Optimizing quantum circuits with Riemannian gradient flow

Roeland Wiersema[1,2,3] and Nathan Killoran[1]
[1]*Xanadu, Toronto, Ontario, Canada M5G 2C8*
[2]*Vector Institute, MaRS Centre, Toronto, Ontario, Canada M5G 1M1*
[3]*Department of Physics and Astronomy, University of Waterloo, Ontario, Canada N2L 3G1*

(Received 2 November 2022; revised 20 April 2023; accepted 7 June 2023; published 27 June 2023)

Variational quantum algorithms are a promising class of algorithms that can be performed on currently available quantum computers. In most settings, the free parameters of a variational circuit are optimized using a classical optimizer that updates parameters in Euclidean geometry. Since quantum circuits are elements of the special unitary group, we can consider an alternative optimization perspective that depends on the structure of this group. In this work, we investigate a Riemannian optimization scheme over the special unitary group and we discuss its implementation on a quantum computer. We illustrate that the resulting Riemannian gradient-flow algorithm has favorable optimization properties for deep circuits and that an approximate version of this algorithm can be performed on near-term hardware. We highlight the connections of our work with previously proposed heuristics like ADAPT-VQE and show that they can be understood as variants of our algorithm.

DOI: 10.1103/PhysRevA.107.062421

I. INTRODUCTION

With quantum computing hardware still in its infancy, variational quantum algorithms offer a way to probe the power of noisy intermediate-scale quantum (NISQ) devices [1,2]. In a typical setup, one calculates gradients with respect to gate parameters in a quantum circuit to minimize a cost function that depends on the variational state. Since these approaches often involve minimizing nonconvex cost functions, the choice of optimizer can greatly affect the result [3]. Unlike in deep learning, where back propagation can remain effective despite a large number of parameters, calculating gradients in a variational quantum circuit quickly becomes inefficient. This is due to the fact that the gradients for single parameters cannot be calculated concurrently, but require additional circuit evaluations for each parameter [4,5].

Gradient-based methods can be improved by considering additional structure of the model under consideration. For instance, when dealing with a statistical model, one can make use of the Fisher information to quantify the statistical distance between probability distributions [6]. This induces a metric on parameter space, which provides the direction of steepest descent with respect to the information geometry [7]. The resulting gradient is called the natural gradient, and forms the basis of a second-order optimization algorithm called natural gradient descent. This method is used in deep learning [8,9] but can also be extended to variational quantum Monte Carlo [10] and variational quantum circuits, where the distance between rays in Hilbert space provides an analog of the Fisher information [11,12].

Optimization algorithms that rely on the Fisher information fall into the category of Riemannian optimization algorithms [13,14]. However, they are limited to optimizing over a real parameter space $\mathbb{R}^n$ with a non-Euclidean metric. Riemannian optimization has a much broader application: We can consider minimizing a function over a differentiable manifold $\mathcal{M}$ equipped with a nondegenerate, positive metric. This construction is more general, and allows one to take the structure of the manifold into account during the optimization. Such applications have been considered in the context of quantum control [15–20], tensor networks [21,22], or optimization of neural networks [23–27]. In the quantum circuit setting, the Riemannian manifold perspective has been considered to study the computational complexity of constructing specific circuits by approximating geodesics on the unitary group [28,29].

In this work, we introduce the optimization of quantum circuits over the special unitary group $\mathrm{SU}(p)$ using Riemannian gradient flows [15]. We show the resulting algorithm can produce quantum circuits with favorable optimization properties but which may be exponentially deep. To obtain a practically feasible circuit optimizer, we make approximations that keep gate costs under control. We explore several toy problems to illustrate the properties of the resulting exact and approximate Riemannian gradient flow.

In Sec. II, we introduce the necessary theory of gradient flows on the special unitary group. Then in Sec. III, we show how these flows can be adapted to the variational quantum circuit setting. To make these algorithms practical, we have to consider approximations in Sec. IV, for which we present numerical results for some toy models in Sec. V. In addition, we argue how some of the literature on adaptive variational approaches can be re-contextualized from the Riemannian optimization point of view. In Sec. VI we summarize our results.

II. BACKGROUND

A. Gradient flows in quantum circuits

An archetypal example of a widely used gradient flow in quantum computing is the variational quantum eigensolver

(VQE) [1]. Consider the cost function $L : \mathbb{R}^n \to \mathbb{R}$,

$$L(\theta) = \mathrm{Tr}\{HU(\theta)\rho_0 U(\theta)^\dagger\} \equiv \langle H\rangle_\theta, \tag{1}$$

where $U(\theta)$ is a parametrized quantum circuit, H a Hamiltonian whose ground state we want to approximate, and $\theta \in [0, 2\pi)^n$ is a vector of gate parameters. Here, $\rho_0 = |\psi_0\rangle\langle\psi_0|$ is some initial state of the system, usually taken to be the zero state $|0\rangle\langle 0|$. We are interested in minimizing $L(\theta)$ with respect to the parameters θ.

To solve the optimization problem $\min_\theta L(\theta)$, we can consider the flow

$$\dot{\theta} = \nabla_\theta L(\theta), \tag{2}$$

where $\nabla_\theta = \sum_i^n \partial_{\theta_i}$ is the standard gradient operator. Equation (2) provides a differential equation for the evolution of the parameters based on the gradient of the function at a point θ. This flow equation can be discretized as

$$\theta_{k+1} = \theta_k - \epsilon \nabla_\theta L(\theta), \tag{3}$$

where ϵ is the step size that controls the precision of the discretization. Using this equation to update the parameters of $L(\theta)$ is called steepest descent since we follow the gradient of the function to a minimum.

To understand why this works, we can look at the level curves of L, i.e., curves through parameter space where the function L is constant. We can define such a curve as $\gamma : (-a, a) \to \mathbb{R}^n$ with $\gamma(0) = \theta$ such that $L(\gamma(t)) = \text{const}$. Differentiating with respect to t then gives

$$\sum_i \partial_{\theta_i} L(\gamma(t))\gamma_i'(t)\Big|_{t=0} = 0, \tag{4}$$

which we identify as $\nabla L \cdot v$, the gradient of L in the direction of $v = \gamma'(t)$. In other words, the gradient of a function produces a vector orthogonal to the level curves through a point. As a result, infinitesimal steps in the direction of the gradient will decrease the function's value until we reach a local minimum [30].

One issue with VQE is that the parametrization of the variational circuit $U(\theta)$ is an arbitrary choice that we have to make. This implies that one must try different *Ansätze* and assume that the state of interest can be expressed with the chosen *Ansatz*. Moreover, due to the nonconvexity of the cost landscape, we have no guarantees that our optimizer can find a good approximation to the desired state.

B. Gradient flows on Lie groups

An N-qubit quantum circuit U is a unitary operation that is an element of the special unitary group $\mathrm{SU}(p)$ where $p = 2^N$. Hence, instead of minimizing the cost Eq. (1) over $\mathbb{R}^n$ for a particular parametrization $U(\theta)$, we can instead directly optimize over $\mathrm{SU}(p)$.

For such a construction to make sense, we need to introduce a gradient on $\mathrm{SU}(p)$. Since $\mathrm{SU}(p)$ is a finite-dimensional Lie group, it carries a differentiable manifold structure. We thus need to use the language of differential geometry to define a gradient on the group [31–34]. In particular, a p-dimensional manifold $\mathcal{M}$ is a set that locally looks like $\mathbb{R}^p$. This local description is given by charts, which smoothly map open subsets of the manifold onto coordinate patches in $\mathbb{R}^p$. If all charts between two subsets of the manifold are compatible, the manifold is differentiable (see Appendix A 1).

The tangent space $T_U\mathrm{SU}(p)$ of the manifold at a point U is a vector space that consists of a collection of vectors $\Omega \in T_U\mathrm{SU}(p)$ that provide the possible directions one can move in on the manifold from point U. The tangent vectors Ω can be defined as derivatives of curves going through the point U (see Appendix A 2). For example, on a sphere, the tangent space at a point p consists of a plane tangent to p.

The introduction of an inner product on the tangent space turns the manifold into a Riemannian manifold, with well-defined notions of angles and distance (see Appendix A 3). Given this metric, the Riemannian gradient $\mathrm{grad}\, L(U)$ of a function $L : \mathrm{SU}(p) \to \mathbb{R}$ at U can be constructed by satisfying two conditions:

(i) The Riemannian gradient $\mathrm{grad}\, L(U)$ at a point U must be an element of the tangent space $T_U\mathrm{SU}(p)$. This ensures that it is always tangential to the manifold at each point, hence $\mathrm{grad}\, L(U) \in T_U\mathrm{SU}(p)$.

(ii) Because there are many different ways to set coordinates on a manifold, and because the function itself should be invariant under a change of coordinates (i.e., its level curves are at the same locations on the manifold), we need to enforce a coordinate-invariant notion of a gradient. This can be achieved with the compatibility condition,

$$\langle \mathrm{grad}\, L(U), \Omega\rangle = \mathrm{Tr}\{\nabla L(U)\Omega\} \tag{5}$$

which expresses the fact that the inner product (under some chosen metric) of the Riemannian gradient with any other tangent vector Ω is independent of the choice of metric (see Fig. 2) [18,23]. Here, we take the reference inner product on the right-hand side to be the Hilbert-Schmidt inner product in the local coordinates $\mathbb{R}^p$.

With the two conditions for the Riemannian gradient given above, we can explicitly construct $\mathrm{grad}\, L(U)$. First, we rewrite Eq. (1) as a scalar function on the special unitary group $L : \mathrm{SU}(p) \to \mathbb{R}$ to obtain

$$L(U) = \mathrm{Tr}\{HU\rho_0 U^\dagger\}, \tag{6}$$

where $U \in \mathrm{SU}(p)$. To solve the optimization problem $\min_U L(U)$, we can consider the Riemannian gradient flow

$$\dot{U} = \mathrm{grad}\, L(U). \tag{7}$$

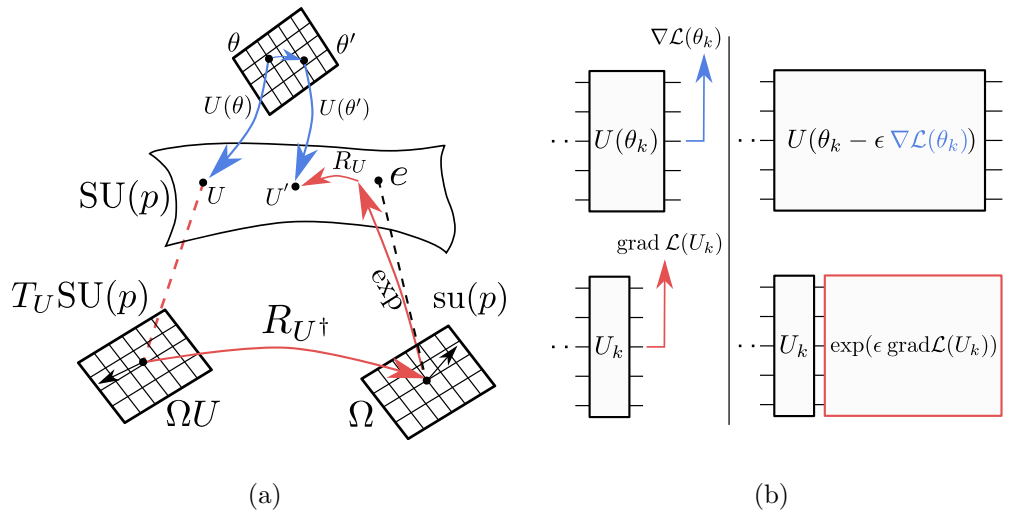
Next, we realize that the tangent space of $\mathrm{SU}(p)$ at the identity element $X_0 = \mathbb{I}$ is given by the Lie algebra $\mathrm{su}(p)$, the set of $p \times p$ skew-Hermitian matrices Ω with $\mathrm{Tr}\{\Omega\} = 0$. The elements of $T_U\mathrm{SU}(p)$ can then be found by right multiplying an element of the Lie algebra with U (see Appendix B):

$$T_U\mathrm{SU}(p) \coloneqq \{\Omega U | \Omega \in \mathrm{su}(p)\}. \tag{8}$$

With the notion of a tangent vector on $\mathrm{SU}(p)$, we can enforce the compatibility condition (see Appendix C 1) and derive the resulting Riemannian gradient flow on $\mathrm{SU}(p)$:

$$\dot{U} = \mathrm{grad}\, L(U) = [U\rho_0 U^\dagger, H]U. \tag{9}$$

Analogous to the gradient in $\mathbb{R}^n$, the Riemannian gradient flow of Eq. (9) converges to a critical point of $L(U)$ on $\mathrm{SU}(p)$ by descending along the level curves of the function [16]. To numerically compute the flow, we need to discretize the gradient steps.



(a) (b)

FIG. 1. Difference between the Riemannian gradient flow and Euclidean gradient flow. (a) At the top, we have a mapping from real parameters $\theta \in \mathbb{R}^n$ to a unitary in $U \in \mathrm{SU}(p)$. Gradient updates in the parameter space from θ to θ' result in a new unitary U' on the group. Starting instead at U, we first obtain the Riemannian gradient at U in the tangent space $T_U\mathrm{SU}(p)$. Since the Riemannian gradient can be written as ΩU with $\Omega \in \mathrm{su}(p)$, we can move to the Lie algebra $\mathrm{su}(p)$ by multiplying the Riemannian gradient with $U^\dagger$ from the right. Then, the exponential map and subsequent right multiplication with U projects the Riemannian gradient back onto the manifold which results in a new unitary U'. (b) At the top, we see how a standard gradient flow optimizes a quantum circuit: The circuit stays fixed and the gradient is calculated via the parameter-shift rule. Next, the free parameters describing the unitary are updated via gradient descent. In the bottom figure, we see that a step of the Riemannian optimization corresponds to appending a new unitary to the original circuit.

The commutator $[U\rho_0 U^\dagger, H]$ is a skew-Hermitian matrix in the tangent space of $\mathrm{SU}(p)$ at U, hence, left multiplication of U with the commutator will in general not keep us on the manifold. To perform a discrete gradient update step, we have to retract the Riemannian gradient from the tangent space onto $\mathrm{SU}(p)$ [35]. In contrast, for the Euclidean case of Eq. (2) where $\mathcal{M} = \mathbb{R}^n$ this is not necessary since the tangent space of $\mathbb{R}^n$ coincides with the manifold at all points: $T_\theta \mathbb{R}^n \cong \mathbb{R}^n$.

The canonical retraction for our setting is the Lie exponential map $\exp_U : T_U\mathrm{SU}(p) \to \mathrm{SU}(p)$, $\Omega \mapsto \exp_U\{\Omega\}$, so that $\exp_U\{t\Omega\}$ for $t \in [0, 1]$ describes a unique geodesic at U with initial "velocity" $\Omega \in T_U\mathrm{SU}(p)$. The operator $\exp_U$ can be decomposed as follows. We realize that $\mathrm{grad}\, L(U) = \Omega U$ with $\Omega = [U\rho_0 U^\dagger, H]$, hence, right multiplication with the inverse $U^\dagger$ yields an element of the Lie algebra. Taking $\exp\Omega$ and multiplying with U from the right then produces the retracted gradient [see Fig. 1(a)]. If we discretize Eq. (9) and perform the retraction, we finally obtain

$$U_{k+1} = \exp \epsilon [U_k \rho_0 U_k^\dagger, H] U_k, \tag{10}$$

where ϵ is the step size and $U_k \in \mathrm{SU}(p)$ the unitary at step k.

To analyze the convergence properties of Eq. (10), we rely on the fact that the map $\rho_0 \mapsto U_k \rho_0 U_k^\dagger$ can be understood as a so-called double-bracket flow on the adjoint orbits of the group [36–38].

Double-bracket flows can be used to solve a variety of tasks such as sorting lists [39], describing Toda flows [40], or diagonalizing Hamiltonians in many-body physics [41–43]. Additionally, they have been studied in the context of quantum gate design [44]. The properties of this optimization scheme are well understood, in particular, if H is nondegenerate there exist exactly $p!$ minima on $\mathrm{SU}(p)$, and $(p-1)!$ global minima. Amazingly, only the global minima are stable attractors of the optimization dynamics, and one can show that almost all points will converge to these minima given a suitable step size [16]. Hence, the Riemannian gradient flow is guaranteed to find the ground state of a nondegenerate Hamiltonian H.

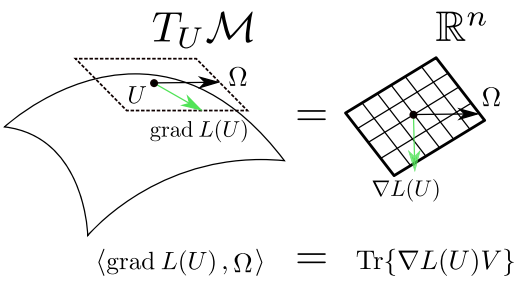


FIG. 2. The compatibility condition. By taking the Euclidean inner product as the reference inner product, we can enforce the invariance of the inner product under a change of metric and explicitly construct $\mathrm{grad}\, L(X)$.

III. EXACT RIEMANNIAN GRADIENT FLOW IN QUANTUM CIRCUITS

If U_k in Eq. (10) is implemented by a quantum circuit, then left multiplication of U_k with the retracted Riemannian gradient is nothing more than appending a set of gates to that circuit, as illustrated in Fig. 1(b). However, it should come as no surprise that an implementation of the Riemannian gradient flow on a quantum computer will require an exponential number of gates as the number of qubits N increases since an element in the Lie algebra $\mathrm{su}[2^N]$ is described by

$4^N - 1$ parameters in general. Nevertheless, we describe an approach for implementing the Riemannian gradient in a circuit in order to set up an approximate scheme that requires only a polynomial number of operations.

An exact approach to implement the Riemannian gradient on a quantum circuit is to decompose the skew-Hermitian operators $[U_k\rho_0 U_k^\dagger, H]$ in terms of a basis of the Lie algebra su$[2^N]$. One such basis is the set of Pauli words $\mathcal{P}^N = \{P^j\}$, where $P^j = \otimes_{l=1}^N \tilde{p}_l$ and $\tilde{p} \in \{I, X, Y, Z\}$ multiplied by i to ensure skew-Hermiticity.

We can write the commutator in the exponent of Eq. (10) in terms of this basis:

$$[U_k\rho_0 U_k^\dagger, H] = -\frac{1}{2^N}\sum_{j=1}^{4^N-1} \mathrm{Tr}\{[U_k\rho_0 U_k^\dagger, H]P^j\}P^j. \tag{11}$$

The coefficients

$$\omega_k^j = \mathrm{Tr}\{[U_k\rho_0 U_k^\dagger, H]P^j\} = \langle [H, P^j]\rangle_{\rho_k}, \tag{12}$$

with $\rho_k = U_k\rho_0 U_k^\dagger$, can then be calculated on a quantum device with the parameter-shift rule [4,5,45,46]

$$\langle [H, P^j]\rangle = i\left\langle V^\dagger\left(\frac{\pi}{2}\right)HV\left(\frac{\pi}{2}\right) - V^\dagger\left(-\frac{\pi}{2}\right)HV\left(-\frac{\pi}{2}\right)\right\rangle_{\rho_k}, \tag{13}$$

with $V(t) = \exp\{itP^j/2\}$ and the expectation value is calculated with respect to the state $U_k\rho_0 U_k^\dagger$. Hence, estimating the coefficients ω_k^j requires taking the gradient of $\langle H\rangle_t$ with respect to t given the state $V(t)U_k|\psi_0\rangle$. The resulting Riemannian gradient flow can be compactly written as

$$U_{k+1} \approx \prod_{j=1}^{4^N-1} \exp -\epsilon\omega_k^j P^j U_k, \tag{14}$$

where we absorbed the exponential factor into ϵ and took the sum out of the product via the Trotter formula at the cost of an error of $O(\epsilon^2)$. In addition to requiring $4^N - 1$ estimates of ω_k^j, this also requires applying the corresponding multiqubit gates generated by all Pauli words of size N, which will be very difficult in practice.

Note that instead of splitting the exponent of the sum with a Trotter decomposition, we could directly use a Cartan decomposition algorithm, e.g., the Khaneja-Glaser or D'Alessandro decomposition to recursively decompose the Riemannian gradient into products of single- and two-qubit unitaries [47–49].

IV. APPROXIMATE RIEMANNIAN GRADIENT FLOW IN QUANTUM CIRCUITS

To circumvent the exponential resources required for the exact Riemannian gradient, we consider an approximation scheme that requires only a polynomial number of parameters and gates. A natural approximation is restricting the Riemannian gradient to a subspace $\mathfrak{k} \subseteq$ su$[2^N]$ via an orthogonal projection onto $\mathfrak{k}$. We show this schematically in Fig. 3. If we

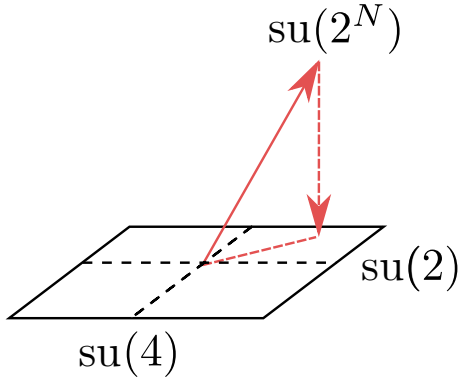


FIG. 3. Restricting the algebra to a subspace and projecting the Riemannian gradient onto this subspace. Schematically, one can also break down the projected subspace into further component subspaces (represented for simplicity as single lines).

let $\{K^j\} \subset \mathcal{P}^N$ for $j = 1, \ldots, k$ be a basis of the subspace $\mathfrak{k}$, then from Eq. (14) we obtain the local Riemannian gradient flow

$$U_{k+1} \approx \prod_{j=1}^{k} \exp -\epsilon\omega_k^j K^j U_k, \tag{15}$$

where now $\omega_k^j = \langle [H, K^j]\rangle_{\rho_k}$. This approximation gives us control over which directions in the Lie algebra we want to explore. Depending on the choice of $\mathfrak{k}$, we append a sequence of k gates at each optimization step. This approximation is an example of a stochastic Riemannian gradient algorithm [50].

Interestingly, the approximate Lie algebra optimization coincides with some VQE approaches for particular choices of $\mathfrak{k}$. For instance, if we restrict the Riemannian gradient to single-qubit Paulis,

$$\mathcal{P}^N_{\text{1-local}} := \{I^{\otimes i-1} \otimes \sigma_i \otimes I^{\otimes N-i} | \sigma_i \in \{X, Y, Z\}\}, \tag{16}$$

where σ_i acts on qubit i, then we are performing a variant of the circuit structure learning algorithm called Rotosolve [51–53], where instead of minimizing the expectation value $\langle H\rangle$ per added gate, we follow the Riemannian gradient with a step ϵ. Additionally, we can choose the subspace $\mathfrak{k}$ in such a way that the terms in the product (15) become two-qubit gates. For example, we could take the subspace to consist only of 2-local Paulis,

$$\mathcal{P}^N_{\text{2-local}} := \{I \otimes \sigma_i \otimes I \ldots I \otimes \sigma_j \otimes I | \sigma_i, \sigma_j \in \{X, Y, Z\}\},$$

which contains $|\mathcal{P}^N_{\text{2-local}}| = 9N(N-1)$ terms. We can also consider the nearest neighbor 2-local Paulis,

$$\mathcal{P}^N_{\text{2-local n.n.}} := \{I \otimes \sigma_i \otimes \sigma_j \otimes \cdots \otimes I | \sigma_i, \sigma_j \in \{X, Y, Z\}\},$$

which contains $|\mathcal{P}^N_{\text{2-local n.n.}}| = 9(N-1)$ terms. If instead of appending all K^j in our set we only append the unitary generated by the K^j with the largest ω_k^j, we are performing a popular metaheuristic first introduced in [54] called Adapt-VQE. The difference being that we do not reoptimize the parameters of previous layers at each step. Additionally, the Lyapunov control strategy FALQON [55] can be understood as a Trotterized time evolution where the step size of the drift Hamiltonian is set to the Riemannian gradient.

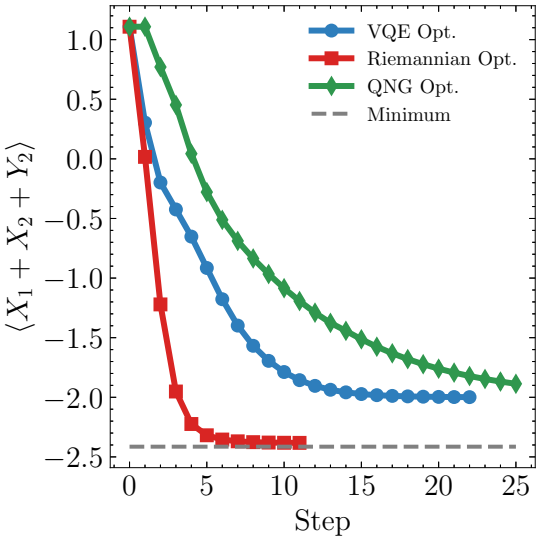


FIG. 4. Two-qubit example for the Hamiltonian $H = X_1 + X_2 + Y_2$. The circuit at initialization for both the VQE and Riemannian optimzer is given by the hardware-efficient *Ansatz* [61]. This circuit *Ansatz* consists of two RY gates with initial parameters (0.1,2.1), followed by two RZ gates with initial parameters (0.3,0.1) and finally a CNOT where the second qubit is the target. The step size for both the Riemannian gradient and parameter-shift VQE are $\epsilon = 0.5$. The VQE optimization gets stuck in a local minimum, whereas the Riemannian gradient-flow optimizer rapidly reaches the optimal solution of $\langle H \rangle \approx -2.40$. We see that the use of the quantum natural gradient (QNG) does not improve the overall convergence [11].

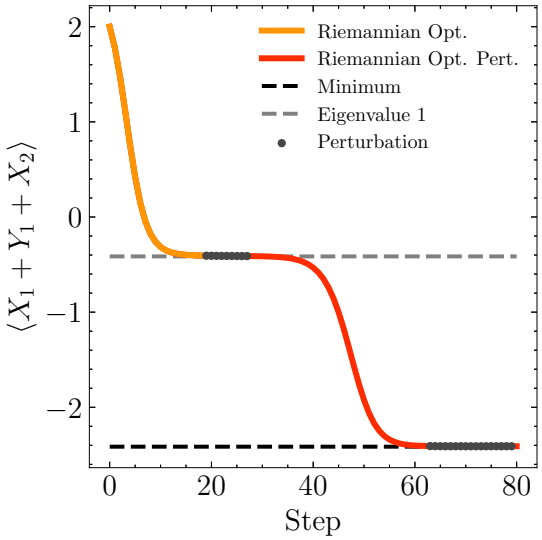


FIG. 5. Two-qubit example for the exact Riemannian gradient for the Hamiltonian $H = X_1 + Y_1 + X_2$. The circuit at initialization consists of two Hadamards on each qubit. The learning rate is set at $\epsilon = 0.2$. After 20 steps, the optimization gets stuck in an eigenstate. We generate a stochastic 4×4 matrix $X \sim \mathcal{N}(0, 0.1)^{4\times4}$ and obtain a random direction in the Lie algebra $K = \frac{i}{2}(X - X^T)$. After five perturbations, we escape the saddle point, and the optimization reaches the ground state of H.

Analogous to these methods, the choice of operator pool that will be appended to the circuit will affect the quality of the ground-state approximation. From the optimal control literature, we know that if $\{K^j\}$ spans $\mathrm{su}(2^N)$ under nested commutation, the system is controllable and any state can be reached given sufficient depth [56,57]. In the ADAPT-VQE setting, there are various proposals for which operators K^j to choose if one considers a fermionic Hamiltonian [58–60]. With the subspace restriction, the fixed-point analysis becomes highly nontrivial. Although we still have the same convergence criterion as before, $\mathrm{grad}\, L(U)|_{\mathfrak{k}} = 0$ can be satisfied if the Riemannian gradient has nonzero components orthogonal to the restricted subspace of the algebra, i.e., $\mathrm{grad}\, L(U)|_{\mathfrak{k}} \in \mathfrak{p}$ where $\mathrm{su}(p) = \mathfrak{p} \perp \mathfrak{k}$. As a result, we lose the global minima guarantees from the exact optimization. However, with the right choice of subspace, it is possible that the local Riemannian gradient information is enough to give a good approximation of the global minimum of Eq. (6).

V. NUMERICAL EXAMPLES

Here, we provide several numerical experiments on toy models to test the Riemannian gradient descent algorithm. Our Riemannian and VQE optimization procedures minimize the costs in Eqs. (6) and (1), respectively. We have implemented the Riemannian optimizer in PennyLane as the `LieAlgebraOptimizer` [62].

First, we consider the exact Riemannian gradient flow, which can be implemented on a circuit for small system sizes. In Fig. 4, we compare the optimizer with the parameter-shift rule for a two-qubit circuit. We see that the Riemannian gradient flow can reach the ground state of a simple Hamiltonian, whereas the VQE optimization can only reach a suboptimal solution.

To further illustrate the optimization properties of the exact Riemannian gradient flow, we study a two-qubit example in Fig. 5 where the optimization gets stuck in an eigenstate, which corresponds to a saddle point in the optimization landscape. After performing a small perturbation in the Lie algebra, we escape the saddle-point minimum and converge to the ground state.

In Fig. 6, we see a simple example of the approximate Riemannian gradient flow, where we restricted the full Lie algebra to a subset of directions. We see that after a few steps, we get close to the minimum of the function. For this example, the Lie algebra restriction still allows us to reach the ground state of the Hamiltonian.

Although the local approximation provides an accurate solution for the previous toy example, we can run into issues for more nontrivial problems, as we see in the final example. We consider the problem of finding the ground state of the transverse field Ising model on four qubits, whose Hamiltonian is given by

$$H = -\sum_i (Z_i Z_{i+1} + g X_i). \tag{17}$$

We assume periodic boundary conditions and set $g = 1$. The ground state of this model can be reached with a depth $N/2$ *Ansatz* for an N-qubit chain using gradient-based VQE

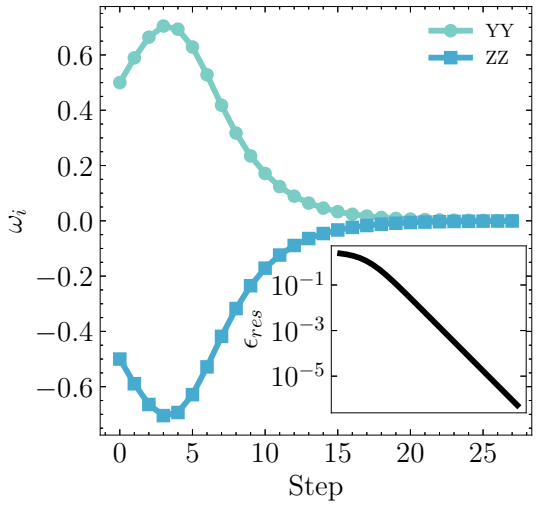


FIG. 6. Nonzero components of the Riemannian gradient versus the optimization step. The initial circuit consists of two Hadamard gates. The Hamiltonian is $H = X_1 + Y_1 Z_2$. At each step in the optimization, the Riemannian gradient $\text{grad}\, L(U) = -[U_k \rho_0 U_k^\dagger, H]$ only has components in the YY and ZZ directions, keeping the state in the submanifold spanned by the states reachable by (XX, YY, ZZ). We can therefore restrict the Lie algebra to the subspace $\mathfrak{k}$ spanned by $\{YY, ZZ\}$ and perform the approximate Riemannian gradient flow. At each step, we need to calculate $\{\omega_k^{YY}, \omega_k^{ZZ}\}$. In the inset we see the residual energy $\epsilon_{\text{res}} = E_0 - \langle H \rangle$ versus the optimization steps. As the optimization progresses, we get exponentially closer to the ground state of H.

[63–65]. We find that the approximate Riemannian gradient optimizer can get close to the ground state. But, unlike standard VQE, we cannot approximate the ground state closer than 1×10^{-2}, as can be seen in Fig. 7.

Here, we see a limitation of the approximate Riemannian gradient flow. If we restrict the Lie algebra to su(2) and su(4) operators, the Riemannian gradient only has a local view of the cost landscape, and cannot access higher-order Lie algebra directions. On the contrary, VQE can access these directions since the *Ansatz* is often universal, i.e., made from a product of single- and two-qubit unitaries. In principle, the unitary that is implemented by such an *Ansatz* could have a generator $W(\theta)$ such that

$$U_{\text{VQE}}(\theta) = \exp -iW(\theta), \tag{18}$$

that can explore additional su(p) directions in the Lie algebra for $p = 8, 16, \ldots$, albeit with a restricted parametrization.

A bottleneck for gradient-based VQE is that the number of circuit evaluations per optimization step scales linearly in the number of parameters, which is difficult in practice since parallel evaluation of quantum gradients requires multiple quantum devices. The approximate Riemannian gradient flow does not suffer from this issue since the amount of circuit evaluations is constant independent of circuit depth: We only require $|\mathfrak{k}|$ gradient calculations at each step. However, the Riemannian gradient flow may produce a circuit that is much deeper than the VQE *Ansatz* since we are appending gates to the circuit at each step k.

Ultimately, the approximate Riemannian gradient flow may not provide an accurate approximation to the ground state of a given Hamiltonian H. However, it could dynamically produce an *Ansatz* that serves as a good starting point for further VQE optimization, similar to ADAPT-VQE.

VI. CONCLUSION

In this work, we proposed Riemannian gradient flows in the context of variational quantum circuits. We showed that one can perform these types of optimizations on a quantum circuit, with strong convergence guarantees holding for exponentially deep variants of this algorithm. The usefulness of the local approximations to the Riemannian gradient flow merits further investigation in order to understand the power of this class of algorithms.

We hope that this alternative optimization paradigm can lead to new variational quantum algorithms, and provide insight for existing variational methods in noisy intermediate-scale quantum hardware. Additionally, we believe that the differential geometry and quantum control perspective can be a fruitful direction of research to further our understanding of the optimization properties such algorithms [67,68]. In particular, these ideas could be used to investigate overparametrization in VQE [65,69–72]. Perhaps the global convergence guarantees of double-bracket flows can be used to understand the convergence properties of deep quantum circuits and provide deeper insight into the power and limitations of VQE.

ACKNOWLEDGMENTS

We would like to thank M. Schuld and E. Gil-Fuster for the discussions and we are grateful to F. Miatto for providing helpful references. R.W. would like to thank D. Wakeham and J. Ceroni from the I.C. for the weekly discussions on differential geometry and J. Izaac for his help with implementing Riemannian gradient flow in PennyLane. We acknowledge the funding provided by MITACS for this project.

APPENDIX A: DIFFERENTIAL GEOMETRY

To establish notation, we briefly summarize some of the key concepts in differential geometry needed for our purposes. There exist many excellent references on the topic: see [31,32] for the physicist-friendly references and [33,34] for the more technical expositions on the subject.

1. Manifolds

A space $\mathcal{M}$ is called an n-dimensional topological manifold if it is locally homeomorphic to $\mathbb{R}^n$. Specifically, there must exist a family of open subsets $U_a \subseteq \mathcal{M}$ such that

(1) the family covers $\mathcal{M}$, i.e., $\bigcup_a U_a = \mathcal{M}$,

(2) $\forall\, a, \exists \varphi_a : U_a \to \varphi_a(U_a) \subset \mathbb{R}^n$ where φ_a is homeomorphic.

The pair (U_a, φ_a) is called a chart and a collection of charts that covers $\mathcal{M}$ is called an atlas $\mathcal{A}$. In order to develop a differential calculus, we require that all charts in $\mathcal{A}$ are C^k compatible. This means that if we have two charts (U_a, φ_a), (U_b, φ_b), we require that $\varphi_a(U_a \cap U_b)$ and $\varphi_b(U_a \cap U_b)$ are

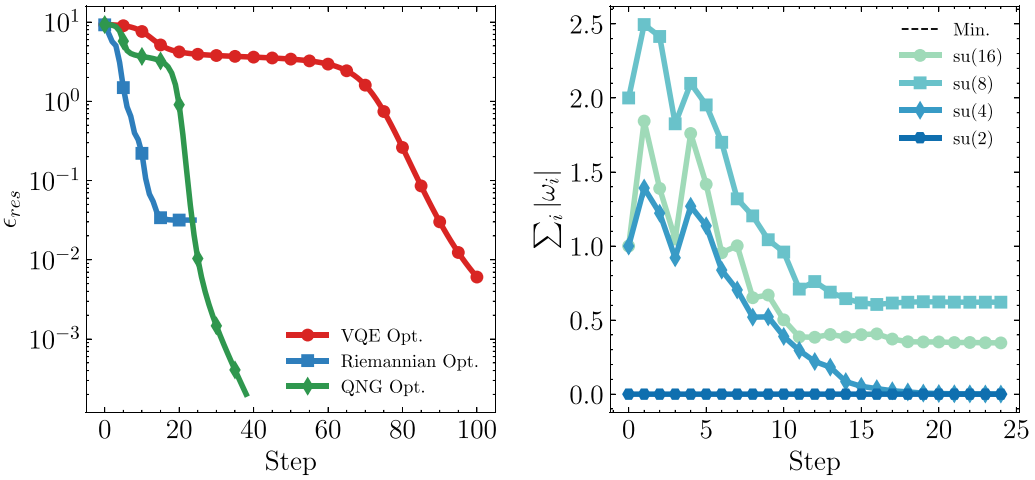


FIG. 7. Comparison of Riemannian gradient optimization versus gradient-based VQE for the four-qubit transverse field Ising model. The Riemannian gradient circuit is initialized with a Hadamard on each qubit. To minimize gate costs, we use an adaptive scheme to reduce the amount of gates appended at each step of the Riemannian optimization. We obtain the ω_k^j's on all qubits or pairs of qubits for su(2) and su(4), respectively. Then, we select the largest ω_k^j and use we use a structure optimization algorithm to calculate the optimal step size ϵ [45,46,52,66]. The gradient-based VQE optimizer has step size $\epsilon = 0.01$. Finally, we append a single gate corresponding to the chosen Lie algebra direction with this step size. (a) The residual energy $\epsilon_{\text{res}} = E_0 - \langle H \rangle$ plateaus for the Riemannian gradient close to the ground-state energy. We verify that the optimizer is not stuck in an eigenstate close to the ground state, and so the optimization gets stuck due to the projection of the gradient onto the local algebra. The VQE optimization on the other hand is still getting closer to the ground state. In addition, the QNG optimizer finds the ground state much more rapidly than the vanilla gradient descent optimizer by taking the geometry of the Hilbert space into account. (b) Here, we plot the magnitude of all components of the Riemannian gradient versus the optimization steps. We see that Riemannian gradient becomes zero in the su(4) direction, but higher-order Lie algebra directions are still nonzero. This explains why we cannot converge close to the ground state: we need to access higher-order elements of the Lie algebra. The transverse field Ising model has symmetries that we can exploit. In particular, we can use the dynamical Lie algebra of the model to construct a Riemannian gradient flow within a subgroup (see Appendix 2 C).

open and that $\varphi_a \circ \varphi_b^{-1}$ is C^k differentiable. The tuple $(\mathcal{M}, \mathcal{A})$ is called a k-differentiable manifold if all charts in $\mathcal{A}$ are C^k compatible. A function $f : \mathcal{M} \to \mathcal{N}$ is said to be C^k differentiable if for all charts (U_a, φ_a) on $\mathcal{M}$, (V_j, ψ_j) on $\mathcal{N}$ in the atlas we have that $\psi_j \circ f \circ \varphi_a^{-1}$ is C^k differentiable.

2. Tangent spaces

We are interested in generalizing the concept of a derivative to arbitrary manifolds. Consider a curve $\gamma : I \to \mathcal{M}$ where $I = (-a, a)$ is an open subset of $\mathbb{R}$ and $\mathcal{M}$ is a differentiable manifold. We can construct a curve on $\mathcal{M}$ so that $\gamma(0) = p$. Then we can ask the following: what is the derivative of a function $f : \mathcal{M} \to \mathbb{R}$ in the direction of this curve? By working in a chart (U, ϕ), $p \in U$ and $\phi(p) = \{x^i\}$ called the coordinate basis where x^i is the ith coordinate of the vector $\phi(p)$, we find

$$\left.\frac{df(\gamma(t))}{dt}\right|_{t=0} = \left.\frac{\partial(f \circ \phi^{-1})}{\partial x^i}\frac{d(\phi \circ \gamma)(t)}{dt}\right|_{t=0} = \left.\frac{\partial f}{\partial x^i}\frac{dx^i(\gamma(t))}{dt}\right|_{t=0}. \tag{A1}$$

This allows us to define a tangent vector at p as

$$v = v^i \frac{\partial}{\partial x^i}, \quad v^i = \left.\frac{dx^i(\gamma(t))}{dt}\right|_{t=0}. \tag{A2}$$

So a tangent vector is an operator that differentiates a function in the direction of some curve $\gamma(t)$ going through a point p as $v(f)(p)$. There exist many such curves, and these curves form an equivalence class. The collection of these equivalence classes is called the tangent space $T_p\mathcal{M}$ of $\mathcal{M}$ at p. The tangent space is then a vector space over linear maps called tangent vectors $v : C^\infty(\mathcal{M}) \to \mathbb{R}$, and can be spanned by a basis of differential operators $\{\partial/\partial x_i\} \equiv \{\partial_i\}$. Since $T_p\mathcal{M}$ is a vector space, there exists a dual vector space $T_p^*\mathcal{M}$ called the cotangent space. Elements of the cotangent space are called cotangent vectors or one-forms $\omega : T_p\mathcal{M} \to \mathbb{R}$, which accept a tangent vector and produce real number. A one-form can be expanded into a basis that is dual to $\{\partial_i\}$,

$$\omega = \omega_i dx^i, \tag{A3}$$

where $\partial_i dx^j = \delta_j^i$. The most important one-form for our purposes is the (exterior) derivative or differential df that takes a function and creates a one-form. The action of df is defined as

$$df(v) = v(f), \tag{A4}$$

$$df = \partial_i f(x^1, \ldots, x^n) dx^i. \tag{A5}$$

3. Riemmanian manifolds

A Riemannian manifold is a manifold $\mathcal{M}$ equipped with a symmetric, nondegenerate metric $g : T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R}$.

Given a basis $\{dx^i\}$ on $T_p^*\mathcal{M}$, the metric can be written as

$$g = g_{ij} dx^i \otimes dx^j. \tag{A6}$$

The metric thus defines an inner product between tangent vectors, which we denote by $\langle \dots, \dots \rangle$. This inner product induces an isomorphism $\flat : T_p\mathcal{M} \to T_p^*\mathcal{M}$ called a musical isomorphism,

$$\flat(v) = \langle v, \rangle ., \quad \forall\, v \in T_p\mathcal{M} \tag{A7}$$

with corresponding inverse, $\sharp : T_p^*\mathcal{M} \to T_p\mathcal{M}$ given by $\sharp(v) = \flat(v)^{-1}$, $\forall\, v \in T_p\mathcal{M}$. If we choose a basis $\{\partial_i\}$ on $T_p\mathcal{M}$ such that $\partial_i dx^j = \delta_i^j$, we see that

$$\flat(v) = \langle v^k \partial_k, \dots \rangle \tag{A8}$$

$$= v^k (g_{ij} dx^i \otimes dx^j) \partial_k \tag{A9}$$

$$= v^i g_{ij} dx^j \tag{A10}$$

and so

$$\sharp(dx^i) = g^{ij} \partial_j, \tag{A11}$$

so that $\sharp(\flat(v)) = v$. Hence, the metric allows us to convert tangent vectors into one-forms and vice versa. More importantly, the metric allows us to talk about distance and angles and provides a natural way to generalize the idea of a gradient to a Riemannian manifold. Remember from the previous section that the differential df creates a one-form from a tangent vector. If we define

$$\sharp(df(v)) \coloneqq \operatorname{grad} f \tag{A12}$$

to be the Riemannian gradient with respect to the metric g, then the differential of a vector $v \in T_p\mathcal{M}$ can be written as

$$df(v) = \langle \operatorname{grad} f, v \rangle. \tag{A13}$$

We can see that, by construction, the Riemannian gradient is an element of the tangent space, $\operatorname{grad} f \in T_p\mathcal{M}$, since the $\sharp$ operation produces a tangent vector. Additionally, $\operatorname{grad} f$ is perpendicular to the level curves at each point $x \in \mathcal{M}$ under the metric. To see this, consider a tangent vector v that points along the level curves of f, clearly we then have $df(v) = 0$ and thus $\operatorname{grad} f \perp v$.

Note that if we take the metric g_{ij} to be the standard Euclidean metric in the standard coordinate basis, we recover the gradient from multivariable calculus:

$$\sharp(df(v)) = (\partial_i f) v^i = \nabla f \cdot v. \tag{A14}$$

Since $df(v)$ is metric independent, we can understand the construction of the Riemannian gradient as requiring that $df(v) = \langle \operatorname{grad} f, v \rangle \equiv \nabla f \cdot v$ in the standard chart. This is called the compatibility condition of the Riemannian gradient.

APPENDIX B: THE GROUP SU(p)

Consider the special unitary Lie group $\mathrm{SU}(p)$:

$$\mathrm{SU}(p) \coloneqq \{X \in \mathbb{C}^{p\times p} | X^\dagger X = I, \det X = 1\}. \tag{B1}$$

This group is equal to $\mathrm{U}(p)$ up to a global phase, and has dimension $p^2 - 1$. Consider now a curve $X(t) : \mathbb{R} \to \mathrm{SU}(p)$, where $\forall\, t$, $X^\dagger X = I$ and $\det X = 1$. If we differentiate this condition with respect to t, we obtain

$$\frac{d}{dt}(X^\dagger(t) X(t)) = 0, \tag{B2}$$

$$\dot{X}^\dagger(t) X(t) + X^\dagger(t) \dot{X}(t) = 0. \tag{B3}$$

If $X(t)$ passes through X at time $t = 0$, then we see that $\dot{X}(0) = V$ must satisfy

$$T_X \mathrm{SU}(p) \coloneqq \{V \in \mathbb{C}^{p\times p} | V^\dagger X + X^\dagger V = 0\}, \tag{B4}$$

so V must be a skew-Hermitian matrix [23]. The Lie algebra is the tangent space of a Lie group at the identity. Hence, for $\mathrm{SU}(p)$,

$$\mathrm{su}(p) \coloneqq \{\Omega \in \mathbb{C}^{p\times p} | \Omega^\dagger = -\Omega\}. \tag{B5}$$

We see that the elements $\Omega \in \mathrm{su}(p)$ are related to Hermitian matrices H by $\Omega = iH$. By multiplying elements X of $\mathrm{SU}(p)$ to the right or left with an element of the algebra, we can move from the tangent space at the identity to the tangent space at X:

$$T_X \mathrm{SU}(p) \coloneqq \{V = \Omega X | \Omega \in \mathrm{su}(p)\}. \tag{B6}$$

APPENDIX C: RIEMANNIAN GRADIENT FLOW

The following is due to Ref. [15].

1. SU(p) flow

For $\mathrm{SU}(p)$ there exists a bi-invariant metric $\langle \dots, \dots \rangle : T_X \mathrm{SU}(p) \times T_X \mathrm{SU}(p) \to \mathbb{R}$ that induces a Riemannian gradient on the group [36]. This bi-invariant metric is given by $\langle WX, VX \rangle = \mathrm{Tr}\{X^\dagger W^\dagger V X\} = \langle W, V \rangle$, $\forall\, W, V \in T_X \mathrm{SU}(p)$. Consider the function

$$h : \mathrm{SU}(p) \to \mathbb{C}^{p\times p}, \quad h(X) \coloneqq C^\dagger X A X^\dagger, \tag{C1}$$

where C and A are Hermitian matrices on $\mathbb{C}^{p\times p}$. For a tangent vector $\Omega X \in T_X \mathrm{SU}(p)$, the derivative D of h is

$$D\,h(X)(\Omega X) = [C^\dagger (DX) A X^\dagger + C^\dagger X A (DX)^\dagger](\Omega X) \tag{C2}$$

$$= C^\dagger \Omega X A X^\dagger - C^\dagger X A X^\dagger \Omega. \tag{C3}$$

Because $DX(\Omega X) = \Omega X$, the derivative of $f : \mathrm{SU}(p) \to \mathbb{R}$, $f(X) = \mathrm{Tr}\{h(X)\}$ at ΩX is

$$D\,f(X)(\Omega X) = \mathrm{Tr}\{D\,h(X)(\Omega X)\} \tag{C4}$$

by the linearity of the trace. Defining $\tilde{A} \coloneqq X A X^\dagger$ we find

$$D\,f(X)(\Omega X) = \mathrm{Tr}\{C^\dagger \Omega \tilde{A} - C^\dagger \tilde{A} \Omega\} \tag{C5}$$

$$= \mathrm{Tr}\{[\tilde{A}, C^\dagger] \Omega\} \tag{C6}$$

$$= \langle [\tilde{A}, C^\dagger]^\dagger, \Omega \rangle \tag{C7}$$

$$= \langle -[\tilde{A}, C^\dagger] X, \Omega X \rangle, \tag{C8}$$

where we used that $\langle V, W \rangle = \langle VX, WX \rangle$ in the final line. We can now identify the Riemannian gradient from the compatibility condition

$$D\,f(X)(\Omega X) = \langle \operatorname{grad} f(X), \Omega X \rangle, \tag{C9}$$

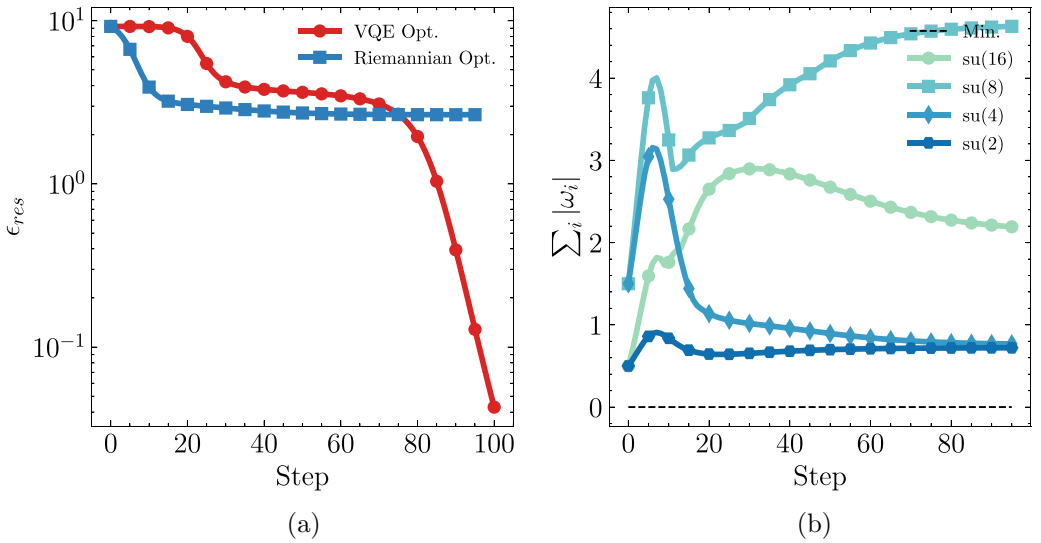


FIG. 8. Comparison of Riemannian gradient optimization versus gradient-based VQE for the four-qubit transverse field Ising model. The Riemannian gradient circuit is initialized with the identity. The VQE circuit is initialized in the same way as in Fig. 7 in the main text. The available directions for the Riemannian optimizer are the Pauli words in the dynamical Lie algebra: $\mathfrak{g} := \{X_i, Z_iZ_j, Y_iZ_j, Z_iY_j, Y_iY_j | 1 \leqslant i, j \leqslant N, i < j\}$, where X_i, Y_i, Z_i are Paulis on location i [68]. Even though the gradient flow stays within the dynamical Lie algebra at every step during the optimization, we see that the ground state is still unreachable and the optimization gets stuck in a local minimum.

so that

$$\mathrm{grad} f(X) = -[\tilde{A}, C^\dagger]X. \tag{C10}$$

Plugging in $\tilde{A} = U\rho_0 U^\dagger$ and $C = H$ and flipping the sign to find the minimum of Eq. (6) instead of the maximum gives the Riemannian gradient flow of Eq. (9).

2. Dynamical Lie algebra gradient flow

Given a Hamiltonian $H = \sum_n O_n$, let $\mathfrak{g}$ denote the set of operators spanned by consecutive applications of the Lie bracket to the set $\{O_n\}$, i.e., the closure of $\{O_n\}$ under commutation. The resulting dynamical (or Hamiltonian) Lie algebra is a subalgebra of $\mathrm{su}(2^N)$ and determines the set of states that can be reached by applying unitaries generated by elements of $\mathfrak{g}$ [56,57,68]. Let $H \in \mathfrak{g}$ and

$$\rho_0 = \frac{1}{2^N} I + \sum_l P_l, \quad \forall\, P_l \in \mathfrak{g}. \tag{C11}$$

We then find that the commutator

$$[\rho_0, H] = \frac{1}{2^N}[I, H] + \sum_l [P_n, H] \tag{C12}$$

$$= \sum_i [P_n, H] \tag{C13}$$

is also an element of $\mathfrak{g}$ because $\mathfrak{g}$ is closed under commutation. Since $\mathfrak{g}$ is a subalgebra, there is a corresponding subgroup G whose elements are generated by exponentiation of elements in $\mathfrak{g}$. Hence,

$$U = \exp \epsilon [\rho_0, H] \tag{C14}$$

is an element of the subgroup G. But then

$$U_{k+1} = \exp \epsilon [\rho_0, H] U_k \tag{C15}$$

will stay in the group G as long as U_0 is an element of G. We therefore see that an appropriate choice of ρ_0 and U_0 will keep the Riemannian gradient flow within the subgroup G. Unfortunately, it is possible that H cannot be diagonalized by elements of G and so the ground state may be unreachable for a flow that stays in the dynamical Lie algebra (see Fig. 8).

[1] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nat. Commun. **5**, 4213 (2014).

[2] E. Farhi, J. Goldstone, and S. Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028.

[3] K. J. Sung, J. Yao, M. P. Harrigan, N. C. Rubin, Z. Jiang, L. Lin, R. Babbush, and J. R. McClean, Using models to improve optimizers for variational quantum algorithms, Quantum Sci. Technol. **5**, 044008 (2020).

[4] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, Quantum circuit learning, Phys. Rev. A **98**, 032309 (2018).

[5] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Phys. Rev. A **99**, 032331 (2019).

[6] R. A. Fisher, On the mathematical foundations of theoretical statistics, Philos. Trans. R. Soc. London, Ser. A **222**, 309 (1922).

[7] S.-I. Amari, Natural gradient works efficiently in learning, Neural Comput. **10**, 251 (1998).

[8] R. Pascanu and Y. Bengio, Revisiting natural gradient for deep networks, in *2nd International Conference on Learning Representations (ICLR)*, edited by Y. Bengio and Y. LeCun (ICLR, 2014).

[9] G. Zhang, S. Sun, D. Duvenaud, and R. Grosse, Noisy Natural Gradient as Variational Inference, in *Proceedings of the 35th International Conference on Machine Learning*, edited by Jennifer Dy and Andreas Krause, Volume 80 of Proceedings of Machine Learning Research (PMLR Press, 2018), pp. 5852–5861.

[10] S. Sorella, Green Function Monte Carlo with Stochastic Reconfiguration, Phys. Rev. Lett. **80**, 4558 (1998).

[11] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, Quantum natural gradient, Quantum **4**, 269 (2020).

[12] A. Katabarwa, S. Sim, D. E. Koh, and P.-L. Dallaire-Demers, Connecting geometry and performance of two-qubit parameterized quantum circuits, Quantum **6**, 782 (2022).

[13] G. Becigneul and O.-E. Ganea, Riemannian adaptive optimization methods, in *7th International Conference on Learning Representations, ICLR, New Orleans, LA, 6-9 May 2019* (ICLR, 2019).

[14] C. Udrişte, *Convex Functions and Optimization Methods on Riemannian Manifolds*, Volume 297 of Mathematics and its Applications (Springer, New York, 1994).

[15] T. Schulte-Herbrüggen, S. J. Glaser, G. Dirr, and U. Helmke, Gradient flows for optimization in quantum information and quantum dynamics: Foundations and applications, Rev. Math. Phys. **22**, 597 (2010).

[16] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*, Communications and Control Engineering (Springer, London, 1994).

[17] A. Oza, A. Pechen, J. Dominy, V. Beltrani, K. Moore, and H. Rabitz, Optimization search effort over the control landscapes for open quantum systems with kraus-map evolution, J. Phys. A: Math. Theor. **42**, 205305 (2009).

[18] I. Luchnikov, M. Krechetov, and S. Filippov, Riemannian geometry and automatic differentiation for optimization problems of quantum physics and quantum technologies, New J. Phys. **23**, 073006 (2021).

[19] U. Helmke, K. Hüper, J. B. Moore, and T. Schulte-Herbrüggen, Gradient flows computing the C-numerical range with applications in NMR spectroscopy, J. Global Optim. **23**, 283 (2002).

[20] S. J. Glaser, T. Schulte-Herbrüggen, M. Sieveking, O. Schedletzky, N. C. Nielsen, O. W. Sørensen, and C. Griesinger, Unitary control in quantum ensembles: Maximizing signal intensity in coherent spectroscopy, Science **280**, 421 (1998).

[21] M. Hauru, M. V. Damme, and J. Haegeman, Riemannian optimization of isometric tensor networks, SciPost Phys. **10**, 040 (2021).

[22] Q. Miao and T. Barthel, A quantum-classical eigensolver using multiscale entanglement renormalization, arXiv:2108.13401.

[23] S. Fiori, Quasi-geodesic neural learning algorithms over the orthogonal group: A tutorial, J. Mach. Learn. Res. **6**, 743 (2005).

[24] S. Fiori, Learning by natural gradient on noncompact matrix-type pseudo-riemannian manifolds, IEEE Trans. Neural Networks **21**, 841 (2010).

[25] S. Wisdom, T. Powers, J. R. Hershey, J. L. Roux, and L. Atlas, Full-capacity unitary recurrent neural networks, in *Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16* (Curran Associates, Red Hook, NY, 2016), pp. 4887–4895.

[26] M. Lezcano-Casado and D. Martínez-Rubio, Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group, In *Proceedings of the 36th International Conference on Machine Learning*, edited by K. Chaudhuri and R. Salakhutdinov, Volume 97 of Proceedings of Machine Learning Research (PMLR Press, 2019), pp. 3794–3803.

[27] Z. Huang, J. Wu, and L. V. Gool, Building Deep Networks on Grassmann Manifolds, in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence* (AAAI Press, Washington, DC, 2018).

[28] M. A. Nielsen, M. R. Dowling, M. Gu, and A. C. Doherty, Quantum computation as geometry, Science **311**, 1133 (2006).

[29] M. R. Dowling and M. A. Nielsen, The geometry of quantum computation, Quantum Inf. Comput. **8**, 861 (2008).

[30] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht, Gradient Descent only Converges to Minimizers, in *29th Annual Conference on Learning Theory*, edited by Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, Volume 49 of Proceedings of Machine Learning Research (Columbia University, New York, 2016), pp. 1246–1257.

[31] J. Baez and J. P. Muniain, *Gauge Fields, Knots and Gravity*, Volume 4 of Series on Knots and Everything (World Scientific, Singapore, 1995).

[32] M. Nakahara, *Geometry, Topology and Physics*, Graduate Student Series in Physics (Taylor and Francis, London, 2003).

[33] M. P. do Carmo, *Riemannian Geometry* (Birkhäuser, Basel, 1992).

[34] J. M. Lee and J. M. Lee, *Introduction to Smooth Manifolds*, Graduate Texts in Mathematics (Springer, Berlin, 2003).

[35] S. Fiori, Lie-group-type neural system learning by manifold retractions, Neural Networks **21**, 1524 (2008).

[36] A. Arvanitoyeorgos, *An Introduction to Lie Groups and the Geometry of Homogeneous Spaces*, Volume 22 of Student Mathematical Library (American Mathematical Society, Providence, RI, 2003).

[37] T.-Y. Tam, Gradient flows and double bracket equations, Differ. Geom. Appl. **20**, 209 (2004).

[38] M. T. Chu and K. R. Driessel, The projected gradient method for least squares matrix approximations with spectral constraints, SIAM J. Numer. Anal. **27**, 1050 (1990).

[39] R. W. Brockett, Dynamical systems that sort lists, diagonalize matrices, and solve linear programming problems, Linear Algebra Appl. **146**, 79 (1991).

[40] A. M. Bloch, R. W. Brockett, and T. S. Ratiu, Completely integrable gradient flows, Commun. Math. Phys. **147**, 57 (1992).

[41] F. Wegner, Flow equations for Hamiltonians, Ann. Phys. **506**, 77 (1994).

[42] S. Kehrein, *The Flow Equation Approach to Many-Particle Systems*, Volume 216 of Springer Tracts in Modern Physics (Springer, Berlin, 2006).

[43] S. D. Głazek and K. G. Wilson, Renormalization of hamiltonians, Phys. Rev. D **48**, 5863 (1993).

[44] C. M. Dawson, J. Eisert, and T. J. Osborne, Unifying Variational Methods for Simulating Quantum Many-Body Systems, Phys. Rev. Lett. **100**, 130501 (2008).

[45] D. Wierichs, J. Izaac, C. Wang, and C. Yen-Yu Lin, General parameter-shift rules for quantum gradients, Quantum **6**, 677 (2022).

[46] O. Kyriienko and V. E. Elfving, Generalized quantum circuit differentiation rules, Phys. Rev. A **104**, 052417 (2021).

[47] N. Khaneja and S. J. Glaser, Cartan decomposition of $SU(2^n)$ and control of spin systems, Chem. Phys. **267**, 11 (2001).

[48] H. N. S. Earp and J. K. Pachos, A constructive algorithm for the Cartan decomposition of $SU(2^N)$, J. Math. Phys. **46**, 082108 (2005).

[49] D. D'Alessandro and R. Romano, Decompositions of unitary evolutions and entanglement dynamics of bipartite quantum systems, J. Math. Phys. **47**, 082109 (2006).

[50] S. Bonnabel, Stochastic gradient descent on riemannian manifolds, IEEE Trans. Autom. Control **58**, 2217 (2013).

[51] K. Mitarai, T. Yan, and K. Fujii, Generalization of the Output of a Variational Quantum Eigensolver by Parameter Interpolation with a Low-depth Ansatz, Phys. Rev. Appl. **11**, 044087 (2019).

[52] M. Ostaszewski, E. Grant, and M. Benedetti, Structure optimization for parameterized quantum circuits, Quantum **5**, 391 (2021).

[53] K. M. Nakanishi, K. Fujii, and S. Todo, Sequential minimal optimization for quantum-classical hybrid algorithms, Phys. Rev. Res. **2**, 043158 (2020).

[54] H. R. Grimsley, S. E. Economou, E. Barnes, and N. J. Mayhall, An adaptive variational algorithm for exact molecular simulations on a quantum computer, Nat. Commun. **10**, 3007 (2019).

[55] A. B. Magann, K. M. Rudinger, M. D. Grace, and M. Sarovar, Feedback-Based Quantum Optimization, Phys. Rev. Lett. **129**, 250502 (2022).

[56] F. Albertini and D. D'Alessandro, Notions of controllability for quantum mechanical systems, in *Proceedings of the 40th IEEE Conference on Decision and Control (Cat. No.01CH37228)*, Vol. 2 (IEEE, Piscataway, NJ, 2001), pp. 1589–1594.

[57] S. G. Schirmer, A. I. Solomon, and J. V. Leahy, Degrees of controllability for quantum systems and application to atomic systems, J. Phys. A: Math. Gen. **35**, 4125 (2002).

[58] H. L. Tang, V. O. Shkolnikov, G. S. Barron, H. R. Grimsley, N. J. Mayhall, E. Barnes, and S. E. Economou, qubit-adapt-vqe: An adaptive algorithm for constructing hardware-efficient ansätze on a quantum processor, PRX Quantum **2**, 020310 (2021).

[59] J. Liu, Z. Li, and J. Yang, An efficient adaptive variational quantum solver of the Schrödinger equation based on reduced density matrices, J. Chem. Phys. **154**, 244112 (2021).

[60] Y. S. Yordanov, V. Armaos, C. H. W. Barnes, and D. R. M. Arvidsson-Shukur, Qubit-excitation-based adaptive variational quantum eigensolver, Commun. Phys. **4**, 228 (2021).

[61] 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 (London) **549**, 242 (2017).

[62] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M. S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi *et al.*, Pennylane: Automatic differentiation of hybrid quantum-classical computations, arXiv:1811.04968.

[63] W. W. Ho and T. H. Hsieh, Efficient variational simulation of non-trivial quantum states, SciPost Phys. **6**, 029 (2019).

[64] D. Wierichs, C. Gogolin, and M. Kastoryano, Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer, Phys. Rev. Res. **2**, 043246 (2020).

[65] R. Wiersema, C. Zhou, Y. de Sereville, J. F. Carrasquilla, Y. B. Kim, and H. Yuen, Exploring entanglement and optimization within the hamiltonian variational ansatz, PRX Quantum **1**, 020319 (2020).

[66] R. M. Parrish, J. T. Iosue, A. Ozaeta, and P. L. McMahon, A jacobi diagonalization and anderson acceleration algorithm for variational quantum algorithm parameter optimization, arXiv:1904.03206.

[67] M. Larocca, P. Czarnik, K. Sharma, G. Muraleedharan, P. J. Coles, and M. Cerezo, Diagnosing barren plateaus with tools from quantum optimal control, Quantum **6**, 824 (2022).

[68] E. Kökcü, T. Steckmann, J. K. Freericks, E. F. Dumitrescu, and A. F. Kemper, Fixed Depth Hamiltonian Simulation via Cartan Decomposition, Phys. Rev. Lett. **129**, 070501 (2022).

[69] B. T. Kiani, S. Lloyd, and R. Maity, Learning unitaries by gradient descent, arXiv:2001.11897.

[70] J. Kim, J. Kim, and D. Rosa, Universal effectiveness of high-depth circuits in variational eigenproblems, Phys. Rev. Res. **3**, 023203 (2021).

[71] M. Larocca, N. Ju, D. García-Martín, P. J. Coles, and M. Cerezo, Theory of overparametrization in quantum neural networks, arXiv:2109.11676.

[72] X. You, S. Chakrabarti, and X. Wu, A convergence theory for over-parameterized variational quantum eigensolvers, arXiv:2205.12481.