to Eq. (7), was $\epsilon_{\text{stop}} = 10^{-12}$. The learning rate η was selected iteratively by conducting some trials with a fixed N and choosing the best-performing values for both methods.

Furthermore, the QNG was computed using both the full QFIM and the diagonal approximation (which utilizes only the diagonal elements of the QFIM). This comparison is crucial for understanding the scenarios in which it is feasible to approximate the QFIM, thus calculating fewer elements. This is significant because the QFIM must be inverted during the parameter update process, which incurs in additional computational overhead.

1. Noiseless case

The noiseless case results are shown in Fig. 4, where two primary figures of merit were analyzed: the *average number of steps* required to reach the ground state and the *convergence rate*, representing the percentage of successful algorithm

completions in finding the ground state. Success in reaching the ground state is defined by achieving an accuracy $\delta E_{\text{opt}} < 10^{-9}$, where δE_{opt} represents the best accuracy obtained after optimization. These values were calculated by running the algorithm from 50 randomly chosen initial points.

An analysis of the average number of steps reveals a performance gap between vanilla and QNG methods (both diagonal approximation and full matrix). Initially, this gap is small for smaller system sizes but grows as N increases. Notably, within the simulated sizes, the difference between QNG with the diagonal approximation and the full matrix appears minimal at first but becomes more pronounced starting from $N = 9$.

Figure 4 also indicates an increase in variance as the number of parameters P increases, attributable to the larger variety of initial points randomly selected. Here the distinction between the classical optimizer and its quantum counterparts becomes apparent as well. Moreover, it is important to highlight that the performance of the vanilla and QNG diagonal approximation is even worse than that of the full matrix approach, as they are less robust against random initialization.

In terms of convergence rate (Fig. 4, right panel), it varies significantly among the methods. Vanilla’s success rate declines from 88% at $N = 5$ to 56% at $N = 12$. In contrast, QNG without approximation maintains nearly 100% success up to $N = 11$, with only a slight reduction to 98% at $N = 12$. The gap between the two quantum methods is more noticeable here, as the success rate for QNG with the diagonal approximation gradually decreases, reaching 88% at $N = 12$.

Figure 5 shows the distribution of 50 accuracy values as a function of system size N . The most frequent accuracy is 10^{-10} , matching the percentage of successful outcomes shown in the right panel of Fig. 4. The accuracy range for vanilla widens as N increases, reaching approximately 10^{-6} by $N = 12$, indicating a performance decline. By contrast, the QNG methods, especially the full matrix, demonstrate a significantly smaller spread, suggesting stability and robustness. This distribution highlights the higher variability and lower consistency of the vanilla method, particularly at larger N , while QNG methods remain narrowly distributed around lower values. Finally, a comparison of the two QNG methods shows that while the full QNG consistently yields lower errors, the diagonal approximation offers a practical

Noiseless

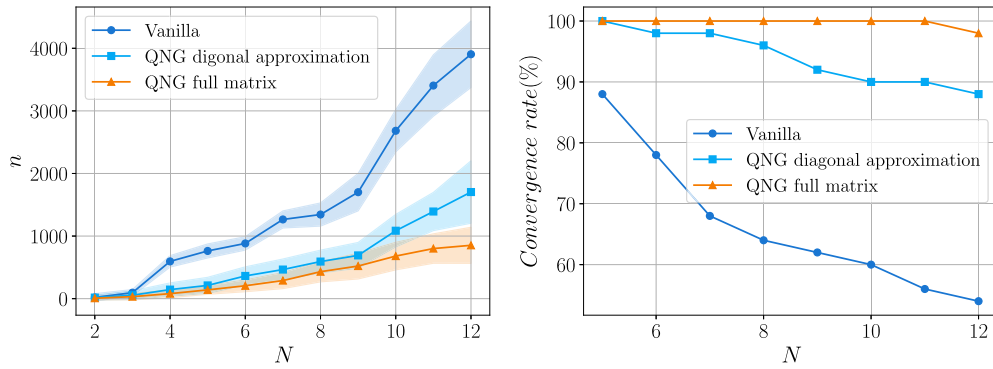


FIG. 4. Comparison of performance metrics for the vanilla optimizer and QNG methods (both diagonal approximation and full matrix) in the noiseless case. The left panel shows the average number of steps, n , required to reach the ground state as a function of the system size $N = [2, 12]$. The right panel displays the convergence rate of each method, reflecting the percentage of successful completions in reaching the ground state. Results demonstrate the growing performance gap between vanilla and QNG as N increases, with the QNG methods maintaining higher success rates and lower step requirements, particularly the QNG full matrix approach.

trade-off. Despite a slight decrease in accuracy as N increases, the diagonal approximation still outperforms vanilla, making it a viable choice when computational resources are limited.

2. Laser noise and SPAM errors cases

In the following subsection, we discuss the numerical results obtained when including both laser noise and SPAM errors in the simulations, using the Python library `pulser` [44], which was developed to simulate Rydberg atom arrays and allows for precise control over parameters characterizing different sources of noise. Specifically, we kept the default temperature $T = 50 \mu\text{K}$ to account for Doppler effects, as well as the Gaussian standard deviation for the laser waist, `laser_waist = 175 μm` , and amplitude fluctuations, `amp_sigma = 0.05`, which are representative values for

current experimental setups (see also the `pulser` documentation [44] for further details). Also for the SPAM errors, we used the default values: $\eta = 0.005$, $\epsilon = 0.01$, and $\epsilon' = 0.05$, which represent the probability of state preparation errors and the probabilities of measurement errors (false positives and false negatives), respectively.

Figures 6 and 7 illustrate the effects of laser noise and SPAM errors on the performance of three methods—vanilla, QNG with diagonal approximation, and QNG with full matrix—as a function of system size N .

Figure 6 highlights the average number of steps n required to reach the convergence. Under both noise types, the vanilla method (blue curves) exhibits a steep increase in n as N grows, along with a wide confidence interval, indicating high sensitivity to noise and inconsistent performance. The QNG diagonal approximation (green curves) provides a notable improvement by reducing the number of steps and

Noiseless accuracies

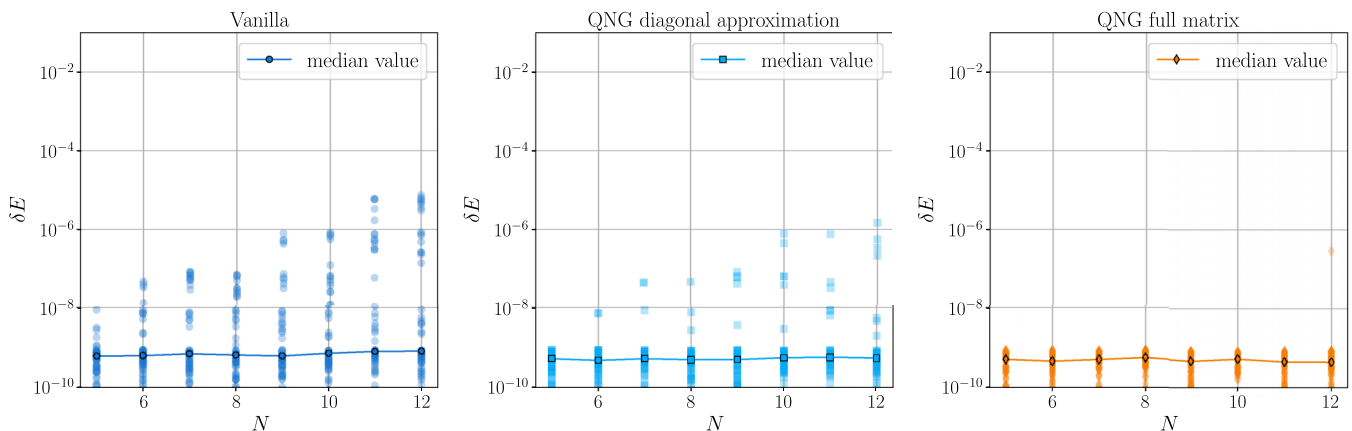


FIG. 5. Comparison of noiseless accuracies for the vanilla, QNG diagonal approximation, and QNG full matrix methods as a function of system size N . Each panel displays the distribution of 50 accuracy values (measured as δE) for each N , obtained from different runs, with the median value of accuracies represented by solid lines. The most populated accuracy sector is at 10^{-10} , indicating successful convergence to the ground state, especially for the QNG Full Matrix which is more consistent as N increases.

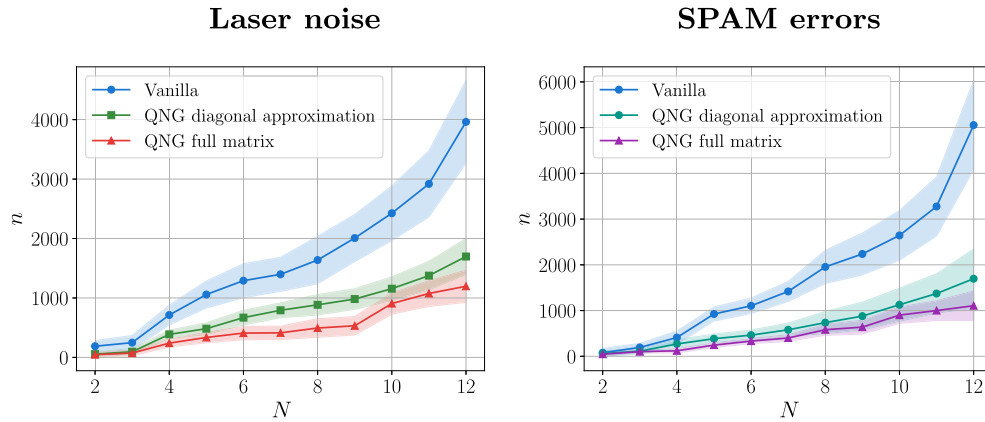


FIG. 6. The effect of laser noise (left) and SPAM errors (right) on the average number of steps n required to reach the cost function minimum as a function of system size N . The three methods compared are vanilla (blue), QNG with diagonal approximation (green), and QNG with full matrix (red/purple). The QNG full matrix method shows the best performance, requiring fewer steps and demonstrating lower sensitivity to noise, while the vanilla method performs the worst, with greater sensitivity and rapid growth of n as N increases.

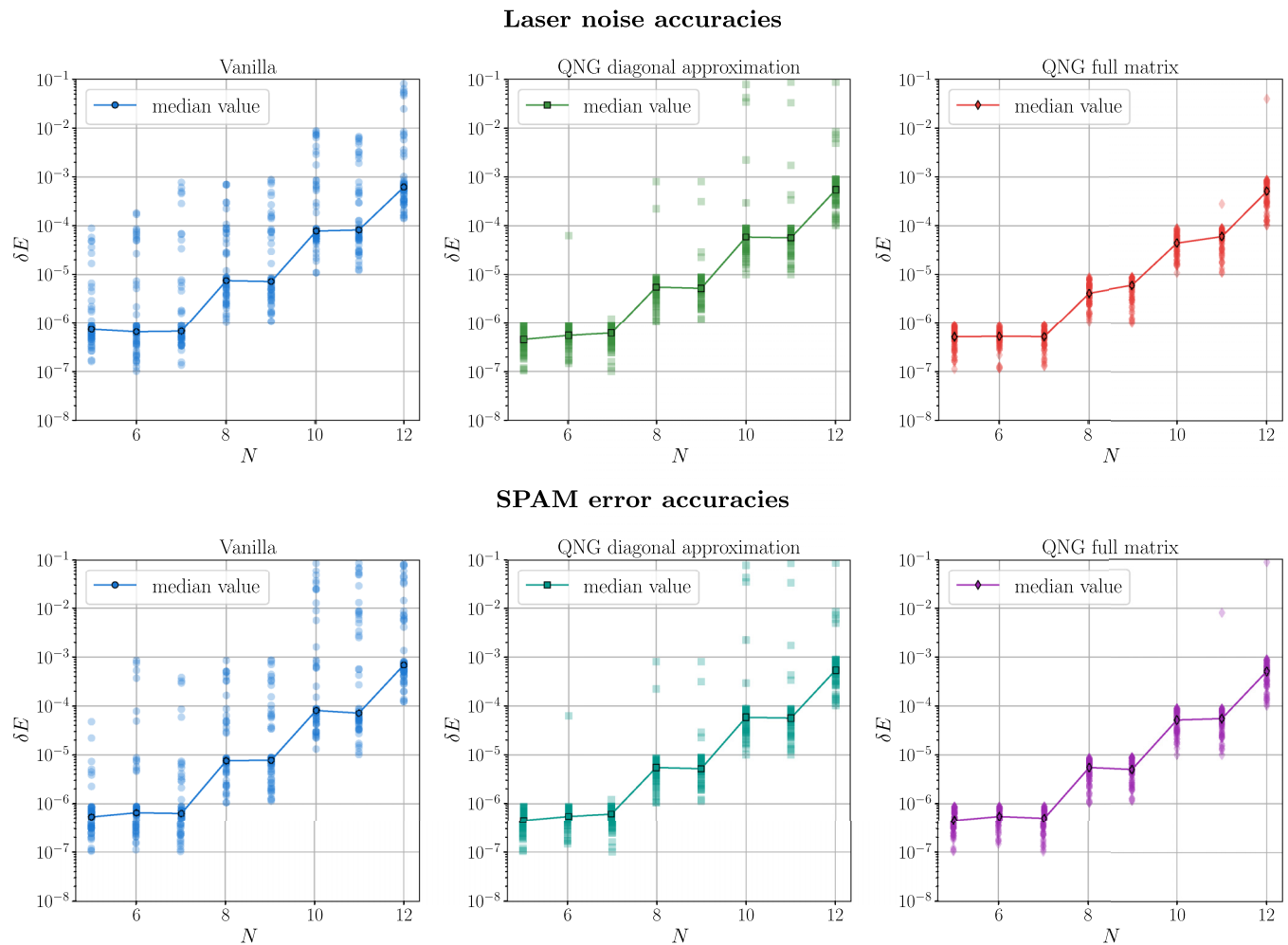


FIG. 7. The accuracy distribution δE across 50 runs as a function of system size N for three methods—vanilla (left), QNG with diagonal approximation (center), and QNG with full matrix (right)—under laser noise (top row) and SPAM errors (bottom row). The QNG full matrix method achieves the lowest δE with minimal variance, indicating superior stability and robustness to noise, while the vanilla method shows significant growth in δE and wide distributions, highlighting its high sensitivity to noise.

showing slower growth of n with N . Its narrower confidence interval suggests moderate robustness to noise. The QNG full matrix method (red/purple curves) demonstrates the best performance, with n remaining relatively low and growing only slightly with N . Additionally, its narrow confidence interval reflects strong resilience to both laser noise and SPAM errors. These results underline the efficiency and stability of the QNG methods, particularly the full matrix variant, in mitigating noise-related challenges during optimization

Figure 7 provides further insights by presenting the accuracy distribution δE across 50 runs for the same three methods. For laser noise (top row), the vanilla method (left) shows a sharp increase in δE with N , along with a wide distribution spanning several orders of magnitude. This indicates its pronounced sensitivity to noise and poor reliability. In contrast, the QNG diagonal approximation (center) reduces variance and the median is slightly below compared to the vanilla one, reflecting enhanced robustness to laser noise. The QNG full matrix method (right) delivers superior performance, maintaining low δE values with minimal spreadness of them, demonstrating very good stability and accuracy.

V. QUANTUM CIRCUIT IMPLEMENTATION AND IBM NOISE MODEL

In this section, we describe the quantum circuit implementation for the QAOA and the integration of a realistic noise model based on calibration data from the IBM quantum device, `ibm_sherbrooke`. This noise modeling approach enables a more realistic assessment of QAOA's performance for the TFIM under hardware constraints. The use of device-specific noise models provides valuable insights into how noise affects the algorithm's convergence and accuracy in practical scenarios.

A. Circuit construction and Hamiltonian implementation

For our QAOA implementation, we employed the PennyLane library [46], which enables efficient construction, simulation, and optimization of quantum circuits. The QAOA ansatz for solving the ground state of the TFIM consists of alternating applications of two Hamiltonians, as described in Sec. II A.

In this circuit-based implementation, we use PennyLane's Ising ZZ two-qubit gates:

$$ZZ(\gamma) = \exp\left(-i\frac{\gamma}{2}\sigma_z \otimes \sigma_z\right),$$

for the interacting terms appearing in the cost Hamiltonian, and single-qubit R_x rotations:

$$R_x(\beta) = \exp\left(-i\frac{\beta}{2}\sigma_x\right),$$

for the transverse terms representing the mixing Hamiltonian. Here the trainable parameters γ and β represent the strength of the interaction and the angle of the rotation, respectively.

Each QAOA layer is composed of a sequence of these gates, where all R_x gates in a layer share a common trainable

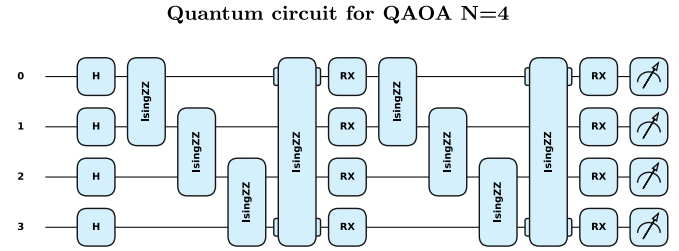


FIG. 8. An example of the quantum circuit for QAOA with $N = 4$, implemented using the PennyLane library.

parameter β , and all ZZ gates share a common trainable parameter θ . Importantly, all ZZ gates within a layer commute, which enables an efficient application of the interaction terms. A particular example of the quantum circuit for QAOA with $N = 4$ is depicted in Fig. 8.

B. Noise model derived from IBM device calibration data

To simulate the realistic behavior of quantum circuits on physical hardware, we incorporated a noise model based on `ibm_sherbrooke`'s calibration data. Using Qiskit's Aer simulator [47], we utilized the backend properties of `ibm_sherbrooke` to define noise characteristics, which were then applied during circuit execution. The calibration data for this specific device is provided in Table I. This noise model accounts for several key factors:

(1) Gate error probability: The likelihood of errors for each basis gate, derived from calibration data, is specific to each qubit and gate type.

(2) Gate duration: Each gate's execution time, contributing to noise exposure, is recorded from device properties.

(3) Relaxation time constants: The T_1 (relaxation) and T_2 (dephasing) times for each qubit are included, allowing for accurate modeling of qubit decoherence during computation.

(4) Readout error probability: This factor represents the probability of classical bit-flip errors in measurement results, based on qubit-specific readout performance.

Model composition

The noise model is implemented with the following components:

(1) Single-qubit gate errors: Simulated as a combination of a single-qubit depolarizing error followed by a thermal relaxation error. The thermal relaxation parameters are obtained from each qubit's T_1 , T_2 , and gate duration.

(2) Two-qubit gate errors: Modeled as a two-qubit depolarizing error followed by thermal relaxation errors applied to both qubits in the gate.

(3) Measurement errors: Single-qubit readout errors, modeled as classical bit-flip probabilities, are applied to the measured outcomes to simulate imperfections in readout fidelity.

To match the real device's noise characteristics, the depolarizing and thermal relaxation parameters are adjusted so that the composite error rate reflects the reported gate error in the backend properties. This noise model dynamically updates based on calibration data from `ibm_sherbrooke`, ensuring

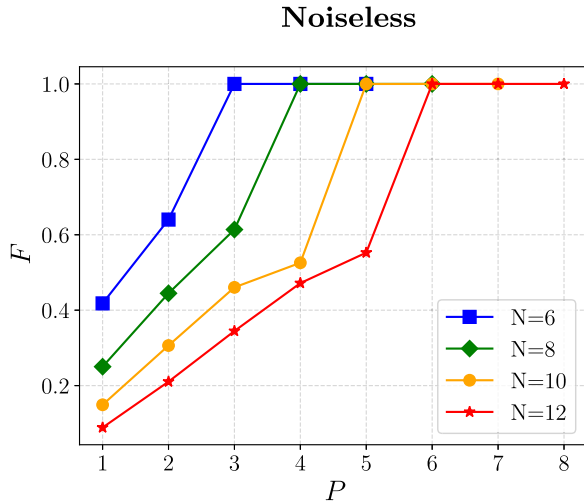


FIG. 9. Fidelity growth as a function of circuit depth P for various system sizes ($N = [6, 8, 10, 12]$) in the noiseless case for the quantum circuit implementation. Consistent with previous numerical studies, the highest fidelity F is achieved at $P = \lfloor N/2 \rfloor$.

it remains accurate across different device calibrations and closely mirrors real device behavior.

VI. NUMERICAL RESULTS

In this section, we report numerical experiments conducted using quantum circuits with a realistic noise model derived from an IBM quantum processor. These simulations are organized into two primary categories: (1) analysis of the fidelity of the QNG optimizer with the full Fubini-Study metric across varying circuit depths and (2) comparison of the QNG optimizer (using both the full Fubini-Study metric and its diagonal approximation) against vanilla gradient descent, both in the presence and absence of noise.

A. Ground-state preparation

In this experiment, we investigate how fidelity varies with the number of layers P for different system sizes under noiseless conditions. The results are shown in Fig. 9. In contrast to results obtained on the Rydberg atoms platform, our noiseless

simulations indicate that the ground state can be achieved with an accuracy of 10^{-10} when $P = \lfloor N/2 \rfloor$, as predicted by numerical results [6]. However, in the presence of noise, we were unable to reproduce the same experiment, as the current Qiskit plugin for the PennyLane library does not support outputting density matrices from the quantum circuit. Nevertheless, we expect a similar trend as in the previous implementation on the Rydberg atom platform: increasing the number of layers also increases the noise in the quantum circuit, leading to lower fidelity.

B. Comparison of QNG against vanilla gradient descent

We compared the performance of the QNG optimizer, using both the full Fubini-Study metric and its diagonal approximation, against Vanilla Gradient Descent in both noiseless (see Fig. 10) and noisy scenarios.

I. Noiseless case

In the noiseless case, we conducted simulations for system sizes $N = [2, 12]$ and ran 50 trials with different initial conditions to assess the robustness of the optimizers against random initializations. Additionally, we aimed to evaluate the extent to which the diagonal approximation allows the QNG optimizer to outperform vanilla gradient descent. For the noiseless simulations, we used a learning rate of $\eta = 0.01$, and the maximum number of iterations for the optimization was 5000. Moreover, the convergence criterion to stop the optimization procedure, according to Eq. (7), was $\epsilon_{\text{stop}} = 10^{-12}$, while the convergence criterion to the true solution was the same as for Rydberg atoms, defined as $\delta E_{\text{opt}} < 10^{-9}$, where δE_{opt} represents the best accuracy obtained after optimization.

As with the simulations on the Rydberg atoms platform, we plotted the average number of steps required for each optimizer to converge, along with the standard deviation across different system sizes. We also plotted the convergence rate for each optimizer across these system sizes, as shown in Fig. 11.

The results indicate that quantum natural gradient consistently outperforms vanilla gradient descent, exhibiting both faster average convergence and greater robustness to random initializations, as demonstrated by reduced variance in the number of optimization steps. Additionally, while the

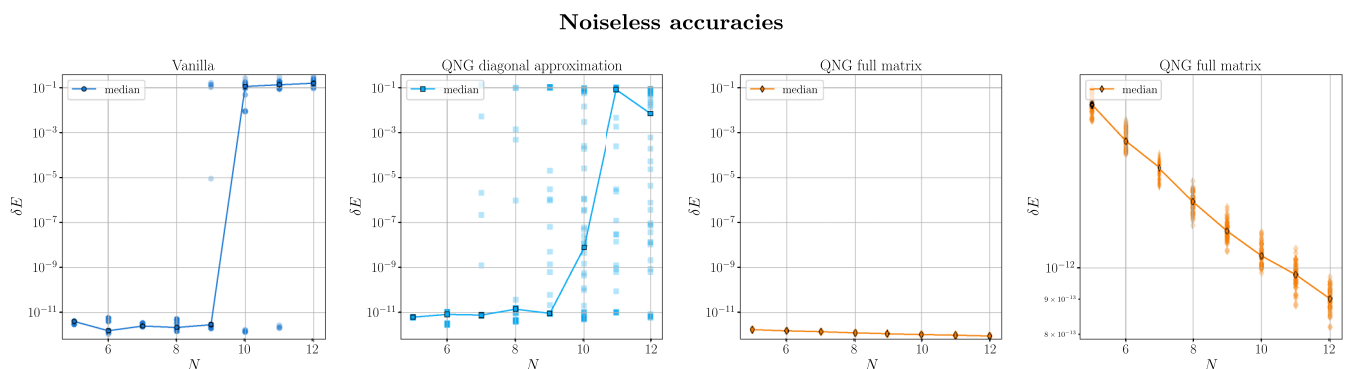


FIG. 10. Comparison of noiseless accuracies for the vanilla, QNG diagonal approximation, and QNG full matrix methods as a function of system size N . Each panel displays the distribution of 50 accuracy values, measured as δE , with the median of the accuracy represented by solid points. The rightmost plot is a zoomed-in view of the QNG full matrix accuracies.

Noiseless

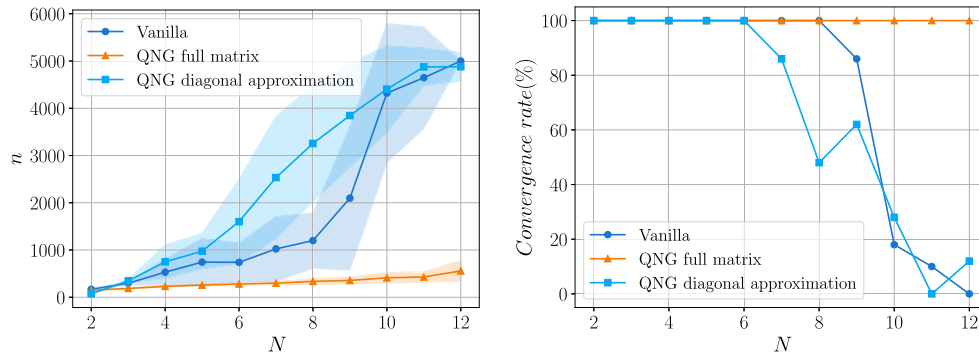


FIG. 11. On the left, the average number of steps n needed to reach the ground state as function of the systems size $N = [2, 12]$ in the noiseless scenario. On the right, the convergence rate of QNG (full and with diagonal approximation) and vanilla in the noiseless scenario as function of the system size $N = [2, 12]$.

convergence rate of quantum natural gradient remains at 100%, the convergence rate of vanilla declines significantly with increasing system size, highlighting the superior performance of quantum natural gradient for larger problem sizes. However, for small system sizes, the performance of QNG with the diagonal approximation is inferior to that of vanilla, exhibiting a similar decline in convergence rate. After $N = 9$ spins, QNG begins to outperform vanilla in terms of convergence rate, achieving a 20% convergence rate compared to the 0% convergence rate of vanilla. Thus, we conclude that in this specific implementation of QAOA applied to the TFIM using quantum circuits, the diagonal approximation only becomes significant in surpassing vanilla when the system size increases, leading to improved convergence. Meanwhile, QNG with the full Fubini-Study metric consistently achieves better results on average across noiseless experiments for all system sizes. This highlights the theoretical advantage of QNG, and its diagonal approximation, over vanilla.

Moreover, we note that the strong robustness of QNG against random initializations and its reduced variance in optimization steps can be attributed to a distance regularization phenomenon, as mentioned earlier. This effect suggests that QNG is effectively regularizing distances within the space of quantum states. Additionally, the superior convergence rate of QNG may also indicate its ability to avoid certain local minima [17].

Additionally, as in the numerical experiments on the Rydberg atom platform, we plotted the distribution of 50 accuracy values as a function of system size N for the different optimizers (Fig. 5). We observe that the QNG optimizer with the full Fubini-Study metric is the only one that maintains low accuracy with minimal variability as the system size increases, remaining below 10^{-12} . In contrast, the vanilla optimizer and QNG with the diagonal approximation perform poorly as the system size increases and exhibit a wider distribution of accuracy values.

Moreover, this reduced variance highlights the robustness of QNG against random initializations—regardless of the initial parameter configurations, QNG consistently converges close to the true solution. Regarding the comparison between the diagonal approximation and vanilla, we observe that al-

though both perform significantly worse than QNG with the full matrix, QNG with the diagonal approximation achieves solutions closer to the true one, with a distribution of values more shifted towards the correct solution. Thus, we conclude that even when using the diagonal approximation, better results can be obtained compared to the vanilla optimizer.

2. Noisy case

For the noisy case, due to technical limitations related to the online connectivity required to obtain the noise model provided by IBM, which had to be loaded at each optimization step, we were only able to run simulations for system sizes $N = [2, 8]$ and for 50 initial conditions. This limitation arose because the prolonged runtime of each optimization step for larger system sizes increased the likelihood of connectivity loss, leading to errors in retrieving the noise model. Nevertheless, the trend is clear enough that additional simulations are unnecessary to conclude that, in the noisy case, the algorithms with the different optimizations perform equally poor, with QNG providing no advantage. In the optimization, we used the same learning rate η and maximum number of iterations as in the noiseless case. However, the convergence criterion

Noisy

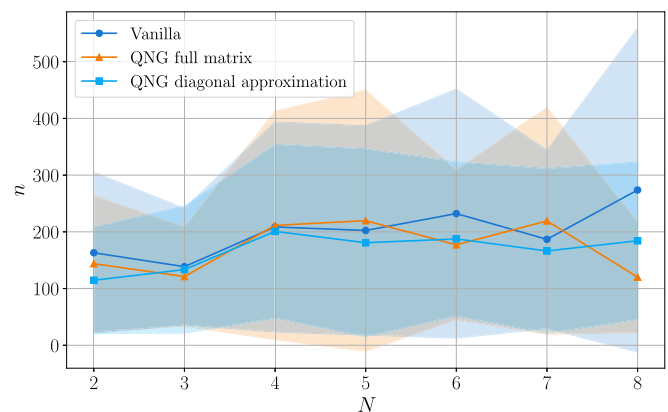


FIG. 12. Average number of steps n needed for convergence as function of the system's size $N = [2, 8]$ in presence of noise.

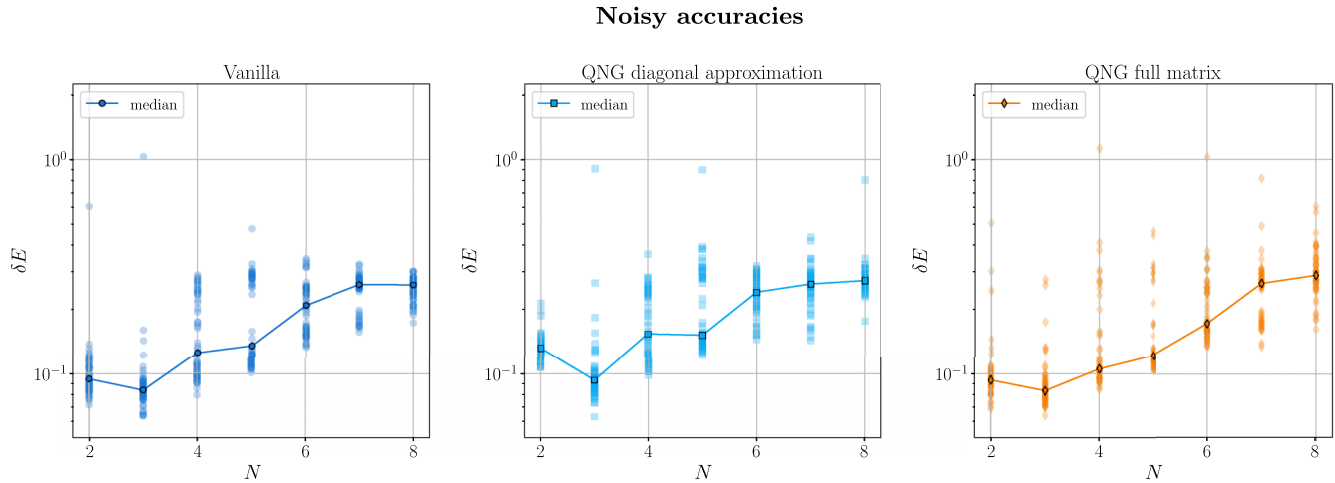


FIG. 13. Comparison of noisy accuracies for the vanilla, QNG diagonal approximation, and QNG full matrix methods as a function of system size N . Each panel displays the distribution of 50 accuracy values, measured as δE , with the median of the accuracy represented by solid points.

and the condition for achieving the true solution were relaxed to account for potential imperfections caused by noise. These were set to $\epsilon_{\text{stop}} = 10^{-8}$ and $\epsilon_{\text{true}} = 10^{-6}$, respectively.

In these experiments, due to the inability of the algorithms to reach the true solution because of noise, we decided not to plot the convergence rate but rather the accuracy [see (3)] obtained from the 50 runs for each optimizer. We can observe the plot for the average number of steps for the different optimizers in Fig. 12. Additionally, the accuracy plots for the different optimizers are shown in Fig. 13.

From the accuracy plot, it is evident that none of the optimizers is able to prepare the true ground state of the TFIM using this realistic noise model, as indicated by the high values of accuracy. As previously mentioned, this may arise because this problem is not well suited for noisy digital quantum computers, resulting in gate decomposition into native gates that leads to overhead. This, combined with high noise levels, yields very poor outcomes for the quantum algorithm. Furthermore, the trend in the average number of steps is relatively similar across the three optimizers. Thus, we can conclude that for QAOA applied to the TFIM using quantum circuits with a realistic noise model from an IBM quantum device, QNG offers no advantage over vanilla gradient descent, as both optimizers yield unsatisfactory results.

VII. CONCLUSIONS AND OUTLOOK

This work evaluated the performance of the QNG optimizer within the QAOA, benchmarking its efficacy on two prominent quantum computing platforms: Rydberg atoms and superconducting circuits. The results provide significant insights into the potential of the QNG optimizer to enhance the optimization of variational quantum algorithms, even in the presence of noise, such as that encountered in NISQ devices.

Across both platforms, QNG consistently outperformed vanilla gradient descent, achieving faster convergence and greater robustness to random initializations. This suggests that QNG effectively regularizes distances within the space of quantum states. Furthermore, QNG demonstrated a higher

convergence rate to the true solution, successfully avoiding certain local minima. On the Rydberg atoms platform, the diagonal approximation of the quantum Fisher information offered a computationally efficient alternative without significantly compromising performance, making it a viable choice for larger systems.

Rydberg atom platforms demonstrated superior performance compared to superconducting circuits. The analog implementation on Rydberg systems eliminates the need for local qubit addressing, a common experimental challenge in digital platforms. Furthermore, Rydberg platforms exhibited remarkable robustness against noise, enabling ground-state preparation with high fidelity under realistic conditions. In contrast, superconducting circuits struggled under noise, with decoherence effects so severe that ground-state preparation became unattainable. This limitation was further exacerbated by the gate decomposition overhead required to implement QAOA on digital hardware, which increases circuit depth and amplifies noise effects.

Simulations on superconducting platforms highlighted their sensitivity to noise. Despite QNG's theoretical advantages, the combined effects of gate decomposition and noise rendered the algorithm ineffective for state preparation in this setting.

While the quantum natural gradient appears to be a very promising method for enhancing variational algorithms, it is important to consider the cost associated with computing the full quantum Fisher information matrix, which becomes impractical for a large number of parameters. Future research should focus on finding methods to reduce the scaling of the computation of the full QFIM. This could include restricting the approach to specific variational architectures with favorable scaling or developing improved protocols to compute it using fewer quantum resources.

Additionally, new approximations to the QFIM should be investigated, particularly for efficiently computing the off-block-diagonal terms. A natural next step for this work would be to incorporate shot noise into the numerical simulations and study how the inverse QFIM behaves under these conditions,

as well as how this affects the performance of QNG. Another potential research direction would be to explore ways to derive a simpler expression for the QFIM in the presence of noise, beyond its current formulation in terms of the diagonalization of the density matrix.

In summary, the QNG optimizer is a highly advantageous method for enhancing the optimization of variational algorithms. It enables faster average convergence, greater robustness against random parameter initializations, and an improved ability to avoid certain local minima. Furthermore, our results demonstrate that QNG exhibits robustness against realistic noise during optimization. Thus, we firmly believe that incorporating QNG for the optimization of variational quantum algorithms is highly promising, provided that significant progress is made in reducing the computational cost of gradient evaluation and QFIM estimation on real hardware.

Finally, we want to emphasize that, despite the strong potential of QNG—offering more stable convergence and improved resilience to noise—these benefits come with significant computational costs. In particular, evaluating the QFIM, whether exactly or approximately, introduces substantial overhead in both classical post-processing and quantum measurement resources. As a result, the practical use of QNG is currently limited to small-scale simulations or hypothetical future scenarios where these costs may become more manageable.

ACKNOWLEDGMENTS

This work was supported by the Project No. PID2023-152724NA-I00, with funding from

MCIU/AEI/10.13039/501100011033 and FSE+, the Severo Ochoa Grant No. CEX2023-001292-S, Generalitat Valenciana Grant No. CIPROM/2022/66, the Ministry of Economic Affairs and Digital Transformation of the Spanish Government through the QUANTUM ENIA project call - QUANTUM SPAIN project, and by the European Union through the Recovery, Transformation and Resilience Plan—NextGenerationEU within the framework of the Digital Spain 2026 Agenda, and by the CSIC Interdisciplinary Thematic Platform (PTI+) on Quantum Technologies (PTI-QTEP+). This project has also received funding from the European Union’s Horizon 2020 research and innovation program under Grant Agreement No. CaLIGOLA MSCA-2021-SE-01-101086123. R.G.-L. is funded by Grant No. CIACIF/2021/136 from Generalitat Valenciana. The work was also partially supported by INFN through the project QUANTUM.

DATA AVAILABILITY

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

APPENDIX: CALIBRATION DATA FROM THE IBM DEVICE `ibm_sherbrooke`

In this Appendix, we present the calibration data from the IBM device `ibm_sherbrooke`, which were used as parameters for our noise model during circuit execution. The calibration data are shown in Table I.

TABLE I. Calibration data from the IBM quantum device `ibm_sherbrooke`, which was used to construct the realistic noise model. The ECR gate, used as a basis gate in IBM quantum processors, is a maximally entangling gate and is equivalent to a CNOT gate up to single-qubit prerotations. The numbers indicate the qubits to which the gate is applied and the order in which it is applied. Note that blank spaces for the ECR gate in some rows indicate that the gate is only available in the specific order given by the table.

Qubit	T1 (μs)	T2 (μs)	Prob		Readout		RZ			ECR error	Gate time (ns)
			meas0 prep1	meas1 prep0	assignment error	ID error	error	SX error	X error		
0	351.63	252.89	0.26	0.23333	0.24667	0.0001205	0	0.0001205	0.0001205		
1	391.24	383.13	0.0172	0.0228	0.02	0.0001565	0	0.0001565	0.0001565	ecr1_2:0.0049;ecr1_0:0.0064	ecr1_2:533.333;ecr1_0:533.333
2	268.69	170.39	0.0194	0.0338	0.0266	0.0002395	0	0.0002395	0.0002395		
3	281.24	259.83	0.53667	0.35	0.44333	0.0002122	0	0.0002122	0.0002122	ecr3_2:0.00619	ecr3_2:533.333
4	410.16	440.13	0.65667	0.2	0.42833	0.0001117	0	0.0001117	0.0001117	ecr4_3:0.00444	ecr4_3:533.333
5	263.77	290.03	0.58	0.30333	0.44167	0.0002235	0	0.0002235	0.0002235	ecr5_4:0.00742	ecr5_4:533.333
6	320.37	263.67	0.56	0.28333	0.42167	0.0002429	0	0.0002429	0.0002429	ecr6_5:0.00575	ecr6_5:533.333
7	342.18	287.59	0.27333	0.46	0.36667	0.0034641	0	0.0034641	0.0034641	ecr7_6:0.00762;ecr7_8:0.01442	ecr7_6:533.333;ecr7_8:533.333
8	246.37	247.17	0.49	0.30667	0.39833	0.0001891	0	0.0001891	0.0001891	ecr8_9:0.0054	ecr8_9:533.333
9	397.17	69.63	0.59333	0.31	0.45167	0.0006543	0	0.0006543	0.0006543		
10	272.59	192.26	0.0134	0.0044	0.0089	0.0002901	0	0.0002901	0.0002901	ecr10_11:0.00676;ecr10_9:0.01087	ecr10_11:533.333;ecr10_9:533.333
11	250.45	93.71	0.45667	0.41333	0.43500	0.0001855	0	0.0001855	0.0001855		

- [1] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, *New J. Phys.* **18**, 023023 (2016).
- [2] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, *Nat. Commun.* **5**, 4213 (2014).
- [3] N. Moll, P. Barkoutsos, L. S. Bishop, J. M. Chow, A. Cross, D. J. Egger, S. Filipp, A. Fuhrer, J. M. Gambetta, M. Ganzhorn *et al.*, Quantum optimization using variational algorithms on near-term quantum devices, *Quantum Sci. Technol.* **3**, 030503 (2018).
- [4] J. Preskill, Quantum computing in the NISQ era and beyond, *Quantum* **2**, 79 (2018).
- [5] S. de Léséleuc, D. Barredo, V. Lienhard, A. Browaeys, and T. Lahaye, Analysis of imperfections in the coherent optical excitation of single atoms to Rydberg states, *Phys. Rev. A* **97**, 053803 (2018).
- [6] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, [arXiv:1411.4028](https://arxiv.org/abs/1411.4028).
- [7] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices, *Phys. Rev. X* **10**, 021067 (2020).
- [8] S. Tibaldi, L. Leclerc, D. Vodola, E. Tignone, and E. Ercolessi, Analog QAOA with Bayesian optimisation on a neutral atom QPU, [arXiv:2501.16229](https://arxiv.org/abs/2501.16229).
- [9] S. Ebadi, A. Keesling, M. Cain, T. T. Wang, H. Levine, D. Bluvstein, G. Semeghini, A. Omran, J.-G. Liu, R. Samajdar *et al.*, Quantum optimization of maximum independent set using Rydberg atom arrays, *Science* **376**, 1209 (2022).
- [10] M. P. Harrigan, K. J. Sung, M. Neeley, K. J. Satzinger, F. Arute, K. Arya, J. Atalaya, J. C. Bardin, R. Barends, S. Boixo *et al.*, Quantum approximate optimization of non-planar graph problems on a planar superconducting processor, *Nat. Phys.* **17**, 332 (2021).
- [11] G. Pagano, A. Bapat, P. Becker, K. S. Collins, A. De, P. W. Hess, H. B. Kaplan, A. Kyprianidis, W. L. Tan, C. Baldwin, L. T. Brady, A. Deshpande, F. Liu, S. Jordan, A. V. Gorshkov, and C. Monroe, Quantum approximate optimization of the long-range Ising model with a trapped-ion quantum simulator, *Proc. Natl. Acad. Sci. USA* **117**, 25396 (2020).
- [12] H.-Y. Huang, R. Kueng, and J. Preskill, Predicting many properties of a quantum system from very few measurements, *Nat. Phys.* **16**, 1050 (2020).
- [13] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, *Nat. Commun.* **9**, 4812 (2018).
- [14] G. Wang, D. E. Koh, P. D. Johnson, and Y. Cao, Minimizing estimation runtime on noisy quantum computers, *PRX Quantum* **2**, 010346 (2021).
- [15] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, and P. J. Coles, Cost function dependent barren plateaus in shallow parametrized quantum circuits, *Nat. Commun.* **12**, 1791 (2021).
- [16] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, Quantum natural gradient, *Quantum* **4**, 269 (2020).
- [17] D. Wierichs, C. Gogolin, and M. Kastoryano, Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer, *Phys. Rev. Res.* **2**, 043246 (2020).
- [18] A. Lopatnikova and M.-N. Tran, Quantum natural gradient for variational Bayes, [arXiv:2106.05807](https://arxiv.org/abs/2106.05807).
- [19] A. Roy, S. Erramilli, and R. M. Konik, Efficient quantum circuits based on the quantum natural gradient, *Phys. Rev. Res.* **6**, 043083 (2024).
- [20] E. Ercolessi, G. Marmo, and G. Morandi, From the equations of motion to the canonical commutation relations, *Riv. Nuovo Cimento* **33**, 401 (2010).
- [21] B. Koczor and S. C. Benjamin, Quantum natural gradient generalized to noisy and nonunitary circuits, *Phys. Rev. A* **106**, 062416 (2022).
- [22] J. R. Johansson, P. D. Nation, and F. Nori, Qutip 2: A python framework for the dynamics of open quantum systems, *Comput. Phys. Commun.* **184**, 1234 (2013).
- [23] E. Lieb, T. Schultz, and D. Mattis, Two soluble models of an antiferromagnetic chain, *Ann. Phys.* **16**, 407 (1961).
- [24] S. Ruder, An overview of gradient descent optimization algorithms, [arXiv:1609.04747](https://arxiv.org/abs/1609.04747).
- [25] S.-i. Amari, Natural gradient works efficiently in learning, *Neural Comput.* **10**, 251 (1998).
- [26] S.-i. Amari, R. Karakida, and M. Oizumi, Fisher information and natural gradient learning in random deep networks, in *The 22nd International Conference on Artificial Intelligence and Statistics* (PMLR, 2019), pp. 694–702.
- [27] S.-i. Amari, Neural learning in structured parameter spaces—natural Riemannian gradient, *Adv. Neural Inf. Process. Syst.* **9**, 127 (1996).
- [28] I. Goodfellow, Y. Bengio, and A. Courville, Regularization for deep learning, *Deep Learning* (MIT Press, Cambridge, MA, 2016), p. 216.
- [29] S.-i. Amari, H. Park, and K. Fukumizu, Adaptive method of realizing natural gradient learning for multilayer perceptrons, *Neural Comput.* **12**, 1399 (2000).
- [30] I. Bengtsson and K. Życzkowski, *Geometry of Quantum States: An Introduction to Quantum Entanglement* (Cambridge University Press, Cambridge, UK, 2006).
- [31] C. W. Helstrom, Quantum detection and estimation theory, *J. Stat. Phys.* **1**, 231 (1969).
- [32] A. S. Holevo, *Probabilistic and Statistical Aspects of Quantum Theory* (North-Holland, Amsterdam, 1982).
- [33] J. Liu, H. Yuan, X.-M. Lu, and X. Wang, Quantum Fisher information matrix and multiparameter estimation, *J. Phys. A: Math. Theor.* **53**, 023001 (2020).
- [34] B. Koczor, Exponential error suppression for near-term quantum devices, *Phys. Rev. X* **11**, 031057 (2021).
- [35] W. J. Huggins, S. McArdle, T. E. O'Brien, J. Lee, N. C. Rubin, S. Boixo, K. B. Whaley, R. Babbush, and J. R. McClean, Virtual distillation for quantum error mitigation, *Phys. Rev. X* **11**, 041036 (2021).
- [36] T.-C. Yen, V. Verteletskyi, and A. F. Izmaylov, Measuring all compatible operators in one series of single-qubit measurements using unitary transformations, *J. Chem. Theory Comput.* **16**, 2400 (2020).
- [37] D. Wierichs, J. Izaac, C. Wang, and C. Y.-Y. Lin, General parameter-shift rules for quantum gradients, *Quantum* **6**, 677 (2022).
- [38] J. Bowles, D. Wierichs, and C.-Y. Park, Backpropagation scaling in parameterised quantum circuits, [arXiv:2306.14962](https://arxiv.org/abs/2306.14962).

- [39] R. Gómez-Lurbe, Efficient protocol to estimate the quantum Fisher information matrix for commuting-block circuits, [arXiv:2505.09818](https://arxiv.org/abs/2505.09818).
- [40] B. van Straaten and B. Koczor, Measurement cost of metric-aware variational quantum algorithms, *PRX Quantum* **2**, 030324 (2021).
- [41] A. Browaeys and T. Lahaye, Many-body physics with individually controlled Rydberg atoms, *Nat. Phys.* **16**, 132 (2020).
- [42] L. Henriët, L. Beguin, A. Signoles, T. Lahaye, A. Browaeys, G.-O. Reymond, and C. Jurczak, Quantum computing with neutral atoms, *Quantum* **4**, 327 (2020).
- [43] M. Saffman, Quantum computing with atomic qubits and Rydberg interactions: Progress and challenges, *J. Phys. B: At. Mol. Opt. Phys.* **49**, 202001 (2016).
- [44] L. Henriët *et al.*, Pulser documentation (2020), <https://pulser.readthedocs.io>.
- [45] H. Silvério, S. Grijalva, C. Dalyac, L. Leclerc, P. J. Karalekas, N. Shammah, M. Beji, L.-P. Henry, and L. Henriët, Pulser: An open-source package for the design of pulse sequences in programmable neutral-atom arrays, *Quantum* **6**, 629 (2022).
- [46] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi *et al.*, Pennylane: Automatic differentiation of hybrid quantum-classical computations, [arXiv:1811.04968](https://arxiv.org/abs/1811.04968).
- [47] A. Javadi-Abhari, M. Treinish, K. Krsulich, C. J. Wood, J. Lishman, J. Gacon, S. Martiel, P. D. Nation, L. S. Bishop, A. W. Cross *et al.*, Quantum computing with qiskit, [arXiv:2405.08810](https://arxiv.org/abs/2405.08810).