PHYSICAL REVIEW RESEARCH **2**, 043246 (2020)

Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer

David Wierichs,[1,*] Christian Gogolin,[1,2] and Michael Kastoryano[1,3,4]
[1]*Institute for Theoretical Physics, University of Cologne, Germany*
[2]*Covestro Deutschland AG, Kaiser Wilhelm Allee 60, 51373 Leverkusen, Germany*
[3]*Amazon Quantum Solutions Lab, Seattle, Washington 98170, USA*
[4]*AWS Center for Quantum Computing, Pasadena, California 91125, USA*

(Received 14 May 2020; accepted 23 October 2020; published 17 November 2020)

We compare the BFGS optimizer, ADAM and NatGrad in the context of VQES. We systematically analyze their performance on the QAQA ansatz for the transverse field Ising and the XXZ model as well as on overparametrized circuits with the ability to break the symmetry of the Hamiltonian. The BFGS algorithm is frequently unable to find a global minimum for systems beyond about 20 spins and ADAM easily gets trapped in local minima or exhibits infeasible optimization durations. NatGrad on the other hand shows stable performance on all considered system sizes, rewarding its higher cost per epoch with reliability and competitive total run times. In sharp contrast to most classical gradient-based learning, the performance of all optimizers decreases upon seemingly benign overparametrization of the ansatz class, with BFGS and ADAM failing more often and more severely than NatGrad. This does not only stress the necessity for good ansatz circuits but also means that overparametrization, an established remedy for avoiding local minima in machine learning, does not seem to be a viable option in the context of VQES. The behavior in both investigated spin chains is similar, in particular the problems of BFGS and ADAM surface in both systems, even though their effective Hilbert space dimensions differ significantly. Overall our observations stress the importance of avoiding redundant degrees of freedom in ansatz circuits and to put established optimization algorithms and attached heuristics to test on larger system sizes. Natural gradient descent emerges as a promising choice to optimize large VQES.

DOI: 10.1103/PhysRevResearch.2.043246

I. INTRODUCTION

Variational quantum algorithms such as the variational quantum eigensolver (VQE) or the quantum approximate optimization algorithm (QAOA) [1] have received a lot of attention of late. They are promising candidates for gaining a quantum advantage already with noisy intermediate-scale quantum (NISQ) computers in areas such as quantum chemistry [2], condensed matter simulations [3], and discrete optimization tasks [4]. A major open problem is that of finding good classical optimizers which are able to guide such hybrid quantum-classical algorithms to desirable minima and to do this with the smallest possible number of calls to a quantum computer backend. In classical machine learning, the adaptive moment estimation (ADAM) optimizer [5] is among the most widely used and recommended algorithms [6,7], and has been one of the most important enablers of progress in deep learning in recent years. Such an accurate and versatile optimizer for quantum variational algorithms is yet to be found.

We are here mostly interested in variational algorithms for quantum many-body problems. To make progress towards finding an efficient and reliable optimizer for this domain, we concentrate on cost functions derived from typical quantum many-body Hamiltonians such as the transverse field Ising (TFIM) and the XXZ model (XXZM) for two reasons. First, their system size can be varied allowing us to systematically study scaling effects. Second, for *integrable* systems, the exact ground states are known and for the TFIM it is possible to construct ansatz classes for VQE circuits that provably contain the global minimum and can be simulated efficiently. Such systems thus allow us to distinguish between the performance of the optimizers and the expressiveness of the ansatz.

As a first result we show that the commonly used optimization strategies ADAM [8] and Broyden-Fletcher-Goldfarb-Shanno (BFGS) [9–18] both run into convergence problems when the system size of a VQE is increased. This happens already for system sizes within the reach of current and near future NISQ devices, which underlines the importance to a systematic search for suitable optimization strategies. The performance of ADAM is shown to depend strongly on the learning rate (the scaling prefactor determining the size of parameter update steps) via multiple effects and the number of epochs required for convergence increases fast with the problem size. Convergence can be improved but only with an expensive fine-tuning of the hyperparameters.

We then study the performance of an optimization strategy known as the quantum natural gradient or NatGrad [19–21]

*wierichs@thp.uni-koeln.de

and introduce Tikhonov regularization to the classical processing step in the VQE [22]. The key characteristic of NatGrad is that it uses the canonical metric on Hilbert space, the *Fubini-Study metric*, to determine improved updates to the variational parameters. While the proposal of NatGrad for VQES includes numerical experiments comparing it to several established optimizers as well as an ADAM variant that uses the natural gradient [19], the presented results extend this comparison in multiple directions: First, we include the BFGS optimizer which is widely used throughout the VQE literature. Second, different models are considered here, each of which is more complex than the example in Ref. [19] as they contain more than one two-qubit term. This extension is essential as is visible by the fact that no qualitative difference between the diagonal approximation of the Fubini-Study metric and the full metric was seen in Ref. [19] but the optimization problems presented here are not solvable with the diagonal approximation at all. Third, we extend the considered problem size from maximally 11 qubits to 40 qubits for the TFIM and 14 qubits for the XXZM. Fourth, our analysis includes the robustness of the investigated optimization algorithms regarding overparametrization, which can be expected to be of relevance in applications. We find that NatGrad does consistently find a global optimum for the largest system sizes we test (40 qubits) and requires significantly fewer epochs to do so than ADAM (in the cases where ADAM converges at all).

Our second set of results concerns the effect of overparametrization in VQES. We study the impact of adding redundant layers to the ideal circuit ansatz. This overparametrization not only increases the optimization cost, it actually appears to make finding the optimum significantly harder. The BFGS algorithm but also the ADAM optimizer, designed to thrive on additional degrees of freedom, fail frequently in this setting. This cannot easily be mitigated by increasing the epoch budget and reducing the learning rate of the ADAM optimizer. While also affected, NatGrad shows much higher resilience against this effect, compensating its higher cost per epoch with a higher chance to succeed. In applications on a relevant scale the circuit ansatz cannot be expected to be minimal making this resilience essential for the success of an optimizer for VQES. This also demonstrates the importance of understanding the role of redundant degrees of freedom in the variational class. When restricting the additional degrees of freedom to the symmetry sector of the model, ADAM does not profit from overparametrization and the BFGS optimizer performs worse whereas NatGrad reliably converges globally.

Our results are in sharp contrast to the usually very good performance of the ADAM optimizer and related (stochastic) gradient descend based techniques in the optimization of classical neural networks. A possible explanation for this good performance in usually overparametrized settings is the following: For common activation functions and random initialization, increasing overparametrization tends to transform local minima into saddle points [23,24]. The optimizer then mainly needs to follow a deep and narrow valley with comparably flat bottom to find a global minimum. The ADAM optimizer is perfectly suitable to pursue this path as it has individual learning rates per parameter that also take into account the average of recent updates (see Sec. II B for details). In this way it avoids side-to-side oscillations in the valley and can build up momentum to slide down the relatively flat bottom of the valley.

The energy landscapes of typical variational quantum algorithms however look very different. First, having deep and wide circuits with many parametrized gates is prohibitive on NISQ computers, which excludes overparametrization as a tool to make the variational space more accessible. Second, the variational parameters usually feed into gates as prefactors of exponentials of Pauli words and thus the cost function is ultimately a combination of trigonometric functions of the parameters. It appears that NatGrad is able to effectively use the information about the ansatz class to navigate the resulting energy landscape with many local minima. Third, it is known that large parts of the parameter space form so-called barren plateaus with very small gradients [25]. A random initialization of the parameters in reasonably deep VQES is thus almost certainly going to leave one stuck in such a plateau. Of course this also implies that one must prevent the optimizer from jumping to a random location in parameter space during optimization. This can be achieved in NatGrad by inhibiting unsuitably large steps by means of Tikhonov regularization. Finally, due to the small number of variational parameters in VQE, the added (classical) computational cost of inverting the Fubini-Study metric, which is used to determine the parameter updates (see Sec. II B), is negligible as compared to the cost of sampling from the quantum backend. This fact, combined with the highly correlated nature of the learning landscape in quantum many-body problems [26], might render second-order methods such as NatGrad more amenable to quantum than to classical settings, where samples are cheap, but there are many variational parameters.

In order to generalize our results, we consider the XXZM together with the Trotterized time evolution operator as circuit ansatz. Indeed we find BFGS to experience the same difficulties in high-dimensional parameter spaces and ADAM to exhibit a similar behavior of the required number of epochs as for the TFIM. The performance of NatGrad mostly is as reliable for this model as for the TFIM.

A. Informal summary of the results

Our main results are the following. First, NatGrad is the most reliable optimization method. This is due to the capability to maneuver high-dimensional search spaces driven by the Natural gradient and its relatively high resilience to overparametrization, both within and outside of the symmetry sector of the solution. The BFGS optimizer fails to navigate towards global minima in large spaces and in the presence of redundant degrees of freedom even in small systems. ADAM suffers significantly from symmetry-breaking overparametrization and is not able to use additional degrees of freedom *within* the symmetry sector for improved performance.

Second, NatGrad has larger quantum computation cost per epoch than the other algorithms by design but the improved learning strategy remedies this via small epoch counts to convergence. Meanwhile, BFGS takes few epochs to convergence at low cost per epoch but produces low-quality results, including local minima and positions in very shallow plateaus

in the cost function. ADAM also has low cost per epoch but for large and complicated problems it takes many epochs to converge and this duration is hard to predict.

Third, the above properties generalize to a certain level. That is, the failure of BFGS and the rapid increase in cost of ADAM appeared at similar parameter counts for different models and ansatz circuits and NatGrad tackled both spin chain systems successfully.

The practical conclusions from the presented work are twofold: On one hand, when solving the ground state energy problem with a VQE on an application-relevant scale, NatGrad appears to be the optimizer of choice for the classical processing step. This holds for both investigated spin chain models and, given the asymptotically vanishing cost overhead of NatGrad for Hamiltonians with many noncommuting terms, probably even more so for quantum chemical systems.

Finally, we observed decisive differences in the cost function landscape and optimizer performance from classical machine learning beyond obvious deviations like the dimension of the parameter space. This implies that heuristics and established methods from machine learning require new evaluation and additional research in order to optimally utilize them for VQES.

II. METHODS

A. Variational quantum eigensolver

The framework of our work is the VQE, a proposal to use parametrized circuits on a quantum computer in combination with classical optimization routines to prepare the ground state of a target Hamiltonian H. In the first part of a VQE, one constructs a quantum circuit that contains parametrized gates. Given input parameters θ for the circuit, a quantum computer can then prepare the corresponding ansatz state and measure an objective function, chosen to be the energy of the Hamiltonian

$$E(\theta) \coloneqq \langle\psi(\theta)|H|\psi(\theta)\rangle \tag{1}$$

and for benchmark problems with known ground state energy E_0, the relative error δ can be calculated as

$$\delta(\theta) \coloneqq \frac{E(\theta) - E_0}{|E_0|}. \tag{2}$$

Additionally one can prepare modified versions of the circuit to determine auxiliary quantities like the energy gradient in the parameter space [27]. The second part of the VQE scheme is an optimization strategy on a classical computer which is granted access to the quantum black box just constructed. In the most straightforward scenario this is a black box minimization scheme, but using auxiliary quantities, more sophisticated optimization methods can be realized as well.

There are two main theoretical challenges for successfully applying VQE. First, the construction of a sufficiently complex, but not overly expensive, circuit that gives rise to an ansatz class containing the ground state-*expressivity*. Second, the choice of a suitable optimizer that is able to search for the ground state within the created parameter space *efficiency*. The two challenges are often seen as independent, but explicit algorithms using information gathered about the variational space during optimization phases for adjusting the ansatz have been proposed as well, some of which are inspired by concrete applications in quantum chemistry or by evolutionary strategies [8,16,28,29].

We now establish some notation for the general VQE setting where we assume the most common objective: Finding the ground state energy of a Hamiltonian H. Starting from an initial product state $|\bar{\psi}\rangle$, we apply parametrized unitaries $\{U_j(\theta_j)\}_{1\leqslant j\leqslant n}$ to construct the ansatz state

$$|\psi(\theta)\rangle \coloneqq \prod_{j=n}^{1} U_j(\theta_j)|\bar{\psi}\rangle. \tag{3}$$

The parameters are typically initialized randomly close to zero to avoid the barren plateau problem [25]. For this work, the unitaries are going to be translationally invariant layers of one- or two-qubit rotations; consider, for instance,

$$L_{zz}(\theta_j) \coloneqq \prod_{k=1}^{N} \exp\left[-\frac{i\theta_j}{2} Z^{(k)} Z^{(k+1)}\right] \tag{4}$$

$$= \exp\left[-\frac{i\theta_j}{2} \sum_{k=1}^{N} Z^{(k)} Z^{(k+1)}\right], \tag{5}$$

where we identified the qubits with index 1 and $N+1$, i.e., we adopt periodic boundary conditions. The ordering of the gates within a layer is not relevant because they commute but for convenience we write them such that terms acting on the first qubits are applied first. $Z^{(k)}$ is the Pauli Z operator acting on the kth qubit and we tacitly assume the tensor product between operators that act on distinct qubits as well as the missing tensor factors of identities. Compared to proposed ansatz circuits that employ full Hamiltonian time evolution $\exp(-i\theta H)$ (see Sec. II A 1 a), such a layer is rather easily implemented on present quantum machines because it only requires linear connectivity and one type of two-qubit rotation. There have been many proposed circuits to generate ansatz classes for a variety of problems, all of which can be boiled down to combining rotational gates and possibly other fixed gates such as the CNOT or SWAP gate (see Sec. II A 1). For the presented optimization methods the derivatives with respect to the variational parameters $\{\theta_j\}_j$ are important and for the above example we observe the special structure of translationally symmetric layers of Pauli rotation gates:

$$\frac{\partial}{\partial\theta_j} L_{zz}(\theta_j) = \left(-\frac{i}{2} \sum_{k=1}^{N} Z^{(k)} Z^{(k+1)}\right) L_{zz}(\theta_j). \tag{6}$$

The derivative only produces an operator-valued prefactor, and all prefactors can be summarized because the single gates commute. While the basic gates composing a unitary $U_j(\theta_j)$ typically take the form of (local) Pauli rotations, the full unitary often is more complex than the above layer and in particular the terms in U_j do not need to commute. However, the structure of rotations enables us in general to evaluate required expressions involving derivatives on a quantum computer, either via measurements of rotation generators or via ancilla qubit schemes.

1. A selection of ansatz classes

Among the ansatz families proposed in the literature we present the following which are used frequently and are directly connected to this work:

(a) QAOA. The quantum approximate optimization algorithm was first proposed by Farhi, Goldstone and Gutmann [1] in 2014 for approximate solutions to (classical) optimization problems by mapping them to a spin chain Hamiltonian. The algorithm looks similar to adiabatic time evolution methods with an inhomogeneous time resolution, which is rather coarse for typical circuit depths. A lot of work has been put into proving properties of the QAOA both in general and for certain problem types, including extensions to quantum cost Hamiltonians [30–33]. At the same time the algorithm has been refined, extended, and characterized on the basis of heuristics and numerical experiments, gaining insight into its properties beyond rigorous statements [14,34–37].

The QAOA circuit is constructed as follows. For a *cost Hamiltonian* H_S and a so-called *mixing Hamiltonian* H_B one alternatingly applies the unitaries $\exp(-i\vartheta_j H_S)$ and $\exp(-i\varphi_j H_B)$ p times, giving rise to a VQE ansatz class with "time" parameters $\{\vartheta_j, \varphi_j\}_{1\leqslant j\leqslant p}$. Originally, the system Hamiltonian would encode a classical optimization problem and thus be diagonal while the mixing Hamiltonian was chosen to be off-diagonal and specifically has been kept fixed to the original $H_B = \sum_{k=1}^{N} X^{(k)}$ for many investigations. However, new choices of mixers have been proposed and investigated as well, giving rise to the more general quantum alternating operator ansatz (QAOa) [15,37,38].

Note that for quantum systems, the terms comprising the Hamiltonian H_S do not commute in general such that very large gate sequences would be necessary to realize the exact QAOA approach including $\exp(-i\vartheta H_S)$. In practice these blocks commonly are broken up in a Trotter-like fashion instead, yielding circuits that are implemented more readily but deviating from the original ansatz. For the TFIM, such a modified QAOA ansatz has been studied intensively [14,34,35] and we are going to use it as a starting point for our investigations.

(b) Adaptive Ansätze. Most prominently for this type of *Ansätze*, ADAPT-VQE tackles both the construction of a suitable ansatz class and the optimization within the constructed parameter space [16].

Instead of a fixed ansatz circuit layout, ADAPT-VQE takes a pool of gates as input and iterates the two steps of the VQE scheme: After rating all gates the most promising one is appended to the circuit (construction) and afterwards all the circuit parameters are optimized (minimization). The optimized parameters from the previous step are then used for both the rating of the gates for the next construction step and the initialization for the following optimization, where newly added gates are initialized close to the identity. For both the concept of allowed gates and the gate rating criteria, there are multiple options and we refer the reader to [16,28] for more detailed descriptions.

Besides ADAPT-VQE, multiple other methods which grow the ansatz circuit in interplay with the optimization have been proposed and demonstrated, including ROTOSELECT [8] and EVQE [29]. These demonstrations include the solution of five-qubit spin chains and small molecules (lithium hydride, beryllium dihydride, and a hydrogen chain) to chemical precision using simulations with and without sampling noise or quantum hardware.

We will not be using any adaptive scheme in our work, but our results on stability and overparametrization raise serious doubts as to the reliability of any adaptive ansatz method (see Sec. III B).

B. Optimizers

A variety of optimizers have been used in the context of variational quantum algorithms. These optimizers are inspired by classical machine learning and can be sorted according to the order of information required about the cost function. Zeroth-order or direct optimization methods only evaluate the function itself, first-order methods need access to the gradient, and second-order optimization need access to the Hessian of the cost function, or some other metric reflecting the local curvature of the learning landscape. As direct optimization is not scalable to problem sizes of relevance we do not include it in our studies. A parameter update step of the optimizer—corresponding to one iteration at the algorithmic level—is called an *epoch* and corresponds to one execution of the update rules described in the following [see Eqs. (7), (10), (11), and (13)]. Most optimization algorithms have one or more hyperparameters, the most common being the learning rate η, which is a scalar prefactor rescaling the parameter update at each epoch.

1. First-order gradient descent

Optimization techniques using the gradient of the cost function are at this point the most widely used in machine learning. Starting from the simple gradient descent method that updates the parameters according to the gradient and a fixed learning rate, a whole family of minimization strategies has been developed. The improved routines are inspired by physical processes like momentum, based on heuristics like adaptive learning rate schedules, or a smart processing of the gradient information as in the Nesterov accelerated gradient. A review of this development can be found e.g., in Ref. [7], here we just present the first-order method we are going to use, the ADAM optimizer.

ADAM, which was proposed in 2014 [5], is probably the most prevalent optimization strategy for deep feed-forward neural networks [6] and has been used in VQE settings as well [8]. For completeness, we briefly outline the ADAM optimizer: Given the cost function $E(\theta)$, where θ recollects all variational parameters, a starting point $\theta^{(0)}$ and a learning rate η, Gradient Descent computes the gradient $\nabla E(\theta^{(t)})$ at the current position and accordingly updates the parameters rescaled by η:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla E(\theta^{(t)}). \tag{7}$$

As the gradient points in the direction of steepest ascend, the parameter update is directed towards the steepest descend of the cost function and for η small enough, the convergence towards a minimum can be understood intuitively. Small learning rates yield slow convergence which increases the cost of the optimization whereas choosing η too large leads to overshooting and oscillations which might prevent con-

vergence. Furthermore, although the optimizer will diagnose convergence to a minimum due to a vanishing gradient, it cannot distinguish between local and global minima.

In order to fix both issues, i.e., the need for an optimally scheduled learning rate and the liability of getting stuck in local minima, various improvements have been proposed and ADAM uses several of these upgrades. The first feature is an *adaptive, componentwise learning rate*, which was introduced in ADAGRAD [39] and improved in RMSPROP [40] to avoid suppressed learning. The second feature ADAM uses is *momentum*, which is inspired by the physical momentum of a ball in a landscape with friction. This is realized by reusing past parameter upgrades weighted with an exponential decay towards the past and enables ADAM to overcome some local minima. The final form of the ADAM algorithm is as follows: Initialize with hyperparameters $\{\eta, \beta_1, \beta_2, \varepsilon\}$, momentum $m^{(0)} = 0$, average squared gradient $v^{(0)} = 0$ and initial position $\theta^{(0)}$. At the tth step, compute the gradient and update the momentum and the cumulated squared gradient as

$$m^{(t)} = \frac{\beta_1 - \beta_1^t}{1 - \beta_1^t} m^{(t-1)} + \frac{1 - \beta_1}{1 - \beta_1^t} \nabla E(\theta^{(t)}), \tag{8}$$

$$v^{(t)} = \frac{\beta_2 - \beta_2^t}{1 - \beta_2^t} v^{(t-1)} + \frac{1 - \beta_2}{1 - \beta_2^t} (\nabla E(\theta^{(t)}))^{\odot 2}, \tag{9}$$

where $x^{\odot 2}$ denotes the elementwise square of a vector x. The parameter update then is computed from these updated quantities via

$$\theta^{(t+1)} = \theta^{(t)} - \frac{\eta}{\sqrt[\odot]{v^{(t)}} + \varepsilon} m^{(t)} \tag{10}$$

with the square root of $v^{(t)}$ taken elementwise. Besides the learning rate η, we identify the hyperparameters β_1 and β_2 as exponential memory decay factors of m and v respectively and the small constant ε as regularizer, which avoids unreasonably large updates in flat regions and division by zero at initialization or for irrelevant parameters.

Because of the advanced features that ADAM uses, it has been very successful at many tasks and even though there are applications for which more basic gradient-based optimizers can be advantageous, we choose ADAM to represent the family of local first-order optimizers.

2. BFGS optimizer

The second optimizer we look at is the BFGS algorithm, which was proposed by its four authors independently in 1970 [9–12]. Using first-order resources only it approximates the Hessian of the cost function and performs global line searches in the direction of the gradient transformed by the Hessian inverse. Therefore it is a global quasi second-order method using local first-order information and its categorization is not obvious. The algorithm is initialized with the starting point $\theta^{(0)}$ and a first guess for the approximate Hessian $H^{(0)}$ of the cost function E, which usually is set to the identity. At each step of the optimization one determines the gradient, computes the direction

$$n^{(t)} = {H^{(t)}}^{-1} \nabla E(\theta^{(t)}) \tag{11}$$

and performs a line search on $\{\theta^{(t)} + \eta\, n^{(t)} | \eta \in \mathbb{R}\}$ which yields the optimal update in that direction and can optionally be restricted to a bounded parameter subspace. Given the new point in parameter space, $\theta^{(t+1)}$, the change in the gradient $D^{(t)} = \nabla E(\theta^{(t+1)}) - \nabla E(\theta^{(t)})$ is calculated and used to update the approximate Hessian via

$$H^{(t+1)} = H^{(t)} + \frac{D^{(t)} {D^{(t)}}^T}{\eta^{(t)} {D^{(t)}}^T n^{(t)}} - \frac{H^{(t)} n^{(t)} {n^{(t)}}^T H^{(t)}}{{n^{(t)}}^T H^{(t)} n^{(t)}}.$$

As the parameter updates are found via line searches, the BFGS algorithm is not strictly local but due to its use of local higher-order information, the global search is much more efficient than direct optimization. The method has been successful in many applications and currently is of widespread use for VQES [13–18].

3. Natural gradient descent

The third optimization strategy we use is the NatGrad [19–21], which due to its increased cost per epoch is not adopted very often in machine learning settings itself but is connected to some successful methods. As an example, stochastic reconfiguration which is closely related to NatGrad [41] recently has been shown to work well for training restricted Boltzmann machines (RBMs) to describe ground states of spin models [42]. Despite this success, the insights into why and under which conditions the method works remain limited and recent work has been put into understanding the learning process for the mentioned application of RBMs and the natural gradient descent [26]. Before discussing NatGrad and its role in the VQE setting, we outline its update rule. Given a starting point $\theta^{(0)}$ and a learning rate η, a step is performed by first constructing the Fubini-Study metric of the ansatz class

$$(F^{(t)})_{ij} := \mathfrak{Re}\{\langle \partial_i \psi^{(t)} | \partial_j \psi^{(t)} \rangle\} - \langle \partial_i \psi^{(t)} | \psi^{(t)} \rangle \langle \psi^{(t)} | \partial_j \psi^{(t)} \rangle \tag{12}$$

at the current position and then updating the parameters via

$$\theta^{(t+1)} = \theta^{(t)} - \eta\, {F^{(t)}}^{-1} \nabla E(\theta^{(t)}), \tag{13}$$

where we abbreviated $|\psi^{(t)})\rangle := |\psi(\theta^{(t)})\rangle$ and $|\partial_i \psi^{(t)}\rangle := \frac{\partial}{\partial \theta_i} |\psi(\theta^{(t)})\rangle$. The Fubini-Study metric is the quantum analog of the Fisher information matrix in the classical natural gradient [20]. It describes the curvature of the ansatz class rather than the learning landscape, but often performs just as well as Hessian based methods. In order to avoid unreasonably large updates caused by very small eigenvalues of F in standard natural gradient descent η has to be chosen very small for an unpredictable number of initial learning steps. Alternatively one can use *Tikhonov* regularization which amounts to adding a small constant to the diagonal of F before inverting it, also see Sec. II E.

Even though NatGrad is simple from an operational viewpoint, it is epochwise the most expensive optimizer of the three presented here (also see Sec. II C). This is due to the fact that it not only uses the gradient but, in order to construct the (Hermitian) matrix F for n parameters, one also needs to evaluate $\frac{1}{2}(n^2 + 3n)$ pairwise overlaps of the set $\{|\psi\rangle, |\partial_1 \psi\rangle, \ldots, |\partial_n \psi\rangle\}$ (all but $\langle \psi | \psi \rangle = 1$). Depending on the gates in the ansatz circuit, each of these overlaps requires

at least one and possibly many individual circuit executions. For circuits containing $\tilde{n}$ simple one- or two-qubit Pauli rotation gates, the number of circuits required is $\frac{1}{2}(\tilde{n}^2 + 3\tilde{n})$, independent of the number of shared parameters. Symmetries of the circuit may reduce the number of distinct terms in which case fewer quantum machine runs suffice.

Taking the jth parametrized unitary to have K_j Hermitian generators P_{j,k_j}, e.g., Pauli words up to prefactors $\{c_{j,k_j}\}$, the factors in the second expression of F take the shape of an expectation value [see also Eq. (6)]

$$\langle\psi|\partial_j\psi\rangle = \langle\bar{\psi}|\prod_{l=1}^{j-1} U_l^\dagger \left[\sum_{k_j=1}^{K_j} c_{j,k_j} P_{j,k_j}\right] \prod_{l=j-1}^{1} U_l|\bar{\psi}\rangle. \quad (14)$$

The first term in Eq. (12) requires slightly more complex circuits using one ancilla qubit and a depth which depends on the indices of the matrix entry [13,17,43,44]. Both for simulation work and for applications on real quantum machines, the construction of the Fubini matrix is expected to take much more time than inverting it—in sharp contrast to typical classical machine learning problems. Given the scaling of the number of required circuits above and the fact that for a fixed number of qubits the depth has to grow at least linearly with the number of parameters, an asymptotic scaling of $\mathcal{O}(\tilde{n}^3)$ is a lower bound for the construction of the full matrix. Standard matrix inversion algorithms do not only show smaller or equal scaling but also exhibit as prefactor the time cost of a FLOP whereas computing the matrix elements scales with prefactors based on sampling for expectation values.

As the number of parameters in a typical VQE circuit is considerably smaller than in neural networks and the circuit chosen in this work exhibits beneficial symmetries, the high cost of the method are expected to be less problematic for our setting and bearable for VQE applications. Indeed, there have been some demonstrations of the natural gradient descent and the imaginary time evolution for small VQE instances [19,45,46] as well as comparisons to standard gradient descent methods and imaginary time evolution for one- and two-qubit systems [47]. Inspired by the classical machine learning context and aiming for reduced cost, modifications of natural gradient descent have been proposed such as a (block) diagonal approximation to the Fubini-Study matrix [19]. We will later show that such simplifications have to be performed with caution and can disturb the optimization.

Finally we want to mention optimizers that treat the variational parameters sequentially, updating only one parameter at each epoch. While such algorithms can be designed to use information about the ansatz class and use the parametrization directly (see, e.g., Ref. [48]), we expect them to behave differently than optimizers updating all parameters simultaneously on which we focus our studies.

C. Optimization cost

To make a fair comparison between the optimization schemes, we briefly lay out the scaling of the required operations and the resulting cost per epoch.

We will use the following notation during the comparison. There are n variational parameters in the circuit, K_H terms in the Hamiltonian and on average $K = \sum_{j=1}^{n} K_j/n$ Pauli generators per variational parameter, with an average of N_M samples required for each expectation value. In practice, one of course would measure whole sets of operators both from the Hamiltonian and from the Pauli generator set simultaneously, such that K and K_H essentially are numbers of bases in which measurements are required. For entries of the Fubini matrix, we assume N_a samples for sufficiently precise measurements, which has been shown to be smaller than N_M numerically;[1] for further discussion see Ref. [45]. Finally, we introduce the timescales

$$t_x := \frac{d}{x} t_{\text{gate}} + t_{\text{wrap}} \quad (15)$$

for integers x that capture effects of averaging the depths of used auxiliary circuits. t_{gate} is the time required by each layer of parallelized gates and t_{wrap} includes the needed time for initializing and measuring the quantum register. Evaluating the gradient of the energy function can be done with different methods yielding a trade-off between precision and cost. On one hand, the analytic gradient can be evaluated up to measurement precision at the expense of an ancilla qubit and a scaling prefactor Kn. On the other hand there is the standard finite difference method, which can be performed symmetrically, asymmetrically or via simultaneous perturbation stochastic approximation (SPSA) [52], with cost prefactors $2n$, $n+1$ and 2, respectively. This means that robustness to imprecise gradients in general is a relevant property of any optimization scheme used for VQES because these gradients are much cheaper to evaluate. Computing the Fubini-Study metric requires two terms and although the measurement cost scales with $\mathcal{O}((Kn)^2)$ for the first and with $\mathcal{O}(Kn)$ for the second, we keep both terms in the overall cost scaling because the VQE regime implies moderate values of Kn.

For the scalings presented in Table I, we assume a homogeneous distribution of the variational gates in the circuit and that similar numbers of samples N_M are required to measure expectation values of the Hamiltonian terms within one basis and each derivative for all gradient methods.

For the full optimization algorithms, the cost are given per epoch as we do not have access to generic scaling of epochs to convergence. Using the cost per epoch one can rescale the optimization cost from epochs to estimated run time on a quantum computer beyond estimates that are based on the classical simulation run times. For the BFGS algorithm, we can not predict the number γ of energy evaluations that are required for the line searches but our numeric experiments and the linear scaling of the cost for nonSPSA gradients suggest that it can be neglected as compared to the gradient computation.

For the quantum run time scalings shown in Figs. 3, 5 and 8, we give the time in units of $t_{\text{eval}} = N_M K_H t_1$, assumed $N_M/N_a \approx 10$ [45] and approximated $t_1 \approx t_2 \approx t_3$.

[1] After submission of this manuscript, analytic bounds on the relative measurement cost of the gradient and the Fubini matrix have been presented in Ref. [60], underlining the numerical results in Ref. [45].

TABLE I. Cost on a quantum computer for selected VQE optimization methods and their subroutines. The optimizer cost are given per epoch, enabling us to compare the techniques beyond their simulation times with different scaling. We neglected terms which are small for $d, n \gg 1$ and used the timescales t_x defined in Eq. (15). The remaining scaling parameters $\{N_M, K, K_H, N_a\}$ are defined in the paragraph above Eq. (15).

	Operation	Quantum cost	
t_{eval}	Energy evaluation	$N_M K_H t_1$	Depending on measurement bases
t_{grad}	Analytic gradient	$(Kn)N_M K_H t_1$	Ancilla qubit required
		$2(Kn)N_M K_H t_1$	Parameter shift rule [27,49,50]
	Numeric gradient (sym.)	$2nN_M K_H t_1$	Sensitive to noise
	Numeric gradient (asym.)	$(n+1)N_M K_H t_1$	
	SPSA gradient	$2N_M K_H t_1$	Additional samples improve precision
t_{Fubini}	Fubini matrix	$(Kn)^2 N_a t_3 + (Kn)N_a t_2$	Ancilla qubit required
		$2(Kn)^2 N_a t_3 + (Kn)N_a t_2$	via projective measurements [51]
	BFGS	$t_{\text{grad}} + \gamma\, t_{\text{eval}}$	$\gamma = \mathcal{O}(n^{0 \leqslant \gamma < 1})$ expected
	ADAM	t_{grad}	
	NatGrad	$t_{\text{grad}} + t_{\text{Fubini}}$	Cost for inverting F can be neglected

1. Epoch count and quantum run time

When comparing the cost of optimizers that access the same resources, the epoch count N_{epoch} is a sufficient figure of merit. The presented algorithms, however, use distinct sets of quantities such that the quantum run time t_Q is a better measure to compare them. It is important to keep the system specific scaling of computing the gradient and the Fubini matrix in mind. The presented spin chain systems and ansätze with translation symmetry contain $\mathcal{O}(1)$ terms to be measured in the Hamiltonian leading to $\mathcal{O}(n)$ cost for the gradient for n parameters in the ansatz. The layered structure and the symmetry of the used circuits leads to $\mathcal{O}(n^3)$ measurements for the Fubini matrix, generating a large overhead in NatGrad. On the other hand, chemical Hamiltonians, which constitute an important application of VQES, contain $\mathcal{O}(N^4)$ terms for N electrons, which can be measured roughly in $\mathcal{O}(N^3)$ bases [53,54] implying cost $\mathcal{O}(nN^3)$ of measuring the gradient. Meanwhile, typical circuit types contain gates with a moderate constant number of generators, leading to $\mathcal{O}(n^2)$ cost of measuring the Fubini matrix, which is considerably smaller than $\mathcal{O}(nN^3)$ for any realistic circuit depth.

In summary, we consider the quantum run time t_Q to deliver a more meaningful comparison between different optimizers but report N_{epoch} as well to characterize the algorithms in a less system-dependent measure. Assuming the epoch count to behave similarly in various VQE landscapes, this enables us to estimate the relative cost of the optimizers when applied to, e.g., quantum chemistry.

D. Models

1. Transverse field Ising model

Our main model is the TFIM on a spin chain with periodic boundary conditions (PBC). Its Hamiltonian reads

$$H_{\text{TFI}} = H_S + H_B := -\sum_{k=1}^{N} Z^{(k)} Z^{(k+1)} - t \sum_{k=1}^{N} X^{(k)}, \tag{16}$$

where we identify the sites 1 and $N+1$ because of the PBC and t is the transverse field. For $t = 0$, the system is the classical Ising chain, which is also called ring of disagrees and is a special case of the MAXCUT problem [1,34]. For $t \neq 0$, the problem is no longer motivated by a classical optimization task and for the critical point $t = 1$, the ground state exhibits long-ranged correlations.

The ground state of the TFIM is found analytically by mapping it to a system of noninteracting fermions, where the transformed Hamiltonian can be diagonalized exactly [55]. The translational invariance of the Hamiltonian is crucial for this step and it will be important that only a small number of different (Pauli) terms can be mapped to *noninteracting* fermions simultaneously. We show the explicit computation via the Jordan-Wigner transformation in Appendix A, it can also be found in, e.g., Ref. [34]. Here we summarize the action of the mapping on the terms in the Hamiltonian which also generate the QAOA circuit [see Eq. (20) for the definition of α_q]:

$$\sum_{k=1}^{N} Z^{(k)} Z^{(k+1)} \longrightarrow (N-2r) + 2 \bigoplus_{q=1}^{r} [\cos\alpha_q\, Z + \sin\alpha_q\, Y], \tag{17}$$

$$\sum_{k=1}^{N} X^{(k)} \longrightarrow (N-2r) + 2 \bigoplus_{q=1}^{r} Z \tag{18}$$

where the expressions on the right are understood in a *fermionic operator basis* and the number of fermions is given by $r = \lfloor \frac{N}{2} \rfloor$. The ground state of H_{TFI} is just the product of the single-fermion ground states in momentum basis and we can write out the state and its energy as

$$E_0 = -E' - 2\sum_{q=1}^{r} \sqrt{1 + t^2 + 2t\cos\alpha_q} \quad \text{with} \tag{19}$$

$$\alpha_q := \begin{cases} (2q-1)\pi/N & \text{for } N \text{ even} \\ 2q\pi/N & \text{for } N \text{ odd} \end{cases}, \tag{20}$$

$$E' := \begin{cases} 0 & \text{for } N \text{ even} \\ 1+h & \text{for } N \text{ odd} \end{cases}. \tag{21}$$

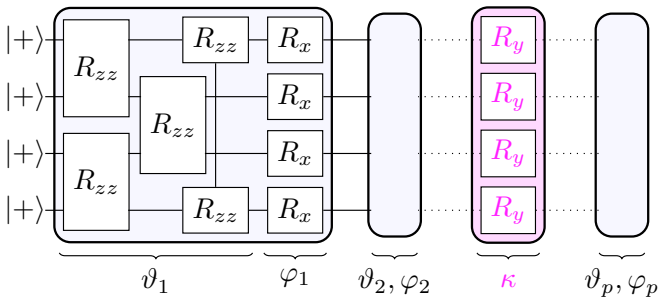

FIG. 1. The QAOA circuit for the TFIM on 4 qubits including an overparametrizing layer $L_y(\kappa)$. The first numerical experiment is performed without any Pauli Y layers L_y and in the second experiment overparametrization is investigated using one or two such layers.

Because of the free fermion mapping, we can not only obtain the exact ground state of the system but also justify the success of the modified QAOA circuit for the TFIM. As mentioned in Sec. II A 1 a, the original QAOA proposal would use the system Hamiltonian and a mixing term as generators for the parametrized gates. For the TFIM, however, separating the nearest-neighbour interaction terms H_S from the transverse field terms H_B recombines the latter with the mixing unitary next to it absorbing one variational parameter per block. The resulting parametrized circuit contains two types of translationally invariant layers, $L_x(\varphi)$ and $L_{zz}(\vartheta)$, of one- and two-qubit rotation gates, respectively. Starting in the ground state of H_B, that is $|\bar{\psi}\rangle = |+\rangle^{\otimes N}$, we alternatingly apply these two layers p times starting with L_{zz}. The resulting QAOA circuit is shown in Fig. 1. In the free fermion picture, this translates to $|\bar{\psi}\rangle = |0\rangle^{\otimes r}$ and to rotations of the r fermionic states about the z axis (L_x) and an axis $e_q = (0, \sin\alpha_q, \cos\alpha_q)^T$ which depends on the fermion momentum q (L_{zz}).

For $t = 0$, one can prove that this circuit can prepare the ground state exactly if and only if $p \geqslant r$ [14], whereas for the case $t \neq 0$ only numerical evidence and a nonrigorous explanation support this claim [35]. This explanation compares the number of independent parameters, $2p$ to the number of constraints from fixing the state of r free fermions, $2r$. While solvability would be implied for a linear system, the given problem is nonlinear and the argument remains on a nonrigorous level.

Finally, the equivalence to a system of free fermions has a practical implication for our simulations of the QAOA circuit: Storing the state of r free fermions just requires memory for $2r$ complex numbers. Applying the entire circuit needs $2pr$ two-dimensional matrix-vector multiplications, which is contrasted by $2pN$ matrix-vector multiplications in 2^N dimensions for a full circuit simulation in the qubit picture. Using the fermionic basis for numerical simulations, results on the VQE optimization problem for up to $N = 200$ and $p > 120$ have been obtained for $t = 0$ [14].

2. Heisenberg XXZ model

As a second model we consider the 1D XXZM with PBC which is defined by

$$H_{\mathrm{XXZ}} = \sum_{k=1}^{N} [X^{(k)}X^{(k+1)} + Y^{(k)}Y^{(k+1)} + \Delta Z^{(k)}Z^{(k+1)}]. \quad (22)$$

Δ is the anisotropy parameter. As in the TFIM, the sites 1 and $N+1$ are identified. The Bethe ansatz reduces the eigenvalue problem for the XXZM to a system of $N/2$ nonlinear equations that can be solved numerically with an iterative scheme [56,57]. This results in polynomial cost for computing the ground state energy but does not yield a simple ansatz class to construct the ground state on a quantum computer or a simulation scheme of reduced complexity.

We therefore use the XXZM as a second benchmark which models the application case more closely: We do not know a finite gate sequence that contains the ground state but instead employ circuits composed of symmetry-preserving layers which we found to be relatively successful in experiments. The ansatz we choose is the first-order Trotterized version of the unitary time evolution with the system Hamiltonian applied to a antiferromagnetic ground state:

$$|\psi(\theta)\rangle = \prod_{j=L}^{1} L_{zz}(\vartheta_j)L_{yy}(\kappa_j)L_{xx}(\varphi_j)|\bar{\psi}\rangle, \quad (23)$$

$$|\bar{\psi}\rangle = \frac{1}{\sqrt{2}}(|01\rangle^{\otimes N/2} \pm |10\rangle^{\otimes N/2}), \quad (24)$$

where we only treat even N and $|\bar{\psi}\rangle$ is chosen symmetric under translation for $(N \mod 4) = 0$ and antisymmetric for $(N \mod 4) = 2$ in anticipation of the exact solution via the Bethe ansatz. We found this circuit to be more successful at finding the ground state than the QAOA circuit. Even though the terms $\sum_{k=1}^{N} X^{(k)}X^{(k+1)}$ and $\sum_{k=1}^{N} Y^{(k)}Y^{(k+1)}$ do not preserve the magnetization in the Z-basis in general they do so within the sector of the above ansatz.

E. Simulation details

The simulations of the QAOA circuit for the TFIM are done in the free fermion picture yielding a quadratic scaling of the energy evaluation in N. The circuits including L_y layers and for the XXZM do not obey the same symmetries and therefore are implemented as a full circuit simulation using PROJECTQ [58]. The depth of the QAOA circuit for the TFIM is fixed to the smallest value containing the exact ground state $p = N/2$, which gives us N variational parameters with one added per L_y in the second main experiment. For the XXZ model, we choose $p = N$ resulting in $3N$ variational parameters. All circuit simulations are performed exactly, i.e., without noise or sampling. Furthermore we use the SCIPY implementation [59] of the BFGS algorithm and in-house routines for ADAM and NatGrad. All variational parameters are initialized uniformly i.i.d. over the interval $[0.0001, 0.05]$ as this corresponds to initializing the circuit close to the identity and symmetric randomization around 0 has shown slightly worse performance in our experiments.

We bound the BFGS optimization to one period of the rotation parameters as this improves the line search efficiency and found only a small dependence on the position of the interval. For the ADAM optimizer we fixed $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-7}$ and vary η in $[0.005, 0.5]$ trying to build heuristics for the particular problems. We found nontrivial behavior of ADAM with respect to the learning rate, observing a strong influence on the optimization duration, for details see Sec. III A. Furthermore, an increased regularization constant ε did not

yield any improvements of ADAM. For NATGRAD, we use learning rates of 0.5, 0.05, and 0.2 and fix the Tikhonov regularization constant to $\varepsilon_T = 10^{-4}$ and 10^{-3} for the TFIM and XXZM, respectively. This is a choice based on numerical experiments in which we explored the hyperparameter spaces of the optimizers. Even though we did not perform a full study on the impact of ε_T regarding the convergence quality or duration, we gained the following intuitive insight on the regularization: Choosing ε_T to be very small or even deactivating the regularization may lead to very large eigenvalues of F^{-1}, which ultimately are bounded artificially by the method of (pseudo-) inverting F. Consequentially, the Natural Gradient might lead to unreasonably large updates when choosing a fixed moderate learning rate η. We confirmed this numerically and observed the jumps generated by this effect to significantly degrade the optimization quality. Choosing a strong regularization on the other hand reduces the impact of the Fubini-Study metric and the (renormalized) limit $\varepsilon_T \to \infty$ corresponds to the standard gradient descent in Eq. (7). We therefore chose ε_T such that NatGrad did not perform excessive jumps in our preliminary experiments while maintaining a significant contribution of F to the optimizer.

Employing (block) diagonal approximations to the Fubini-Study matrix as suggested in [19] was not successful due to long-range correlations between the variational parameters in the circuit.

III. MAIN RESULTS

In this section, we state and assess the main numerical results of the paper. For a detailed description of the optimizers and circuit models, see the Methods section above (Sec. II).

A. QAOA circuits for the TFIM

We start our numerical investigation with the QAOA circuit for the TFIM on N qubits with critical transverse field $t = 1$ and analyze the *accuracy, speed and stability* of all three optimizers BFGS, ADAM and NatGrad (see Sec. II A 1 a for the ansatz and Sec. II D 1 for the model). We consider circuits with a depth of $p = N/2$ blocks corresponding to $n = N$ parameters, which are sufficiently expressive to contain the ground state and respect the symmetries of the Hamiltonian. For each system size, we sample 20 points close to the origin in parameter space and initialize each optimizer at these positions (see Sec. II E for simulation details). This leads to statistically distributed performances of the algorithms and as we perform exact simulations without sampling and noise it is the only source of stochasticity. The minimal relative error $\delta_{\min}$ and the number of required epochs for each initial point and optimizer are shown in Fig. 2.

Before we analyze the results, recall that the optimization problem can be solved exactly, i.e., the ansatz contains the true ground state. This enables us to identify optimization results with precisions $\delta_{\min} \geqslant 10^{-3}$ as local minima and we consider them to be unsuccessful as they deviate from the ground state on a physically relevant scale. In practical applications, the precision reached in both local and global minima would be much lower and in particular results with $\delta \approx 10^{-10}$ are

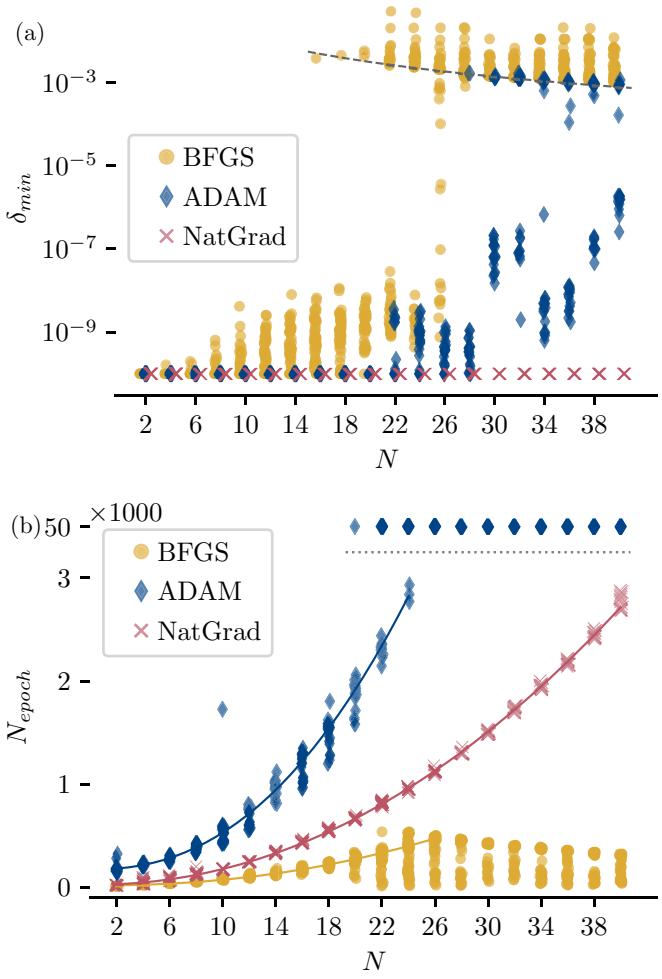

FIG. 2. Relative error $\delta_{\min}$ and epoch count N_{epoch} for the three optimizers initialized at 20 randomly chosen points close to the origin for the QAOA circuit with $n = N$ variational parameters. The ADAM optimizer is chosen with a learning rate of $\eta = 0.06$. (a) NatGrad reaches the ground state for all instances and all system sizes, while BFGS and ADAM start systematically getting stuck in local minima close to the first excited state (dashed line) beyond a system size of $N = 20$. (b) The monomial fits to the mean number of epochs to global minimization yield the scalings $N^{2.1}$ (BFGS), $N^{2.3}$ (ADAM), and $N^{2.1}$ (NatGrad). ADAM experiences a transition around $N = 22$ qubits, where the number of epochs to convergence jumps by an order of magnitude (separated by dotted line).

unreasonable to measure in quantum machines. This choice of benchmark is made in order to clearly reveal intrinsic features of the optimizers. For realistic applications, a systematic study of noise needs to be taken into account as well.

Our first observation is that the BFGS optimizer systematically fails to converge for systems sizes larger than $N = 20$. For small system sizes, however, it reaches a global minimum in the smallest number of epochs and at low cost per epoch (see Table I). The fast convergence is preserved for failed runs, which demonstrates that BFGS gets stuck in local minima, and can be attributed to the flexible parameter update size based on the line search subroutine. The runs of BFGS interrupted at a $\delta < 10^{-6}$ level could be improved to reach the

goal of $\delta = 10^{-10}$ by tuning the interrupt criterion. Therefore these runs are considered successful.

For ADAM, we here show the optimization results with $\eta = 0.06$, which similarly display a deterioration in accuracy for system sizes beyond $N = 26$. It is important to note that the failed ADAM runs are interrupted after 5×10^4 epochs and convergence with additional run time is not excluded in general. The question is then: How many update steps are needed for convergence? We observe a polynomial scaling of the required epochs in the system size up to a transition point $N^*(\eta)$, which depends on the chosen learning rate. Above this system size *both* successful and failing runs take much longer and exceed the set budget of 5×10^4 epochs.

The learning rate η imposes two main effects on the run time of the ADAM optimizer: On one hand, the transition point described above marks the system size at which a given learning rate leads to unpredictably high epoch numbers and increasing η shifts this point to smaller system sizes. On the other hand, a reduced learning rate slows down the optimization significantly, prolonging the optimization duration unnecessarily for all $N < N^*(\eta)$. This makes the choice of the learning rate for ADAM a system-dependent fine tuning problem, requiring additional heuristics and hyperparameter optimization. We present a more detailed analysis of the influence of the learning rate on the performance of ADAM in Appendix B.

In Fig. 2, we present the ADAM runs for a medium learning rate in order to demonstrate the described behavior but not the best possible performance of the ADAM optimizer.

NatGrad shows reliable convergence to a global minimum for all sampled initial parameters. The number of epochs to convergence scales polynomially with the system size and there is little variance in the required number of epochs.

For most of the unsuccessful runs, the relative error is very close to the (relative) gap of the Hamiltonian demonstrating that these local minima of the energy landscape correspond to excited states. This has been observed before in the context of digitized quantum annealing and QAOA [4] where the transition from ground to excited state is caused by a small energy gap of the (time-dependent) annealing Hamiltonian. The convergence to a local minimum reproduces this transition and demonstrates that the failed optimization runs yield deviations from the ground state not only on a level of numerical imprecisions but on a physically relevant scale, leading to wrong results of the VQE. For transverse fields other than $t = 1$, the similarity between the gap and the error due to local minima was not confirmed (see Appendix C) and, in particular, the latter is too small for the presented optimizer comparison for $t > 1$ and the optimization becomes too easy for $t < 1$.

Using the scalings as discussed in Sec. II C and taking the translation symmetry of the TFIM into account, we show the expected optimization durations on a quantum computer in Fig. 3. Due to the increased cost per epoch and a similar scaling of the number of epochs for all optimizers, the cost for NatGrad are considerably higher than those for BFGS and ADAM in the regimes in which they converge and ADAM does not suffer from the sudden increase in required epochs. We expect the scaling for ADAM, which is truncated in Fig. 2 due to our epoch budget, to yield quantum run times comparable to those of NatGrad. As we show in Appendix B, reducing the learning rate makes bigger system sizes accessible to ADAM, but also rather drastically increases run times because of slower convergence.

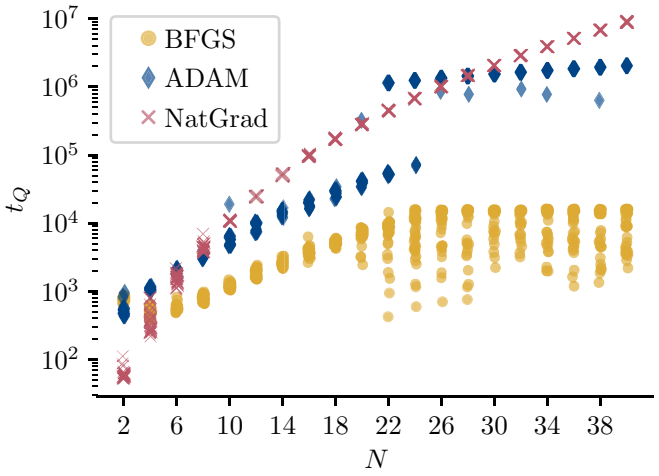

FIG. 3. Estimated run times t_Q on a quantum computer for the optimization runs shown in Fig. 2 based on the scalings in Table I. We note that none of the ADAM optimizations for $N \geqslant 30$ attained the full precision of 10^{-10} such that the scaling is truncated based on the epoch budget.

In summary, we find the BFGS optimizer to run into convergence problems already for medium sized systems, ADAM to take a large number of epochs with a transition into unpredictable cost at a certain system size and NatGrad to exhibit reliable convergence. While the estimated cost for running NatGrad on a real quantum computer are high, the number of epochs is much smaller than for ADAM. This implies that in applications like quantum chemistry which exhibit a more favourable scaling for measuring the Fubini matrix as compared to the gradient, NatGrad can be expected to be significantly cheaper overall (cf. Sec. II C 1).

Furthermore, the success of both commonly used optimizers, BFGS and ADAM, strongly depends on the initial parameters whereas NatGrad shows stable convergence and a small variance of the optimization duration.

B. Overparametrization by adding Y layers

We now extend the optimal QAOA circuit for the TFIM by adding redundant layers of Pauli Y rotations. These additional rotations can be deactivated by setting their variational parameter κ to zero. This means in particular that the new ansatz classes still contain the ground state and simply introduce a form of overparametrization. Alternatively one can introduce additional degrees of freedom to the circuit by using more blocks in the QAOA circuit than minimally required, maintaining the symmetry of the model, which is shown in Sec. III C.

As single-qubit Pauli Y rotations cannot be represented in the free fermion basis of the Hamiltonian [see Eq. (17)], the overparametrized class can be seen as breaking a symmetry. This means that for any given $\kappa \neq 0$, the ansatz state will not be a global minimum and it will be crucial for an optimization

algorithm to find the subspace with $\kappa = 0$. This is clear for a single additional layer of gates, but we expect it to hold for multiple nonadjacent layers as well. Although the present situation is artificially constructed and the broken symmetry is manifest, similar behavior is expected in systems where we do not have an analytical solution. More generally, even for an ansatz class which is suitable to express the ground state a very specific configuration of the variational parameters is necessary to find that state and the chosen optimization algorithm consequentially should be resilient to local minima.

Our choice of overparametrization leads to such local minima, constructing an optimization problem that can be used as a test for the resilience of the optimizer. It furthermore is comparable to overparametrizing a classical neural network respecting translational symmetry but outside of the sector equivalent to free fermions as presented in, e.g., Ref. [42]. The presented experiment thus can be used to compare the optimizer performance for classical machine learning of quantum states and VQES.

We look at two configurations of the extended circuits with y-rotation layers included at positions $\{\lfloor \frac{N}{4} \rfloor\}$ and $\{\lfloor \frac{N}{4} \rfloor, \lfloor \frac{N}{2} \rfloor - 1\}$, respectively. With this choice we avoid special points in the circuit and expect these setups to properly emulate the problem of (additional) local minima.

Again we sample 20 positions in parameter space close to the origin and initialize the three optimizers at these points, resulting in the precisions and success ratios shown in Fig. 4 together with the estimated quantum computer run times in Fig. 5. We observe a clear distinction between the optimizations that succeed to find a global minimum and those which converge to a local minimum only, which makes the success ratio for this numerical experiment well-defined. In contrast to the results for the minimal QAOA circuit, no intermediate precisions caused by a finite epoch budget occur. All optimizers suffer from the introduced gates as they show convergence to local minima for system sizes they tackled successfully without overparametrization. The error of these attained local minima lies on a relevant scale but is smaller than the gap of the model by a factor of ∼0.4.

For BFGS, this effect appears for some system sizes for one layer of Pauli Y rotations but is much stronger for two additional layers, reducing the fraction of globally minimized runs to less than 50% for multiple system sizes. We do not claim a scaling behavior with the system size but note an alternating pattern for the configuration with two Y layers, demonstrating large fluctuations of the success ratio (cf. in particular system sizes 10 and 12 for two Y layers).

For the ADAM optimizer, we use a comparably small learning rate of $\eta = 0.02$, which pushes the jump of the optimization duration that we observed before well out of the treated system size range. Nonetheless, we observe runs stuck in local minima already for small systems without exceeding the epoch budget so in contrast to Sec. III A allowing for a longer run time would not improve the performance. Also for ADAM, the fraction of successful instances fluctuates with the system size but in particular for two Pauli Y rotation layers the effect becomes stronger for bigger systems and no successful runs were observed for $N \geqslant 14$.

The performance of NatGrad on the other hand, for which we reduced the learning rate to $\eta = 0.05$, is more reliable

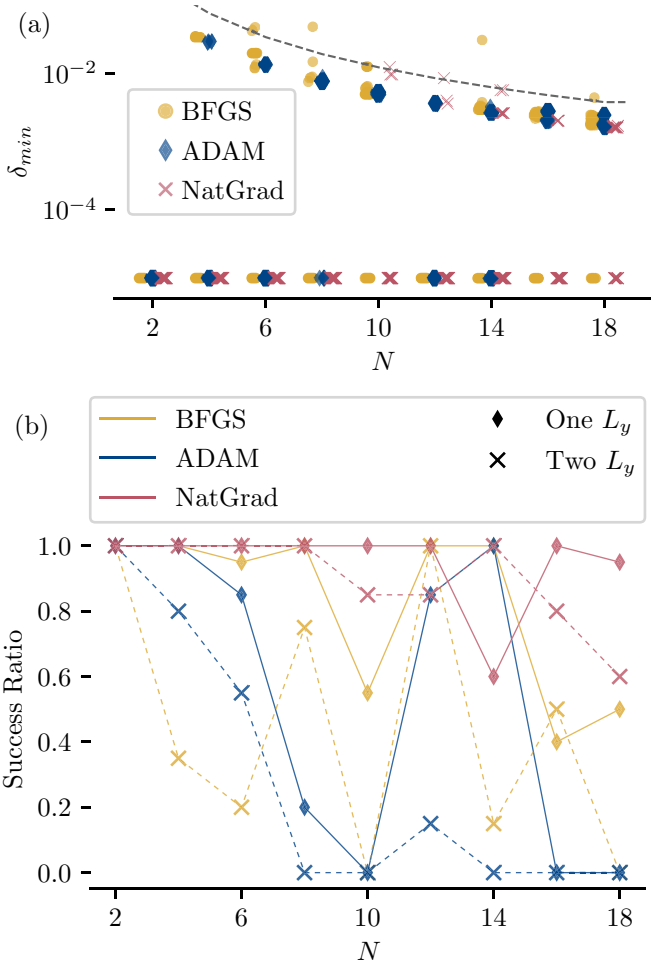

FIG. 4. (a) Achieved precisions δ_{min} and (b) fraction of successful optimizations out of 20 runs with the three optimizers on QAOA circuits extended by one or two Pauli Y-rotation layers. Successful optimization runs and those only converging locally are separated by a gap in the attained minimal precision, which is smaller but on the scale of the gap of the model, and in contrast to Fig. 2 the epoch budget is almost never consumed entirely. Instead the optimization is completed, yielding either a global or a local minimum.

and the success rate is the best for most of the circuits, with few exceptions. In particular, there are only few system sizes with local convergence for one and two additional degrees of freedom each and overall the success rate of NatGrad does not drop below 60%. For 10 and 18 qubits and two additional layers, NatGrad solves 85% and 60% of the task instances, respectively, while BFGS and ADAM fail in *all* of them.

For all optimizers, we confirm that successful runs deactivate the additional Pauli Y rotation layers by setting the corresponding parameters to 0 and that all optimizations with worse precision failed to do so, leading to a local minimization only. The quantum run times demonstrate the expected scaling with NatGrad as the most expensive optimizer, where the small epoch count compensates the increased cost per epoch for small systems. However, the increased effort is rewarded with significantly higher success rates, making NatGrad a strong choice for (potentially) overparametrized VQE optimization. We want to stress that the relative cost of the Fubini

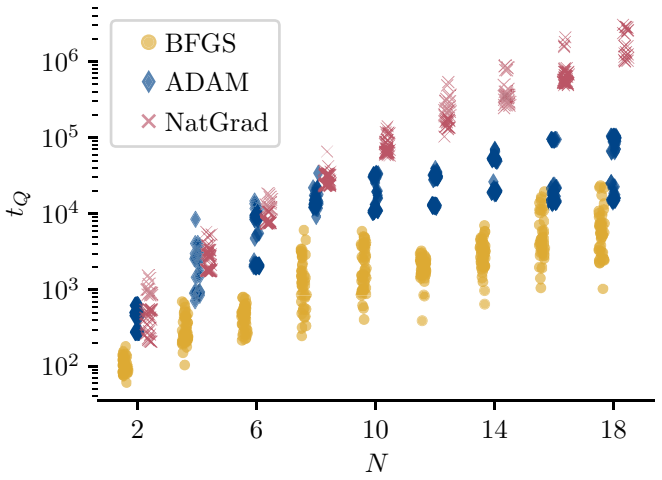

FIG. 5. Estimated run times t_Q on a quantum computer for the optimizations in Fig. 4 based on Table I and the same assumptions as in Fig. 3. For the ADAM optimizer the lower branch of data points corresponds to successful minimizations.

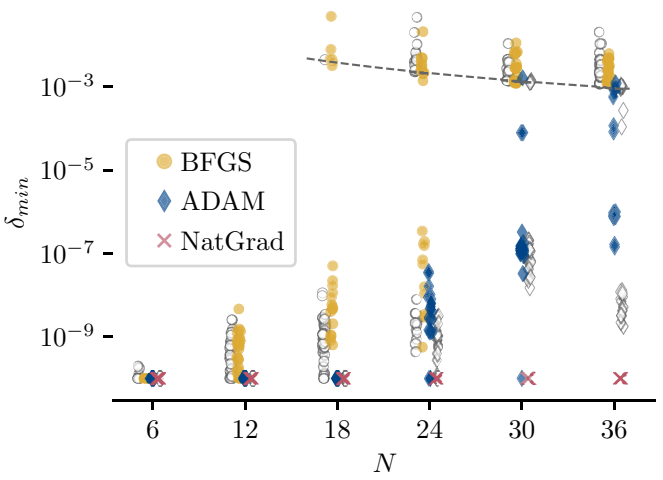

FIG. 6. Minimal attained relative errors δ_{min} for the TFIM as in Fig. 2 but with the enlarged QAOA circuit containing 2 additional blocks. The empty markers show the results from Fig. 2 for comparison.

matrix are high for spin chain systems and that the reduced number of epochs required by NatGrad will have a bigger impact in other systems (see also Sec. II C 1).

Overall our numerical experiments with the extended QAOA circuits for the TFIM demonstrate the fragility of the three tested optimizers to perturbations of the ansatz class. A significant decrease in performance is caused by overparametrization outside of the symmetry sector of the model and the QAOA ansatz class. All algorithms were successful for the original QAOA circuits on the considered system sizes implying that the reduced success ratio can directly be attributed to the extension of the ansatz class. This is in contrast to machine learning settings where heavy overparametrization is essential to make the cost function landscape tractable to local optimizers like ADAM. The strong fluctuations over the tested system sizes indicate that more repetitions of the optimization would be required to resolve systematic behavior.

We note that the BFGS algorithm in some instances converges to a local minimum although it has access to nonlocal information via its line search subroutine. In particular, in the presence of two misleading parameters in the search space, the local information determining the one-dimensional subspace does not seem to suffice any longer to find the global minimum, even though the approximated Hessian is used. For the ADAM optimizer, the initial gradient leads to an activation of symmetry breaking layers and due to the restriction to local information the algorithm is not able to leave the resulting sector of the search space with local minima it enters initially. NatGrad also is affected by the limitation to local information but because of the access to geometric properties of the ansatz state class it was on average less likely to leave the Pauli Y-rotation layers activated. We attribute this to the fact that NatGrad performs the optimization in the locally undeformed Hilbert space by extracting the influence of the parametrization. As a consequence the optimizer does not follow the incentive to activate the Pauli Y rotations at the beginning when given the same gradient as ADAM, but stays within the minimal parameter subspace. A better foundation for this intuition and the observed exceptions will be subject to further investigations of NatGrad.

Our results also hint at possible hurdles for adaptive optimization strategies which construct the circuit ansatz iteratively: to obtain viable scaling with the problem size and parameter count, such algorithms have to rate the available gates based on local information in order to estimate their usefulness for the VQE. This rating however might suggest gates which introduce problematic local minima as in the case demonstrated here. When testing ADAPT-VQE [16] for the TFIM we indeed observed that rating gate layers by their gradient suggests using L_y, which—as demonstrated above—is harmful for the VQE.

C. Symmetry-preserving overparametrization

Here we discuss the effect of overparametrizing the QAOA ansatz for the TFIM with symmetry-preserving layers, i.e., by choosing the number of blocks p bigger than the minimum $\lfloor \frac{N}{2} \rfloor$ required to achieve the exact solution. To this end, we optimized the QAOA ansatz on the critical TFIM with two additional blocks, corresponding to four additional variational parameters while keeping all hyper- and simulation parameters fixed and present the attained relative precisions in Fig. 6.

All optimizers perform similarly to the optimizations of the minimal QAOA circuit (displayed with empty markers). The BFGS optimizer achieves slightly less precise results, ADAM obtains similar precisions within statistical fluctuations, showing singular improved convergence but many results with worse precision, and NatGrad solves all instances to requested precision as before. In particular, this means that overparametrization does not facilitate the optimization task but even tends to make it more difficult for the established optimizers. For the BFGS algorithm, this is in accordance with the intuition for large systems which links the poor performance to the high dimensionality of the parameter space and the unfit information access via line searches (see Sec. III A). For ADAM however, the results show a decisive difference

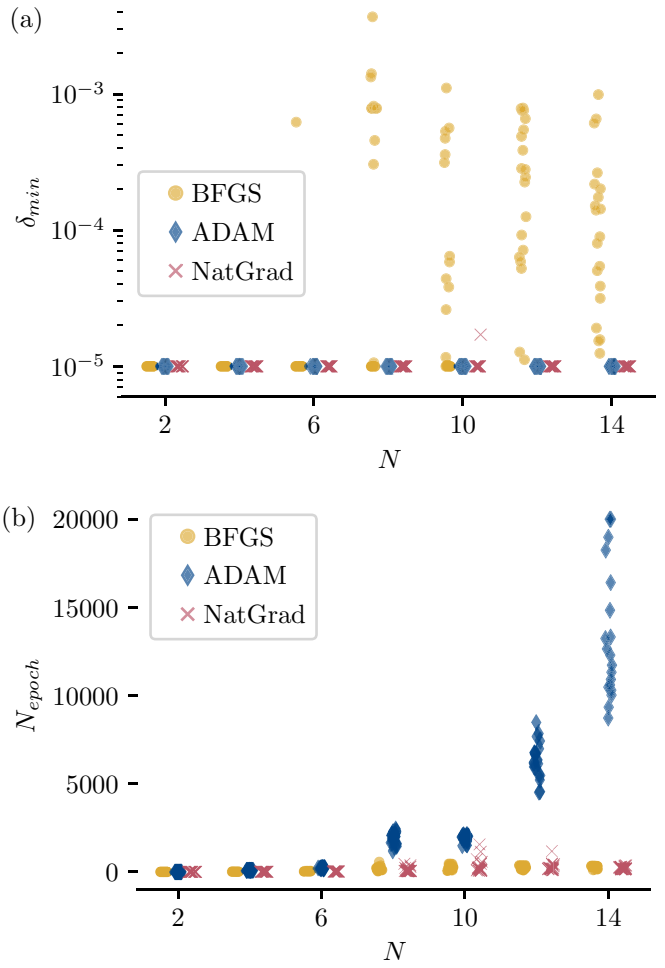

FIG. 7. (a) Minimal achieved precisions and (b) epoch count of the three optimizers and 20 runs on the ansatz in Eq. (23) for the XXZM at depth $p = N$. The circuit contains $n = 3N$ parameters and the learning rates are 0.03 and 0.2 for ADAM and NATGRAD, respectively. The epoch count is displayed on a logarithmic scale for these results.

between the classical machine learning setting and VQE as ADAM thrives on overparametrization in classical cost functions but struggles to exploit additional degrees of freedom in the ansatz circuit. Results of experiments on smaller systems (up to $N = 24$) indicate, that the optimizers behave as described above for stronger overparametrization (up to $n = \frac{4N}{3}$) as well.

D. Results on the Heisenberg model

To complement the study on scaling and overparametrization in the integrable TFIM, we present here numerical results on the XXZM with the ansatz discussed in detail in Sec. II D 2. The performance of the three optimizers, again initialized at 20 distinct points close to 0, is shown in Fig. 7 together with the number of epochs.

The BFGS optimizer shows problems in convergence for increasing circuit sizes but there seems to be a continuous transition between local and global minimum precisions such that a success rate can not be defined as easily. The low number of epochs to convergence required by BFGS—for both global minima and low-quality results—makes it the cheapest optimizer but the unreliable optimization outcomes underline its infeasibility for large-scale VQES.

The behavior of ADAM is comparable to the one observed on the TFIM when using sufficiently small learning rates (cf. Appendix B): while the target precision of 10^{-5} is reached systematically for all problem sizes, the epoch count exhibits a rapid increase. It does not only appear to be exponential but additionally shows abrupt jumps e.g., when increasing the size from 6 to 8 and from 10 to 12 qubits.

The number of variational parameters at which the loss of precision of BFGS and the increase in epochs for ADAM occur is similar to that in the TFIM: The BFGS optimizer starts failing to reach the target precision at $n = 24$ and $n = 22$ for the XXZM and the TFIM, respectively. Likewise the cost of ADAM in Fig. 7 jumps abruptly at $n = 24$ and $n = 36$ and the runs with comparable learning rate for the TFIM show (less clear) transitions at $n = 26$ and $n = 30$ (see Fig. 9). The Hilbert space dimension however clearly differs at the transition points. While it is intuitively clear that the main influence should be due to the properties of the parameter space, the physical system size in general could affect the performance, too.

The reliable performance of NatGrad was confirmed for the XXZM, failing to converge globally only once for 10 qubits. These high quality results were obtained by modifying the regularization constant ε_T from 10^{-4} to 10^{-3} and setting the learning rate $\eta = 0.2$. This improvement is based on the observation that runs with a smaller learning rate and regularization were interrupted prematurely due to slow learning. We would like to emphasize that the presented choice is not the result of an extensive hyperparameter optimization but the best of a few tested settings, out of which only two were benchmarked on the full set of optimization tasks. The epoch count for the NatGrad optimizer shows more fluctuations than before but is much smaller than for ADAM. For 14 qubits, ADAM takes between 8721 and 20 000 epochs, while the count for NatGrad ranges from 132 to 361.

For a fair comparison of the optimizer cost, we again look at the estimated quantum computing run times t_Q in Fig. 8. Due to the small epoch count and comparably low cost per epoch, the unsuccessful BFGS runs clearly are cheapest. More interestingly, the difference in the number of epochs between NatGrad and ADAM discussed above equalizes the overhead in the cost per epoch of NatGrad due to the Fubini-Study matrix computation. This trend was indicated in the minimal QAOA circuit results for the TFIM (cf. Fig. 3) but distorted by the finite epoch budget.

The results for the Heisenberg model overall confirm the observations on the TFIM: NatGrad exhibits a favourable scaling in the epoch count which remedies the increased effort per epoch that is required to determine the Fubini matrix as compared to ADAM. Meanwhile, ADAM shows unpredictable behavior in its optimization cost but consistently attains the target precision whereas BFGS suffers from high dimensional search spaces, rendering it a cheap but unreliable method for VQES. We emphasize that the relative cost for measuring the Fubini-Study matrix in NatGrad is smaller for Hamiltonians with many terms as discussed in Sec. II C 1. This means that

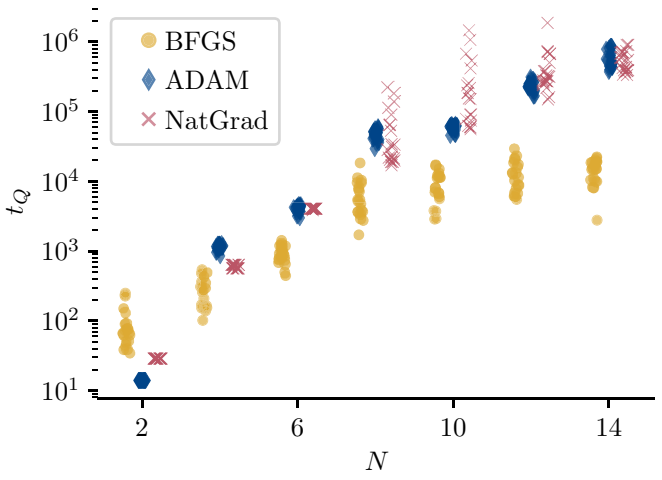

FIG. 8. Estimated quantum run times t_Q for the optimization tasks in Fig. 7 based on Table I and the same assumptions as in Fig. 3 and 5.

the quantum run times for NatGrad can be significantly better for such systems and the relative scaling of t_Q is comparable to the drastic distinction seen for N_{epoch} in Fig. 7 as the cost per epoch approach the ones for any gradient-based method.

IV. CONCLUSION

Our first main result shows that the BFGS optimizer, while quick and reliably for small systems, has an increased chance getting stuck in local minima already in medium sized VQES that are comparable to present day and near future NISQ devices. This may be surprising as it has access to nonlocal information due to its line search subroutine. We suspect that this aspect of the algorithm becomes less helpful for finding a global minimum because of its sparsity in high-dimensional parameter spaces.

The ADAM optimizer on the other hand is able to find global minima also in larger parameter spaces (up to 42) for suitably small learning rates but this comes at the cost of a quickly increasing number of epochs to complete the optimization. In particular we observed two effects of the learning rate η on the run time of ADAM: On the one hand, there is a threshold size of the parameter space that depends on η above which the epoch count rapidly increases, which means that a small enough value of the learning rate is essential to avoid extremely long run times. On the other hand, the optimization duration for sizes below the threshold is significantly increased when reducing η making it undesirable to choose the learning rate smaller than strictly necessary. It thus appears that tedious hyperparameter tuning is necessary to balance these two effects.

The NatGrad optimizer recently proposed for VQE shows very reliable convergence to a global minimum for all tested system sizes within fewer epochs but at high cost per epoch. The problem of jumping into barren plateaus even after a suitable initialization can be fixed via Tikhonov regularization, which can be tuned with a continuous parameter to gradually trade the benefit from the information geometry for stability. This makes the algorithm a promising, although more expensive, candidate for the optimization of future VQES. The increased cost for determining the Fubini matrix at each step have a particularly strong effect on the estimated quantum run time for spin chain systems, for other systems with more favourable scaling NatGrad might not only be more reliable but additionally exhibit lower cost.

Our second main experiment treats overparametrization in VQE ansatz classes including an example of additional rotation gates that break the symmetry of the Hamiltonian as well as symmetry-preserving overparametrization. The BFGS optimizer fails to find a global minimum in some instances even for very small systems and in general exhibits a strongly fluctuating performance which decreases considerably with the number of additional gate layers.

Also ADAM showed strong susceptibility to the additional degrees of freedom. Beyond the implications on applications, this is interesting because overparametrization is heavily used in machine learning to make the cost function tractable for optimizers like ADAM and we therefore appear to observe a fundamental difference between classical machine learning and VQES.

Finally, NatGrad showed some failed optimization runs for selected system sizes as well but mostly remained successful even for multiple additional gate layers. It therefore rewards its increased cost per epoch with higher success rates and is the only tested optimization strategy that showed resilience to both big search spaces and local minima caused by overparametrization.

We therefore conclude that overparametrization which extends the effective Hilbert space is a serious problem for standard optimizers and even NatGrad as most resilient algorithm is disturbed by this issue. The simulation cost restricted the maximal system size for this second experiment but there is no reason to assume that a stronger overparametrization with more symmetry breaking layers would resolve these problems. This implies difficulties for adaptive ansatz techniques because standard rating strategies cannot detect this property and the gate set therefore has to be minimal in order to prevent this type of overparametrization.

For overparametrized ansatz classes *within* the symmetry sector of the TFIM, all optimizers behave similar to the minimal parametrization or show slightly worse convergence. This demonstrates that the optimization problem within VQES differs significantly from optimizations in classical machine learning, where overparametrization enhances the performance of ADAM.

In general, one could expect the cost function of VQES to behave differently than those in common machine learning models as the parameters enter in a very nonlinear manner via rotation gates. The restriction of NISQ devices to rather shallow circuits implies much smaller numbers of variational parameters than in machine learning and therefore NatGrad can be considered a viable option for VQE optimization while using second order resources.

The extension of our analysis to the XXZM confirmed the problems of the BFGS optimizer with big search spaces and the rapid run time growth for ADAM. NatGrad performed reliably on the XXZM as well and the reduced number of epochs compensates the cost per epoch such that the cost of the convergent optimizers ADAM and NatGrad are similar for

the tested system sizes. Additional experiments are in order to show further generalization to nonintegrable models, which would imply that a full VQE optimization on big systems in general is most affordable using NatGrad.

Our investigations have shown that NatGrad might enable VQES to solve more complex and bigger problems as it performs well on a test model with challenges representative of those in potential future applications of VQES. If reliability is more important than minimizing the quantum run time of a single optimization run we recommend NatGrad as optimizer of choice. Alternatively, whenever the Hamiltonian of interest contains many terms and thus is expensive to measure, the relative additional cost of obtaining the Fubini matrix become small (see Sec. II C 1) and the high reliability and low number of required epochs of NatGrad again make it the best method.

The observed differences between classical machine learning and VQES show that insights and heuristics from the former do not necessarily apply in the latter case and demonstrate the importance of understanding the optimization problem in VQES and the properties of the optimization algorithms.

ACKNOWLEDGMENTS

We would like to thank Chae-Yeun Park, David Gross, Gian-Luca Anselmetti, and Thorben Frank for helpful discussions. Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy-Cluster of Excellence Matter and Light for Quantum Computing (ML4Q) EXC 2004/1-390534769. The authors would like to thank Covestro Deutschland AG, Kaiser Wilhelm Allee 60, 51373 Leverkusen, for the support with computational resources. The work was conducted while all three authors were affiliated with the Institute for Theoretical Physics of the University of Cologne.

APPENDIX A: EXACT SOLUTION OF THE TFIM

Here we derive the analytic solution of the TFIM by mapping it to noninteracting fermions, also see Ref. [15]. We start with the linear combinations $a_k \coloneqq \frac{1}{2}(Z^{(k)} + iY^{(k)})$, which fulfill

$$X^{(k)} = 2a_k^\dagger a_k - 1 \ , \ Z^{(k)} = a_k^\dagger + a_k \tag{A1}$$

and map them to the operators

$$b_k \coloneqq \prod_{l=1}^{k-1} \mathcal{N}_l \, a_k \ , \ \mathcal{N}_l \coloneqq \exp[i\pi a_l^\dagger a_l], \tag{A2}$$

which satisfy fermionic anticommutation relations:

$$\{b_k^\dagger, b_l\} = \delta_{kl} \ , \ \{b_k, b_l\} = \{b_k^\dagger, b_l^\dagger\} = 0. \tag{A3}$$

For the transformation of the Hamiltonians H_S and H_B, which comprise both the TFIM Hamiltonian and the generators for the unitaries in the QAOA ansatz, note that

$$\mathcal{N}_l^2 = \mathbb{1} \ , \ \mathcal{N}_l^\dagger = \mathcal{N}_l = \mathcal{N}_l^{-1}, \tag{A4}$$

$$\mathcal{N}_k b_k = b_k \ , \ \mathcal{N}_k b_k^\dagger = -b_k^\dagger. \tag{A5}$$

Using Eq. (A1) and the above properties the transformed Hamiltonians read

$$H_S = -\left[\sum_{k=1}^{N-1}(b_k^\dagger - b_k)b_{k+1}^\dagger - (b_N^\dagger - b_N)b_1^\dagger \ \mathcal{G}\right] + \text{H.c.}, \tag{A6}$$

$$H_B = -t\sum_{k=1}^{N} 2b_k^\dagger b_k - 1, \tag{A7}$$

where we denote by $\mathcal{G} \coloneqq \prod_{l=1}^{N}\mathcal{N}_l$ the gauge factor in the term generated by the periodic boundary conditions and the nonlocal transformation (A3), which also has a reversed sign. $\mathcal{G}$ interacts with the initial state of the QAOA ansatz $|\bar{\psi}\rangle$ and the Hamiltonian terms in the following way:

$$\mathcal{G}|\bar{\psi}\rangle = \exp\left[\frac{i\pi}{2}\left(-\frac{1}{t}H_B + N\right)\right]|+\rangle^{\otimes N} = e^{i\pi N}|\bar{\psi}\rangle, \tag{A8}$$

$$[\mathcal{G}, H_B] = 0 = [\mathcal{G}, H_S], \tag{A9}$$

where we used the ground state energy $-tN$ of H_B and Eq. (A5). This means that the reversed sign is canceled for odd N. Therefore we introduce an additional phase via the transformation

$$c_k \coloneqq e^{ik\nu}b_k \ , \ \nu \coloneqq \begin{cases} \pi/N & \text{for } N \text{ even} \\ 0 & \text{for } N \text{ odd} \end{cases}, \tag{A10}$$

$$H_S = -\left[\sum_{k=1}^{N} e^{i\nu}\left(c_k^\dagger e^{i2k\nu} - c_k\right)c_{k+1}^\dagger\right] + \text{H.c.}\,, \tag{A11}$$

$$H_B = -t\sum_{k=1}^{N} 2c_k^\dagger c_k - 1, \tag{A12}$$

where we defined ν such that the result holds for both odd and even N. The last mapping we perform is a Fourier transformation with shifted momenta:

$$d_q \coloneqq \frac{1}{\sqrt{N}}\sum_{k=1}^{N} e^{2\pi i(q-1)k/N}c_k \ , \tag{A13}$$

$$H_S = -\left[\sum_{q=1}^{N} e^{-i\alpha_q}d_q^\dagger d_{-q}^\dagger - e^{i\alpha_q}d_q d_q^\dagger\right] + \text{H.c.}\,, \tag{A14}$$

$$H_B = t\sum_{q=1}^{N} 2d_q^\dagger d_q - 1 \tag{A15}$$

with mode-dependent angles and relabeled Fourier modes

$$\alpha_q \coloneqq \begin{cases} (2q-1)\pi/N & \text{for } N \text{ even} \\ 2q\pi/N & \text{for } N \text{ odd} \end{cases}, \tag{A16}$$

$$d_{-q} \coloneqq \begin{cases} d_{N+1-q} & \text{for } N \text{ even} \\ d_{N+2-q} & \text{for } N \text{ odd} \end{cases}. \tag{A17}$$

We finally can split up the sums, recollect the terms corresponding to the pairs $\{d_q, d_{-q}\}$ and rewrite the Hamiltonians

in a fermionic operator basis:

$$H_S = H_S' - 2\Bigg[\sum_{q=1}^{r} \cos\alpha_q\left(d_q^\dagger d_q - d_{-q}d_{-q}^\dagger\right) - i\sin\alpha_q\left(d_q^\dagger d_{-q}^\dagger - d_{-q}d_q\right)\Bigg] \tag{A18}$$

$$= -2\sum_{q=1}^{r}\begin{pmatrix} d_q^\dagger & d_{-q}\end{pmatrix}\begin{pmatrix}\cos\alpha_q & -i\sin\alpha_q \\ i\sin\alpha_q & -\cos\alpha_q\end{pmatrix}\begin{pmatrix} d_q \\ d_{-q}^\dagger\end{pmatrix} + H_S', \tag{A19}$$

$$H_B = H_B' - 2t\sum_{q=1}^{r} d_q^\dagger d_q - d_{-q}d_{-q}^\dagger \tag{A20}$$

$$= H_B' - 2t\sum_{q=1}^{r}\begin{pmatrix} d_q^\dagger & d_{-q}\end{pmatrix}\begin{pmatrix}1 & 0 \\ 0 & -1\end{pmatrix}\begin{pmatrix} d_q \\ d_{-q}^\dagger\end{pmatrix}, \tag{A21}$$

where $H_B' = H_S' = 0$ and $H_B/t = H_S' = -1$ for even and odd N, respectively, using $d_1^\dagger d_1|\bar{\psi}\rangle = 1$ and $d_1 d_1^\dagger|\bar{\psi}\rangle = 0$ for the odd case.

In this shape, the simple structure of the model becomes apparent as we identify $r = \lfloor\frac{N}{2}\rfloor$ pairs of fermionic modes in momentum space which interact within but not between the pairs. The Hamiltonian can thus be written as a direct sum

$$H_{\text{TFI}} = -2\bigoplus_{q=1}^{r}(t+\cos\alpha_q)Z + \sin\alpha_q Y - (1+t)(N-2r). \tag{A22}$$

Due to the fact that H_B and H_S not only constitute H_{TFI} but also generate the (modified) QAOA ansatz, the simulation of the circuit can be carried out on a $2r$-dimensional space that decomposes into the direct sum above. On the Bloch spheres of the free fermions, the two time evolution operators $e^{-i\vartheta H_S}$ and $e^{-i\varphi H_B}$ correspond to rotations about the individual axes $e_q = (0, \sin\alpha_q, \cos\alpha_q)$ and the z axis, respectively. Furthermore we can manually solve for the ground state of the TFIM by computing the ground state in each subspace individually:

$$E_0 = E' - 2\sum_{q=1}^{r} E_q,\ |\psi_0\rangle = \bigoplus_{q=1}^{r}|\psi_{q,0}\rangle, \tag{A23}$$

$$E_q = \sqrt{1+t^2+2t\cos\alpha_q}, \tag{A24}$$

$$|\psi_{q,0}\rangle = \frac{1}{\sqrt{2E_q(E_q-\cos\alpha_q-t)}}\begin{pmatrix} i\sin\alpha_q \\ E_q-\cos\alpha_q-t\end{pmatrix} \tag{A25}$$

where E' is the eigenvalue of $H_B' + H_S'$.

APPENDIX B: LEARNING RATE INFLUENCE ON PERFORMANCE OF ADAM

In order to evaluate the systematically large optimization durations of the ADAM optimizer for the QAOA circuit of the TFIM, we tested it at multiple learning rates from the interval $[0.005, 0.1]$ observing a major influence on the run time, see Fig. 9. For a given learning rate η, the required number

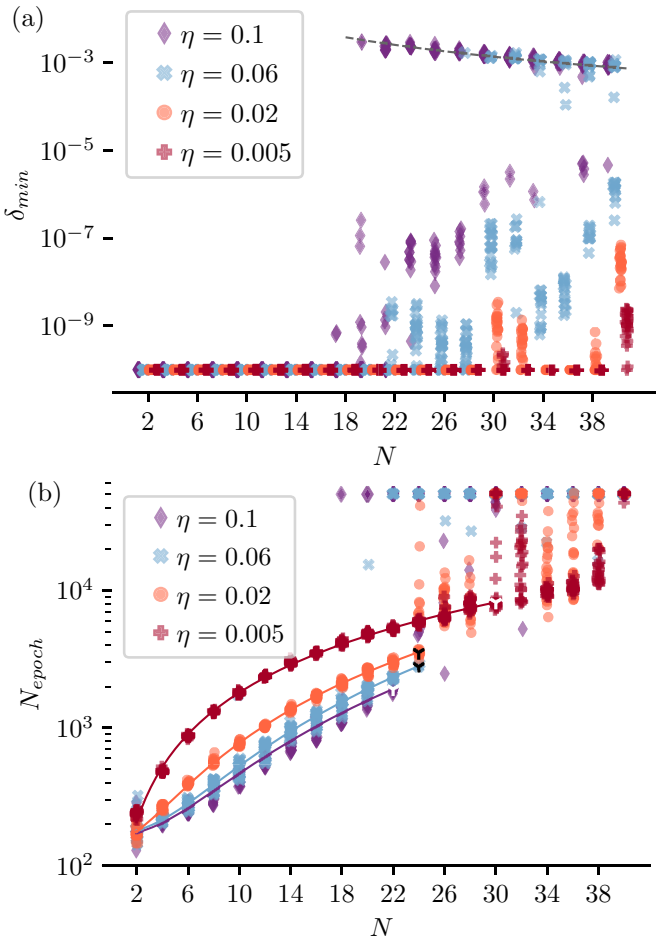

FIG. 9. Minimal attained relative errors δ_{min} and epoch count N_{epoch} for the ADAM optimizer initialized at 20 distinct points close to zero and with different learning rates η. (a) The threshold size beyond which ADAM fails can be shifted by reducing η, delaying local convergence and output of an excited state (dashed line) to bigger systems. (b) The shown fits are based on *filtered* data in order to determine the apparent scaling for small system sizes and thus do *not* aim at describing the entire data. The biggest system size partially included in the fit is marked. For the shown learning rates in descending order, we obtain the exponents 2.3, 2.3, 1.9, and 1.4 but prefactors 1.8, 1.9, 7.3, and 74.7.

of epochs grows polynomially with the system size up to a size N^* above which ADAM takes much longer, exceeding the budget of 5×10^4 epochs. In this second phase, we find the optimizer to require excessively many epochs both when succeeding and when getting stuck in a local minimum (see, e.g., $\eta = 0.06$), which prevents us from systematically distinguishing the two cases before convergence. The observed transition point $N^*(\eta)$ can be shifted towards bigger system sizes by decreasing the learning rate, i.e., $N^*(\eta)$ is monotonically decreasing. Meanwhile, reducing η increases the epoch count significantly for smaller system sizes without disrupting the convergence as is expected for well-behaved systems. Even though the scaling exponent is smaller for lower learning rates the optimization requires more epochs which is due to a large prefactor, increasing the cost for all system sizes before the jump. The observed dependencies of the run time on η

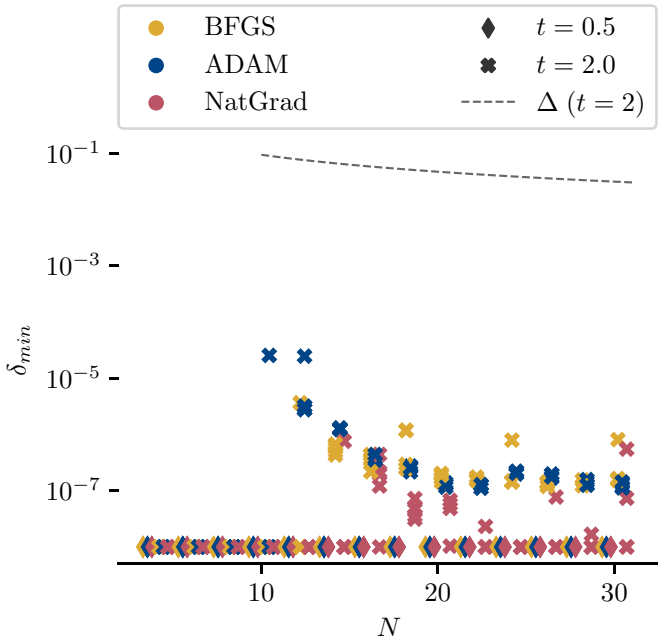

FIG. 10. Minimal attained relative errors δ_{min} for the TFIM at sub- ($t = 0.5$) and supercritical ($t = 2$) transverse fields and energy gap Δ of the supercritical model.

result in a system size dependent optimal learning rate which trades off the systematically increased epoch counts for small η against the position of the jump in optimization duration. This demonstrates that heuristics for ADAM are needed in order to achieve systematic global optimization and that the required number of optimization steps can be unpredictably large depending on the hyperparameters.

APPENDIX C: NONCRITICAL TFIM

In this section, we present numerical results for the optimization of the QAOA ansatz for the noncritical TFIM and demonstrate why the critical transverse field strength was chosen for the main investigations. As these experiments are performed for exploratory purposes, the maximal system size is reduced to 30, we choose one field strength for each phase and we sample 5 (instead of 20) initial parameter positions. As shown in Fig. 10, all optimizers succeed in finding global minima to the required precision for the subcritical transverse field strength but the supercritical model is harder to solve than both the sub- and the critical model. Convergence to local minima is observed at $t = 2$ for systems as small as 10 spins. However, we found that the error caused by convergence to local minima is three to five orders of magnitude smaller than the gap of the model, whereas optimizations for the critical model show errors very close to the gap (see Fig. 2). This improved separation of successful and failed optimization runs in the critical model makes it more suitable for the optimizer comparison.

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

[2] Y. Cao, J. Romero, J. P. Olson, M. Degroote, P. D. Johnson, M. Kieferová, I. D. Kivlichan, T. Menke, B. Peropadre, N. P. Sawaya *et al.*, Quantum chemistry in the age of quantum computing, Chem. Rev. **119**, 10856 (2019).

[3] A. Smith, M. S. Kim, F. Pollmann, and J. Knolle, Simulating quantum many-body dynamics on a current digital quantum computer, npj Quantum Inf. **5**, 106 (2019).

[4] L. Zhou, S.-T. Wang, S. Choi, H. Pichler, and M. D. Lukin, Quantum Approximate Optimization Algorithm: Performance, Mechanism, and Implementation on Near-Term Devices, Phys. Rev. X **10**, 021067 (2020).

[5] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv:1412.6980.

[6] A. Karpathy, A peek at trends in machine learning. Blog post, April (2017).

[7] S. Ruder, An overview of gradient descent optimization algorithms, arXiv:1609.04747.

[8] M. Ostaszewski, E. Grant, and M. Benedetti, Quantum circuit structure learning, arXiv:1905.09692.

[9] C. G. Broyden, The convergence of a class of double-rank minimization algorithms 1. General considerations, IMA J. Appl. Math. **6**, 76 (1970)

[10] R. Fletcher, A new approach to variable metric algorithms, Comput. J. **13**, 317 (1970) .

[11] D. Goldfarb, A family of variable-metric methods derived by variational means, Math. Comput. **24**, 23 (1970).

[12] D. F. Shanno, Conditioning of quasi-Newton methods for function minimization, Math. Comput. **24**, 647 (1970).

[13] G. Giacomo Guerreschi and M. Smelyanskiy, Practical optimization for hybrid quantum-classical algorithms, arXiv:1701.01450.

[14] G. B. Mbeng, R. Fazio, and G. Santoro, Quantum annealing: A journey through digitalization, control, and hybrid quantum variational schemes, arXiv:1906.08948v3.

[15] Z. Wang, N. C. Rubin, J. M. Dominy, and E. G. Rieffel, *XY* mixers: Analytical and numerical results for the quantum alternating operator Ansatz, Phys. Rev. A **101**, 012320 (2020).

[16] 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).

[17] J. Romero, R. Babbush, J. R. McClean, C. Hempel, P. J. Love, and A. Aspuru-Guzik, Strategies for quantum computing molecular energies using the unitary coupled cluster Ansatz, Quantum Sci. Tech. **4**, 014008 (2018).

[18] 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 Inf. **6**, 10 (2020).

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

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

[21] A. Harrow and J. Napp, Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms, arXiv:1901.05374.

[22] J. Martens and I. Sutskever, Training deep and recurrent networks with hessian-free optimization, in *Neural Networks: Tricks of the Trade* (Springer, Berlin, Heidelberg, 2012), pp. 479–535.

[23] R. Livni, S. Shalev-Shwartz, and O. Shamir, On the computational efficiency of training neural networks, in *Advances in Neural Information Processing Systems* (Curran Associates, Inc., New York, NY, 2014), pp. 855–863.

[24] D. Li, T. Ding, and R. Sun, Over-parameterized deep neural networks have no strict local minima for any continuous activations, arXiv:1812.11039.

[25] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, Barren plateaus in quantum neural network training landscapes, Nat. Commun. **9**, 4812 (2018).

[26] C.-Y. Park and M. J. Kastoryano, Geometry of learning neural quantum states, Phys. Rev. Res. **2**, 023232 (2020).

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

[28] H. Lun Tang, E. Barnes, H. R. Grimsley, N. J. Mayhall, and S. E. Economou, qubit-ADAPT-VQE: An adaptive algorithm for constructing hardware-efficient Ansatze on a quantum processor, arXiv:1911.10205.

[29] A. G. Rattew, S. Hu, M. Pistoia, R. Chen, and S. Wood, A domain-agnostic, noise-resistant, hardware-efficient evolutionary variational quantum eigensolver, arXiv:1910.09694.

[30] M. E. S. Morales, J. Biamonte, and Z. Zimborás, On the universality of the quantum approximate optimization algorithm, Quantum Info. Process. **19**, 291 (2020).

[31] S. Lloyd, Quantum approximate optimization is computationally universal, arXiv:1812.11075.

[32] M. B. Hastings, Classical and quantum bounded depth approximation algorithms, arXiv:1905.07047.

[33] E. Farhi and A. W. Harrow, Quantum Supremacy through the Quantum Approximate Optimization Algorithm, arXiv:1602.07674.

[34] Z. Wang, S. Hadfield, Z. Jiang, and E. G. Rieffel, Quantum approximation optimization algorithm for MaxCut: A fermionic view, Phys. Rev. A **97**, 022304 (2018).

[35] W. W. Ho and T. H. Hsieh, Efficient variational simulation of nontrivial quantum states, SciPost Physics **6**, 029 (2019).

[36] M. Y. Niu, S. Lu, and I. Chuang, Optimizing QAOA: Success probability and runtime dependence on circuit depth, arXiv:1905.12134.

[37] V. Akshay, H. Philathong, M. E. Morales, and J. D. Biamonte, Reachability Deficits in Quantum Approximate Optimization, Phys. Rev. Lett. **124**, 090504 (2020).

[38] S. Hadfield, Z. Wang, B. O'Gorman, E. G. Rieffel, D. Venturelli, and R. Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator Ansatz, Algorithms **12**, 34 (2019).

[39] J. Duchi, E. Hazan, and Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, J. Machine Learning Res. **12**, 61 (2011).

[40] G. Hinton, N. Srivastava, and K. Swersky, Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning (2012).

[41] F. Becca and S. Sorella, *Quantum Monte Carlo Approaches for Correlated Systems* (Cambridge University Press, Cambridge, 2017).

[42] G. Carleo and M. Troyer, Solving the quantum many-body problem with artificial neural networks, Science **355**, 602 (2017).

[43] Y. Li and S. C. Benjamin, Efficient Variational Quantum Simulator Incorporating Active Error Minimization, Phys. Rev. X **7**, 021050 (2017).

[44] P.-L. Dallaire-Demers, J. Romero, L. Veis, S. Sim, and A. Aspuru-Guzik, Low-depth circuit Ansatz for preparing correlated fermionic states on a quantum computer, Quantum Science Technology **4**, 045005 (2019).

[45] S. McArdle, T. Jones, S. Endo, Y. Li, S. C. Benjamin, and X. Yuan, Variational Ansatz-based quantum simulation of imaginary time evolution, npj Quantum Inf. **5**, 75 (2019).

[46] B. Koczor and S. C. Benjamin, Quantum natural gradient generalised to nonunitary circuits, arXiv:1912.08660.

[47] N. Yamamoto, On the natural gradient for variational quantum eigensolver, 2019, arXiv:1909.05074.

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

[49] J. Li, X. Yang, X. Peng, and C.-P. Sun, Hybrid Quantum-Classical Approach to Quantum Optimal Control, Phys. Rev. Lett. **118**, 150503 (2017).

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

[51] K. Mitarai and K. Fujii, Methodology for replacing indirect measurements with direct measurements, Phys. Rev. Res. **1**, 013006 (2019).

[52] J. C. Spall, Multivariate stochastic approximation using a simultaneous perturbation gradient approximation, IEEE Trans. Autom. Control **37**, 332 (1992).

[53] P. Gokhale and F. T. Chong, $O(N^3)$ measurement cost for variational quantum eigensolver on molecular hamiltonians, arXiv:1908.11857.

[54] T.-C. Yen, V. Verteletskyi, and A. F. Izmaylov, Measuring all compatible operators in one series of single-qubit measurements using unitary transformations, J. Chem. Theory Comput. **16**, 2400 (2020).

[55] E. Lieb, T. Schultz, and D. Mattis, Two soluble models of an antiferromagnetic chain, Ann. Phys. **16**, 407 (1961).

[56] M. Karbach, G. Müller, H. Gould, and J. Tobochnik, Introduction to the bethe Ansatz i, Comput. Phys. **11**, 36 (1997).

[57] M. Karbach, K. Hu, and G. Müller, Introduction to the bethe Ansatz ii, Comput. Phys. **12**, 565 (1998).

[58] D. S. Steiger, T. Häner, and M. Troyer, Projectq: an open source software framework for quantum computing, Quantum **2**, 49 (2018).

[59] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright *et al.*, SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python, Nat. Methods **17**, 261 (2020).

[60] B. van Straaten and B. Koczor, Measurement cost of metric-aware variational quantum algorithms, arXiv:2005.05172.