Variational quantum algorithms with exact geodesic transport

André J. Ferreira-Martins,[1,*] Renato M. S. Farias,[1,†] Giancarlo Camilo,[1] Thiago O. Maciel,[1] Allan Tosta,[1] Ruge Lin,[2,‡] Abdulla Alhajri,[1] Tobias Haug,[1] and Leandro Aolita[1]

[1] *Quantum Research Center, Technology Innovation Institute, Abu Dhabi, UAE*

[2] *Departament de Física Quàntica i Astrofísica and Institut de Ciències del Cosmos, Universitat de Barcelona, Barcelona, Spain.*

Variational quantum algorithms (VQAs) are promising candidates for near-term applications of quantum computers, but their training represents a major challenge in practice. We introduce *exact-geodesic VQAs*, a curvature-aware framework that enables analytic Riemannian optimization of variational quantum circuits through a convenient choice of circuit ansatz. Our method exploits the exact metric to find a parameter optimization path based on exact geodesic transport with conjugate gradients (EGT-CG). This supersedes the quantum natural gradient method, in fact recovering it as its first-order approximation. Further, the exact-geodesic updates for our circuit ansatz have the same measurement cost as standard gradient descent. This contrasts with previous metric-aware methods, which require resource-intensive estimations of the metric tensor using quantum hardware. In numerical simulations for electronic structure problems of up to 14 spin-orbitals, our framework allows us to achieve up to a 20x reduction in the number of iterations over Adam or quantum natural gradient methods. Moreover, for degenerate cases, which are notoriously difficult to optimize with conventional methods, we achieve rapid convergence to the global minima. Our work demonstrates that the cost of VQA optimization can be drastically reduced by harnessing the Riemannian geometry of the manifold expressed by the circuit ansatz, with potential implications at the interface between quantum machine learning, differential geometry, and optimal control theory.

I. INTRODUCTION

Variational quantum algorithms (VQAs) [1, 2] are hybrid quantum-classical methods particularly well suited for noisy intermediate-scale quantum devices. The idea is to use a classical computer to iteratively optimize the parameters of a parametrized quantum circuit (PQC) to minimize a cost function encoding the problem of interest, whose evaluation is done via the quantum circuit's output. However, this classical-quantum optimization loop represents a major challenge in practice, largely due to the large number of iterations required and the encounter of flat optimization landscapes [3]. The *quantum natural gradient* (QNG) [4, 5] is a first-order Riemannian optimization method [6, 7] that improves gradient-based VQA optimization by accounting for the geometry of the space of quantum states reachable by a PQC. The QNG moves in the steepest descent direction as given by the metric tensor $\boldsymbol{g}$ of the space. Moreover, second-order geodesic corrections to the QNG [8] as well as the addition of *conjugate gradient* methods [9] have been proposed to further improve the curvature corrections to the direction of descent in the QNG optimizer.

However, given that for most VQA ansatze the metric tensor is not known in analytic form, implementing the QNG in practice comes with an experimental overhead of empirically estimating it on hardware [10]. For a generic PQC with M free parameters, this overhead is $\mathcal{O}(M^2)$ quantum hardware measurements at each optimization step. Moreover, the overhead increases when second-order corrections are added, which in general requires $\mathcal{O}(M^3)$ measurement to empirically estimate second-order geometric quantities [8]. Further attempts to lower the cost to estimate $\boldsymbol{g}$ to $\mathcal{O}(M)$ have relied on block-diagonal approximations of the metric tensor [4, 8, 9, 11–14], though it has been shown that they may lead to suboptimal optimization schemes due to loss of information on parameter correlation [15]. There are several approaches proposing metric-tensor estimations at a constant cost [16–18]. In particular, two alternative methods to estimate a stochastic approximation of $\boldsymbol{g}$ using quantum hardware are based on the simultaneous perturbation stochastic approximation algorithm [17] and Stein's identity [18]. Even though they reduce the estimation overhead to $\mathcal{O}(1)$ in M, numerical evidence suggests that this approximation decreases the performance when compared to the QNG with the exact metric tensor [17–19].

In this work, we remove the need for empirical estimation of the metric tensor $\boldsymbol{g}$ by using a class of PQCs whose set of parameterized pure quantum states are hyperspheres, using their hyperspherical coordinates as circuit parameters. In these coordinates, the metric tensor $\boldsymbol{g}$ is a diagonal matrix that can be calculated in closed form, leading to analytical solutions for geodesic paths and exact parameter update rules for geodesic-based descent, which we name *Exact Geodesic Transport* (EGT). The procedure is schematically illustrated in Fig. 1. Consequently, we show how to implement exact (all-orders) Riemannian optimization at the same quantum hardware measurement costs as standard gradient descent. The first- [4] and second-order [8] parameter update rules are recovered, respectively, as the first and second-order Taylor series approximations of the exact geodesic paths under the assumption of small learning rates. Here, we ap-

* andre.jfm.sci@gmail.com

† renato.msf@gmail.com

‡ Contributions were done when at TII[1].

ply the Riemannian conjugate gradient method (CG) introduced in [20] specifically for Exact Geodesic Transport on the sphere, choosing the learning rates appropriately to guarantee global convergence (*i.e.*, the optimization reaches a stationary point regardless of the initialization). We call this new proposed optimizer *EGT-CG*.

We study the performance of our optimizer on the task of ground state preparation for the electronic structure problem and spin chains, and benchmark it against other state-of-the-art optimizers. We show numerical evidence that EGT-CG consistently presents superior performance in terms of optimization iterations required for convergence, even in the challenging case of degenerate ground states. Additionally, for molecular Hamiltonians, our ansatz allows an efficient initialization with a problem-informed warm start, which has high Hartree state overlap, allowing for even faster and more stable convergence. Also, using the one-dimensional XXZ spin-chain Hamiltonian as an example, we display the resource scaling of EGT-CG optimization as a function of system size and benchmark it against other combinations of optimizers and VQA ansatze. Moreover, the flexibility of the method to handle different optimization subspaces requiring no auxiliary qubits and with circuit depth linear on the number of non-zero amplitudes is illustrated using the transverse-field Ising Hamiltonian.

We provide three technical contributions that may be relevant beyond this work. We show: (*i*) how to estimate gradients with fewer quantum resources compared to the standard parameter-shift rule [21] for the circuit ansatze used in this work; (*ii*) how this circuit ansatz admits a more efficient regularization technique of the metric $\boldsymbol{g}$ beyond standard techniques; (*iii*) how to fully characterize the (probabilistic) barren plateaus [3] within our ansatz by analytically calculating the exact variances of the loss function and its gradient components, showing that the barren plateaus do not occur at poly(n) circuit depth.

The remainder of the manuscript is structured as follows. In Sec. II, we briefly review Riemannian optimization methods for VQA. In Sec. III, we introduce our EGT and EGT-CG optimizers based on exact geodesic descent. Section IV presents the results of numerical demonstrations, while Sec. V contains the concluding remarks.

II. PRELIMINARIES

Notation. We use the shorthand notation $[d] := \{1, 2, \ldots, d\}$. For any bitstring $b \in \{0,1\}^n$ of length n, we denote by $|b|$ its Hamming weight (HW) and define the HW of a computational basis state $|b\rangle$ as that of b. We use superscripts to represent a bit value that gets repeated (*e.g.* $0^3 1^2 \equiv 00011$). Boldface symbols denote multi-dimensional objects (*e.g.* $\mathbf{x} := \{x_j\}_{j\in[d]}$ and $\boldsymbol{g} := \{g_{j\ell}\}_{j,\ell\in[M]}$), the shorthand $\partial_{y_j} := \partial/\partial y_j$ denotes partial derivatives with respect to a variable y_j, and $\boldsymbol{\partial}_{\mathbf{y}} := \{\partial_{y_j}\}_{j\in[|\mathbf{y}|]}$ is used for the vector of all partial derivatives over d variables. Moreover, we denote $(\boldsymbol{\partial}_{\mathbf{y}} f)_{\mathbf{y}'}$ as the derivative of a function f w.r.t. $\mathbf{y}$ and evaluated at $\mathbf{y} = \mathbf{y}'$.

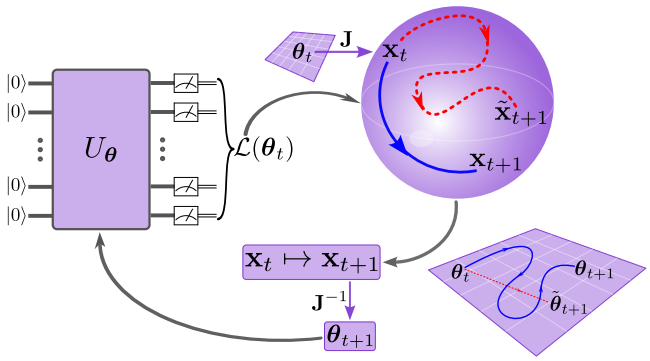

Figure 1. **Schematic illustration of our method.** VQA pipeline with standard gradient descent (GD, red dashed curves) update rule *vs.* the proposed descent with exact geodesic transport (EGT, blue solid curves). In GD, the search for a minimum of the loss function $\mathcal{L}(\boldsymbol{\theta})$ is done via steps along straight lines on a flat parameter space, whereby the parameters $\boldsymbol{\theta}$ are updated directly. In contrast, EGT implements an amplitude-based update rule, based on paths along geodesics on the curved space – manifold – that contains the state vector $|\psi(\boldsymbol{\theta})\rangle$ output by the variational circuit $U_{\boldsymbol{\theta}}$. For our circuit Ansatz, such paths define great-circle arcs on a hypersphere. Both spaces are related via the Jacobian $\mathbf{J}$ of the (exact) coordinate transformation $\boldsymbol{\theta} = \boldsymbol{\theta}(\mathbf{x})$ between circuit parameters $\boldsymbol{\theta}$ and the amplitudes $\mathbf{x}$ of $|\psi(\boldsymbol{\theta})\rangle$ in the computational basis. This allows one to recover the updated parameters $\boldsymbol{\theta}_{t+1}$ from the amplitudes $\mathbf{x}_{t+1}$ updated via EGT.

In the following, we use the standard setup of variational quantum algorithms (VQA) [1]: one is given an n-qubit parametrized quantum circuit $U_{\boldsymbol{\theta}}$ (the *VQA ansatz*) with a set of parameters $\boldsymbol{\theta}$ that prepares a quantum state $|\psi(\boldsymbol{\theta})\rangle = U_{\boldsymbol{\theta}} |0^n\rangle$, and the goal is to find the parameters $\boldsymbol{\theta}^*$ that minimize a given loss function $\mathcal{L}(\boldsymbol{\theta})$ of interest. The end-to-end algorithm is a classical/quantum hybrid method in that the loss function value at any specific $\boldsymbol{\theta}$ can be queried using the quantum device, while a classical computer guides the search for $\boldsymbol{\theta}^*$. Here, we focus on ground state estimation problems, where $\mathcal{L}(\boldsymbol{\theta}) \equiv \langle\psi(\boldsymbol{\theta})| H |\psi(\boldsymbol{\theta})\rangle$ is the expectation value of a Hamiltonian H, although we emphasize the methods generalize to arbitrary $\mathcal{L}(\boldsymbol{\theta})$.

Given a VQA ansatz $U_{\boldsymbol{\theta}}$, the set of all reachable quantum states defines a hypersurface on the Hilbert space. Whenever this surface is a manifold, we can define a Riemannian metric $\boldsymbol{g}$ by taking the real part of the quantum geometric tensor [4, 22, 23]

$$g_{j\ell}(\boldsymbol{\theta}) = \mathrm{Re}\Big\{ \langle \partial_{\theta_j}\psi \,|\, \partial_{\theta_\ell}\psi\rangle - \langle \partial_{\theta_j}\psi \,|\, \psi\rangle\langle\psi \,|\, \partial_{\theta_\ell}\psi\rangle \Big\}, \quad (1)$$

which is also known as the quantum Fisher information metric or Fubini-Study metric. From $\boldsymbol{g}$, various geometrical properties, such as geodesics and the exponential map describing parallel transport along geodesics, can

be derived (see App. A). In principle, gradient-descent methods based on geodesic transport are ideal for optimization on curved manifolds. In practice, they may become prohibitive given that the metric components in Eq. (1) are often unknown and need to be estimated on quantum hardware, causing an overhead in quantum resources. Moreover, exact transport along the geodesic flow requires access to the exponential map, which is unknown for arbitrary manifolds. With these two bottlenecks, the effect of curvature in VQA optimizations is often ignored altogether or only approximated via QNG [4, 8, 9], though its implementation is still hindered by the high cost of estimating the metric.

The main contribution of this work is the development of a VQA parameter optimization routine that efficiently implements exact geodesic transport with no quantum overhead by using a particular choice of circuit ansatz, hence solving the aforementioned issues.

III. VQA WITH EXACT GEODESIC TRANSPORT

In this section, we introduce our VQA for gradient-based optimization over arbitrary subspaces with exact geodesic transport. To achieve that, the key ingredient is a VQA ansatz that parametrizes the state using hyperspherical coordinates [24–27]. In particular, we use the exact amplitude encoders for arbitrary subspaces introduced in Ref. [27], which allows the exploration of eventual symmetries of the loss function $\mathcal{L}(\boldsymbol{\theta})$ to restrict the optimization to an effective subspace of computational basis states.

Given an arbitrary subset B of $d \leq 2^n$ computational basis states of a n-qubit system, Ref. [27] showed how to construct a parameterized quantum circuit with the minimum number of parameters $\boldsymbol{\theta}$ that prepares an arbitrary superposition $|\psi(\boldsymbol{\theta})\rangle = \sum_{j=1}^{d} x_j(\boldsymbol{\theta})\,|b_j\rangle$ of states in B, with ℓ_2-normalized amplitudes $\mathbf{x} \coloneqq \{x_j(\boldsymbol{\theta})\}_{j\in[d]}$. For simplicity, here we focus on the case of real-valued amplitudes. In this case, the circuit consists of exactly $M = d-1$ parameterized gates whose angles $\boldsymbol{\theta} \equiv \boldsymbol{\theta}(\mathbf{x})$ are the hyperspherical coordinates of the amplitude vector $\mathbf{x}$ on the $(d-1)$-dimensional sphere (see Ref. [27] for details). As a result, using the amplitude encoder from Ref. [27] as ansatz guarantees full expressivity, *i.e.* overparametrization [28–30] in the subspace B with the minimum possible number of parameters. Here we use the encoder for real-valued amplitudes since, for all the examples analyzed here, the Hamiltonian H has real-valued components in the computational basis. However, the method extends straightforwardly to the case of complex amplitudes (see App. A for details).

The manifold spanned by the set of all states $|\psi(\boldsymbol{\theta})\rangle$ reachable by the encoder is isomorphic to the $(d-1)$-dimensional sphere $\mathbb{S}^{d-1}$, whose metric $\boldsymbol{g}$ is diagonal in the coordinate basis, with components $g_{11} = 1$ and $g_{jj} = \prod_{\ell=1}^{j-1} \sin^2(\theta_\ell)$ for $j \in [2,\, d-1]$. Consequently, the Riemannian exponential map describing geodesic transport on $\mathbb{S}^{d-1}$ can be computed in closed form [6, 7]. For a starting point $\mathbf{x}$ with tangent vector $\boldsymbol{v}$, the exponential map is given by

$$\mathrm{ExpMap}_{\mathbf{x}}(\eta\,\boldsymbol{v}) = \cos(\eta\,\|\boldsymbol{v}\|)\,\mathbf{x} + \sin(\eta\,\|\boldsymbol{v}\|)\,\frac{\boldsymbol{v}}{\|\boldsymbol{v}\|}\,, \quad (2)$$

where η is an affine parameter, and $\|\boldsymbol{v}\| \coloneqq \sqrt{\langle \boldsymbol{v},\, \boldsymbol{v}\rangle_{\mathbf{x}}}$, with $\langle \boldsymbol{u},\, \boldsymbol{v}\rangle_{\mathbf{x}} \coloneqq \boldsymbol{u}^T \boldsymbol{v}$ being the induced metric on the tangent space of $\mathbf{x} \in \mathbb{S}^{d-1}$.

The analytical form of the exponential map in Eq. (2) opens up the possibility of gradient-based parameter optimization with exact geodesic descent, solving at once the two main bottlenecks of previous QNG approaches to curvature-aware VQA optimization mentioned in Sec. II. With that in mind, we introduce the Exact Geodesic Transport (EGT) parameter-update rule, illustrated in Fig. 1 (see App. B for details),

$$\mathbf{x}_{t+1} = \mathrm{ExpMap}_{\mathbf{x}_t}(\eta_t\,\boldsymbol{v}_t)\,, \quad \boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}(\mathbf{x}_{t+1})\,, \quad (3)$$

where η_t is the learning rate at step t, and $\boldsymbol{v}_t \coloneqq -\mathbf{J}(\boldsymbol{\theta}_t)\,\boldsymbol{g}^{-1}(\boldsymbol{\theta}_t)\,(\boldsymbol{\partial}_{\boldsymbol{\theta}}\mathcal{L})_{\boldsymbol{\theta}_t}$. Here, $\boldsymbol{\theta}(\mathbf{x})$ is the standard hyperspherical-coordinate transformation (see Ref. [27, Eq. (7)]), $\boldsymbol{g}^{-1}(\boldsymbol{\theta}_t)\,(\boldsymbol{\partial}_{\boldsymbol{\theta}}\mathcal{L})_{\boldsymbol{\theta}_t}$ is the *natural gradient* of $\mathcal{L}(\boldsymbol{\theta})$, and $\mathbf{J}(\boldsymbol{\theta})$ is the Jacobian matrix of the transformation $\boldsymbol{\theta}(\mathbf{x})$. In App. C, we show an efficient method to calculate $\boldsymbol{v}_t$ numerically assuring that the metric $\boldsymbol{g}$ remains invertible at each step when the gradient $(\boldsymbol{\partial}_{\boldsymbol{\theta}}\mathcal{L})_{\boldsymbol{\theta}_t}$ is estimated using quantum hardware measurements. For $\eta_t\,\|\boldsymbol{v}_t\| \ll 1$, first- and second-order Taylor approximations of the exponential map in Eq. (3) recover, up to a coordinate transformation, the standard QNG update rule [4] and its correction introduced in [8], respectively.

Having analytical access to geometric quantities on the sphere allows us to improve the EGT update rule in Eq. (3) by adding memory from the previous descent direction. We do so by using the Riemannian conjugate gradient (CG) method [20]. The resulting Exact Geodesic Transport with Conjugate Gradient (EGT-CG) optimizer takes the same form as Eq. (3), replacing $\boldsymbol{v}_t$ by a new tangent vector $\boldsymbol{u}_t$ given recursively by

$$\boldsymbol{u}_0 \coloneqq \boldsymbol{v}_0\,, \quad \boldsymbol{u}_t \coloneqq \boldsymbol{v}_t + \beta_t\,\mathcal{T}_{\eta_t \boldsymbol{u}_t}(\boldsymbol{u}_{t-1})\,, \quad (4)$$

where $\beta_t \geq 0$ is defined depending on a particular CG method, and the expression $\mathcal{T}_{\eta_t \boldsymbol{u}_t}(\boldsymbol{u}_{t-1}) \coloneqq \cos(\eta_t\,\|\boldsymbol{u}_{t-1}\|)\,\boldsymbol{u}_{t-1} - \sin(\eta_t\,\|\boldsymbol{u}_{t-1}\|)\,\|\boldsymbol{u}_{t-1}\|\,\mathbf{x}_{t-1}$ is the exact vector transport of the previous direction $\boldsymbol{u}_{t-1}$ from the tangent space at $\mathbf{x}_{t-1}$ to the tangent space at $\mathbf{x}_t$ (see App. A). The choice of η_t and β_t has a strong impact on the performance of the optimizer, and $\beta_t = 0$ for all t recovers the EGT update rule in Eq. (3). For the ground state estimation problem, EGT-CG allows convergence guarantees by the precise scheduling of η_t using the so-called *strong Wolfe conditions* [31, 32] as well as β_t using the hybrid conjugate gradient, as discussed in detail in App. B. In App. D, we compare these convergence guarantees to heuristic choices of η_t and β_t using

Bayesian optimization [33, 34], showing through an example that the path where the Wolfe conditions hold can lead to significantly better performance in cases where the ground state is degenerate, whilst still using fewer resources in terms of loss function evaluations.

The particular state implemented by the circuits from Ref. [27] offers two extra benefits for the optimization. First, we can estimate gradients using quantum hardware with fewer resources when compared to the standard parameter-shift rule (PSR) [21], as described in the App. E. Second, we are also able to fully characterize the presence of (probabilistic) barren plateaus in the optimization landscape by exactly calculating the variance of the loss function $\mathcal{L}$ and its (natural) gradient components [3]. In particular, the results imply the absence of barren plateaus for poly(n)-dimensional (sub)spaces, as shown in detail in App. F.

IV. NUMERICAL DEMONSTRATIONS

In this section, we show numerical benchmarking of the performance of the EGT-CG optimizer against state-of-the-art optimizers on the problem of ground-state preparation. As mentioned in the previous sections, we minimize the loss function $\mathcal{L}(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})| H |\psi(\boldsymbol{\theta})\rangle$, with H being the Hamiltonian of the system of interest. In Sec. IV A, we show results for the electronic structure problem of 14 different molecules. The Hamiltonians were taken from `PennyLane`'s Quantum Dataset [35, 36] in the STO-3G basis set and at the equilibrium bond length. In Sec. IV B, we analyze the scaling of the performance with system size on the XXZ spin-chain Hamiltonian. We also apply the optimizer to the transverse-field Ising model (TFIM) as an example of ground states containing different Hamming weights (HW). All simulations were run using an integration of the software packages `Qibo` [37] and `PyTorch` [38]. Gradients were calculated using `PyTorch`'s backpropagation method, a process equivalent to estimating gradients up to machine precision using PSR. We used `Qibo`'s backend that specializes in simulations of Hamming-weight-preserving circuits, and we used Ref. [39] for the Bayesian optimization.

We benchmarked the EGT-CG optimizer against different optimizers and learning rate schedulers (referred to as *optimization schemes*): (i) a CG optimizer with standard gradient descent [40] and learning rate schedules based on the strong Wolfe conditions [20]; (ii) the QNG approximated to first [4] and second [8] orders with learning rates chosen via Bayesian optimization [33, 34]; and (iii) the *Adam* optimizer [41] with constant learning rates $\eta \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$. The performance of each optimization scheme was assessed using the relative error of ground-state energy estimation as a function of the number of iterations, here referred to as *epochs*. In all cases, we set a maximum limit of 10^3 epochs, and the optimization was either halted 15 epochs after reaching chemical accuracy or if an early stop was triggered by either 20 consecutive epochs of loss function decrease (in magnitude) smaller than 10^{-4}, or 10 consecutive epochs of loss function increase.

A. Electronic structure problem

We first consider the problem of finding the ground state of molecular Hamiltonians for the electronic structure problem. Given a molecule having k electrons and a basis set describing its orbitals, the Jordan-Wigner transform [42–44] maps its fermionic Hamiltonian into a qubit Hamiltonian H where each qubit corresponds to one of the n spin-orbitals. The ground state of the qubitized Hamiltonian has real-valued amplitudes and is supported on the $\binom{n}{k}$-dimensional subspace of HW-k computational basis states. To explore this particle-preserving symmetry while parametrizing the entire subspace in hyperspherical coordinates, we use the aforementioned fixed-Hamming weight encoder, hereinafter denoted as HWE$_k$.

1. Molecular warm start

We take advantage of the HWE$_k$ ansatz to initialize the circuit parameters $\boldsymbol{\theta}$ such that the corresponding state has high fidelity with the Hartree state $|1^k0^{n-k}\rangle$, which is known to provide a rough approximation to the true ground state of some molecules [45]. We propose, as a warm start, the superposition state

$$|\psi_{\text{warm}}\rangle = \sqrt{\alpha}\,|1^k0^{n-k}\rangle + \sqrt{\frac{1-\alpha}{\binom{n}{k}-1}} \sum_{|b\rangle\in B'_k} |b\rangle\ , \quad (5)$$

where $\alpha \in (0,1)$ is a free parameter (in all the simulations here, we used $\alpha = 0.9$), B_k is the set of all n-qubit computational basis states of HW-k, and $B'_k := B_k \setminus \{|1^k0^{n-k}\rangle\}$. This state has fidelity α with the Hartree state, is supported over all the HW-k basis states, and can be prepared deterministically using the HWE$_k$ ansatz, but not with other standard VQA ansatze. The motivation for (5) is two-fold: while the high fidelity naturally reduces the overall number of optimization steps, populating all the remaining directions yields a sizable gradient that is beneficial for the gradient-based optimizer [46].

In Fig. 2(d) (dark gray bars), we plot the infidelity of the warm start $|\psi_{\text{warm}}\rangle$ relative to the true ground state of the molecules (found by numerical diagonalization). The fidelities where calculated as $\sum_{j=1}^{s} |\langle\psi_{\text{warm}}|\alpha_j\rangle|^2$, where $\{|\alpha_j\rangle\}_{j=1}^{s}$ are the eigenstates with degeneracy s ($s = 2$ and $s = 3$ respectively for H_5 and CH_2, and $s = 1$ for all others). The small infidelities highlight the fact that the proposed warm start is indeed adequate for all molecules considered except for CH_2 (see App. G for a discussion). Even in the case of CH_2, for which the Hartree state is not a good warm start — it is nearly

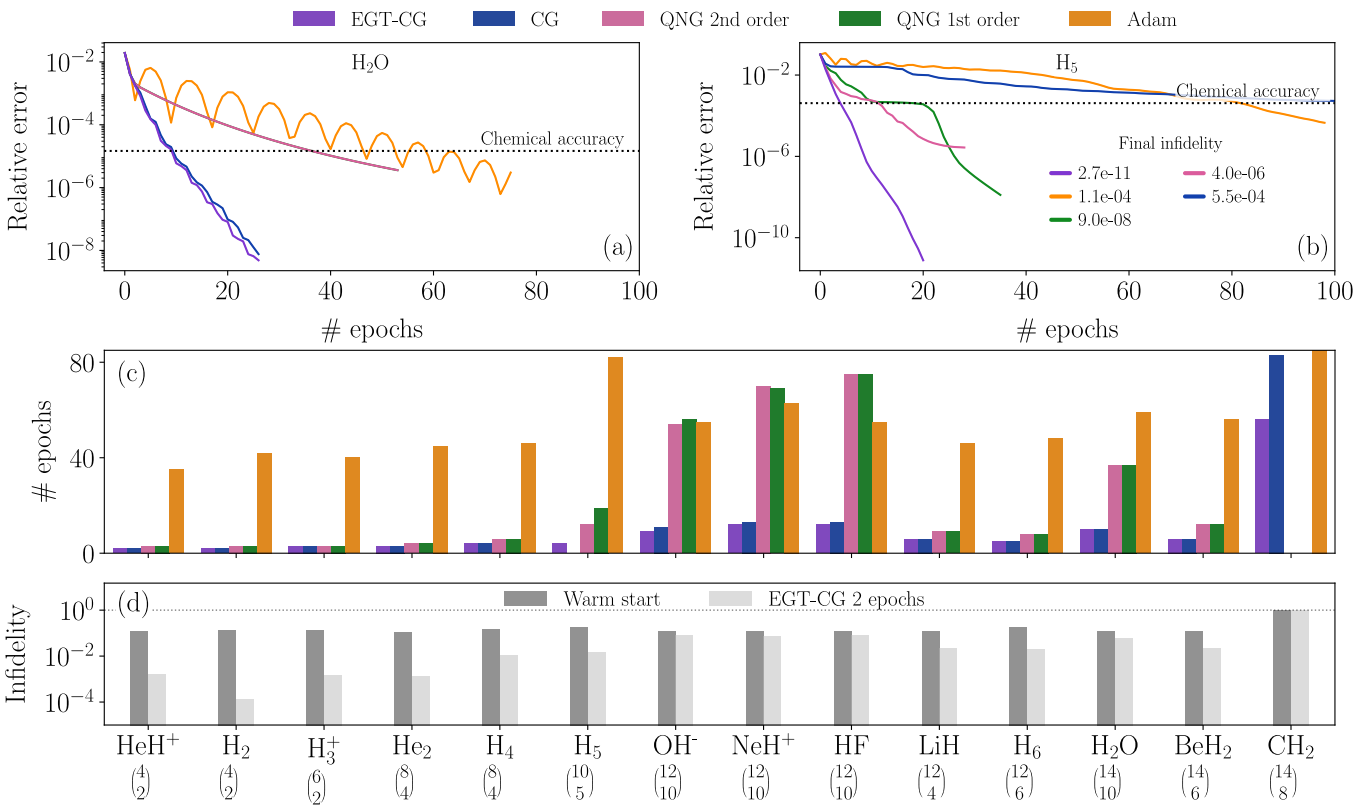

Figure 2. **Ground-state optimization for molecules.** Performance of different optimization schemes using as VQA ansatz the Hamming-weight-k amplitude encoder (HWE$_k$) from Ref. [27] with the warm start in Eq. (5) with $\alpha = 0.9$. The different optimizers are: exact geodesic transport with conjugate gradients (EGT-CG, purple), conjugate gradient method with flat-space gradients (CG, blue), quantum natural gradient (QNG) of first (green) and second (pink) orders, and the standard Adam optimizer (orange). The learning rates for EGT-CG and CG satisfy the strong Wolfe conditions (see App. B); For QNG, they were chosen using Bayesian optimization techniques; For Adam, we used a constant η. (a) Relative ground-state energy error for H_2O *vs.* number of epochs. Using the Jordan-Wigner transformation and the STO-3G basis set, the ground state is represented by $n = 14$ spin-orbitals as the active space with $k = 10$ electrons. EGT-CG and CG display a similar performance, both comfortably beating the other methods ($\eta = 0.005$ for Adam here). (b) Relative ground-state energy error for H_5, represented by $n = 10$ spin-orbitals and $k = 5$ electrons. The color code is the same as in (a), with the orange curve now corresponding to Adam with $\eta = 0.05$. The infidelity $1 - F_{\text{final}}$ of the state prepared by each scheme at the end of the optimization relative to the true ground state is displayed in the inset. (c) Number of epochs to achieve chemical accuracy for 14 molecules and different optimizers. For H_5 and CH_2, some optimization schemes did not converge to chemical accuracy, in which case no bar is shown. CH_2 with Adam reached chemical accuracy in 130 epochs (y-axis was limited for better presentation). (d) Infidelities relative to the true ground state for the initial state $|\psi_{\text{warm}}\rangle$ with $\alpha = 0.9$ and for the state after 2 epochs of EGT-CG, for the same molecules of (c). Below each molecule label in the x-axis, the size of the corresponding parametrized subspace is indicated by $\binom{n}{k}$.

orthogonal to the ground state subspace —, the global convergence guaranteed by EGT-CG with strong Wolfe conditions plays an important role and our optimizer still successfully converges (see Fig. 2(c)). Also, it is important to emphasize that the HWE$_k$ ansatz allows the preparation of any other heuristic initial state inspired by domain knowledge.

In addition, we could enforce spin symmetries to further reduce the effective subspace in which $|\psi_{\text{warm}}\rangle$ is supported. However, we numerically verified that this reduction was not significant for the system sizes analyzed in this manuscript. We note that spin-projected matrix product states [47] are known to provide good warm starts as well. However, their preparation requires circuits of challenging sizes, in particular with an extensive number of auxiliary qubits. For example, recent estimates for industry-relevant molecules give between 4 and 5 times more auxiliary qubits required than system qubits themselves [48, Table 1]. In contrast, HWE$_k$ allows us to prepare $|\psi_{\text{warm}}\rangle$ with circuits of depth $\binom{n}{k}$ but without extra qubits. Moreover, performing only 2 epochs of EGT-CG allows us to move significantly in the direction of the ground state (see the light-gray bars in Fig. 2(d)). This indicates the usefulness of our method as the preparation of initial states for fault-tolerant methods, as it is further discussed in Sec. V.

2. Results

Here, we present the results of the benchmarks for the molecular Hamiltonians. In Fig. 2(a), we show the results of our numerical simulations for the H_2O molecule. We see that the EGT-CG optimizer (purple), as well as the CG optimizer with flat-space gradients (blue), achieves chemical accuracy after 10 epochs. This convergence performance is 3.7 times faster than the QNG optimizers of first (green) and second (pink) orders, and 4.7 times faster than the fastest Adam optimizer tested ($\eta = 0.005$, orange). From these results, one might be tempted to conclude that the performance advantage in the EGT-CG optimizer is mainly attributed to the conjugate gradient method and not to the properties of the curved manifold. However, we now show numerical evidence that this is not the case.

In Fig. 2(b), we show the results for the H_5 molecule using the same color scheme as Fig. 2(a). The H_5 molecule, if it exists as a metastable species, would be a free radical with a degenerate ground state at the equilibrium bond length. We see that the EGT-CG optimizer performs very well, achieving chemical accuracy after only 4 epochs, which is 3 and 4.75 faster than the first (12 epochs) and second (19 epochs) order QNG methods, respectively. Meanwhile, the CG optimizer reaches a plateau near chemical accuracy without ever crossing it, being early-stopped at epoch 128, after 20 epochs of stationary loss. The Adam optimizer with constant $\eta = 0.05$ (orange) reaches chemical accuracy after 82 epochs, 20.5 times slower than EGT-CG. In Fig. 2(b) inset, we also display the infidelities $1 - F_{\text{final}}$ of the final state achieved by each optimization scheme, relative to the true ground states found by direct diagonalization, where $F_{\text{final}} = \sum_{j=1}^{s} |\langle \psi_{\text{final}} | \alpha_j \rangle|^2$. This shows that EGT-CG achieves by far the best accuracy, both in terms of ground state fidelity as well as its energy.

The analysis above was repeated for 12 other molecules, for a total of 14 molecules. All displayed similar patterns to those observed in Figs. 2(a)-(b). Fig. 2(c) shows a comparison of the number of epochs needed by each optimizer to achieve chemical accuracy in ground-state energy estimation for the 14 molecules. The color scheme is the same as in Figs. 2(a)-(b). EGT-CG has the best performance overall, consistently reaching chemical accuracy in fewer epochs than the other optimizers. For 12 of the 14 molecules, the CG optimizer with flat-space gradients performs comparably to the EGT-CG optimizer. However, for molecules with a degenerate ground state — namely, H_5 and CH_2 —, either the CG or the QNG optimizers fail to achieve chemical accuracy while the EGT-CG still shows the fastest convergence. In addition to the benefit of incorporating exact geodesic descent, this also highlights the practical importance of the global convergence guarantees of EGT-CG. The optimization using Adam with a constant learning rate achieves chemical accuracy in all cases, though after significantly more epochs than EGT-CG. Adam converges faster when using larger learning rates; however, the optimization trajectories become non-monotonic (see, *e.g.* the orange curve in Fig.2(a)). In the case of CH_2, we show in App. G that the non-monotonicity helps Adam to avoid local minima, while QNG optimizers (without CG) get stuck.

B. Spin chains

Here, we discuss the application of the EGT-CG optimizer to finding the ground state of one-dimensional spin chains. We begin by considering the n-qubit XXZ Hamiltonian $H_{\text{XXZ}} := \sum_{j=0}^{n-1} (X_j X_{j+1} + Y_j Y_{j+1} + \Delta Z_j Z_{j+1})$ with closed-boundary condition, with Δ being the anisotropy strength, and $\{X_j, Y_j, Z_j\}$ the usual single-qubit Pauli operators acting on the jth qubit. For simplicity, we focus on the case $\Delta = 1/2$ and investigate n even in the interval $n \in [4, 16]$. The ground state is known to have real-valued amplitudes supported on the *half-filling* subspace, *i.e.*, the $\binom{n}{n/2}$-dimensional subspace of computational basis states with Hamming weight $k = n/2$ [49]. Due to symmetries in the system (including cyclic translational and parity), there are $d = \mathcal{O}\Big(n^{-1}\binom{n}{n/2}\Big)$ unique amplitudes in absolute value in the ground state [49], leading to the possibility of restricting the circuit implementation to a smaller subspace. To explore these symmetries, we choose again the $\text{HWE}_{k=n/2}$ ansatz, but now we only parametrize the unique amplitudes. This parametrizes the aforementioned effective Hilbert subspace in hyperspherical coordinates using the minimum number $(d-1)$ of gate parameters.

We again assess the number of epochs taken by each optimization scheme to achieve chemical accuracy in ground-state energy estimation. We benchmark EGT-CG against flat-space CG, first- and second-order QNG, and Adam with the same respective learning rate choices described in Sec. IV A. Additionally, we also test three other circuit architectures using the Adam optimizer: (*i*) another HW-preserving ansatz composed of *quantum orthogonal layers* (QOL) of nearest- and next-nearest-neighbor connectivity [50, Fig. 2]. The number of free parameters is $M = \binom{n}{n/2} - 1$; (*ii*) the brickwork *hardware-efficient* (HE) ansatz [51] that parametrizes the full Hilbert space with $M = 2^n - 1$ parameters; (*iii*) an overparametrized HE ansatz, hereinafter referred to as HE OP, with $M = 2\,(2^n - 1)$ free parameters.

In Fig. 3(a), we plot the number of epochs to reach chemical accuracy in energy estimation as a function of the system size n, averaged over 50 Haar-random initializations (as no natural warm start is available here), with error bars indicating the 95% confidence interval. As before, optimization was halted 15 epochs after reaching chemical accuracy, or if an early stopping condition was triggered. Missing points for QOL, HE, and HE OP indicate that they did not reach chemical accuracy. This

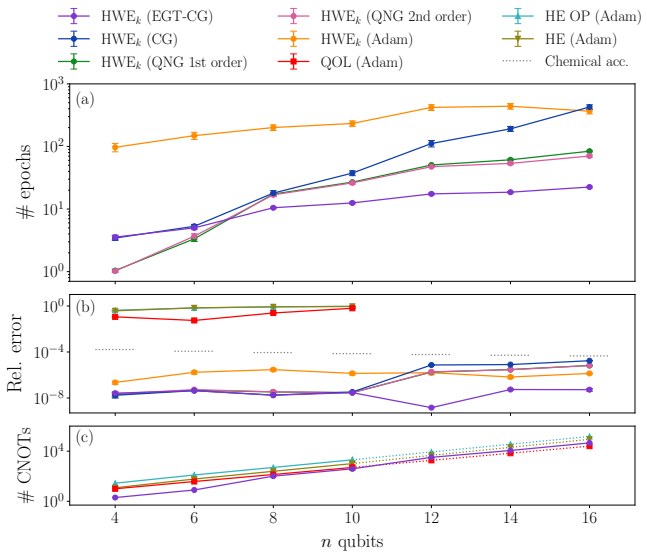

Figure 3. **Ground-state estimation for the** XXZ **model.** Performance of different circuit ansatze and optimization schemes *vs.* system size n. We show the Hamming-weight-encoder (HWE) with 5 optimization schemes: EGT-CG (purple), flat space CG (dark blue), QNG of first (pink) and second (green) orders, and Adam (orange); moreover, we show also the quantum orthogonal layer (QOL) ansatz (red), the brickwork hardware-efficient (HE) ansatz with $M = 2^n - 1$ parameters (dark gold), and an overparametrized HE with $M = 2\,(2^n - 1)$ parameters (HE OP, light blue), all of which used the Adam optimizer. For HWE we went up to $n = 16$ qubits since the ansatz allows exploiting additional XXZ symmetries [49], while simulations were run up to $n = 10$ for the other ansatze. For EGT-CG and CG, the learning rates satisfy the strong Wolfe conditions (see App. B); For QNG, they were chosen using Bayesian optimization; For Adam, we used a constant $\eta \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$ and presented the curve for the value that displayed faster convergence. Each point is the average over 50 Haar-random initializations, with error bars indicating the 95% confidence interval. The optimization was halted 15 epochs after reaching chemical accuracy, or if an early stopping condition was triggered. (a) Average number of epochs to reach chemical accuracy in energy estimation. Missing points for QOL, HE and HE OP indicate that they did not reach chemical accuracy. (b) Relative energy estimation error at the end of the training, indicating that the optimization using QOL, HE , and HE OP never achieved chemical accuracy for all tested n. (c) Number of CNOT gates in each ansatz. Although we did not run demonstrations for ansatze other than HWE for $n > 10$, we plot the CNOT counts for these circuits for a scaling comparison. Numerical details can be found in Tables II and III in App. H.

is further explicit in Fig. 3(b), where we plot the relative energy estimation error at the end of training for all ansatze and optimization schemes, indicating that the optimization using QOL, HE and HE OP with Adam halted order of magnitudes before chemical accuracy. We see that the EGT-CG optimizer (purple) once again proves superior to all other optimization schemes, consistently achieving chemical accuracy orders of magnitude faster and with a milder increase rate in the number of epochs when compared to the other methods as the system size increases.

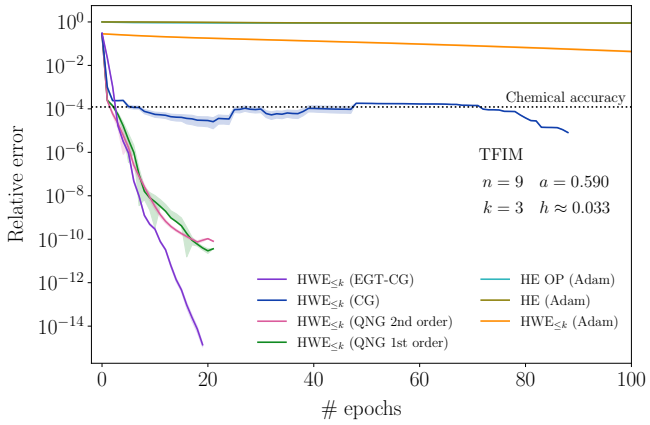

Figure 4. **Ground-state estimation for the** TFIM **Hamiltonian.** Optimization curves comparing different circuit ansatze and optimization schemes. With HWE$_{\leq k}$, we used EGT-CG (purple), flat space CG (dark blue), QNG of first (pink) and second (green) orders, and Adam (orange); moreover, we also used the brickwork hardware-efficient (HE) ansatz with $M = 2^n - 1$ parameters (dark gold), and an overparametrized HE with $M = 2\,(2^n - 1)$ parameters (HE OP, light blue), both with the Adam optimizer, for which we tried constant $\eta \in \{0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$ and present the curve for the value that displayed faster convergence. Each point is the average over 50 Haar-random initializations, with error shades indicating the 95% confidence interval. The optimization was halted 15 epochs after reaching chemical accuracy, or if an early stopping condition was triggered.

In Fig. 3(c), we analyze the quantum resources by plotting the number of CNOTs used by each circuit type. In fact, QOL (which is also a maximally expressive ansatz) has a more favorable scaling in terms of CNOTs when compared to HWE$_k$. However, it did not converge to chemical accuracy even for small system sizes. This further supports the power of using HWE$_k$ as a circuit ansatz, as it is the only circuit among the ones we tested that allows the implementation of EGT-CG.

Next, we consider the one-dimensional n-qubit transverse-field Ising model (TFIM), with Hamiltonian $H_{\text{TFIM}} := -\sum_{j=0}^{n-1} (Z_j Z_{j+1} + h X_j)$ and closed-boundary condition, where h is the transverse field strength. As demonstrated in Ref. [52], if $h = \mathcal{O}((k/n)^a)$ with $a > 1/2$ and sufficiently large n, the ground state of TFIM in the computational basis is fully supported on states of HW at most k. Thus, the ground state can be searched using a circuit ansatz given by the union of HWE$_k$ ansatze with all the Hamming weights up to k, *i.e.* HWE$_{\leq k} := \bigcup_{j=0}^{k}$ HWE$_j$ [27, 53]. Using this, we benchmark the EGT-CG optimizer against the flat-space CG, the QNG of first and second orders, as well as the Adam optimizer. We also train the aforementioned

HE and HE OP ansatze using the Adam optimizer.

Figure 4 shows the average relative error in ground-state energy estimation for a chain of $n = 9$ qubits as a function of the number of epochs. We chose the parameters $a = 0.59$ and $h \approx 0.033$, with the ground state having support on computational basis states of HW up to $k = 3$. We use the same 50 Haar-random states as starting points of the optimizations for all combinations of circuits and optimization schemes. We see that all versions of the natural gradient method, as well as the two conjugate gradient methods, achieve relative chemical accuracy, on average, in 3 to 5 epochs. While the geometry-aware methods continue to improve relative error estimation, the flat-space CG optimizer, on average, does not provide any improvement beyond the chemical accuracy level until around the 80th epoch. Meanwhile, all three geometry-aware methods continuously improve accuracy until being stopped at around the 20th epoch. The EGT-CG again displays the best performance at the end of the training, reaching an average relative energy-estimation error lower than 10^{-14}. The first- and second-order QNGs reach an average relative error of $\sim 10^{-10}$, which is 5 orders of magnitude better than flat-space CG at the end of training but still 4 orders of magnitude worse than the results using the EGT-CG optimizer. As in the example of the H_5 molecule in the previous section, the EGT-CG performs better than flat-space CG around small-gradient landscape regions. Even though it is not displayed in Fig. 4 for convenience, the combination of $\mathrm{HWE}_{\leq k}$ ansatz and Adam optimizer reached chemical accuracy around the 311th epoch on average. That is approximately 60 times more epochs than the geometry-aware optimizers and flat-space CG. The HE and HE OP ansatze with Adam optimizer did not achieve chemical accuracy, being early-stopped, on average, at the 128th and 159th epochs, respectively.

V. CONCLUSIONS

Our framework enables optimizing quantum circuits with parameter updates along exact geodesics in the space of parametrized states. We introduce EGT-CG to optimize circuit parameters along geodesic paths, which can be applied to any circuit ansatz where the geodesics on the spanned manifold are easily computable. This opens up new possibilities for quantum machine learning theory at the nexus with differential geometry and optimal control theory, which may in turn lead to a new generation of practical VQAs. Importantly, we emphasize that, although we have focused here on HW-preserving Hamiltonians and loss functions given by simple expectation values due to space constraints, the framework is fully versatile, extending also to generic ground states (not restricted to fixed-HW subspaces) and non-linear loss functions [54]. We also highlight the importance of using the strong Wolfe conditions to schedule the learning rate. This guarantees that the optimizer is globally convergent (*i.e.*, the optimization reaches a stationary point regardless of the initialization) even in cases where the barren plateaus are present and the gradients vanish (sub-)exponentially.

An important question is the impact of our framework on quantum-advantage regimes of VQAs. A precondition for quantum advantage is that the circuit ansatz is hard to simulate classically. We can achieve that using a circuit ansatz as follows: assume we have a classically intractable state $V\,|1^k0^{n-k}\rangle$, where V is an arbitrarily fixed unitary (*i.e.*, independent of the optimization parameters) with polynomial circuit depth, and $k = \mathcal{O}(\log n)$. One can imagine that this state is given and known to be an acceptable solution for some target problem. Then, we can variationally improve such a solution by choosing as circuit ansatz the state $|\psi_{\mathrm{qa}}(\boldsymbol{\theta})\rangle = V\,U_{\boldsymbol{\theta}}\,|1^k0^{n-k}\rangle$, where $U_{\boldsymbol{\theta}}$ is our HWE_k ansatz with polynomial circuit depth, *i.e.* with $k = \mathcal{O}(\log(n))$. Notably, $|\psi_{\mathrm{qa}}(\boldsymbol{\theta})\rangle$ is classically hard, yet has an analytic metric since fixed V leaves the metric invariant, allowing for efficient optimization using EGT-CG.

Alternatively, HWE can serve as a powerful initial state with efficient preparation. In particular, one can use EGT-CG to optimize a sparsified version of HWE (acting on a polynomial subspace of an exponentially large Hilbert space), where the optimized state then serves as a warm start for other quantum algorithms. For example, the optimized state obtained with our method can be used as inputs for quantum imaginary time evolution [55–58], eigenvalue thresholding [57, 59, 60], or quantum phase estimation [48]. This is particularly appealing for molecular Hamiltonians, where good warm-start states are known — such as spin-projected matrix-product states [47] — but whose preparation requires an extensive number of auxiliary qubits [48]; whereas our method requires none. These are some of the exciting questions for future explorations.

ACKNOWLEDGMENTS

We thank Ilia Luchnikov for insightful discussions.

[1] M. Cerezo, A. Arrasmith, R. Babbush, S. C. Benjamin, S. Endo, K. Fujii, J. R. McClean, K. Mitarai, X. Yuan, L. Cincio, and P. J. Coles, *Variational quantum algorithms*, Nature Reviews Physics **3**, 625–644 (2021).

[2] K. Bharti, A. Cervera-Lierta, T. H. Kyaw, T. Haug, S. Alperin-Lea, A. Anand, M. Degroote, H. Heimonen, J. S. Kottmann, T. Menke, *et al.*, *Noisy intermediate-scale quantum algorithms*, Rev. Mod. Phys. **94**, 015004

(2022).
[3] M. Larocca, S. Thanasilp, S. Wang, K. Sharma, J. Biamonte, P. J. Coles, L. Cincio, J. R. McClean, Z. Holmes, and M. Cerezo, *Barren plateaus in variational quantum computing*, Nature Reviews Physics **7**, 174–189 (2025).
[4] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, *Quantum natural gradient*, Quantum **4**, 269 (2020).
[5] B. Koczor and S. C. Benjamin, *Quantum natural gradient generalized to noisy and nonunitary circuits*, Phys. Rev. A **106**, 062416 (2022).
[6] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds* (Princeton University Press, 2008).
[7] H. Sato, *Riemannian optimization and its applications*, Vol. 670 (Springer, 2021).
[8] M. Halla, *Quantum natural gradient with geodesic corrections for small shallow quantum circuits*, Physica Scripta **100**, 055121 (2025).
[9] M. Halla, *Modified conjugate quantum natural gradient* (2025), arXiv:2501.05847 [quant-ph].
[10] T. Haug and M. S. Kim, *Natural parametrized quantum circuit*, Physical Review A **106** (2022).
[11] Y. Yao, P. Cussenot, R. A. Wolf, and F. Miatto, *Complex natural gradient optimization for optical quantum circuit design*, Physical Review A **105** (2022).
[12] N. Meyer, D. D. Scherer, A. Plinge, C. Mutschler, and M. J. Hartmann, *Quantum natural policy gradients: Towards sample-efficient reinforcement learning*, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2023) p. 36–41.
[13] J. Qi and M.-H. Hsieh, *Federated quantum natural gradient descent for quantum federated learning*, in *Federated Learning* (Academic Press, 2024) pp. 329–341.
[14] D. Fitzek, R. S. Jonsson, W. Dobrautz, and C. Schäfer, *Optimizing variational quantum algorithms with qBang: Efficiently interweaving metric and momentum to navigate flat energy landscapes*, Quantum **8**, 1313 (2024).
[15] D. Wierichs, C. Gogolin, and M. Kastoryano, *Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer*, Physical Review Research **2** (2020).
[16] B. van Straaten and B. Koczor, *Measurement cost of metric-aware variational quantum algorithms*, PRX Quantum **2** (2021).
[17] J. Gacon, C. Zoufal, G. Carleo, and S. Woerner, *Simultaneous perturbation stochastic approximation of the quantum Fisher information*, Quantum **5**, 567 (2021).
[18] M. Halla, *Estimation of quantum Fisher information via Stein's identity in variational quantum algorithms* (2025), arXiv:2502.17231 [quant-ph].
[19] J. Gacon, C. Zoufal, G. Carleo, and S. Woerner, *Stochastic approximation of variational quantum imaginary time evolution*, in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)* (IEEE, 2023) p. 129–139.
[20] H. Sato, *Riemannian conjugate gradient methods: General framework and specific algorithms with convergence analyses*, SIAM Journal on Optimization **32**, 2690–2717 (2022).
[21] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, *Evaluating analytic gradients on quantum hardware*, Phys. Rev. A **99**, 032331 (2019).
[22] J. Liu, H. Yuan, X.-M. Lu, and X. Wang, Quantum fisher information matrix and multiparameter estimation, Journal of Physics A: Mathematical and Theoretical **53**, 023001 (2020).
[23] J. J. Meyer, *Fisher information in noisy intermediate-scale quantum applications*, Quantum **5**, 539 (2021).
[24] B. T. Gard, L. Zhu, G. S. Barron, N. J. Mayhall, S. E. Economou, and E. Barnes, *Efficient symmetry-preserving state preparation circuits for the variational quantum eigensolver algorithm*, npj Quantum Information **6**, 10 (2020).
[25] J. Landman, N. Mathur, Y. Y. Li, M. Strahm, S. Kazdaghli, A. Prakash, and I. Kerenidis, *Quantum methods for neural networks and application to medical image classification*, Quantum **6**, 881 (2022).
[26] P. Bermejo, B. Aizpurua, and R. Orús, *Improving gradient methods via coordinate transformations: Applications to quantum machine learning*, Phys. Rev. Res. **6**, 023069 (2024).
[27] R. M. Farias, T. O. Maciel, G. Camilo, R. Lin, S. Ramos-Calderer, and L. Aolita, *Quantum encoder for fixed-Hamming-weight subspaces*, Phys. Rev. Appl. **23**, 044014 (2025).
[28] M. Larocca, N. Ju, D. García-Martín, P. J. Coles, and M. Cerezo, *Theory of overparametrization in quantum neural networks*, Nature Computational Science **3**, 542–551 (2023).
[29] T. Haug, K. Bharti, and M. Kim, *Capacity and quantum geometry of parametrized quantum circuits*, PRX Quantum **2**, 040309 (2021).
[30] T. Haug and M. Kim, *Generalization of quantum machine learning models using quantum Fisher information metric*, Physical Review Letters **133**, 050603 (2024).
[31] P. Wolfe, *Convergence conditions for ascent methods*, SIAM Review **11**, 226 (1969).
[32] P. Wolfe, *Convergence conditions for ascent methods. II: Some Corrections*, SIAM Review **13**, 185 (1971).
[33] J. Močkus, *On Bayesian methods for seeking the extremum*, in *Optimization Techniques IFIP Technical Conference Novosibirsk, July 1–7, 1974 6* (Springer, 1975) pp. 400–404.
[34] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, *Recent advances in Bayesian optimization*, ACM Computing Surveys **55**, 1 (2023).
[35] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. Akash-Narayanan, A. Asadi, *et al.*, *PennyLane: Automatic differentiation of hybrid quantum-classical computations* (2022), arXiv:1811.04968 [quant-ph].
[36] U. Azad and S. Fomichev, *PennyLane quantum chemistry datasets* (2023).
[37] S. Efthymiou, S. Ramos-Calderer, C. Bravo-Prieto, A. Pérez-Salinas, D. García-Martín, A. Garcia-Saez, J. I. Latorre, and S. Carrazza, *Qibo: A framework for quantum simulation with hardware acceleration*, Quantum Science and Technology **7**, 015018 (2021).
[38] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, *PyTorch: An imperative style, high-performance deep learning library* (2019), arXiv:1912.01703 [cs.LG].
[39] F. Nogueira, *Bayesian optimization: Open source constrained global optimization tool for Python* (2014).
[40] Y. H. Dai and Y. Yuan, *An efficient hybrid conjugate gradient method for unconstrained optimization*, Annals of Operations Research **103**, 33 (2001).

[41] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization* (2017), arXiv:1412.6980 [cs.LG].

[42] P. Jordan and E. Wigner, *Über das Paulische Äquivalenzverbot*, Zeitschrift für Physik **47**, 631 (1928).

[43] A. Tranter, P. J. Love, F. Mintert, and P. V. Coveney, *A comparison of the Bravyi–Kitaev and Jordan–Wigner transformations for the quantum simulation of quantum chemistry*, Journal of Chemical Theory and Computation **14**, 5617–5630 (2018).

[44] J. Tilly, H. Chen, S. Cao, D. Picozzi, K. Setia, Y. Li, E. Grant, L. Wossnig, I. Rungger, G. H. Booth, and J. Tennyson, *The variational quantum eigensolver: A review of methods and best practices*, Physics Reports **986**, 1–128 (2022).

[45] A. Szabo and N. S. Ostlund, *Modern quantum chemistry: Introduction to advanced electronic structure theory* (Dover Publications, Mineola, 1996).

[46] H. Mhiri, R. Puig, S. Lerch, M. S. Rudolph, T. Chotibut, S. Thanasilp, and Z. Holmes, *A unifying account of warm start guarantees for patches of quantum landscapes* (2025), arXiv:2502.07889 [quant-ph].

[47] Z. Li and G. K.-L. Chan, *Spin-projected matrix product states: Versatile tool for strongly correlated systems*, Journal of Chemical Theory and Computation **13**, 2681 (2017).

[48] D. W. Berry, Y. Tong, T. Khattar, A. White, T. I. Kim, S. Boixo, L. Lin, S. Lee, G. K.-L. Chan, R. Babbush, and N. C. Rubin, *Rapid initial state preparation for the quantum simulation of strongly correlated molecules* (2024), arXiv:2409.11748 [quant-ph].

[49] C. Gomez, G. Sierra, and M. Ruiz-Altaba, *Quantum groups in two-dimensional physics*, Cambridge Monographs on Mathematical Physics (Cambridge University Press, 2011).

[50] M. Robbiati, E. Pedicillo, A. Pasquale, X. Li, A. Wright, R. M. S. Farias, K. U. Giang, J. Son, J. Knörzer, S. T. Goh, *et al.*, *Double-bracket quantum algorithms for high-fidelity ground state preparation* (2024), arXiv:2408.03987 [quant-ph].

[51] A. Kandala, A. Mezzacapo, K. Temme, M. Takita, M. Brink, J. M. Chow, and J. M. Gambetta, *Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets*, Nature **549**, 242 (2017).

[52] J. Yu, J. R. Moreno, J. T. Iosue, L. Bertels, D. Claudino, B. Fuller, P. Groszkowski, T. S. Humble, P. Jurcevic, W. Kirby, *et al.*, *Quantum-centric algorithm for sample-based Krylov diagonalization* (2025), arXiv:2501.09702 [quant-ph].

[53] S. Raj and B. Coyle, *Hyper compressed fine-tuning of large foundation models with quantum-inspired adapters* (2025), arXiv:2502.06916 [cs.LG].

[54] M. Sciorilli, L. Borges, T. L. Patti, D. García-Martín, G. Camilo, A. Anandkumar, and L. Aolita, *Towards large-scale quantum optimization solvers with few qubits*, Nature Communications **16** (2025).

[55] M. Motta, C. Sun, A. T. K. Tan, M. J. O'Rourke, E. Ye, A. J. Minnich, F. G. S. L. Brandão, and G. K.-L. Chan, *Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution*, Nature Physics **16**, 205–210 (2019).

[56] T. L. Silva, M. M. Taddei, S. Carrazza, and L. Aolita, *Fragmented imaginary-time evolution for early-stage quantum signal processors*, Scientific Reports **13** (2023).

[57] A. Tosta, T. de Lima Silva, G. Camilo, and L. Aolita, *Randomized semi-quantum matrix processing*, npj Quantum Information **10** (2024).

[58] T. de Lima Silva, L. Borges, and L. Aolita, *Partition function estimation with a quantum coin toss* (2024), arXiv:2411.17816 [quant-ph].

[59] L. Lin and Y. Tong, *Heisenberg-limited ground-state energy estimation for early fault-tolerant quantum computers*, PRX Quantum **3**, 010318 (2022).

[60] K. Wan, M. Berta, and E. T. Campbell, *Randomized quantum algorithm for statistical phase estimation*, Physical Review Letters **129** (2022).

[61] B. O'Neill, *Elementary differential geometry*, Rev. 2nd ed. (Elsevier Academic Press, 2006).

[62] L. E. Blumenson, *A derivation of n-dimensional spherical coordinates*, The American Mathematical Monthly **67**, 63 (1960).

[63] Which is the same Jacobian matrix as the one found in Sec.II just below Eq.3, but without the index $\mathbf{x}$, which is dropped for convenience. For explicit formulas, see Eq.(C1) in App.C.

[64] I. Bengtsson and K. Życzkowski, *Geometry of quantum states: An introduction to quantum entanglement* (Cambridge University Press, 2017).

[65] J. M. Lee, *Introduction to Riemannian Manifolds*, Graduate Texts in Mathematics, Vol. 176 (Springer International Publishing, 2018).

[66] Y. H. Dai and Y. Yuan, *A nonlinear conjugate gradient method with a strong global convergence property*, SIAM Journal on Optimization **10**, 177 (1999).

[67] M. R. Hestenes, E. Stiefel, *et al.*, *Methods of conjugate gradients for solving linear systems*, Vol. 49 (NBS Washington, DC, 1952).

[68] T. T. Truong and H.-T. Nguyen, *Backtracking gradient descent method and some applications in large scale optimisation. Part 2: Algorithms and experiments*, Applied Mathematics & Optimization **84**, 2557 (2021).

[69] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes: The art of scientific computing*, 3rd ed. (Cambridge University Press, 2007).

[70] B. N. Datta, *Numerical Linear Algebra and Applications, 2nd Edition*, 2nd ed. (Society for Industrial and Applied Mathematics, Philadelphia, PA, 2010).

[71] A. Abbas, R. King, H.-Y. Huang, W. J. Huggins, R. Movassagh, D. Gilboa, and J. McClean, *On quantum backpropagation, information reuse, and cheating measurement collapse*, in *Advances in Neural Information Processing Systems*, Vol. 36 (Curran Associates, Inc., 2023) pp. 44792–44819.

[72] G.-L. R. Anselmetti, D. Wierichs, C. Gogolin, and R. M. Parrish, *Local, expressive, quantum-number-preserving VQE ansätze for fermionic systems*, New Journal of Physics **23**, 113010 (2021).

[73] J. S. Kottmann, A. Anand, and A. Aspuru-Guzik, *A feasible approach for automatically differentiable unitary coupled-cluster on quantum computers*, Chem. Sci. **12**, 3497 (2021).

[74] K. W. Fang, *Symmetric multivariate and related distributions* (Chapman and Hall/CRC, 2018).

Appendix A: The differential geometry of hyperspheres

Here, we introduce the main concepts of differential geometry used in this paper, as they apply to the study of hyperspheres as subspaces of Euclidean spaces. We give an elementary account of the subject, as given in Ref. [61], and introduce also a few sophisticated results that we need.

Additional notations: Besides the conventions given in Sec. II we define $\mathbf{e}_\ell \coloneqq (\delta_{\ell,j})_{j=1}^d \in \mathbb{R}^d$, where $\delta_{\ell,j}$ is the Kronecker delta symbol, for the canonical basis vectors in $\mathbb{R}^d$, use the letters γ and τ for functions defining differentiable curves on manifolds, and write γ', γ'' for the first and second derivatives of a real function of a single variable γ. We use the symbol $\langle\ ;\ \rangle$ to denote the canonical Euclidean inner product function $\langle\ ;\ \rangle : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ given by $\langle \boldsymbol{v}; \boldsymbol{u}\rangle \coloneqq \sum_{\ell=1}^d v_\ell u_\ell$ with v_ℓ, u_ℓ being the components of the basis vector $\mathbf{e}_\ell$ in $\boldsymbol{v}, \boldsymbol{u}$ respectively, and we use the symbol $\|\ \|$ for the induced Euclidean two-norm on $\mathbb{R}^d$.

Tangent spaces: Consider the vector space $\mathbb{R}^d$ with its standard Euclidean inner product and recall that the unit hypersphere is the set $\mathbb{S}^{d-1}$ of points $\mathbf{x} \in \mathbb{R}^d$ that satisfy the quadratic equation $\|\mathbf{x}\|^2 = 1$. Our approach is to define the geometric properties of $\mathbb{S}^{d-1}$ by studying the properties of sets of differentiable curves $\gamma : [0,\, 1] \to \mathbb{S}^{d-1}$ on them. But first, let us build intuition by considering sets of differentiable curves in $\mathbb{R}^d$. Let $\mathbf{a} \in \mathbb{R}^d$ be some point and let $\tau : [0, 1] \to \mathbb{R}^d$ be an arbitrary differentiable curve with $\tau(0) = \mathbf{a}$. Since τ is differentiable, there exists a unique tangent vector to τ at the point $\mathbf{a}$ given by its derivative $\tau'(0)$. We call the set of all possible tangent vectors to curves of this form the *tangent space of* $\mathbb{R}^d$ *at* $\mathbf{a}$, denoted by $T_{\mathbf{a}}\mathbb{R}^d$. Since a tangent vector to a curve in $\mathbb{R}^d$ defines a unique tangent line to that curve, every element of $T_{\mathbf{a}}\mathbb{R}^d$ can be obtained by computing tangent vectors to lines of the form $\mathbf{a} + t\boldsymbol{v}$ for some $\boldsymbol{v} \in \mathbb{R}^d$. This makes $T_{\mathbf{a}}\mathbb{R}^d$ an isomorphic copy of $\mathbb{R}^d$ but centered at the point $\mathbf{a}$. Therefore, the points in $\mathbb{R}^d$ that represent canonical basis vectors form a basis for the tangent spaces of $\mathbb{R}^d$ at every point. This justifies using the same notation $\mathbf{e}_j$ for the jth canonical basis vector of any tangent space of $\mathbb{R}^d$. Similarly, let $T_{\mathbf{a}}\mathbb{S}^{d-1}$ be the set of all the tangent vectors to the curves $\gamma : [0, 1] \to \mathbb{S}^{d-1}$ with $\gamma(0) = \mathbf{a}$ at the point $\mathbf{a} \in \mathbb{S}^{d-1}$. Since for every $t \in [0, 1]$ we must have $\langle \gamma(t); \gamma(t)\rangle = 1$, we get by differentiation and the properties of the Euclidean inner product that $\langle \gamma(t); \gamma'(t)\rangle = 0$ for all $t \in [0, 1]$, and in particular for $t = 0$ we have $\langle \mathbf{a}; \gamma'(0)\rangle = 0$, *i.e.* the tangent space $T_{\mathbf{a}}\mathbb{S}^{d-1}$ is the subspace of all $\boldsymbol{v} \in T_{\mathbf{a}}\mathbb{R}^d$ such that $\langle \mathbf{a}; \boldsymbol{v}\rangle = 0$.

Natural gradient: Tangent spaces allow us to define linear maps that approximate the effect of smooth maps at a point. For example, let $\mathcal{L} : \mathbb{R}^d \to \mathbb{R}$ be a smooth function. We know from vector calculus that near a point $\mathbf{a}$, we have $\mathcal{L}(\mathbf{a} + \boldsymbol{v}) \approx \mathcal{L}(\mathbf{a}) + \sum_{j=1}^n (\partial_{x_j}\mathcal{L})_{\mathbf{a}} v_j$. The second term of this sum can be interpreted as the action of the linear functional $\mathrm{d}\mathcal{L}_{\mathbf{a}} : T_{\mathbf{a}}\mathbb{R}^d \to \mathbb{R}$ on $\boldsymbol{v} \in T_{\mathbf{a}}\mathbb{R}^d$ defined by $\mathrm{d}\mathcal{L}_{\mathbf{a}}(\boldsymbol{v}) \coloneqq \langle (\boldsymbol{\partial}_{\mathbf{x}}\mathcal{L})_a; \boldsymbol{v}\rangle = \sum_{j=1}^n (\partial_{x_j}\mathcal{L})_{\mathbf{a}} v_j$, where the last equation is the expression of the inner product in the canonical basis of $T_{\mathbf{a}}\mathbb{R}^d$. The linear functional $\mathrm{d}\mathcal{L}_{\mathbf{a}}$ is called the *differential* of $\mathcal{L}$ at $\mathbf{a}$, and its generalizations play a major role in differential geometry. The vector $\mathrm{grad}^{\mathrm{euc}}_{\mathbf{a}}(\mathcal{L}) \coloneqq (\boldsymbol{\partial}_{\mathbf{x}}\mathcal{L})_a \in T_{\mathbf{a}}\mathbb{R}^d$ is called the (euclidean) *gradient* of $\mathcal{L}$ at $\mathbf{a}$, and gives the direction of steepest ascent of $\mathcal{L}$ at $\mathbf{a}$. If $\mathbf{a} \in \mathbb{S}^{d-1}$, we can define the "spherical" gradient of $\mathcal{L}$ at $\mathbf{a}$, also called *natural gradient*, as the vector of steepest ascent of $\mathcal{L}$ at $\mathbf{a}$ among the vectors tangent to $\mathbb{S}^{d-1}$. We can calculate it by computing the differential $\mathrm{d}\mathcal{L}_{\mathbf{a}}$ on the vectors in $T_{\mathbf{a}}\mathbb{S}^{d-1}$. Since $T_{\mathbf{a}}\mathbb{S}^{d-1}$ consists of vectors in $T_{\mathbf{a}}\mathbb{R}^d$ that are orthogonal to $\mathbf{a}$, for any $\boldsymbol{v} \in T_{\mathbf{a}}\mathbb{R}^d$ the vector $\Pi_{\mathbf{a}}\boldsymbol{v}$ is in $T_{\mathbf{a}}\mathbb{S}^{d-1}$, where $\Pi_{\mathbf{a}}$ is the projection operator on $T_{\mathbf{a}}\mathbb{S}^{d-1}$ with a matrix representation given by $\mathbb{1} - [\mathbf{a}][\mathbf{a}]^T$. Then, since projection operators are symmetric, we have $\mathrm{d}\mathcal{L}_{\mathbf{a}}(\Pi_a\boldsymbol{v}) = \langle (\boldsymbol{\partial}_{\mathbf{x}}\mathcal{L})_{\mathbf{a}}; \Pi_a\boldsymbol{v}\rangle = \langle (\Pi_a(\boldsymbol{\partial}_{\mathbf{x}}\mathcal{L})_{\mathbf{a}}); \boldsymbol{v}\rangle$, from which we get

$$\mathrm{grad}^{\mathrm{sph}}_{\mathbf{a}}(\mathcal{L}) = \Pi_{\mathbf{a}}(\boldsymbol{\partial}_{\mathbf{x}}\mathcal{L})_{\mathbf{a}}\,. \tag{A1}$$

Metric and geodesics: So far, we have been assuming that the tangent spaces $T_{\mathbf{a}}\mathbb{R}^d$ have linear and Euclidean structures, in the sense that each $T_{\mathbf{a}}\mathbb{R}^d$ naturally inherits the same inner product function as $\mathbb{R}^d$ for every point $\mathbf{a}$ under its isomorphism with $\mathbb{R}^d$. It turns out that this fact is the distinguishing feature of Euclidean geometry. A function that associates a positive definite inner product with each point in $\mathbb{R}^d$ is called a *metric*, and the metric that chooses the standard Euclidean inner product for every $T_{\mathbf{a}}\mathbb{R}^d$ is called the *standard metric of* $\mathbb{R}^d$. Fixing a choice of metric for $\mathbb{R}^d$ transforms it into a *Riemannian manifold*. Among other things, the standard metric allows us to define the (standard) length of a differentiable curve $\tau : [0,\, 1] \to \mathbb{R}^d$ by integrating $\|\tau'\|$ over its domain, where the norm is defined using the inner product function given above for the tangent space of every point in τ. Since each $T_{\mathbf{a}}\mathbb{S}^{d-1}$ is a linear subspace of $T_{\mathbf{a}}\mathbb{R}^d$ for every $\mathbf{a} \in \mathbb{S}^{d-1}$, it inherits the inner product function of $T_{\mathbf{a}}\mathbb{R}^d$, allowing us to define a metric for $\mathbb{S}^{d-1}$ called the *round metric*. Therefore, the length of a curve $\gamma : [0, 1] \to \mathbb{S}^{d-1}$ in the round metric has the same form as if we computed it as a curve in $\mathbb{R}^d$, given by

$$L(\gamma) \coloneqq \int_0^1 \mathrm{d}\eta\ \sqrt{\langle \gamma'(\eta); \gamma'(\eta)\rangle}\,. \tag{A2}$$

Using this definition of length, we can define a curve that is the analog of a straight line but in $\mathbb{S}^{d-1}$, called a *geodesic*, which is a curve on $\mathbb{S}^{d-1}$ having minimal length between any pair of its points. For each point $\mathbf{a} \in \mathbb{S}^{d-1}$ and each tangent vector $\boldsymbol{v} \in T_{\mathbf{a}}\mathbb{S}^{d-1}$, there is a unique geodesic curve $\gamma : [0,1] \to \mathbb{S}^{d-1}$ with $\gamma(0) = \mathbf{a}$ and $\gamma'(0) = \boldsymbol{v}$, and it is obtained by minimizing $L(\gamma)$ over the set of all possible curves that satisfy these conditions. It turns out that this problem is equivalent to finding the minimum of the functional

$$E(\gamma) \coloneqq \int_0^1 \mathrm{d}\eta \, \left(\frac{1}{2} \langle \gamma'(\eta); \gamma'(\eta) \rangle + \lambda \left(\langle \gamma(\eta); \gamma(\eta) \rangle - 1 \right) \right), \tag{A3}$$

where the curve γ represents the trajectory of a free particle, with the first term in parentheses being the particle's energy and the second term being a constraint term imposing that this particle should stay on $\mathbb{S}^{d-1}$ at all times, with λ being a Lagrange multiplier. The solution to this problem is given by solving the Euler-Lagrange equation $\gamma'' = \lambda\, \gamma$ with the initial conditions $\gamma(0) = \mathbf{a}$ and $\gamma'(0) = \boldsymbol{v}$, resulting in

$$\gamma(\eta) = \cos(\eta \, \|\boldsymbol{v}\|) \, \mathbf{a} + \sin(\eta \, \|\boldsymbol{v}\|) \, \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} \, . \tag{A4}$$

Equation (A4) shows that the geodesics of $\mathbb{S}^{d-1}$ are great circles, *i.e.* the intersection of the hypersphere with a hyperplane that contains the initial point and the center of the hypersphere.

Exponential map and parallel transport: As we have already said, the geodesics of the Riemannian manifold are analogs of straight lines in Euclidean spaces. This allows us to define the analog of the straight motion of geometric objects on the manifold. Formally, when the transported object is a point this is described by the *Riemannian exponential map* $\mathrm{ExpMap}_{\mathbf{a}} : T_{\mathbf{a}}\mathbb{S}^{d-1} \to \mathbb{S}^{d-1}$, that takes a velocity vector $\boldsymbol{v} \in T_{\mathbf{a}}\mathbb{S}^{d-1}$ at the point $\mathbf{a}$ and maps it to the endpoint $\gamma(1) \in \mathbb{S}^{d-1}$ of the geodesic defined by the initial conditions $\gamma(0) = \mathbf{a}$ and $\gamma'(0) = \boldsymbol{v}$. To see how to transport tangent vectors, note that the equation of the geodesic curve corresponds to the rotation formula for an arbitrary point in $\mathbb{R}^d$ on the 2-plane generated by the unit vectors $\mathbf{a}$ and $\boldsymbol{v}/\|\boldsymbol{v}\|$. If we look at the effect of this rotation on the hyperplane $T_{\mathbf{a}}\mathbb{S}^{d-1}$, as a subset of $\mathbb{R}^d$, we see that it not only changes the tangency point but also changes the relative orientation of the hyperplane so that it remains tangent to $\mathbb{S}^{d-1}$. Under this rotation, the straight line generated by a vector $\boldsymbol{u} \in T_{\mathbf{a}}\mathbb{S}^{d-1}$ that is orthogonal to the vector $\boldsymbol{v}$ is mapped to a parallel line. This observation motivates the definition of a *parallel transport map* $\mathcal{T}_{\eta\boldsymbol{v}} : T_{\mathbf{a}}\mathbb{S}^{d-1} \to T_{\gamma(\eta)}\mathbb{S}^{d-1}$ given by the

$$\begin{aligned} \mathcal{T}_{\eta\boldsymbol{v}}(\boldsymbol{u}) &\coloneqq \left(\boldsymbol{u} - \frac{\langle \boldsymbol{v}; \boldsymbol{u} \rangle}{\|\boldsymbol{v}\|^2} \, \boldsymbol{v} \right) + \frac{\langle \boldsymbol{v}; \boldsymbol{u} \rangle}{\|\boldsymbol{v}\|^2} \, \gamma'(\eta) \\ &= \boldsymbol{u} - \sin(\eta \, \|\boldsymbol{v}\|) \, \frac{\langle \boldsymbol{v}; \boldsymbol{u} \rangle}{\|\boldsymbol{v}\|} \, \mathbf{a} + (\cos(\eta \, \|\boldsymbol{v}\|) - 1) \, \frac{\langle \boldsymbol{v}; \boldsymbol{u} \rangle}{\|\boldsymbol{v}\|^2} \, \boldsymbol{v} \, , \end{aligned} \tag{A5}$$

where the expression in parentheses is the component of $\boldsymbol{u}$ that is invariant under this rotation. This map is called parallel transport because it changes the tangent vectors in such a way that their relative orientation w.r.t. the tangent hyperplane is the same before and after moving along a geodesic.

The differential: To use these results in our method, we need to describe how to connect the description of $\mathbb{S}^{d-1}$ as a subspace of $\mathbb{R}^d$ and its parameterization in hyperspherical coordinates. As we will show shortly, a parameterization of a d_1-dimensional surface M in $\mathbb{R}^d$ can be realized as a bijection between it and a space of parameters, which is usually some subset of $\mathbb{R}^{d_1}$. We can use this function to translate back and forth between parameters and points in our surface, so it would be convenient if we could also use this function to somehow translate geometric objects defined in points of our surface into geometric objects in the space of parameters. This is done using the differential, which we will now proceed to define. Let $\mathbf{f} : \mathbb{R}^{d_1} \to \mathbb{R}^{d_2}$ be a smooth map and $\mathbf{a} \in \mathbb{R}^{d_1}$ be a point, the *differential of* $\mathbf{f}$ at $\mathbf{a}$ is the linear map $\mathrm{d}\mathbf{f}_{\mathbf{a}} : T_{\mathbf{a}}\mathbb{R}^{d_1} \to T_{\mathbf{f}(\mathbf{a})}\mathbb{R}^{d_2}$ defined by $\mathrm{d}\mathbf{f}_{\mathbf{a}}(\boldsymbol{v}) \coloneqq (\partial_t \mathbf{f}(\mathbf{a} + t\boldsymbol{v}))_0$ (in case the meaning of this expression is unclear, recall the notational conventions for derivatives given in Sec.II). By the chain rule, we get $\mathrm{d}\mathbf{f}_{\mathbf{a}}(\boldsymbol{v}) = \langle (\boldsymbol{\partial}_{\mathbf{x}}\mathbf{f})_{\mathbf{a}}; \boldsymbol{v} \rangle$. Expanding both $\mathbf{f}$ and $\boldsymbol{v}$ in their respective canonical bases, we obtain the vector $(\sum_{\ell=1}^{n} (\partial_{x_\ell} f_j)_{\mathbf{a}} v_\ell)_{j=1}^{n}$, from which we can read the matrix representation $\mathbf{J}_{\mathbf{f}}(\mathbf{a})$ of $\mathrm{d}\mathbf{f}_{\mathbf{a}}$ with coefficients (j, ℓ) given by $(\partial_{x_\ell} f_j)_{\mathbf{a}}$ called the *Jacobian matrix of* $\mathbf{f}$ at $\mathbf{a}$.

Hyperspherical coordinates: To show how the differential does the translation of geometric objects between a surface and its parameter space, we will use as an example the hyperspherical coordinate system on $\mathbb{S}^{d-1}$. These coordinates are defined by choosing the ordered sequence $(\mathbf{e}_1, \ldots, \mathbf{e}_d)$ of basis vectors and applying the iterative process described in Ref. [62], producing a map $\mathbf{x} : D \subset \mathbb{R}^{d-1} \to \mathbb{S}^{d-1}$ where

$$\mathbf{x}(\boldsymbol{\theta}) \coloneqq \cos(\theta_1) \, \mathbf{e}_1 + \sum_{j=1}^{d-2} \left(\prod_{\ell=1}^{j} \sin(\theta_\ell) \right) \cos(\theta_{j+1}) \, \mathbf{e}_{j+1} + \left(\prod_{j=1}^{d-1} \sin(\theta_j) \right) \mathbf{e}_d, \tag{A6}$$

such that $D \coloneqq [0,\,\pi]^{d-2} \times [0,\,2\pi)$ with $\boldsymbol{\theta} = (\theta_j)_{j=1}^{d-1}$ being a vector in D representing hyperspherical angles. By construction, this map is smooth and has a continuous inverse, but it does not have a smooth inverse over all of D. To see this, we can compute the differential $\mathrm{d}\mathbf{x}_{\boldsymbol{\phi}}$ at a generic point $\phi \in D \subset \mathbb{R}^{d-1}$ and see if the Jacobian matrix $\mathbf{J}_{\mathbf{x}}(\boldsymbol{\phi})$ [63] has full rank there.

Tangent vectors and *pushforwards*: From now on, we will only consider the points of $\boldsymbol{\phi} \in \mathbb{R}^{d-1}$ for which the Jacobian has full rank. For these points, the action of the differential on the canonical basis vectors of $T_{\boldsymbol{\phi}}\mathbb{R}^{d-1}$ defines a basis on the tangent spaces $T_{\mathbf{x}(\boldsymbol{\phi})}\mathbb{S}^{d-1}$ called the *hyperspherical coordinate basis*. Therefore, the differential "pushes" of any vector in $T_{\boldsymbol{\phi}}\mathbb{R}^{d-1}$ to some vector in $T_{\mathbf{x}(\boldsymbol{\phi})}\mathbb{S}^{d-1}$, which is why $\mathrm{d}\mathbf{x}_{\boldsymbol{\phi}}$ it is also called the *pushforward* of $\mathbf{x}$ at the point $\boldsymbol{\phi}$, where "forward" refers to the choice of initial and final spaces being the same as the ones for the function $\mathbf{x}$.

Pullback of the metric: To see what the differential does to the metric on $\mathbb{S}^{d-1}$, let $\mathbf{a} \in \mathbb{S}^{d-1}$ be a point for which there is some $\boldsymbol{\phi}$ with $\boldsymbol{\phi} \coloneqq \mathbf{x}^{-1}(\mathbf{a})$. Since we can push any pair of vectors $\boldsymbol{v}, \boldsymbol{u} \in T_{\boldsymbol{\phi}}\mathbb{R}^{d-1}$ to a pair of vectors $\mathrm{d}\mathbf{x}_{\boldsymbol{\phi}}(\boldsymbol{v}), \mathrm{d}\mathbf{x}_{\boldsymbol{\phi}}(\boldsymbol{u}) \in T_{\mathbf{a}}\mathbb{S}^{d-1}$, we can define an inner product function $\langle\ ;\ \rangle_{\boldsymbol{\phi}} : (T_{\boldsymbol{\phi}}\mathbb{R}^{d-1})^2 \to \mathbb{R}$ by $\langle \boldsymbol{v}; \boldsymbol{u}\rangle_{\boldsymbol{\phi}} \coloneqq \langle \mathrm{d}\mathbf{x}_{\boldsymbol{\phi}}(\boldsymbol{v}); \mathrm{d}\mathbf{x}_{\boldsymbol{\phi}}(\boldsymbol{u})\rangle_{\mathbf{a}}$ for all $\boldsymbol{v}, \boldsymbol{u} \in T_{\boldsymbol{\phi}}\mathbb{R}^{d-1}$. This inner product is called the *pullback* of the Euclidean inner product give in $T_{\mathbf{a}}\mathbb{S}^{d-1}$, since it is defined by "pulling back" the Euclidean inner product from $T_{\mathbf{a}}\mathbb{S}^{d-1}$ to $T_{\boldsymbol{\phi}}\mathbb{R}^{d-1}$. The matrix representation of this inner product is the *metric tensor for the hyperspherical coordinates* and is given by $\mathbf{g}(\boldsymbol{\phi}) \coloneqq \mathbf{J}_{\mathbf{x}}(\boldsymbol{\phi})^T\mathbf{J}_{\mathbf{x}}(\boldsymbol{\phi})$, which is the same matrix as the one given in Eq.1 when evaluated on the fixed HW ansatz state.

Pullback of the natural gradient: With the pullback of the metric, we can also express the natural gradient of a function h on $\mathbb{S}^{d-1}$ at a point $\mathbf{a}$ as a vector in $T_{\boldsymbol{\phi}}\mathbb{R}^{d-1}$, which amounts to defining a "pullback" for the natural gradient. First, let $\tilde{\mathcal{L}}(\boldsymbol{\theta}) \coloneqq (\tilde{\mathcal{L}} \circ \mathbf{x})(\boldsymbol{\theta}) = \mathcal{L}(\mathbf{x}(\boldsymbol{\theta}))$, which means that $\tilde{\mathcal{L}} : \mathbb{R}^{d-1} \to \mathbb{R}$ is a smooth function between Euclidean spaces. By the definition of the differential we have $\mathrm{d}\tilde{\mathcal{L}}_{\boldsymbol{\phi}}(\boldsymbol{v}) = \langle(\boldsymbol{\partial}_{\boldsymbol{\theta}}\tilde{\mathcal{L}})_{\boldsymbol{\phi}}; \boldsymbol{v}\rangle$ for all $\boldsymbol{v} \in T_{\boldsymbol{\phi}}\mathbb{R}^{d-1}$. It is tempting to read the vector $(\boldsymbol{\partial}_{\boldsymbol{\theta}}\tilde{\mathcal{L}})_{\boldsymbol{\phi}}$ as the expression of the natural gradient on $\mathbb{S}^{d-1}$ as a vector in $T_{\boldsymbol{\phi}}\mathbb{R}^{d-1}$. However, since the Euclidean norm in $T_{\boldsymbol{\phi}}\mathbb{R}^{d-1}$ is not the same as the pullback over $\mathbf{x}$ of the Euclidean inner product on $\mathbb{S}^{d-1}$, the correct expression for the natural gradient must be a vector $\boldsymbol{u}$ that satisfies the equation $\langle \boldsymbol{u}; \boldsymbol{v}\rangle_{\boldsymbol{\phi}} = \langle(\boldsymbol{\partial}_{\boldsymbol{\theta}}\tilde{\mathcal{L}})_{\boldsymbol{\phi}}; \boldsymbol{v}\rangle$. This guarantees that the image of $\boldsymbol{u}$ under the action of $\mathrm{d}\mathbf{x}_{\boldsymbol{\phi}}$ is exactly given by $\mathrm{grad}^{sph}_{\mathbf{a}}(\mathcal{L})$, or in other words, that $\boldsymbol{u}$ is the *pullback* of $\mathrm{grad}^{sph}_{\mathbf{a}}(\mathcal{L})$ under $\mathbf{x}$. Notice that it is a pullback instead of a pushforward. This is a consequence of the fact that the natural gradient is the dual vector of a linear functional, a.k.a. a pseudovector, and as such, it is transformed via a pullback under $\mathbf{x}$, not a pushforward.

Natural gradient in different basis: Using the metric tensor $\mathbf{g}(\boldsymbol{\phi})$, we can see that the matrix used to express the natural gradient on $\mathbb{S}^{d-1}$ as an element in $T_{\boldsymbol{\phi}}\mathbb{R}^{d-1}$ is given by $\mathbf{g}^{-1}(\boldsymbol{\phi})(\boldsymbol{\partial}_{\boldsymbol{\theta}}\tilde{\mathcal{L}})_{\boldsymbol{\phi}}$, where we do a slight abuse of notation by identifying a vector with its matrix representation. This allows us to write the matrix representation of the natural gradient on $\mathbb{S}^{d-1}$, now again as a vector in $T_{\mathbf{a}}\mathbb{S}^{d-1}$, but expressed in terms of hyperspherical coordinates, by applying the pushforward under $\mathbf{x}$ giving us $\mathbf{J}_{\mathbf{x}}(\boldsymbol{\phi})\mathbf{g}^{-1}(\boldsymbol{\phi})(\boldsymbol{\partial}_{\boldsymbol{\theta}}\tilde{\mathcal{L}})_{\boldsymbol{\phi}}$ which is the same expression found in Sec.II just below Eq.3 where the index $\mathbf{x}$ was dropped and the substitution $\boldsymbol{\phi} \to \boldsymbol{\theta}_t$ was made.

This finishes our discussion of all of the necessary background information to understand how our approach to VQA works. Next, we will briefly discuss how to extend our method to states with complex amplitudes. It will be slightly more technical and will invoke some results without stating them.

Spheres and the space of quantum states: It is well known that the space of pure quantum states over d level is the *complex projective space* $\mathbb{CP}^{d-1}$, which is a Riemannian manifold with the Fubini-Study metric [64]. It is also a well known result that $\mathbb{CP}^{d-1} = \mathbb{S}^{2d-1}/U(1)$, where $U(1)$ acts as multiplication by a global phase. The Fubini-Study metric is the quotient metric associated with the round metric on $\mathbb{S}^{2d-1}$ [65, Example 2.30]. If we use the general HWE encoder as our ansatz [27] to prepare a parametrized quantum state on $\mathbb{CP}^{d-1}$, we can use the fact that a global phase φ_{global} is not an observable to prove that for any loss function $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi})$, $\partial_{\varphi_{global}}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\varphi}) = 0$. This implies that the natural gradient of $\mathcal{L}$ in $\mathbb{S}^{2d-1}$ does not have any component in the direction of the global phase, so the natural gradient on $\mathbb{CP}^{d-1}$ is given by the same vector [6, Section 3.6.2], and a geodesic on $\mathbb{S}^{2d-1}$ that is defined by some natural gradient in $\mathcal{L}$ is mapped into a geodesic on $\mathbb{CP}^{d-1}$ since the differential of the quotient map from $\mathbb{S}^{2d-1}$ to $\mathbb{CP}^{d-1}$ is a Riemannian isometry [64]. Therefore, we can do our optimization completely by embedding the state vectors in the parameterization of $\mathbb{S}^{2d-1}$ given by the ansatz after removing a global phase from the initial data vector.

Algorithm 1: Riemannian Conjugate Gradient

Input: Loss function $\mathcal{L}$, transport function $\mathscr{T}^{(t)}$, tolerance $\varepsilon > 0$, maximum number of iterations $N_{\max}$
Output: Critical point $\boldsymbol{\theta}_t^\star$ and $\mathcal{L}(\boldsymbol{\theta}_t^\star)$

$\boldsymbol{v}_0 \leftarrow -\mathbf{J}(\boldsymbol{\theta}_0)\, \boldsymbol{g}^{-1}(\boldsymbol{\theta}_0)\, (\boldsymbol{\partial}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}))_{\boldsymbol{\theta}_0}$
$\boldsymbol{u}_0 \leftarrow \boldsymbol{v}_0$
for $t \leftarrow 0, 1, \ldots, N_{\max}$ **do**
  $\text{res} \leftarrow \langle -\boldsymbol{v}_t, \boldsymbol{u}_t \rangle_{\mathbf{x}_t}^2 \,/ \|\boldsymbol{u}_t\|^2$
  if $|\text{res}| < \varepsilon$ **then**
    break
  $\eta_t \leftarrow$ OptStepSize($\mathcal{L}$, $\mathscr{T}^{(t)}$, $\mathbf{x}_t$, $\boldsymbol{v}_t$, $\boldsymbol{u}_t$)
  $\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}\left(\text{ExpMap}_{\mathbf{x}_t}(\eta_t\, \boldsymbol{u}_t)\right)$ (see Eq. (3))
  $\boldsymbol{v}_{t+1} \leftarrow -\mathbf{J}(\boldsymbol{\theta}_{t+1})\, \boldsymbol{g}^{-1}(\boldsymbol{\theta}_{t+1})\, (\boldsymbol{\partial}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}))_{\boldsymbol{\theta}_{t+1}}$
  $\beta_{t+1} \leftarrow$ from Eqs. (B6)
  $\boldsymbol{u}_{t+1} \leftarrow$ from Eq. (B3)
  $\boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t+1}$; $\boldsymbol{v}_t \leftarrow \boldsymbol{v}_{t+1}$; $\boldsymbol{u}_t \leftarrow \boldsymbol{u}_{t+1}$; $\mathbf{x}_t \leftarrow \mathbf{x}_{t+1}$
return $\boldsymbol{\theta}_t$, $\mathcal{L}(\boldsymbol{\theta}_t)$

Appendix B: Riemannian optimization in the hypersphere

In optimization tasks, the gradient-descent method is the simplest algorithm to find a minimum of sufficiently smooth functions. We first recall that our circuit ansatze [27] parametrize quantum states $|\psi(\boldsymbol{\theta}(\mathbf{x}))\rangle$ as points in a manifold that is the surface of a $(d-1)$-dimensional unit sphere $\mathbb{S}^{d-1}$. We also recall that geodesics that connect two points $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ are not a straight line. One way to move along a geodesic is to consider the great circle $\gamma(\eta_t)$ such that $\gamma(0) = \mathbf{x}_t$ and $\left(\frac{d}{d\eta_t}\gamma(\boldsymbol{v}_t)\right)_{\eta_t=0} = \zeta_t$. Thus, we can use the great circle given by the *exponential map* [6, 7]:

$$\mathbf{x}_{t+1} = \text{ExpMap}_{\mathbf{x}_t}(\eta_t\, \boldsymbol{v}_t) = \cos(\eta_t\, \|\boldsymbol{v}_t\|)\, \mathbf{x}_t + \sin(\eta_t\, \|\boldsymbol{v}_t\|)\, \frac{\boldsymbol{v}_t}{\|\boldsymbol{v}_t\|}\,, \tag{B1}$$

where $\|\boldsymbol{v}_t\| \coloneqq \sqrt{\langle \boldsymbol{v}_t, \boldsymbol{v}_t \rangle_{\mathbf{x}_t}}$, and, in this context, the scaling factor η_t can be understood as a learning rate. Equation (3) defines a closed line on $\mathbb{S}^{d-1}$, with the geodesic coming back to the original point when $\eta_t\, \|\boldsymbol{v}_t\| = 2\pi j$, $\forall\, j \in \mathbb{Z}$.

We enhance the optimization scheme in Eq. (B1) using a Riemannian conjugate gradient (CG) method with convergence guarantees for lower bounded and sufficiently smooth objective functions. This addition distances the above optimizer even further from the first- [4] and second-order [8] approximations in the literature. In flat geometry, we formulate the CG as follows. For $t = 0$, the direction of descent is given by the natural gradient, $\boldsymbol{v}_0 = -\mathbf{J}(\boldsymbol{\theta}_0)\, \boldsymbol{g}^{-1}(\boldsymbol{\theta}_0)(\boldsymbol{\partial}_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}))_{\boldsymbol{\theta}_0}$. For $t > 0$, CG uses the natural gradient $\boldsymbol{v}_t$ and a memory from the previous direction, $\boldsymbol{u}_t$, *i.e.*

$$\boldsymbol{u}_{t+1} = \boldsymbol{v}_{t+1} + \beta_{t+1}\, \boldsymbol{u}_t, \tag{B2}$$

for some choice of $\beta_{t+1} \geq 0$. Since the memory direction $\boldsymbol{u}_t$ in Eq. (B2) was calculated in the previous iteration, the equivalent update in curved space requires the transport of the vector $\boldsymbol{u}_t$ along the geodesic connecting $\mathbf{x}_t$ and $\mathbf{x}_{t+1}$ to properly perform the addition of these two vectors. Ref. [20] considers a general framework of Riemannian CG methods where they allow for a general map $\mathscr{T}^{(t)} : T_{\mathbf{x}_t}\mathcal{M} \to T_{\mathbf{x}_{t+1}}\mathcal{M}$ to transport the direction $\boldsymbol{u}_t \in T_{\mathbf{x}_t}\mathcal{M}$ to $T_{\mathbf{x}_{t+1}}\mathcal{M}$,

$$\boldsymbol{u}_{t+1} = \boldsymbol{v}_{t+1} + \beta_{t+1} s_t \mathscr{T}^{(t)}(\boldsymbol{u}_t), \tag{B3}$$

where s_t is a scaling parameter satisfying

$$0 < s_t \leq \min\left\{1, \frac{\sqrt{\langle \boldsymbol{u}_t, \boldsymbol{u}_t \rangle_{\mathbf{x}_t}}}{\sqrt{\langle \mathscr{T}^{(t)}(\boldsymbol{u}_t),\, \mathscr{T}^{(t)}(\boldsymbol{u}_t) \rangle_{\mathbf{x}_{t+1}}}}\right\}. \tag{B4}$$

While any choice of s_t in the above interval suffices, here we choose s_t to saturate the inequality on the right-hand side of Eq. (B4). In this work, we generate the sequence $\{\mathbf{x}_t\}$ on the manifold $\mathcal{M}$ from an initial point $\mathbf{x}_0 \in \mathcal{M}$ using

the exponential map in Eq. (3), and the search direction $\boldsymbol{u}_t$ lies in the tangent space $\mathrm{T}_{\mathbf{x}_t}\mathcal{M}$. Here, we use the exact geodesic transport of an arbitrary vector $\zeta_t \in \mathrm{T}_{\mathbf{x}_t}\mathcal{M}$ on $\mathbb{S}^{d-1}$ along the geodesic (see Eq. (A5)),

$$\mathscr{T}^{(t)}(\zeta_t) \coloneqq \mathcal{T}_{\eta_t \boldsymbol{u}_t}(\zeta_t) \coloneqq \zeta_t - \sin(\eta_t \|\boldsymbol{u}_t\|)\, \frac{\langle \boldsymbol{u}_t,\, \zeta_t\rangle_{\mathbf{x}_t}}{\|\boldsymbol{u}_t\|}\, \mathbf{x}_t + (\cos(\eta_t \|\boldsymbol{u}_t\|) - 1)\, \frac{\langle \boldsymbol{u}_t,\, \zeta_t\rangle_{\mathbf{x}_t}}{\|\boldsymbol{u}_t\|^2}\, \boldsymbol{u}_t\,. \tag{B5}$$

In addition, we need to choose β_{t+1} accordingly. Here, we consider one hybrid Riemannian CG method from Ref. [20] (initially proposed for Euclidean optimization in Ref. [40]), using the choices β_{t+1} from Dai and Yuan [66] as well as Hestenes and Stiefel [67],

$$\begin{aligned}
\beta_{t+1} &\coloneqq \max\left\{0,\, \min\{\beta_{t+1}^{\mathrm{DY}},\, \beta_{t+1}^{\mathrm{HS}}\}\right\};\\
\beta_{t+1}^{\mathrm{DY}} &\coloneqq \frac{\langle -\boldsymbol{v}_{t+1},\, -\boldsymbol{v}_{t+1}\rangle_{\mathbf{x}_{t+1}}}{\langle -\boldsymbol{v}_{t+1},\, s_t\, \mathcal{T}_{\eta_t \boldsymbol{u}_t}(\boldsymbol{u}_t)\rangle_{\mathbf{x}_{t+1}} - \langle -\boldsymbol{v}_t,\, \boldsymbol{u}_t\rangle_{\mathbf{x}_t}};\\
\beta_{t+1}^{\mathrm{HS}} &\coloneqq \frac{\langle -\boldsymbol{v}_{t+1},\, -\boldsymbol{v}_{t+1}\rangle_{\mathbf{x}_{t+1}} - \langle -\boldsymbol{v}_{t+1},\, \ell_t\, \mathcal{T}_{\eta_t \boldsymbol{u}_t}(-\boldsymbol{v}_t)\rangle_{\mathbf{x}_{t+1}}}{\langle -\boldsymbol{v}_{t+1},\, s_t\, \mathcal{T}_{\eta_t \boldsymbol{u}_t}(\boldsymbol{u}_t)\rangle_{\mathbf{x}_{t+1}} - \langle -\boldsymbol{v}_t,\, \boldsymbol{u}_t\rangle_{\mathbf{x}_t}},
\end{aligned} \tag{B6}$$

where ℓ_t has a similar role to Eq. (B4), but for the transport of the gradient.

So far, we assumed we have an appropriate step size $\eta_t > 0$ — also known as the learning rate. However, to guarantee the convergence properties at each step, we need a step size that satisfies the $\mathscr{T}^{(t)}$-Wolfe conditions (*cf.* Ref. [20]),

$$\mathcal{L}\left(\boldsymbol{\theta}(\mathbf{x}_{t+1})\right) - \mathcal{L}\left(\boldsymbol{\theta}(\mathbf{x}_t)\right) \leq c_1\, \eta_t\, \langle -\boldsymbol{v}_t,\, \boldsymbol{u}_t\rangle_{\mathbf{x}_t}\,, \tag{B7}$$

$$\left|\langle -\boldsymbol{v}_{t+1},\, \mathcal{T}_{\eta_t\, \boldsymbol{u}_t}(\boldsymbol{u}_t)\rangle_{\mathbf{x}_{t+1}}\right| \leq c_2 \left|\, \langle -\boldsymbol{v}_t,\, \boldsymbol{u}_t\rangle_{\mathbf{x}_t}\,\right|, \tag{B8}$$

where $0 < c_1 < c_2 < 1$. In our numerical runs, we chose $c_1 = 0.485$ and $c_2 = 0.999$, and the subroutine OptStepSize in Alg. 1 can be seen as a standard backtracking algorithm [68] outputting a learning rate that satisfies the strong Wolfe conditions in Eqs. (B7)-(B8). We can modify this subroutine in problem-specific cases. We considered three heuristics: (*i*) search for a learning rate in a region given by the power method, *i.e.*, we start with

$$\eta_t^{PM} = c_3 \|\boldsymbol{u}_t\|^{-1} \arccos\left[\left(1 + \left(\frac{\|\boldsymbol{u}_t\|}{2\,\mathcal{L}(\theta_t)}\right)^2\right)^{-1/2}\right], \tag{B9}$$

and then backtrack until Eqs. (B7)-(B8) are satisfied, with $c_3 > 1$ being problem-dependent — for the spin chains, $c_3 = 1$ behaved well; (*ii*) in the case of molecules, a heuristic choice for the backtrack starting point that proved to work was $\max\left(\eta_t^{PM}, \frac{\pi}{4\|\boldsymbol{u}_t\|}\right)$; (*iii*) perform a *golden-section search* [69] on $\mathcal{L}\left(\boldsymbol{\theta}(\mathbf{x}_{t+1})\right)$ to guarantee Eq. (B7) and then backtrack until both conditions are satisfied. Although all the strategies work, the amount of resources necessary for each one may vary in a problem-dependent way, and we reported the results that yielded the best results with the minimum resources. A description of the entire procedure is presented in Alg. 1.

Appendix C: Efficient implementations of coordinate transformation and regularization

As mentioned in Sec. III, there are two steps required to estimate the natural gradient $\boldsymbol{v}_t$ in $\mathbf{x}$ coordinates. The first is the estimation of the Euclidean gradient $(\boldsymbol{\partial}_{\boldsymbol{\theta}}\mathcal{L})_{\boldsymbol{\theta}_t}$ obtained using quantum hardware measurements, while the second involves the computation of the product $\mathbf{J}(\boldsymbol{\theta}_t)\, \boldsymbol{g}^{-1}(\boldsymbol{\theta}_t)\, (\boldsymbol{\partial}_{\boldsymbol{\theta}}\mathcal{L})_{\boldsymbol{\theta}_t}$. We now calculate the cost of the two remaining products to numerically calculate $\boldsymbol{v}_t$. The cost to classically calculate the natural gradient $\boldsymbol{g}^{-1}(\boldsymbol{\theta}_t)\, (\boldsymbol{\partial}_{\boldsymbol{\theta}}\mathcal{L})_{\boldsymbol{\theta}_t}$ is of d floating-point multiplications. Since the metric $\boldsymbol{g}$ is a diagonal matrix in the coordinate basis, it can be element-wise inverted efficiently. Moreover, the aforementioned product has the same numerical cost as an element-wise multiplication between the two $(d-1)$-dimensional arrays $(\boldsymbol{\partial}_{\boldsymbol{\theta}}\mathcal{L})_{\boldsymbol{\theta}_t}$ and $\mathrm{diag}(\boldsymbol{g}^{-1})$.

We now estimate the numerical cost of applying the coordinate transformation using the Jacobian matrix $\mathbf{J}$. Under the assumption of $\mathbf{x} \in \mathbb{S}^{d-1}$, the Jacobian matrix for the hyperspherical coordinate transformation can be

straightforwardly calculated using its definition as

$$\mathbf{J}(\boldsymbol{\theta}) = \begin{pmatrix} \partial_{\theta_1}\cos(\theta_1) & \partial_{\theta_2}\cos(\theta_1) & \cdots & \partial_{\theta_{d-1}}\cos(\theta_1) \\ \partial_{\theta_1}\sin(\theta_1)\cos(\theta_2) & \partial_{\theta_2}\sin(\theta_1)\cos(\theta_2) & \cdots & \partial_{\theta_{d-1}}\sin(\theta_1)\cos(\theta_2) \\ \vdots & \vdots & \ddots & \vdots \\ \partial_{\theta_1}\prod_{\ell=1}^{d-2}\sin(\theta_\ell)\cos(\theta_{d-1}) & \partial_{\theta_2}\prod_{\ell=1}^{d-2}\sin(\theta_\ell)\cos(\theta_{d-1}) & \cdots & \partial_{\theta_{d-1}}\prod_{\ell=1}^{d-2}\sin(\theta_\ell)\cos(\theta_{d-1}) \\ \partial_{\theta_1}\prod_{\ell=1}^{d-1}\sin(\theta_\ell) & \partial_{\theta_2}\prod_{\ell=1}^{d-1}\sin(\theta_\ell) & \cdots & \partial_{\theta_{d-1}}\prod_{\ell=1}^{d-1}\sin(\theta_\ell) \end{pmatrix}$$

$$= \begin{pmatrix} -\sin(\theta_1) & 0 & \cdots & 0 \\ \cos(\theta_1)\cos(\theta_2) & -\sin(\theta_1)\sin(\theta_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \cos(\theta_1)\prod_{l=2}^{d-2}\sin(\theta_l)\cos(\theta_{d-1}) & \sin(\theta_1)\cos(\theta_2)\prod_{l=3}^{d-2}\sin(\theta_l)\cos(\theta_{d-1}) & \cdots & -\prod_{l=1}^{d-2}\sin(\theta_l)\sin(\theta_{d-1}) \\ \cos(\theta_1)\prod_{l=2}^{d-1}\sin(\theta_l) & \sin(\theta_1)\cos(\theta_2)\prod_{l=3}^{d-1}\sin(\theta_l) & \cdots & \prod_{l=1}^{d-2}\sin(\theta_l)\cos(\theta_{d-1}) \end{pmatrix}. \quad \text{(C1)}$$

Equation (C1) shows that $\mathbf{J}(\boldsymbol{\theta}) \in \mathbb{R}^{d\times d-1}$ is a *tall lower Hessenberg* matrix, *i.e.* a tall matrix that has nonzero elements only in the lower triangle and the first diagonal above the main diagonal (known as the first *superdiagonal*) [70, Chapter 2]. Then, the cost of the product between the Jacobian and the natural gradient array can be implemented efficiently by element-wise multiplication of the nonzero elements of each row of $\mathbf{J}$ with the corresponding element(s) of the natural gradient. Since the number of nonzero elements in $\mathbf{J}$ goes from 1 in the first row to $d-1$ in the last two rows, the cost of the matrix multiplication is $(d+2)(d-1)/2$ multiplications and $(d-2)(d+1)/2$ additions. Hence, the overall time complexity of numerically calculating $\boldsymbol{v}_t$ is $\mathcal{O}(d^2)$. Additionally, the nonzero elements of the jth Jacobian row can be reused in the calculation of the $(j+1)$th row. Therefore, the entire matrix product can be performed while only one Jacobian row is stored in memory at a time, giving a space complexity of $\mathcal{O}(d)$.

For the aforementioned procedure to work, it is necessary to guarantee the invertibility of the metric tensor $\boldsymbol{g}$ during training. Here, we describe how this can be efficiently guaranteed for the EGT optimizer. As stated in Sec. III, the metric of $\mathbb{S}^{d-1}$ is diagonal in the coordinate basis, with components $g_{11} = 1$ and $g_{jj} = \prod_{\ell=1}^{j-1}\sin^2(\theta_\ell)$ for $j \in [2, d-1]$. The eigenvalues of $\boldsymbol{g}$ are a function of $\boldsymbol{\theta}$ and, more importantly, each eigenvalue g_{jj} is associated with all angles $\{\theta_\ell\}_{\ell\in[j]}$. If any of the angles decrease in magnitude such that $\theta_\ell \ll 1$ during training, then all eigenvalues $g_{jj} \to 0\,, \forall\, j \geq \ell$. Such a metric then becomes degenerate, and all directions $\theta_j\,, \forall\, j \geq \ell$ become unexplored during training even under the use of pseudo-inversion. Instead of resorting to standard techniques, *e.g.* Tikhonov regularization, to solve this issue, we use the connection between the eigenvalues g_{jj} and angles θ_j to create the following heuristic rule: if $\sin(\theta_j) \approx \theta_j \lesssim \tau$, then $\theta_j \to \pi/2$. The constant τ is a threshold used to reset training in the jth direction and maintain the invertibility of $\boldsymbol{g}$. We empirically verified that $\tau = 10^{-3}$ yielded stable training for all numerical demonstrations presented in Sec. IV.

Bond length (Å)	Learning rate scheduling	Calls to loss (per epoch)	Calls to loss (total)	Gradient evaluations
0.964	Bayesian	25	669	27
	Bayesian	50	1198	25
	Strong Wolfe	≈ 18	461	24
1.860	Bayesian	25	5115	209
	Bayesian	50	5665	122
	Strong Wolfe	≈ 17	1245	70

Table I. Resource comparison for OH^- at two bond lengths — degenerate and non-degenerate cases. The total counts reflect values for 15 consecutive epochs after chemical accuracy was achieved. In all cases, only one gradient evaluation per epoch was performed (*i.e.*, the second Wolfe condition was always satisfied when the first was). Here, "calls to loss" refers to the loss evaluations necessary to test the strong Wolfe conditions or the queries to perform the Bayesian trials only, *i.e.*, it does not include the loss calls necessary for gradient estimation.

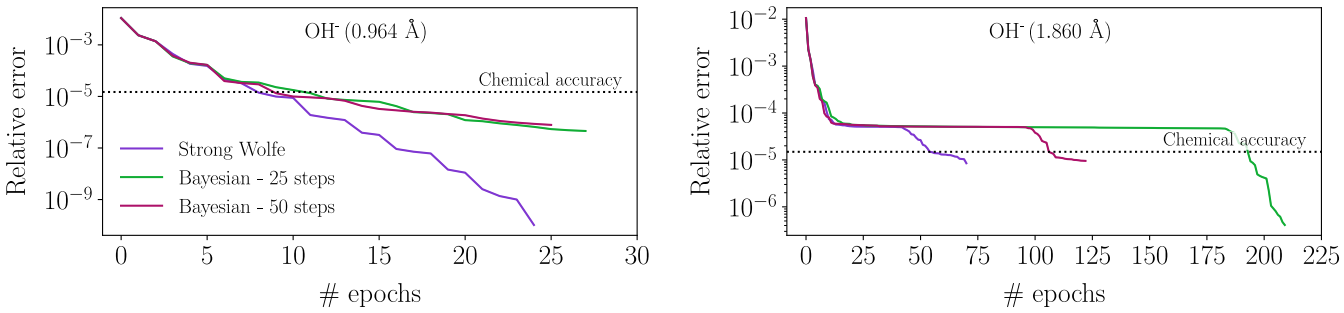

Figure 5. **Relaxation of the global convergence guarantees.** Relative ground-state energy estimation error for the OH^- molecule as a function of the number of epochs, at two different bond lengths. Using the Jordan-Wigner transformation and the STO-3G basis set, the ground state is represented by $n = 12$ spin-orbitals as the active space with $k = 10$ electrons. Quantum circuit ansatz is the HWE_k. The optimizer is the EGT-CG with learning-rate scheduling based on the strong Wolfe conditions (purple) and Bayesian optimization with 25 (green) or 50 (red) steps. Optimization is halted 15 epochs after chemical accuracy is reached. (*left*) Energy estimation at bond length of 0.964Å. In this case (bond length close to equilibrium), the ground state is not degenerate, and the spectral gap is 0.49% of the ground-state energy. The three schedulings perform well, with the strong-Wolfe scheduling slightly outperforming both Bayesian approaches. (*right*) Energy estimation at bond length of 1.860Å. In this case, the ground-state has degeneracy 6, and the spectral gap is 0.005% of the ground-state energy. In this scenario, the strong-Wolfe scheduling reaches chemical accuracy almost 3 times faster than both schedulings that use Bayesian optimization.

Appendix D: Global convergence guarantee *vs.* heuristic method

In this appendix, we analyze the numerical performance of the EGT-CG defined in Eqs. (2) - (4) when the learning rate(s) η_t and memory scaling factor β_{t+1} are chosen using (*i*) the strong Wolfe conditions described in App. B; or (*ii*) Bayesian optimization techniques [33]. We tested the Bayesian approach with 25 and 50 calls to the loss function to determine the learning rate η_t per epoch t.

In Fig. 5(*left*), we show the relative energy-estimation error as a function of the number of epochs in the training procedure. The OH^- molecule is at equilibrium bond length (0.964Å) and the spectral gap in this case is $\sim 0.49\%$ of the ground energy. We see that the EGT-CG optimizer reaches chemical accuracy around the same number of epochs (~ 10) for all three methodologies, even though the training method using the strong Wolfe conditions reaches relative energy errors more than 3 orders of magnitude smaller by the end of training. We also highlight that for this non-degenerate ground state, the strong Wolfe conditions perform better both in terms of calls to loss per epoch as well as calls to loss overall, since it reaches better precision in the same number of epochs. The number of calls to the gradient function is similar in all three methods. The numbers are displayed in Table I.

In Fig. 5(*right*), we show the relative energy-estimation error *vs.* the number of epochs for the same molecule but a different bond length, 1.86Å. At this bond length, the spectral gap is only 0.005% of the ground-state energy. The proximity between ground and first-excited states makes training more difficult. We see that all methods display a plateau-like behavior after approximately 10 epochs. However, the method that uses the strong Wolfe conditions significantly improves the relative error before the 50th epoch. Meanwhile, the Bayesian optimization scheme with 50 calls to the loss function to estimate the best η_t takes 2x more epochs to start training significantly. The Bayesian optimization with 25 calls to the loss function takes approximately 3.5x more epochs than the strong Wolfe conditions. In Table I, we also see that this increases the number of calls to the loss function for the Bayesian methods by a factor bigger than 4, while the strong Wolfe conditions require less than 3x more calls to the loss function.

Appendix E: Estimating gradients with HWE_k ansatz without using the parameter-shift rule

In this appendix, we derive an explicit expression for the gradient of the loss $\mathcal{L}_\psi(\boldsymbol{\theta}) = \langle\psi(\boldsymbol{\theta})| H |\psi(\boldsymbol{\theta})\rangle$ on the state prepared by the ansatze used in this manuscript, presenting an improvement in terms of quantum resources when compared to the standard parameter-shift rule (PSR). For this appendix, we introduce the subscript ψ to $\mathcal{L}$ indicating that the loss function is calculated w.r.t. state $|\psi(\boldsymbol{\theta})\rangle$. This distinction will become relevant later.

Given n qubits, the ansatze prepares quantum states with support on d-dimensional (sub)spaces. The ansatz explores the effective Hilbert space using $M = d - 1$ angles represented by the vector $\boldsymbol{\theta} \in \mathbb{R}^{d-1}$. We can write the

prepared state $|\psi(\boldsymbol{\theta})\rangle$ as

$$|\psi(\boldsymbol{\theta})\rangle = \cos(\theta_1)\ |b_1\rangle + \sum_{j=2}^{d-1}\left(\prod_{m=1}^{j-1}\sin(\theta_m)\right)\cos(\theta_j)\ |b_j\rangle + \left(\prod_{j=1}^{d-1}\sin(\theta_j)\right)|b_d\rangle\ . \tag{E1}$$

The ℓth component of the gradient of the state in Eq. (E1) is then given by

$$|\partial_{\theta_\ell}\psi(\boldsymbol{\theta})\rangle = \delta_{\ell 1}\cos\Big(\theta_\ell+\frac{\pi}{2}\Big)|b_1\rangle + \sum_{j=2}^{d-1}\omega_{\ell j}\prod_{m=1}^{j-1}\sin\Big(\theta_m+\delta_{\ell m}\frac{\pi}{2}\Big)\cos\Big(\theta_j+\delta_{\ell j}\frac{\pi}{2}\Big)|b_j\rangle + \prod_{j=1}^{d-1}\sin\Big(\theta_j+\delta_{\ell j}\frac{\pi}{2}\Big)|b_d\rangle\,, \tag{E2}$$

where δ_{lj} is the Kronecker delta and $\omega_{\ell j}$ is the discrete step function $\omega_{\ell j}=1$ if $\ell\leq j$, and zero otherwise. From the definition of the metric $\boldsymbol{g}$ for $\mathbb{S}^{d-1}$ in Sec. III, we see that $|\varphi_\ell(\boldsymbol{\theta})\rangle \coloneqq g_{\ell\ell}^{-1/2}(\boldsymbol{\theta})\ |\partial_{\theta_\ell}\psi(\boldsymbol{\theta})\rangle$ is a well-normalized quantum state $\forall\,\ell\in[d-1]$, with nonzero jth components in $j\geq\ell$. Defining the quantum state $|\phi_\ell(\boldsymbol{\theta})\rangle \coloneqq \left(|\psi(\boldsymbol{\theta})\rangle + |\varphi_\ell(\boldsymbol{\theta})\rangle\right)/\sqrt{2}$, we obtain

$$(\partial_{\theta_l}\mathcal{L})_{\boldsymbol{\theta}_t} = g_{\ell\ell}^{1/2}\ \left(2\,\mathcal{L}_{\phi_\ell}(\boldsymbol{\theta}_t) - \mathcal{L}_{\varphi_\ell}(\boldsymbol{\theta}_t) - \mathcal{L}_{\psi}(\boldsymbol{\theta}_t)\right)\ , \tag{E3}$$

where, as introduced at the beginning of this appendix, the terms $\mathcal{L}_{\phi_\ell}$, $\mathcal{L}_{\varphi_\ell}$, and $\mathcal{L}_\psi$ indicate the loss function calculated w.r.t. the quantum states $|\phi_\ell(\boldsymbol{\theta}_t)\rangle$, $|\varphi_\ell(\boldsymbol{\theta}_t)\rangle$, and $|\psi(\boldsymbol{\theta}_t)\rangle$. Despite the similarity with PSR due to the parameters θ_ℓ being shifted by $\frac{\pi}{2}$, Eq. E3 is not a proper PSR since it lacks opposing, negative shifts, *i.e.* shifts by $-\frac{\pi}{2}$.

Now, we follow Ref. [71] in the way we analyze the time complexity of estimating $(\boldsymbol{\partial}_\theta\mathcal{L})_{\boldsymbol{\theta}_t}$ using quantum hardware measurements. There, the time complexity to execute the loss function once is denoted as TIME($\mathcal{L}$). Meanwhile, Refs. [72, 73] derived parameter-shift rules for quantum circuits composed of M Givens rotations and showed that the time complexity of estimating the entire gradient vector is TIME(PSR) $= 4\,M$ TIME($\mathcal{L}$). With that in mind, we now analyze the time complexity for Eq. (E3). Since the last term on the right-hand side (r.h.s.) of Eq. (E3), $\mathcal{L}_\psi$, is constant, it can be estimated only once and therefore has a time complexity of TIME($\mathcal{L}$) independent of ℓ. The first term on the r.h.s. of Eq. (E3) is the most interesting of the three. As mentioned above, the state $|\varphi_\ell(\boldsymbol{\theta})\rangle$ only has nonzero amplitudes in the components $j\geq\ell$. This means that this state does not require all M parameters to be prepared, instead requiring $d-\ell$ parametrized gates. Hence, if the number of parametrized rotations needed to estimate each $\mathcal{L}_{\varphi_\ell}$ is m, then the complexity of estimating all ℓ elements is $\sum_{m=1}^{M} m = \frac{M+1}{2}$ TIME($\mathcal{L}$). The first term on the r.h.s. of Eq. (E3) depends on the preparation of a superposition between $|\psi(\boldsymbol{\theta})\rangle$ and $|\varphi_\ell(\boldsymbol{\theta})\rangle$. Even though $|\varphi_l(\boldsymbol{\theta})\rangle$ can be, in general, individually prepared with a reduced number of parametrized gates, the same cannot be said about $|\psi(\boldsymbol{\theta})\rangle$. Therefore, the time complexity of estimating all components $\mathcal{L}_{\phi_l}$ is M TIME($\mathcal{L}$). Finally, the sum of the three components gives us the time complexity of estimating the full gradient vector, *i.e.* TIME($(\boldsymbol{\partial}_{\boldsymbol{\theta}}\mathcal{L})_{\boldsymbol{\theta}_t}$) $= \frac{3\,(M+1)}{2}$ TIME($\mathcal{L}$). For sufficiently large M, we have

$$\text{TIME}((\boldsymbol{\partial}_{\boldsymbol{\theta}}\mathcal{L})_{\boldsymbol{\theta}_t}) \approx \frac{3}{8}\ \text{TIME(PSR)}\,. \tag{E4}$$

Equation (E4) demonstrates that estimating gradient vectors using Eq. (E3) instead of PSR provides a 62.5% reduction in time complexity and quantum resources used. We also highlight that even though significant, the resource reduction presented in Eq. (E4) is constant and therefore it is still in line with the complexity results obtained in Ref. [71].

Appendix F: Barren-plateau analysis

In this appendix, we show that the amplitude encoder in hyperspherical coordinates allows exact computation of the variance of a loss function of the type $\mathcal{L} = \langle\psi|\,H\,|\psi\rangle$. The result is summarized in the following theorem:

Theorem F.1. Let H be a real-valued Hamiltonian, $|\psi\rangle$ an n-qubit quantum state amplitude-encoded in hyperspherical coordinates and with support in a d-dimensional (sub)space, and $\mathcal{L} = \langle\psi|\ H\ |\psi\rangle$ a loss function. Denote by $\boldsymbol{v}$ the Riemannian gradient of $\mathcal{L}$, and by v_j its jth component. Then,

$$\text{Var}[\mathcal{L}] = \frac{d\ \text{tr}(H^2) - \text{tr}(H)^2}{d^2\,(d+2)/2}\,, \tag{F1}$$

and

$$\mathrm{Var}[v_j] = \frac{d\,(d+2)\,\|h_j\|_2^2 - 2\,(d+2)\,H_{jj}\;\mathrm{tr}(H) + \mathrm{tr}(H)^2 + 2\;\mathrm{tr}(H^2)}{d\,(d+2)\,(d+4)/4}\,, \tag{F2}$$

where $\|\cdot\|_2$ is the Euclidean norm.

Proof. In terms of the amplitudes $\mathbf{x} \in \mathbb{S}^{d-1}$, the loss function takes the quadratic form $\mathcal{L}(\mathbf{x}) = \mathbf{x}^T H\,\mathbf{x}$ and the Riemannian gradient is $\boldsymbol{v} = \Pi_{\mathbf{x}}\,\boldsymbol{\partial}_{\mathbf{x}}\mathcal{L} = 2\,(\mathbb{1} - \mathbf{x}\mathbf{x}^T)\,H\,\mathbf{x}$, where $\boldsymbol{\partial}_{\mathbf{x}}\mathcal{L} = 2\,H\,\mathbf{x}$ is the Euclidean gradient and $\Pi_{\mathbf{x}} \coloneqq \mathbb{1} - \mathbf{x}\mathbf{x}^T$ is the orthogonal projector onto the tangent space to the sphere at $\mathbf{x}$. The variance of $\mathcal{L}$ and v_j, are, respectively, $\mathrm{Var}[\mathcal{L}] = \mathbb{E}[\mathcal{L}^2] - \mathbb{E}[\mathcal{L}]^2$ and $\mathrm{Var}[v_j] = \mathbb{E}[v_j^2] - \mathbb{E}[v_j]^2$. The expectation values $\mathbb{E}[f(\mathbf{x})] \coloneqq \int_{\mathbb{S}^{d-1}} \mathrm{d}\mu(\mathbf{x})\,f(\mathbf{x})$ are w.r.t. the uniform measure $\mathrm{d}\mu(x)$ on $\mathbb{S}^{d-1}$ [64]. All the expectation values of interest can be expressed as linear combinations of moments of the random variables x_j (*e.g.*, see Ref. [74]). All *odd* moments of $\mathbf{x}$ vanish, *i.e.*

$$\mathbb{E}[x_{i_1} x_{i_2} \cdots x_{i_\ell}] = 0 \quad \forall\, \ell \text{ odd}\,, \tag{F3}$$

while all the even moments have a closed-form expression. In particular,

$$\begin{aligned}
\mathbb{E}[x_i\,x_j] &= \frac{\delta_{ij}}{d} \\
\mathbb{E}[x_i\,x_j\,x_k\,x_\ell] &= \frac{\delta_{ij}\,\delta_{k\ell} + \delta_{ik}\,\delta_{j\ell} + \delta_{i\ell}\,\delta_{jk}}{d\,(d+2)} \\
\mathbb{E}[x_i\,x_j\,x_k\,x_\ell\,x_p\,x_q] &= \frac{\delta_{ij}\,\delta_{k\ell}\,\delta_{pq} + \text{permutations}}{d\,(d+2)\,(d+4)}\,,
\end{aligned} \tag{F4}$$

where δ_{ij} is the Kronecker symbol and the permutations in the last line denote 14 similar terms corresponding to all the ways of partitioning the indices (i, j, k, ℓ, p, q) into three unordered and disjoint pairs of indices. The calculation of $\mathrm{Var}[\mathcal{L}]$ is then straightforward:

$$\mathrm{Var}[\mathcal{L}] = \sum_{i,j,k,\ell} H_{ij}\,H_{k\ell}\,\mathbb{E}[x_i\,x_j\,x_k\,x_\ell] - \left(\sum_{ij} H_{ij}\,\mathbb{E}[x_i\,x_j]\right)^2 = \frac{2\;\mathrm{tr}(H^2) + \mathrm{tr}(H)^2}{d\,(d+2)} - \frac{\mathrm{tr}(H)^2}{d^2}\,, \tag{F5}$$

from which Eq. (F1) follows.

For the j-th component of the Riemannian gradient, v_j, we first notice that $\mathbb{E}\big[\nabla_j\mathcal{L}\big] = 0$ follows immediately from (F3), while $\mathbb{E}\big[(\nabla_j\mathcal{L})^2\big]$ can be computed using (F4) after noticing that $\frac{1}{4}v_j^2 = (H\mathbf{x})_j^2 - 2(H\mathbf{x})_j\,x_j\,(\mathbf{x}^T H\mathbf{x}) + x_j^2\,(\mathbf{x}^T H\mathbf{x})^2$. Namely, the first term is

$$\mathbb{E}\big[(H\mathbf{x})_j^2\big] = \sum_{ik} H_{ji}\,H_{jk}\,\mathbb{E}[x_i\,x_k] = \frac{1}{d}\sum_i H_{ji}^2 = \frac{\|h_j\|_2^2}{d}, \tag{F6}$$

where h_j is the j-th row of H. The second term is

$$\mathbb{E}\big[(H\mathbf{x})_j\,x_j\,(\mathbf{x}^T H\mathbf{x})\big] = \sum_{i,k,\ell} H_{ji}\,H_{k\ell}\,\mathbb{E}[x_j\,x_i\,x_k\,x_\ell] = \frac{H_{jj}\;\mathrm{tr}(H) + 2\,\|h_j\|_2^2}{d\,(d+2)}, \tag{F7}$$

and the last term reads

$$\mathbb{E}[x_j^2(\mathbf{x}^T H\mathbf{x})^2] = \sum_{i,k,p,q} H_{ik}\,H_{pq}\,\mathbb{E}\big[x_j\,x_j\,x_i\,x_k\,x_p\,x_q\big] = \frac{1}{d\,(d+2)\,(d+4)}\sum_{i,k,p,q} H_{ik}\,H_{pq}\big(\delta_{ik}\,\delta_{k\ell}\,\delta_{pq} + \text{permutations}\big).$$

There are 15 different pairings of the indices (j_1, j_2, i, k, p, q) into triples of unordered disjoint pairs (here we use subscripts j_1, j_2 just distinguish the two occurrences of the same symbol j for counting purposes). Of these, 3 have j_1, j_2 paired together: one of the type $(j_1,\, j_2)(i,\, k)(p,\, q)$, leading to $\sum_{i,k,p,q} H_{ik}\,H_{pq}\,\delta_{ik}\,\delta_{pq} = \mathrm{tr}(H)^2$, and 2 of the type $(j_1, j_2)\,(i, p)\,(k, q)$, which give $\sum_{i,k,p,q} H_{ik}\,H_{pq}\,\delta_{ip}\,\delta_{kq} = \sum_{i,k} H_{ik}\,H_{ik} = \mathrm{tr}(H^2)$; 4 pairings have each j paired with a different index and the leftovers pair together, *e.g.* $(j_1, i)\,(j_2,\, k)(p,\, q)$, which yield $\sum_{i,k,p,q} H_{ik}\,H_{pq}\,\delta_{ji}\,\delta_{jk}\,\delta_{pq} =$

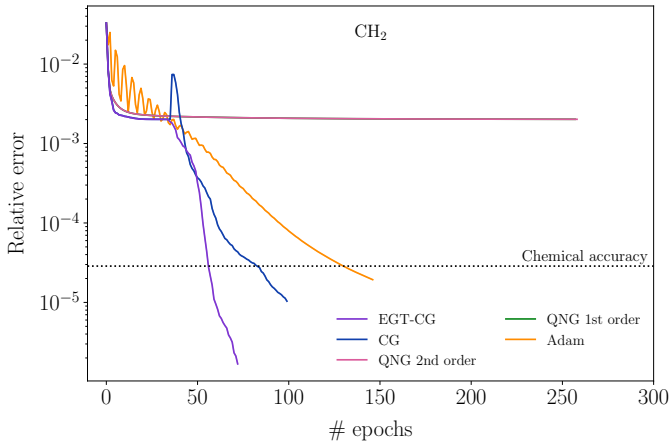

Figure 6. Relative ground-state energy estimation error for the CH_2 molecule as a function of the number of epochs. Using the Jordan-Wigner transformation and the STO-3G basis set, the ground state is represented by $n = 14$ spin-orbitals as the active space with $k = 8$ electrons. The same warm start $|\psi_{\text{warm}}\rangle$ ($\alpha = 0.9$) was used, despite it not having a large overlap with the true degenerate ground-space. The ground-state has degeneracy 3, and a spectral gap of 0.2% of the ground-state energy, which favors most optimizer schemes to plateau at the first excited-state energy, whilst the EGT-CG escapes the plateau and converges to chemical accuracy monotonically, given the guarantees of the Strong Wolfe conditions. Despite the non-monotonic path, Adam ($\eta = 0.01$) escapes the plateau and manages to achieve chemical accuracy.

$H_{jj}\ \mathrm{tr}(H)$; and, finally, 8 pairings have each j paired with a different index with the left-over pair linking the two H factors, *e.g.* $(j_1,\, i)\,(j_2,\, p)\,(k,\, q)$, which give $\sum_{i,k,p,q} H_{ik}\, H_{pq}\, \delta_{ji}\, \delta_{jp}\, \delta_{kq} = \sum_k H_{jk}\, H_{jk} = \left\|h_j\right\|_2^2$. It follows that

$$\mathbb{E}[x_j^2\, (\mathbf{x}^T H \mathbf{x})^2] = \frac{\mathrm{tr}(H)^2 + 2\ \mathrm{tr}(H^2) + 4\, H_{jj}\ \mathrm{tr}(H) + 8\left\|h_j\right\|_2^2}{d\,(d+2)\,(d+4)}\,, \tag{F8}$$

and putting all together leads to $\mathrm{Var}[v_j]$ given by Eq. (F2). □

Appendix G: CH_2 optimization curves

As discussed in Sec. IV A, the proposed warm start (Eq. (5)) is not the most adequate choice for CH_2 — indeed, as one can verify via the brute-force diagonalization, the Hartree state $|1^8 0^6\rangle$ is not the major contribution in none of its three degenerate ground-states —, as seen in Fig. 2(d), the overlap of $|\psi_{\text{warm}}\rangle$ with the degenerate ground-states subspace is of $\sim 10^{-5}$. Despite this, as can be seen in Fig. 6, the conjugate methods as well as Adam with a relatively large learning rate were able to achieve chemical accuracy, although even EGT-CG took more epochs than usual when compared to the other molecules (see Fig. 2(c)). This indicates that using the simple proposed warm start may not be the optimal choice for all molecules. However, we stress that if there is any other initialization inspired by domain knowledge, preparing it with HWE_k remains trivial, whilst impractical for other ansatze.

As observed for other molecules as well, the optimization curve of Adam is non-monotonic, exhibiting an oscillating behavior which, although may be an indication of an exceedingly large initial learning rate that could lead to divergence, has been shown to lead to faster convergence to chemical accuracy in all the performed simulations — here, for CH_2, $\eta = 0.01$, and the initial oscillatory behavior was essential for the optimizer to scape the plateau in the first excited states, which the QNG methods were not able to do. Also, we note that the CG optimization scheme, despite achieving chemical accuracy, follows a non-monotonic path, while the loss for EGT-CG is monotonously decreasing.

Appendix H: Quantum resource estimation

Here, we present an estimation of the quantum resources necessary to run the ground-state estimation demonstrations described in Sec. IV B. We focus on the number of calls to the loss function required to determine the learning rate η_t per epoch t per optimization scheme (Table II). We also provide explicit CNOT and parameter counts in each one of the circuit ansatze considered in the numerical demonstrations (Table III).

Number of qubits (n)	HWE (EGT-CG)	HWE (CG)	HWE (QNG 1st)	HWE (QNG 2nd)
	Calls to loss (average per epoch)			
4	6.4	6.4	50.0	50.0
6	5.5	5.5	50.0	50.0
8	6.1	5.5	50.0	50.0
10	5.4	4.4	50.0	50.0
12	5.2	3.4	50.0	50.0
14	5.0	3.0	50.0	50.0
16	4.2	3.1	50.0	50.0

Table II. Resource comparison for XXZ ground state estimation, in terms of calls to the loss in each epoch necessary to determine the learning rate of each optimization scheme: The satisfaction of the strong Wolfe conditions for EGT-CG and CG; per-epoch greedy Bayesian optimization for QNG 1st and 2nd; constant initial learning rate for Adam (which does not require any additional calls to the loss). The values are averages over 50 different Haar-random initializations. For the CG methods, the testing of Wolfe conditions requires a variable number of calls to loss per epoch, so the reported number is the average number of calls throughout the optimization.

Number	**# of CNOTs in ansatz**				**# of parameters in ansatz**			
of qubits (n)	HWE	QOL	HE OP	HE	HWE	QOL	HE OP	HE
4	2	10	28	12	1	5	30	15
6	8	38	126	60	2	19	126	63
8	102	138	504	248	7	69	510	255
10	390	502	2040	1020	15	251	2046	1023
12	3230	1846	8184	4092	49	923	8190	4095
14	11372	6862	32760	16380	132	3431	32766	16383
16	45738	25738	131056	65520	439	12869	131070	65535

Table III. Number of CNOTs and parameters in each ansatz used for simulations of the XXZ ground state. Although we did not run experiments for ansatze other than HWE for $n > 10$, we plot the CNOT counts for these circuits for a scaling comparison. Numerical details can be found in Tables II and III in App. H.