Simultaneous Perturbation Stochastic Approximation of the Quantum Fisher Information

Julien Gacon,[1,2] Christa Zoufal,[1,3] Giuseppe Carleo,[2] and Stefan Woerner[1,*]

[1] *IBM Quantum, IBM Research – Zurich*
[2] *Institute of Physics, École Polytechnique Fédérale de Lausanne (EPFL)*
[3] *Institute for Theoretical Physics, ETH Zurich*

(Dated: March 18, 2021)

The Quantum Fisher Information (QFI) is a central metric in promising algorithms, such as Quantum Natural Gradient Descent and Variational Quantum Imaginary Time Evolution. Computing the full QFI matrix for a model with d parameters, however, is computationally expensive and generally requires $\mathcal{O}(d^2)$ function evaluations. To remedy these increasing costs in high-dimensional parameter spaces, we propose using simultaneous perturbation stochastic approximation techniques to approximate the QFI at a constant cost. We present the resulting algorithm and successfully apply it to prepare Hamiltonian ground states and train Variational Quantum Boltzmann Machines.

I. INTRODUCTION

Quantum computing promises potential advances in many fields, such as quantum chemistry and physics [1–3], biology [4–6], optimization [7–10], finance [11], and machine learning [12–14]. A computational paradigm particularly suitable for near-term, noisy quantum devices is that of variational quantum algorithms. These consist in considering parameterized quantum circuits and optimize suitable objective functions to solve a variety of tasks [15].

In this context, Variational Quantum Imaginary Time Evolution (VarQITE) techniques received a lot of attention [16–18]. These approaches approximate Quantum Imaginary Time Evolution (QITE) by mapping the quantum state evolution to the evolution of parameters in a parameterized quantum circuit, which serves as an ansatz for the evolved state. The interest in these approaches stems from the fact that QITE is an integral part of many quantum algorithms and can, for instance, be used to find ground states of given Hamiltonians [16], but also to prepare corresponding Gibbs states [17–20]. The former is important, e.g., for quantum chemistry or combinatorial optimization, while the latter finds applications, e.g., in the simulation of many-body systems [21], quantum semi-definite program solvers [22], as well as in evaluating and training Quantum Boltzmann Machines (QBMs) [23]. Variational formulations of time evolution and Variational QBMs (VarQBMs) are very interesting, as they allow us to study these important tasks with near-term quantum computers.

One significant drawback of VarQITE, and of the closely related Quantum Natural Gradient (QNG) descent [24], is that it requires evaluating the Quantum Fisher Information (QFI) at every iteration. This operation has a cost scaling quadratically in the number of circuit parameters, and it is computationally expensive in the presence of many variational parameters. Proposed methods to lower the computational cost to linear complexity are to approximate the QFI by a (block-) diagonal matrix [24] or to avoid computing the QFI by a global projection of the evolved state onto the manifold of a variational ansatz [25].

In this paper, we propose a new approach to approximate VarQITE that only requires a constant number of circuit evaluations per iteration. This is achieved by applying ideas originally developed for the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithm [26]. Our approach is particularly efficient if no precise state evolution is required, as is the case, e.g., for ground state approximation. However, by allowing additional circuit evaluations, our algorithm is able to approximate the exact path of VarQITE arbitrarily close.

The remainder of this paper is structured as follows. Sec. II reviews first- and second-order SPSA and introduces the required concepts. Sec. III adapts second-order SPSA to provide stochastic approximations of the QFI and shows how this can be used to approximate QNG and to train QBMs. Lastly, we show numerical results for both applications in Sec. IV and conclude our paper in Sec. V.

II. SPSA

Minimizing a function's value by selecting optimal input parameters is an ubiquitous problem in computational science, for example, in neural networks or variational quantum algorithms. Using an initial guess for the function's parameters, a highly efficient and widely used family of methods to find the minimum of the function is gradient descent. There, the parameters are updated iteratively by following the direction of the negative gradient of the function with respect to the parameters. Since the negative gradient points to the direction of steepest descent, the idea is that this update rule will eventually lead to a (local) minimum.

Calculating gradients of a function scales linearly in

* wor@zurich.ibm.com

the number of parameters for both, analytic gradients, as well as finite difference approximations. Gradient descent techniques require that gradients must be evaluated at each iteration step, possibly leading to a computational bottleneck when applied to problems with high-dimensional parameter spaces.

Simultaneous perturbation methods provide a solution to this so-called curse of dimensionality. Instead of considering each parameter dimension individually, SPSA uses a stochastic approximation for the gradient by simultaneously perturbing all parameters in a random direction. This results in an unbiased estimator for the gradient if the random directions are sampled from a suitable distribution, for instance, uniformly from $\{1, -1\}$ for each parameter. In addition to the computational efficiency, SPSA is also well suited for optimizing noisy objective functions which usually appear in near-term variational quantum algorithms [26].

Let $f : \mathbb{R}^d \to \mathbb{R}$ be a function with d parameters. For an initial point $\theta^{(0)} \in \mathbb{R}^d$ and a small learning rate $\eta > 0$, the k-th iteration of standard–also called vanilla–gradient descent to minimize f is defined by

$$\theta^{(k+1)} = \theta^{(k)} - \eta \nabla f(\theta^{(k)}). \tag{1}$$

In contrast, the k-th iteration of SPSA first samples a random direction $\Delta^{(k)} \sim \mathcal{U}(\{1, -1\}^d)$ and then approximates the gradient $\nabla f(\theta^{(k)})$ by

$$\nabla f(\theta^{(k)}) \approx \frac{f(\theta^{(k)} + \epsilon\Delta^{(k)}) - f(\theta^{(k)} - \epsilon\Delta^{(k)})}{2\epsilon} \Delta^{(k)}, \tag{2}$$

for some small displacement $\epsilon > 0$. This update uses only two evaluations of f, as opposed to the $\mathcal{O}(d)$ evaluations required for analytic gradients or finite difference approximations.

SPSA can be extended to a second order-method, i.e., to approximate the Hessian in addition to the gradient [27], and we denote this algorithm as 2-SPSA. In second order-methods, the gradient descent update rule is given by

$$\theta^{(k+1)} = \theta^{(k)} - \eta H^{-1}(\theta^{(k)}) \nabla f(\theta^{(k)}), \tag{3}$$

where $H \in \mathbb{R}^{d \times d}$ is the Hessian, or approximated Hessian, of f.

Instead of computing all d^2 entries of H in each iteration, 2-SPSA samples the Hessian using two random directions $\Delta_1^{(k)}$ and $\Delta_2^{(k)}$. The resulting symmetric point-sample is

$$\hat{H}^{(k)} = \frac{\delta f}{2\epsilon^2} \frac{\Delta_1^{(k)} \Delta_2^{(k)T} + \Delta_2^{(k)} \Delta_1^{(k)T}}{2}, \tag{4}$$

where

$$\begin{aligned} \delta f = & f(\theta^{(k)} + \epsilon\Delta_1^{(k)} + \epsilon\Delta_2^{(k)}) \\ & - f(\theta^{(k)} + \epsilon\Delta_1^{(k)}) \\ & - f(\theta^{(k)} - \epsilon\Delta_1^{(k)} + \epsilon\Delta_2^{(k)}) \\ & + f(\theta^{(k)} - \epsilon\Delta_1^{(k)}). \end{aligned} \tag{5}$$

The point sample $\hat{H}^{(k)}$ is then combined with all previous samples in a smoothed estimator

$$\bar{H}^{(k)} = \frac{k}{k+1} \bar{H}^{(k-1)} + \frac{1}{k+1} \hat{H}^{(k)}. \tag{6}$$

To evaluate the gradient, the first-order SPSA technique is used. In total, this update step uses six function evaluations instead of $d^2 + d$ for an analytic second order method, assuming access to the corresponding derivatives.

In Eq. (4), the Hessian estimate is based on the sampling of two random directions and the resulting point-estimate $\hat{H}^{(k)}$ is an unbiased estimator of the full Hessian. By re-sampling additional directions and averaging over all point-estimates, the stochastic approximation of the Hessian can be systematically improved to arbitrary accuracy. In the limit of infinitely many samples, this converges to the exact Hessian [27].

Close enough to a minimum, the Hessian of a function is positive semi-definite and our approximation should reflect this. One possibility to impose this property is to replace $\bar{H}^{(k)}$ with $\sqrt{\bar{H}^{(k)} \bar{H}^{(k)}}$, whose eigenvalues correspond to the absolute values of the eigenvalues of $\bar{H}^{(k)}$. Since we also need to ensure invertibility of the Hessian estimate, we further add a small positive constant $\beta > 0$ to the diagonal and obtain the regularization

$$\sqrt{\bar{H}^{(k)} \bar{H}^{(k)}} + \beta \mathbb{I}, \tag{7}$$

where $\mathbb{I} \in \mathbb{R}^{d \times d}$ denotes the identity matrix. To further mitigate instabilities that may arise from a close-to-singular estimate, a blocking condition can be invoked that only accepts update steps if the loss at the candidate parameters is smaller than the current loss, plus a tolerance, and otherwise re-samples from the Hessian and gradient. If the loss function is not evaluated exactly, such as in case of a sample-based estimation through measurements from a quantum circuit, Ref. [27] suggests to choose the tolerance as twice the standard deviation of the loss.

Moreover, the convergence of SPSA and 2-SPSA is guaranteed if a set of conditions on the noise in the loss function evaluation, the differentiability of the loss function, and the meta-parameters η, ϵ and Δ are satisfied. For details as well as the proof of convergence, we refer to Sec. 3 in Ref. [27].

III. SPSA OF THE QFI

In this section, we present the Quantum Natural SPSA (QN-SPSA) algorithm by extending 2-SPSA to estimate the QFI instead of the Hessian of the loss function. We show how QN-SPSA efficiently approximates the QNG algorithm for preparing Hamiltonian ground states and, then, we leverage this idea to approximate Gibbs state preparation, which we use for the evaluation and training of VarQBMs. These are all algorithms that rely on

accessing the QFI in every iteration and any algorithm with this reliance can be significantly sped up with our approach.

A. Quantum Natural Gradient

Assume a parameterized model $p_\theta : \mathbb{R}^d \to \mathcal{H}$, mapping d parameters to a Hilbert space $\mathcal{H}$, and a loss function $f : \mathcal{H} \to \mathbb{R}$, such that the loss for parameters $\theta \in \mathbb{R}^d$ is given as $f(p_\theta)$. Now, the goal is to find the optimal parameters that minimize the loss, given a starting point $\theta^{(0)} \in \mathbb{R}^d$.

Vanilla gradient descent chooses the update step proportional to the gradient $\nabla f(p_\theta)$ at the current parameter values. A rescaling of all parameters by a fixed factor is, therefore, directly reflected in the magnitude of the gradient descent step. If the learning rate is not properly adjusted, multiplying the parameters by a large constant leads to overshooting the desired values, while multiplying by a small constant can strongly slow down the speed of convergence.

An elegant solution to this rescaling problem is to choose the update step such that the change in the model p_θ instead of the parameters θ remains small. To illustrate the difference to vanilla gradient descent, it helps to rewrite the update rule from Eq. (1) as

$$\theta^{(k+1)} = \arg\min_{\theta\in\mathbb{R}^d} \langle \theta - \theta^{(k)}, \nabla f(\theta^{(k)})\rangle + \frac{1}{2\eta}||\theta-\theta^{(k)}||_2^2. \quad (8)$$

In this representation, we clearly see that vanilla gradient descent chooses the parameter update aligned with the direction of the gradient and determines the size of the update step by limiting the change compared to the previous parameters values, $\theta^{(k)}$.

We now replace the Euclidean norm $||\cdot||_2$ by $||\cdot||_{g(\theta)} = \langle \cdot, g(\theta)\cdot\rangle$ where $g(\theta) \in \mathbb{R}^{d\times d}$ denotes the metric tensor induced by the model p_θ [24, 28]. In doing so, instead of specifying the limit in the update step by using changes in the *parameter space*, we now consider changes in the *model space*. The update rule changes to

$$\theta^{(k+1)} = \arg\min_{\theta\in\mathbb{R}^d} \langle \theta - \theta^{(k)}, \nabla f(\theta^{(k)})\rangle + \frac{1}{2\eta}||\theta-\theta^{(k)}||^2_{g(\theta^{(k)})}, \quad (9)$$

which can be solved exactly by

$$\theta^{(k+1)} = \theta^{(k)} - \eta g^{-1}(\theta^{(k)})\nabla f(\theta^{(k)}). \quad (10)$$

This is known as Natural Gradient Descent [28].

We now consider the case where p_θ is given by a parameterized quantum circuit. Let $|\psi_\theta\rangle$ describe a parameterized pure quantum state on n qubits for classical parameters $\theta \in \mathbb{R}^d$. Then, the metric tensor $g(\theta) \in \mathbb{R}^{d\times d}$ is given by the QFI with elements [24]

$$g_{ij}(\theta) = \mathrm{Re}\left\{ \left\langle \frac{\partial\psi_\theta}{\partial\theta_i} \middle| \frac{\partial\psi_\theta}{\partial\theta_j} \right\rangle - \left\langle \frac{\partial\psi_\theta}{\partial\theta_i} \middle| \psi_\theta \right\rangle \left\langle \psi_\theta \middle| \frac{\partial\psi_\theta}{\partial\theta_j} \right\rangle \right\}. \quad (11)$$

The required expectation values can be computed by using a linear combination of unitaries or by parameter shift techniques [29].

Computing g in general requires evaluating $\mathcal{O}(d^2)$ expectation values. By using the 2-SPSA algorithm, we can replace $g(\theta^{(k)})$ by a stochastic approximation $\bar{g}^{(k)}$, requiring only the evaluation of four expectation values, i.e., constant and independent of d.

To exploit 2-SPSA, we use a different representation of the QFI than in Eq. (11), namely, the Hessian of the Fubini-Study metric [24, 30]

$$g_{ij}(\theta) = -\frac{1}{2}\frac{\partial}{\partial\theta_i}\frac{\partial}{\partial\theta_j}|\langle\psi_{\theta'}|\psi_\theta\rangle|^2\Big|_{\theta'=\theta}, \quad (12)$$

see Appendix B for more details. We generalize 2-SPSA for the Hessian of a metric instead of a function by applying perturbations only to the second argument of the metric and keeping the first argument fixed. Concretely, Eqs. (4) and (5) change to

$$\hat{g}^{(k)} = -\frac{1}{2}\frac{\delta F}{2\epsilon^2}\frac{\Delta_1^{(k)}\Delta_2^{(k)T} + \Delta_2^{(k)}\Delta_1^{(k)T}}{2}, \quad (13)$$

where

$$\begin{aligned}\delta F = &F(\theta^{(k)}, \theta^{(k)} + \epsilon\Delta_1^{(k)} + \epsilon\Delta_2^{(k)})\\ &- F(\theta^{(k)}, \theta^{(k)} + \epsilon\Delta_1^{(k)})\\ &- F(\theta^{(k)}, \theta^{(k)} - \epsilon\Delta_1^{(k)} + \epsilon\Delta_2^{(k)})\\ &+ F(\theta^{(k)}, \theta^{(k)} - \epsilon\Delta_1^{(k)}),\end{aligned} \quad (14)$$

and $F(\psi,\phi) = |\langle\psi|\phi\rangle|^2$. The smoothing of the point-estimates $\hat{g}^{(k)}$ into $\bar{g}^{(k)}$ and the technique to ensure the estimate is positive semi-definite remains the same as in the previous section.

Evaluating the Fubini-Study metric requires calculation of the absolute value of the overlap of $|\psi_\theta\rangle$ with parameter values θ and slightly shifted parameters $\theta + \epsilon\Delta$. The overlap of two quantum states can be estimated using the swap test [31], where both states are prepared in separate qubit registers. If the states are given by $|\psi_\theta\rangle = U(\theta)|0\rangle$ for a parameterized unitary U, and we only need the absolute value of the overlap, we can prepare $U^\dagger(\theta+\epsilon\Delta)U(\theta)|0\rangle$ and estimate the probability of measuring $|0\rangle$, which is equal to $|\langle\psi_\theta|\psi_{\theta+\epsilon\Delta}\rangle|^2$. If our state has n qubits and the circuit corresponding to U has depth m, the swap test requires a circuit width of $2n$, but only leads to a depth of $m + \mathcal{O}(1)$ [32]. In contrast, the compute-uncompute method [13] uses circuits of width n, but instead needs twice the depth, $2m$. Depending on the unitary and the structure of the available hardware, either method can be advantageous.

B. Quantum Boltzmann Machines

QBMs are energy-based machine learning models that encode information in the parameters ω of a parameterized n-qubit Hamiltonian $\hat{O}_\omega$ [23]. This Hamiltonian

defines a Gibbs state

$$\rho^{\text{Gibbs}}(\hat{O}_\omega) = \frac{e^{-\hat{O}_\omega/(\mathrm{k_B}T)}}{Z}, \tag{15}$$

with the Boltzmann constant $\mathrm{k_B}$, system temperature T and partition function $Z = \mathrm{Tr}[e^{-\hat{O}_\omega/(\mathrm{k_B}T)}]$. Depending on the construction of $\hat{O}_\omega$ and the choice of the loss function, QBMs can be used for various machine learning tasks, such as generative or discriminative learning [18]. Throughout the training, the Gibbs state $\rho^{\text{Gibbs}}(\hat{O}_\omega)$ is repeatedly prepared and measured for different parameter values ω. The obtained samples are then used to evaluate the loss function.

Preparing quantum thermal states, such as Gibbs states, is difficult and several techniques have been proposed to solve this task [33–40]. In the following, we consider approximate construction of the Gibbs state using VarQITE, which follows the time evolution of a maximally mixed state under $\hat{O}_\omega$ for the time $(2\mathrm{k_B}T)^{-1}$.

To construct the initial maximally mixed state on n qubits in a quantum circuit, we use n additional *environmental* qubits. Each of the n qubits encoding the Gibbs state is assigned to one environmental qubit and each qubit pair is prepared in the Bell state $(|00\rangle+|11\rangle)/\sqrt{2}$. If the environmental qubits are now traced out, the remaining n qubits are in a maximally mixed state. Moreover, the Hamiltonian is extended such that it acts trivially on the environmental qubits

$$\hat{O}_\omega \to \hat{O}_\omega \otimes I^{\otimes n}.$$

In the variational approach, the state of the $2n$ qubits is represented by a parameterized quantum circuit with parameters θ. For VarQBMs, the initial parameter values $\theta^{(0)}$ must be chosen such that each qubit pair is in a Bell state [18].

With the correct initial state prepared, we can now apply VarQITE. The update rule for the circuit parameters are governed by McLachlan's variational principle [41]

$$g_{ij}(\theta^{(t)})\frac{\partial\theta_j^{(t)}}{\partial t} = -\mathrm{Re}\left\{\left\langle \frac{\partial\psi_{\theta^{(t)}}}{\partial\theta_i}\right| \hat{O}_\omega \left|\psi_{\theta^{(t)}}\right\rangle\right\}, \tag{16}$$

where g is the QFI from Eq. (11). We obtain the time-evolution of the parameters by integrating with an arbitrary ODE solver, such as an explicit Euler scheme

$$\theta^{(t+\delta\tau)} = \theta^{(t)} + \delta\tau\frac{\partial\theta^{(t)}}{\partial t}, \tag{17}$$

where $\delta\tau$ is $(2\mathrm{k_B}T)^{-1}$ divided by the number of time steps.

Now, we can apply the same idea as before and replace $g(\theta^{(t)})$ in the linear system of equations given in Eq. (16) with the approximation $\bar{g}^{(t)}$ obtained with QN-SPSA and, thus, significantly reduce the costs associated with Gibbs state preparation, while sacrificing some accuracy.

IV. NUMERICAL RESULTS

In this section, we apply the introduced technique to different problem instances. First, we analyze how QN-SPSA performs compared with analytic QNG for ground state approximation, and, second, we show how VarQBMs perform when the Gibbs states are prepared with VarQITE when the QFI is approximated using 2-SPSA.

A. Quantum Natural Gradient

We compare the performance of the following optimization routines on problems from the related literature to illustrate the speed of convergence, as well as the robustness of convergence with respect to the initial choice of parameters: vanilla gradient descent, SPSA, QNG, and QN-SPSA.

1. *Speed of convergence*

To investigate the speed of convergence, we compare the value of the loss function against the number of iterative parameter updates. As in Ref. [24] we use a Pauli two-design circuit as the parameterized quantum circuit. The circuit consists of an initial layer of $R_\mathrm{Y}(\pi/4)$ gates followed by alternating rotation and entanglement layers. The rotation layers apply uniformly at random selected $R_\mathrm{X}, R_\mathrm{Y}$ or R_Z gates and the entanglement layers apply controlled-Z gates between nearest neighbors. An example of the circuit for five qubits is schematically shown in Fig. 1(a). The loss function is the expectation value with respect to the observable $\hat{O} = Z_1Z_2$, i.e., it only involves the first two qubits.

In this benchmark, we use 11 and 25 qubits and repeat the rotation and entanglement layers three times, leading to 44 and 100 parameters, respectively. The learning rate for all optimizers is chosen to be $\eta = 10^{-2}$ and the displacement for the finite different approximation in the SPSA methods is $\epsilon = 10^{-2}$. The regularization constant for QN-SPSA is set to $\beta = 10^{-3}$. The analytic optimizers are deterministic and only run once, while for both SPSA techniques we show the mean and standard deviation of 25 independent runs. The circuits are implemented in Qiskit [42] and executed using statevector simulation.

Fig. 1(b) shows the loss per iteration for each method for 11 qubits. As previously presented in Ref. [24], the analytic QNG converges faster than vanilla gradient descent. The mean of the standard SPSA algorithm coincides almost exactly with the vanilla gradient descent, which is the expected behavior since SPSA is an unbiased estimator of the gradient. We further observe that QN-SPSA outperforms SPSA and vanilla gradient descent and approaches the loss achieved by the analytic QNG, although with a larger variance than standard SPSA. The mean of the QN-SPSA loss is close to the QNG loss, but it cannot reach it due to the regularization constant $\beta > 0$

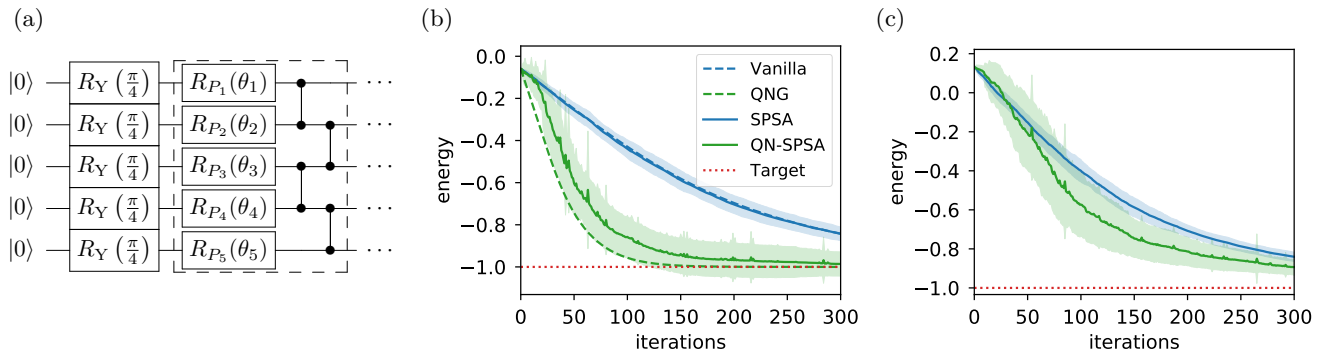

FIG. 1: Investigation of the loss for the Pauli two-design circuit with the observable $\hat{O} = Z_1 Z_2$. (a) The circuit for five qubits, where in each layer the dashed box is repeated. At then end we add a final rotation layer, not shown here. For each rotation gate R_{P_i} the rotation axis is chosen uniformly at random, i.e., $P_i \sim \mathcal{U}(\{X, Y, Z\})$. (b) The loss for 11 qubits and three rotation layers for vanilla gradient descent, QNG and the respective SPSA variants. (c) The same problem scaled to 25 qubits. The analytic optimizers are not shown since they are computationally too costly to evaluate for 100 parameters.

that we add for numerical stability. With this regularization constant, we can interpolate between the natural gradient ($\beta = 0$) and vanilla gradient ($\beta \gg 0$), see Appendix C for more details.

In Fig. 1(c), we repeat the experiment for 25 qubits. With 100 parameters, this example already manifests the advantage of SPSA-based optimizers over analytic gradients. While analytic QNG requires the execution of approximately 1.5 million circuits, QN-SPSA needs only 2100 circuits and still produces very similar results. Due to the large computational cost, the analytic gradients are not presented in the 25 qubit case.

In Appendix D, we compare the convergence of the different optimization schemes with respect to the number of function evaluations and discuss the efficiency and true costs of the different optimizers in more detail.

2. *Region of convergence*

The advantage of natural gradients is not just a faster convergence, which–for problems with a simple loss landscape–might also be achieved with vanilla gradient descent or SPSA if the learning rate is carefully calibrated. But, since QNG (or VarQITE) approximates imaginary time evolution, we have the guarantee that QNG always converges to the ground state if the initial state has a non-zero overlap with it and if a sufficiently powerful ansatz and small stepsize are chosen [16]. Even with an ansatz that cannot follow QITE exactly, QNG and QN-SPSA have superior convergence properties to vanilla gradient descent and SPSA.

To illustrate this, we use the same problem as in Ref. [16] with the ansatz

$$|\psi_\theta\rangle = e^{i\theta_0}(|0\rangle\langle 0| \otimes \mathbb{I} + |1\rangle\langle 1| \otimes R_Y(\theta_2))(R_X(\theta_1) \otimes \mathbb{I})\,|00\rangle\,,$$

prepared by the circuit in Fig. 2(a) and try to minimize the energy with respect to the Hamiltonian

$$\hat{O} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

A variational global phase is added to account for phase differences between the target state and the ansatz, which does not impact the expectation value but can lead to incorrect gradients [17]. We choose different initial points in the same loss landscape and test if vanilla gradient, natural gradient, and the SPSA variants converge to the optimal solution.

In this example, we choose an equidistant grid of 15×15 points in $[-\pi, \pi]^2$ for the initial values of θ_1 and θ_2. The initial global phase is set to zero, i.e., $\theta_0 = 0$. As in Ref. [16] we use constant learning rates of $\eta = 0.886$ for vanilla gradient descent and $\eta = 0.225$ for QNG and all methods do 200 iterations. We consider an optimization run as converged if the final absolute error is below 10^{-4}. For the SPSA methods, we execute 10 optimization runs for each initial point and label the point converged if at least one out of the 10 runs converged. The vanilla methods are deterministic and only run once. Standard SPSA and QN-SPSA use the same learning rates as the corresponding analytic versions, i.e., 0.886 and 0.225, respectively.

The results are shown in Fig. 2(b). The analytic vanilla gradient descent and QNG reproduce the results from Ref. [16]: QNG converges from all sampled points except when one of the initial angles is exactly 0, where at least one gradient component vanishes and the parameter update cannot move towards one of the minima in the corners of the plot. Vanilla gradient additionally fails to converge in a diamond-shaped region around the saddle point $(0, 0)$. The regions of convergence for the SPSA

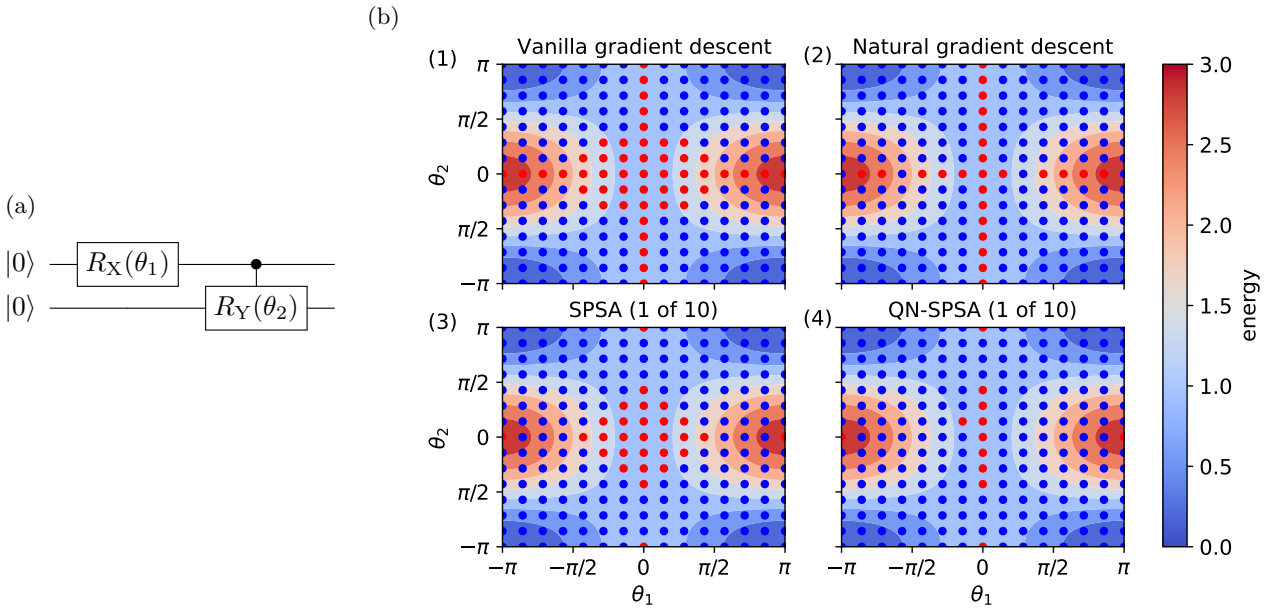

FIG. 2: Convergence tests for a simple loss function. (a) The parameterized quantum circuit used as ansatz. (b) The performance of (1) vanilla gradient descent, (2) QNG, (3) SPSA and (4) QN-SPSA are compared. Each dot marks an initial point, which is marked blue if the method converged and red otherwise. For SPSA and QN-SPSA we repeat the optimization 10 times and consider the point as converged if at least one out of the 10 runs converged to the global optimum. The global phase parameter θ_0 is not shown since it does not affect the loss function value.

methods are similar to the analytic versions, however, they do not suffer as much from vanishing gradient components. Due to the random selection of the direction of the gradient, the stochastic methods have a chance to move to a region where both gradient components are nonzero. To conclude, QN-SPSA outperforms all other methods and impressively converges for almost all initial points.

B. Quantum Boltzmann Machines

In the following, we show that QN-SPSA enables the realization of an approximate VarQBM implementation, where the computational complexity is reduced compared to standard VarQITE-based Gibbs state preparation. We choose to apply the suggested method to a generative learning example, investigated in Ref. [18], where the aim is to prepare a Gibbs state whose sampling probabilities correspond to a given target probability density function.

The learning task in this example is to reproduce the sampling statistics of the Bell state $(|00\rangle + |11\rangle)/\sqrt{2}$. Given a parameterized Hamiltonian $\hat{O}_\omega$, we aim to find a set of parameters, ω, such that the sampling probabilities of the corresponding Gibbs state $\rho^{\text{Gibbs}}(\hat{O}_\theta)$ are close to the target sampling probabilities

$$p^{\text{Bell}} = (0.5, 0, 0, 0.5), \tag{18}$$

measured in the computational basis. The distance to the target distribution is assessed using the relative entropy, which is a measure that describes the distance between two probability distributions. Minimizing the distance between two distributions with respect to the relative entropy is equivalent to minimizing the cross-entropy of the respective probability distributions

$$\ell(\theta) = -\sum_{x=0}^{3} p_x^{\text{Bell}} \log\left(p_x^{\text{Gibbs}}(\hat{O}_\omega)\right),$$

where x corresponds to the computational basis states and

$$p_x^{\text{Gibbs}}(\hat{O}_\omega) = \text{Tr}\left[\rho^{\text{Gibbs}}(\hat{O}_\omega)\,|x\rangle\,\langle x|\right]. \tag{19}$$

This example uses the parameterized Hamiltonian

$$\hat{O}_\omega = \omega_1 Z_1 Z_2 + \omega_2 Z_1 + \omega_3 Z_2.$$

The system temperature is set to $k_B T = 1$ which results in the evolution time $(2k_B T)^{-1} = 0.5$. Furthermore, the approximate QN-SPSA Gibbs state preparation uses forward Euler with 10 equidistant time steps and the ansatz circuit shown in Fig. 3(a). To start the evolution in a maximally mixed state, the initial parameters for the circuit are

$$\forall i \in \{1, \ldots, d\} : \theta_i^{(0)} = \begin{cases} \frac{\pi}{2}, & \text{if } i \in \{9, 10\}, \\ 0, & \text{otherwise.} \end{cases}$$

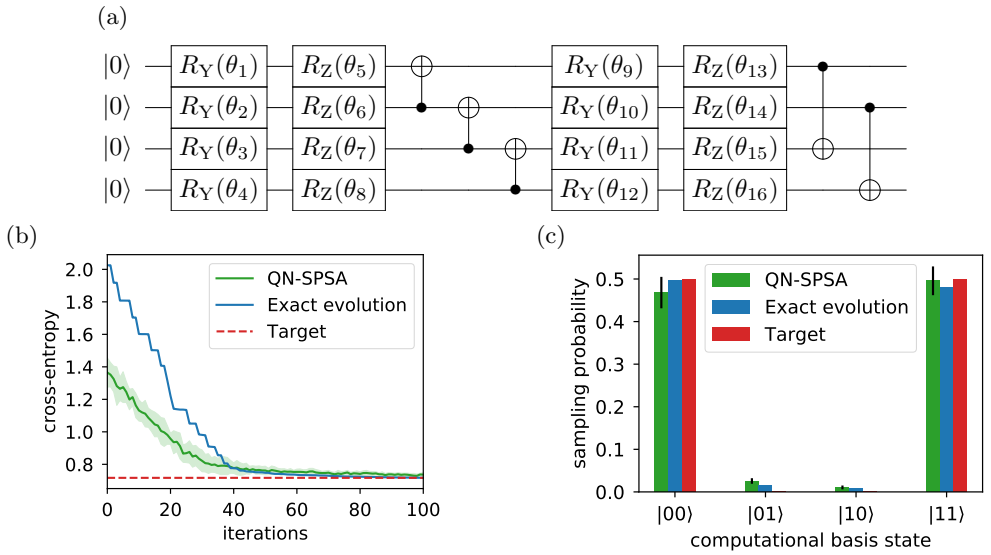

FIG. 3: Generative learning with VarQBMs. (a) The parameterized circuit encoding the Gibbs state. (b) The mean and standard deviation of the QN-SPSA training loss and loss of the exact evolution via matrix exponentiation. The target loss is the final value the training with exact evolution. (c) The final probabilities of the trained Gibbs states. For each of the 10 optimization runs we prepare the final state 10 times to approximate the standard deviation on the final sampling statistics.

The initial parameters for the Hamiltonian $\omega^{(0)}$ are chosen uniformly at random from $[-2, 2]$. We optimize the Hamiltonian parameters ω with 100 iterations of SPSA with a learning rate and perturbation of 0.1.

For QN-SPSA, we choose a perturbation of $\epsilon = 0.01$ and a regularization constant of $\beta = 0.1$. Numerical tests reveal that these experiments perform well with 10 re-samplings of the Hessian per iteration and an averaging of 10 Gibbs state preparation per Hamiltonian parameters. This additional averaging might be necessary due to ill-conditioning of the underlying linear system of equations. In practice, the number of re-samplings and averages can be traded off against a larger standard deviation of the loss function.

Fig. 3(b) shows the development of the loss function of 10 optimization runs of VarQBM with QN-SPSA along with the loss if the exact Gibbs state preparation by means of matrix exponentiation is used. Though they are subject to noise, the VarQBMs reliably converge to the same loss as the optimization with exact evolution. By using more time steps, decreasing the regularization constant β, and using more Hessian re-samplings, the loss of QN-SPSA, we can attempt to track the exact evolution more closely.

The sampling statistics of the final, trained Gibbs states are presented in Fig. 3(c). The output sampling distribution of the Gibbs state prepared with QN-SPSA approximates the Bell distribution well: the target sampling probabilities for the states $|00\rangle$ and $|11\rangle$ are within the standard deviation of the trained state, and–though they are not exactly 0–the states $|01\rangle$ and $|10\rangle$ have only minuscule amplitudes. These non-zero amplitudes are, however, also present in the final state obtained with exact Gibbs state preparation and might, thus, also be a limitation of the chosen system Hamiltonian $\hat{O}_\omega$.

V. CONCLUSION

In this paper, we presented how SPSA can be used to approximate the QFI, whereby we reduce the cost of circuit executions from being quadratic in the number of parameters to constant. We tested the resulting algorithm, QN-SPSA, on ground state preparation and VarQBMs and reproduced existing results from literature where the analytic QFI was used.

In the ground state calculations, we observed that QN-SPSA inherits the fast convergence and robustness of QNG with respect to the initial parameters, while having the computational cost benefits of SPSA, overall, leading to the most effective optimization method of the tested algorithms. With the reduced number of circuits required to evaluate the QNG, our approach enables the simulation and investigation of much larger systems than previously possible.

For generative learning with VarQBMs, we successfully trained a Gibbs state to reproduce a Bell-state target distribution. The speed-up on this small example was not as significant as the ground state calculations due to the required re-sampling and the performance of QN-SPSA for more difficult distributions and more parameters requires further investigation. Another application to consider is

VarQBMs for discriminative learning.

QN-SPSA allows incorporating several adaptive strategies to tailor the optimization routine to the problem or improve convergence. These include using first-order SPSA in the beginning of the optimization to construct a stable QFI estimate, calibrating the perturbation for the finite difference gradients, and dynamically adjusting the number of sampled dimensions according to the rejected or accepted steps.

A caveat of QN-SPSA is the required evaluation of the overlap of two variational ansatze with different parameters to compute the point-estimates of the QFI. Current available algorithms to compute the overlap of two states either require duplication of the circuit depth or of the circuit width. Finding options to reduce this overhead is a relevant open question for further research, particularly, for running this algorithm on real noisy quantum devices.

To conclude, QN-SPSA provides a promising and efficient new method for parameter optimization in variational quantum algorithms. Given the enormous reduction in the number of evaluations needed for many relevant applications compared with the original QNG, this is an important step towards scaling these quantum algorithms to practically relevant problem sizes.

VI. ACKNOWLEDGEMENTS

We would like to thank Ali Javadi-Abhari for insightful discussions throughout the project, in particular regarding the evaluation of state overlaps. We are grateful for Amira Abbas, who generously shared her helpful intuition and knowledge on the Quantum Fisher Information, as well as reviews on this manuscript. Also, we thank Daniel Egger and Panagiotis Barkoutsos for their ideas of challenging loss functions for the optimizers in this work.

Christa Zoufal acknowledges the support of the National Centre of Competence in Research *Quantum Science and Technology* (QSIT).

[1] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Simulated Quantum Computation of Molecular Energies, Science **309**, 1704 (2005).

[2] A. Peruzzo *et al.*, A variational eigenvalue solver on a photonic quantum processor, Nature Communications **5**, 4213 (2014).

[3] M. C. Bañuls *et al.*, Simulating lattice gauge theories within quantum technologies, European Physical Journal D **74**, 165 (2020).

[4] A. Perdomo-Ortiz, N. Dickson, M. Drew-Brook, G. Rose, and A. Aspuru-Guzik, Finding low-energy conformations of lattice protein models by quantum annealing, Scientific Reports **2**, 571 (2012).

[5] M. Fingerhuth, T. Babej, and C. Ing, A quantum alternating operator ansatz with hard and soft constraints for lattice protein folding, Preprint at https://arxiv.org/abs/1810.13411 (2018).

[6] A. Robert, P. K. Barkoutsos, S. Woerner, and I. Tavernelli, Resource-efficient quantum algorithm for protein folding, npj Quantum Information **7**, 38 (2021).

[7] E. Farhi, J. Goldstone, and S. Gutmann, A Quantum Approximate Optimization Algorithm, Preprint at https://arxiv.org/abs/1411.4028 (2014).

[8] A. Gilliam, S. Woerner, and C. Gonciulea, Grover Adaptive Search for Constrained Polynomial Binary Optimization, Preprint at https://arxiv.org/abs/1912.04088 (2019).

[9] L. Braine, D. J. Egger, J. Glick, and S. Woerner, Quantum Algorithms for Mixed Binary Optimization applied to Transaction Settlement, Preprint at https://arxiv.org/abs/1910.05788 (2019).

[10] J. Gacon, C. Zoufal, and S. Woerner, Quantum-enhanced simulation-based optimization, in *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)* (2020) pp. 47–55.

[11] D. J. Egger *et al.*, Quantum computing for finance: State-of-the-art and future prospects, IEEE Transactions on Quantum Engineering **1**, 1 (2020).

[12] J. S. Otterbach *et al.*, Unsupervised Machine Learning on a Hybrid Quantum Computer (2017), preprint at https://arxiv.org/abs/1712.05771.

[13] V. Havlíček *et al.*, Supervised learning with quantum-enhanced feature spaces, Nature (London) **567**, 209 (2019).

[14] M. Schuld, Quantum machine learning models are kernel methods, Preprint at https://arxiv.org/abs/2101.11020 (2021).

[15] N. Moll *et al.*, Quantum optimization using variational algorithms on near-term quantum devices, Quantum Science and Technology **3**, 030503 (2018).

[16] S. McArdle *et al.*, Variational ansatz-based quantum simulation of imaginary time evolution, npj Quantum Information **5**, 10.1038/s41534-019-0187-2 (2019).

[17] X. Yuan, S. Endo, Q. Zhao, Y. Li, and S. C. Benjamin, Theory of variational quantum simulation, Quantum **3**, 191 (2019).

[18] C. Zoufal, A. Lucchi, and S. Woerner, Variational quantum boltzmann machines, Quantum Machine Intelligence **3**, 7 (2020).

[19] T. Matsui, Quantum statistical mechanics and Feller semigroup, Quantum Probability Communications (1998).

[20] M. Khalkhali and M. Marcolli, *An Invitation to Noncommutative Geometry* (WORLD SCIENTIFIC, 2008).

[21] J. Eisert, M. Friesdorf, and C. Gogolin, Quantum many-body systems out of equilibrium, Nature Physics **11**

(2015).
[22] F. G. S. L. Brandão *et al.*, Quantum SDP Solvers: Large speed-ups, optimality, and applications to quantum learning, Preprint at https://arxiv.org/abs/1710.02581 (2017).
[23] M. H. Amin, E. Andriyash, J. Rolfe, B. Kulchytskyy, and R. Melko, Quantum Boltzmann Machine, Phys. Rev. X **8** (2018).
[24] J. Stokes, J. Izaac, N. Killoran, and G. Carleo, Quantum natural gradient, Quantum **4**, 269 (2020).
[25] S. Barison, F. Vicentini, and G. Carleo, An efficient quantum algorithm for the time evolution of parameterized circuits, Preprint at https://arxiv.org/abs/2101.04579 (2021).
[26] J. C. Spall, An overview of the simultaneous perturbation method for efficient optimization, Johns Hopkins APL Technical Digest **19**, 11 (1998).
[27] J. C. Spall, Accelerated second-order stochastic optimization using only function measurements, in *Proceedings of the 36th IEEE Conference on Decision and Control*, Vol. 2 (1997) pp. 1417–1424 vol.2, iSSN: 0191-2216.
[28] S. Amari and S. C. Douglas, Why natural gradient?, in *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP '98 (Cat. No.98CH36181)*, Vol. 2 (1998) pp. 1213–1216 vol.2.
[29] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, Evaluating analytic gradients on quantum hardware, Phys. Rev. A **99**, 032331 (2019).
[30] A. Mari, T. R. Bromley, and N. Killoran, Estimating the gradient and higher-order derivatives on quantum hardware, Phys. Rev. A **103**, 012405 (2021).
[31] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, Quantum fingerprinting, Phys. Rev. Lett. **87**, 167902 (2001).
[32] L. Cincio, Y. Subaşı, A. T. Sornborger, and P. J. Coles, Learning the quantum algorithm for state overlap, Preprint at http://arxiv.org/abs/1803.04114 (2018).
[33] K. Temme, T. J. Osborne, K. G. H. Vollbrecht, D. Poulin, and F. Verstraete, Quantum Metropolis Sampling, Nature **471** (2011).
[34] M.-H. Yung and A. Aspuru-Guzik, A quantum–quantum Metropolis algorithm, Proceedings of the National Academy of Sciences **109** (2012).
[35] D. Poulin and P. Wocjan, Sampling from the Thermal Quantum Gibbs State and Evaluating Partition Functions with a Quantum Computer, Phys. Rev. Lett. **103** (2009).
[36] M. Motta and et al., Determining eigenstates and thermal states on a quantum computer using quantum imaginary time evolution, Nature Physics **16** (2020).
[37] F. G. S. L. Brandão and M. J. Kastoryano, Finite Correlation Length Implies Efficient Preparation of Quantum Thermal States, Communications in Mathematical Physics **365** (2019).
[38] M. J. Kastoryano and F. G. S. L. Brandão, Quantum Gibbs Samplers: The Commuting Case, Communications in Mathematical Physics **344** (2016).
[39] J. Wu and T. H. Hsieh, Variational Thermal Quantum Simulation via Thermofield Double States, Phys. Rev. Lett. **123** (2019).
[40] A. Chowdhury, G. H. Low, and N. Wiebe, A Variational Quantum Algorithm for Preparing Quantum Gibbs States, Preprint at https://arxiv.org/abs/2002.00055 (2020).
[41] A. McLachlan, A variational solution of the time-dependent Schrödinger equation, Molecular Physics **8** (1964).
[42] H. Abraham *et al.*, Qiskit: An open-source framework for quantum computing (2019).
[43] A. Kandala *et al.*, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature (London) **549**, 242 (2017).

Appendix A: Variational Quantum Imaginary Time Evolution

For a Hamiltonian $\hat{O}$, the Wick-rotated Schrödinger equation

$$\frac{\partial \left|\psi(t)\right\rangle}{\partial t} = -(\hat{O} - E_t)\left|\psi(t)\right\rangle$$

describes the normalized form of Quantum Imaginary Time Evolution (QITE) where

$$\left|\psi(t)\right\rangle = \frac{e^{-\hat{O}t}}{\sqrt{\left\langle\psi(0)\right| e^{-\hat{O}t}\left|\psi(0)\right\rangle}}\left|\psi(0)\right\rangle,$$

and $E_t = \langle\psi(t)|\hat{O}|\psi(t)\rangle$.

A time-discretized approximation of this evolution can be implemented using a variational ansatz state $|\psi(\theta^{(t)})\rangle$ with parameters $\theta^{(t)}$ associated with time t. This VarQITE can be realized using McLachlan's variational principle [41]

$$\delta \left\| \frac{\partial \left|\psi(\theta^{(t)})\right\rangle}{\partial t} + (\hat{O} - E_\tau)\left|\psi(\theta^{(t)})\right\rangle \right\|_{\ell^2}^2 = 0,$$

which aims to minimize the distance between the left and right side of the Wick-rotated Schrödinger equation w.r.t. the variational space given by $|\psi(t)\rangle$.

This variational principle leads to the following linear system of equations

$$g_{ij}(\theta^{(t)})\frac{\partial\theta_j^{(t)}}{\partial t} = -\mathrm{Re}\left\{\left\langle \frac{\partial\psi(\theta^{(t)})}{\partial\theta_i}\right| \hat{O}\left|\psi(\theta^{(t)})\right\rangle\right\}$$

which together with an initial value for θ defines an initial value problem that can be numerically solved with an ODE solver.

Given that the time steps in the ODE solver are chosen sufficiently small and the variational quantum circuit is sufficiently expressive, then VarQITE steps can only decrease the system energy or keep it constant [16]. It follows that this approach offers good theoretical properties for searching for the state corresponding to the minimum energy eigenstate.

VarQITE may also be used for approximate Gibbs state preparation, see Sec. III B for further details, as well as for ground state evaluation. The latter case, can be motivated as follows. Suppose the initial state $|\psi(\theta^{(0)}))\rangle$ has an overlap with the ground state and the evolution

time $t \to \infty$. In this case, all contributions in $|\psi(\theta^{(t)}\rangle$ which correspond to eigenvalues bigger than the minimum are being damped with time and $\lim_{t\to\infty} |\psi(\theta^{(t)})\rangle$ is equal to the ground state. Since, in practice, an infinite time cannot be simulated one needs to find a sufficiently big, finite time. Notably, a VarQITE ground state search coincides with a special case of QNG [24] where

$$f(\theta) = -\frac{1}{2} \langle\psi(\theta)|\hat{O}|\psi(\theta)\rangle \,.$$

Appendix B: Comparison of the QFI Formulas

Eqs. (11) and (12) show different ways to compute the QFI. Here we show the equivalence and justify the coefficient of 1/2 in Eq. (12). A single QFI element according to Eq. (12) is

$$\begin{aligned}
&-\frac{1}{2}\frac{\partial^2}{\partial\theta_i\partial\theta_j} |\langle\psi_{\theta'}|\psi_\theta\rangle|^2 \Big|_{\theta'=\theta} \\
&= -\frac{1}{2}\frac{\partial^2}{\partial\theta_i\partial\theta_j} \langle\psi_{\theta'}|\psi_\theta\rangle\langle\psi_\theta|\psi_{\theta'}\rangle \Big|_{\theta'=\theta} \\
&= -\frac{\partial}{\partial\theta_i} \mathrm{Re}\left\{\langle\psi_{\theta'}|\psi_\theta\rangle \left\langle\psi_{\theta'}\middle|\frac{\partial\psi_\theta}{\partial\theta_j}\right\rangle\right\}\Big|_{\theta'=\theta} \\
&= -\mathrm{Re}\left\{\left\langle\psi_{\theta'}\middle|\frac{\partial^2\psi_\theta}{\partial\theta_i\partial\theta_j}\right\rangle + \left\langle\psi_{\theta'}\middle|\frac{\partial\psi_\theta}{\partial\theta_j}\right\rangle\left\langle\frac{\partial\psi_\theta}{\partial\theta_i}\middle|\psi_{\theta'}\right\rangle\right\}\Big|_{\theta'=\theta} \\
&= \mathrm{Re}\left\{-\left\langle\psi_\theta\middle|\frac{\partial^2\psi_\theta}{\partial\theta_i\partial\theta_j}\right\rangle - \left\langle\frac{\partial\psi_\theta}{\partial\theta_i}\middle|\psi_\theta\right\rangle\left\langle\psi_\theta\middle|\frac{\partial\psi_\theta}{\partial\theta_j}\right\rangle\right\}.
\end{aligned}$$

We can rewrite the first summand using the identity we obtain from differentiating both sides of the equation $1 = \langle\psi_\theta|\psi_\theta\rangle$ with respect to θ,

$$\begin{aligned}
0 &= \frac{\partial^2}{\partial\theta_i\partial\theta_j}\langle\psi_\theta|\psi_\theta\rangle \\
&= 2\mathrm{Re}\left\{\frac{\partial}{\partial\theta_i}\left\langle\frac{\partial\psi_\theta}{\partial\theta_j}\middle|\psi_\theta\right\rangle\right\} \\
&= 2\mathrm{Re}\left\{\left(\left\langle\frac{\partial^2\psi_\theta}{\partial\theta_i\partial\theta_j}\middle|\psi_\theta\right\rangle + \left\langle\frac{\partial\psi_\theta}{\partial\theta_i}\middle|\frac{\partial\psi_\theta}{\partial\theta_j}\right\rangle\right)\right\} \\
\Leftrightarrow -\mathrm{Re}&\left\{\left\langle\frac{\partial^2\psi_\theta}{\partial\theta_i\partial\theta_j}\middle|\psi_\theta\right\rangle\right\} = \mathrm{Re}\left\{\left\langle\frac{\partial\psi_\theta}{\partial\theta_i}\middle|\frac{\partial\psi_\theta}{\partial\theta_j}\right\rangle\right\}.
\end{aligned}$$

Replacing the second derivative of $|\psi_\theta\rangle$ with the two first order derivatives we obtain

$$g_{ij}(\theta) = \mathrm{Re}\left\{\left\langle\frac{\partial\psi_\theta}{\partial\theta_i}\middle|\frac{\partial\psi_\theta}{\partial\theta_j}\right\rangle - \left\langle\frac{\partial\psi_\theta}{\partial\theta_i}\middle|\psi_\theta\right\rangle\left\langle\psi_\theta\middle|\frac{\partial\psi_\theta}{\partial\theta_j}\right\rangle\right\},$$

which is the same as Eq. (11).

Appendix C: Influence of QFI Regularization

In each iteration step QN-SPSA constructs a (up to) rank-2 estimate $\hat{g}^{(k)}$ of the QFI. Even though we start with a full rank matrix, $\bar{g}^{(0)} = \mathbb{I}$ this can lead to a singular approximation. Since we have to invert the estimated QFI, or solve a LSE, this can be problematic.

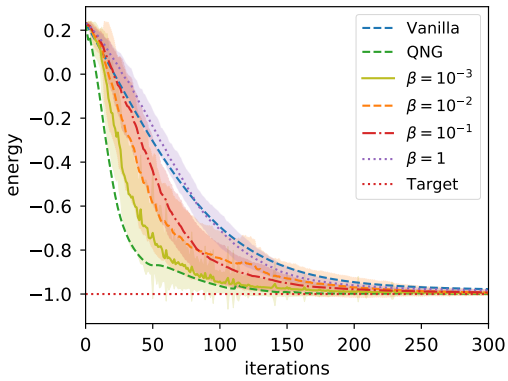

FIG. 4: Loss function for the Pauli two-design for vanilla and natural gradient descent and QN-SPSA. QN-SPSA is shown for different regularization constants $\beta \in \{10^{-3}, 10^{-2}, 10^{-1}, 1\}$.

To solve this problem we ensure that the estimate is symmetric-positive definite in each iteration by taking the absolute value of the eigenvalues and adding a regularization constant $\beta > 0$ on the diagonal, i.e., $\sqrt{A\bar{A}}+\beta\mathbb{I}$. To better understand the influence of the regularization for larger values we further normalize the expression as $(\sqrt{A\bar{A}} + \beta\mathbb{I})/(1+\beta)$. Without the normalization, a large regularization decreases the step size such that for $\beta \to \infty$ the update step approaches zero.

If the QFI estimate is faithful the eigenvalues are already positive and taking the absolute value of the eigenvalues does not change anything. However, adding the regularization constant always has an impact. A small regularization constant leads to an update closer to the QNG while a large constant neglects the QFI approximation and leads to an update closer to vanilla gradient descent. On the other hand, using a small constant is more prone to numerical instabilities than using a large one.

The regularization β is thus a hyper-parameter to trade off numerical stability for faithful QFI approximation. Fig. 4 shows the Pauli two-design example from Sec. IV A with 9 qubits for different values of β and visualizes how the regularization can be used to interpolate between natural and vanilla gradient descent.

Appendix D: Convergence efficiency of natural gradients

In Sec. IV A and Fig. 1 we show how fast different gradient descent techniques converge in the number of required iteration steps. We observed, that natural gradients have a clear advantage over vanilla gradient approaches. However, this comparison might not be en-

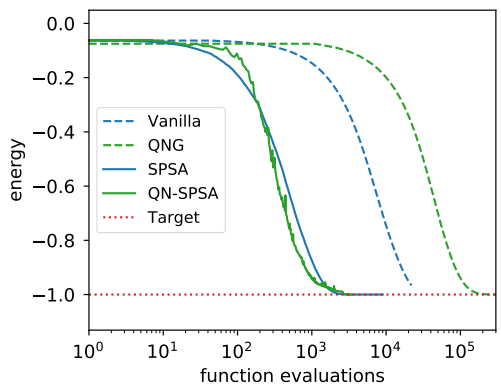

FIG. 5: The loss of the Pauli two-design example from Sec. IV A if the number of function evaluations are considered instead of iterations.

tirely fair since natural gradient methods, both analytic and SPSA-based, are more expensive to evaluate than their vanilla counterparts.

In Fig. 5, we change the perspective to see how fast each method converges with respect to the number of function evaluations instead of iterations. One function evaluation corresponds here to the calculation of an expectation value. Vanilla gradient descent and QNG require $\mathcal{O}(d)$ and $\mathcal{O}(d^2)$ function evaluations, respectively, while both SPSA techniques only require $\mathcal{O}(1)$. We see that the analytic methods require about one, respectively two, orders of magnitude more evaluations than the SPSA-based techniques. Since QN-SPSA requires seven function evaluations per step and SPSA only three, both methods perform very similarly in this example.

For problems with a more challenging loss landscape, however, natural gradient techniques provide a much more stable convergence and outperform vanilla gradients. In difficult landscapes, where small parameter changes can lead to large changes in function values, natural gradient methods show a stable convergence since they limit how much a parameter step can change the model. Standard gradients are agnostic to model sensitivity and easily overshoot if the learning rate is not adjusted to the loss function.

To investigate this behaviour, we consider an interesting loss landscape motivated from a QAOA ansatz [7] for a MAXCUT problem on a five-node random graph with random integer weights sampled from $\mathcal{U}([-10, 10])$. The observable for this particular application is

$$\begin{aligned}\hat{O} =& Z_4Z_5 + 2.5Z_3Z_5 + 2.5Z_4Z_5 - 0.5Z_2Z_5 \\ & - 0.5Z_2Z_3 - 4.5Z_1Z_5 + 3.5Z_1Z_3\end{aligned}$$

and the mixer Hamiltonian is $\sum_{i=1}^{5} X_i/20$. The QAOA ansatz is shown in Fig. 6(a).

The resulting loss function has numerous local extrema and has spiky regions close to the global minima, where the gradients are several magnitudes larger than in the flatter surroundings. The paths of the different optimizers in this landscape is shown in Fig. 6(b). QN-SPSA uses a learning rate of $\eta = 10^{-2}$, a displacement of $\epsilon = 10^{-2}$ and a regularization constant of $\beta = 10^{-3}$. SPSA is run three times: first with the same settings as QN-SPSA, then with the automatic calibration introduced in [43], and lastly with a manually adjusted calibration. The automated calibration chooses the learning rate and displacements as power series with optimal exponents for SPSA [26] and calibrates the constant coefficients of the power series such that in the first step the magnitude of the parameter update is $|\theta_i^{(1)} - \theta_i^{(0)}| \approx 2\pi/10$. However, in practice, fixing the parameter update can be problematic as it does not take into account how sensitive the model is with respect to a rescaling of the parameters. This becomes obvious in Fig. 6(b), where SPSA with this automatic calibration acts on too large length scales and starts to oscillate heavily. Thus, in the second run, we manually tested different parameter magnitude updates and selected the best at $|\theta_i^{(1)} - \theta_i^{(0)}| \approx 0.1$.

In Fig. 6(c), the mean and standard deviation of the loss for 25 runs is shown for each of the methods. We clearly see that QN-SPSA outperforms SPSA, even with manual calibrations and the additional evaluation costs taken into account.

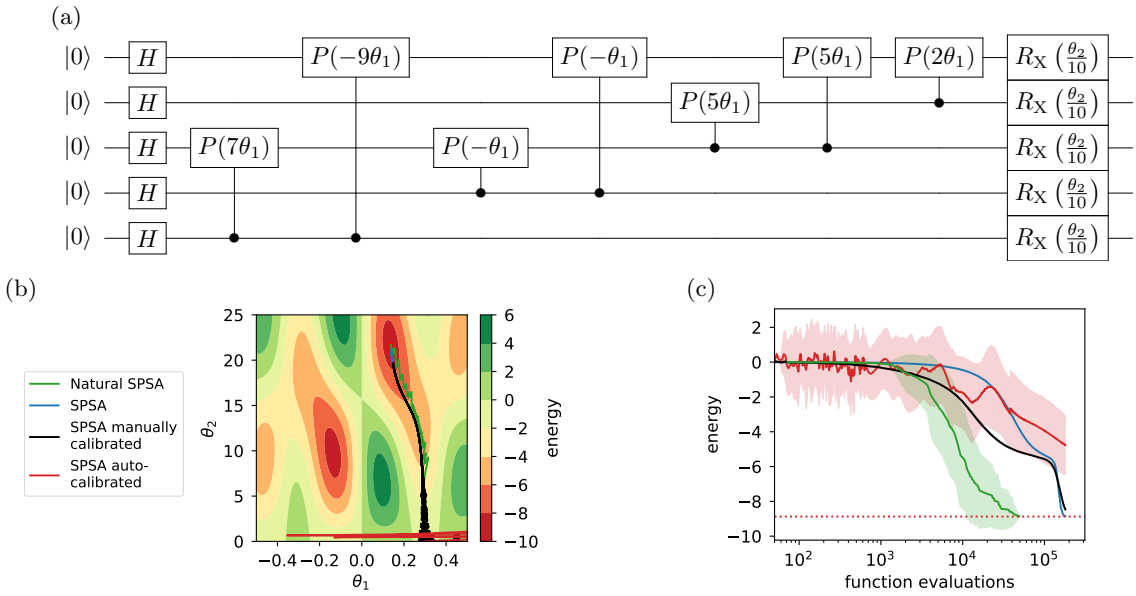

FIG. 6: The weighted MAXCUT problem. (a) The QAOA ansatz for the problem instance. (b) The loss landscape and the path of different optimization routines through the landscape. (c) The convergence of the investigated methods with respect to the required number of function evaluations.