ARTICLE OPEN

Check for updates

Stochastic gradient line Bayesian optimization for efficient noise-robust optimization of parameterized quantum circuits

Shiro Tamiya[1]✉ and Hayata Yamasaki[2,3]✉

Optimizing parameterized quantum circuits is a key routine in using near-term quantum devices. However, the existing algorithms for such optimization require an excessive number of quantum-measurement shots for estimating expectation values of observables and repeating many iterations, whose cost has been a critical obstacle for practical use. We develop an efficient alternative optimization algorithm, stochastic gradient line Bayesian optimization (SGLBO), to address this problem. SGLBO reduces the measurement-shot cost by estimating an appropriate direction of updating circuit parameters based on stochastic gradient descent (SGD) and further utilizing Bayesian optimization (BO) to estimate the optimal step size for each iteration in SGD. In addition, we formulate an adaptive measurement-shot strategy and introduce a technique of suffix averaging to reduce the effect of statistical and hardware noise. Our numerical simulation demonstrates that the SGLBO augmented with these techniques can drastically reduce the measurement-shot cost, improve the accuracy, and make the optimization noise-robust.

npj Quantum Information (2022)8:90 ; https://doi.org/10.1038/s41534-022-00592-6

INTRODUCTION

Advances in technologies of quantum hardware lead to intensive research on finding practical applications on noisy intermediate-scale quantum (NISQ) devices[1]. Variational quantum algorithms (VQAs)[2–4] are a class of promising candidates of quantum algorithms that are implementable on the NISQ devices. The VQAs can be used for a variety of computational tasks including quantum chemistry calculations[5–8], combinatorial optimization[9–11], and training of machine-learning models[12–15]. These tasks are achieved by minimizing task-specific cost functions usually defined as a sum of expectation values of observables. The optimization of minimizing the cost function is performed through updating parameters of a parameterized quantum circuit using a classical optimizer in a feedback loop. In particular, VQAs employ a quantum device to prepare quantum states that the parameterized quantum-circuit outputs. We perform a shot of quantum measurement on each output state to extract classical information, which is useful for estimating the expectation values of the cost function. The measurement outcomes are fed to the classical optimizer, with which we improve the circuit parameters so as to minimize the cost function iteratively.

But problematically, if we try to estimate the expectation values with high precision in the VQAs, we usually need an excessive number of measurement shots until minimizing the cost function[16,17]. In practice, a user of a quantum computer often needs to access a distant server of a quantum computer to query measurement shots, while the classical optimizer can be performed locally by the user at a negligible cost compared to the cost of using the quantum computer in terms of time and money[18]; in this setting, the number of measurement shots crucially dominates the cost of VQAs, which we aim to minimize here. In previous research, problems of reducing computational resources in VQAs have often been tackled by estimating an expectation-value efficiently[19–22] and reducing the number of iterations until convergence[23–30]. By contrast, to overcome a dominant obstacle in the above setting of VQAs, we here study the problem of reducing the overall cost of measurement shots in the optimization, that is, how we can optimize the circuit parameters at as little cost of the total number of measurement shots as possible. A difficulty of this problem stems from the nature of quantum mechanics: it is costly to extract expectation values as classical information from quantum states, yet the optimization would be hard without the assistance of classical information obtained from measurements on the quantum states. We stress that the problem here is not the estimation of the expectation values themselves; rather, a fundamental question that we ask is how efficiently we can use classical information of the measurement outcomes to optimize the circuit parameters without extracting the expectation values with high precision.

In this work, we address this problem by establishing a framework for the classical optimizer that combines two different optimization approaches, namely, stochastic gradient descent (SGD) and Bayesian optimization (BO). SGD is a standard algorithm in machine learning for training models, using an estimator of gradient at each optimization step rather than the exact value of the gradient[31,32]. Among a variety of existing optimizers proposed for VQAs[23–30,33–36], gradient-based optimizers have been studied intensively, motivated by the fact that the use of gradient information improves convergence[37]. Recently, SGD for VQAs has been investigated as a class of gradient-based optimizers[33]. The SGD for VQAs often uses a fixed small number of measurement shots to estimate the gradient, which may successfully avoid measuring expectation values with high precision. However, SGD has major shortcomings that may make the algorithm inefficient. First, instead of the low cost of each iteration, SGD may need a larger number of iteration until convergence than optimization algorithms using the exact gradient; second, SGD requires careful control of the step size of updating the parameters in each iteration, which may crucially affect the efficiency of the algorithm, but an appropriate choice of the step size is often difficult. On the other hand, BO is another common algorithm for optimization of a black-box function without

[1]Department of Applied Physics, Graduate School of Engineering, The University of Tokyo, 7-3-1 Hongo, Bynkyo-ku, Tokyo 113-8656, Japan. [2]Institute for Quantum Optics and Quantum Information—IQOQI Vienna, Austrian Academy of Sciences, Boltzmanngasse 3, 1090 Vienna, Austria. [3]Atominstitut, Technische Universität Wien, Stadionallee 2, 1020 Vienna, Austria. ✉email: tamiya@qi.t.u-tokyo.ac.jp; hayata.yamasaki@gmail.com

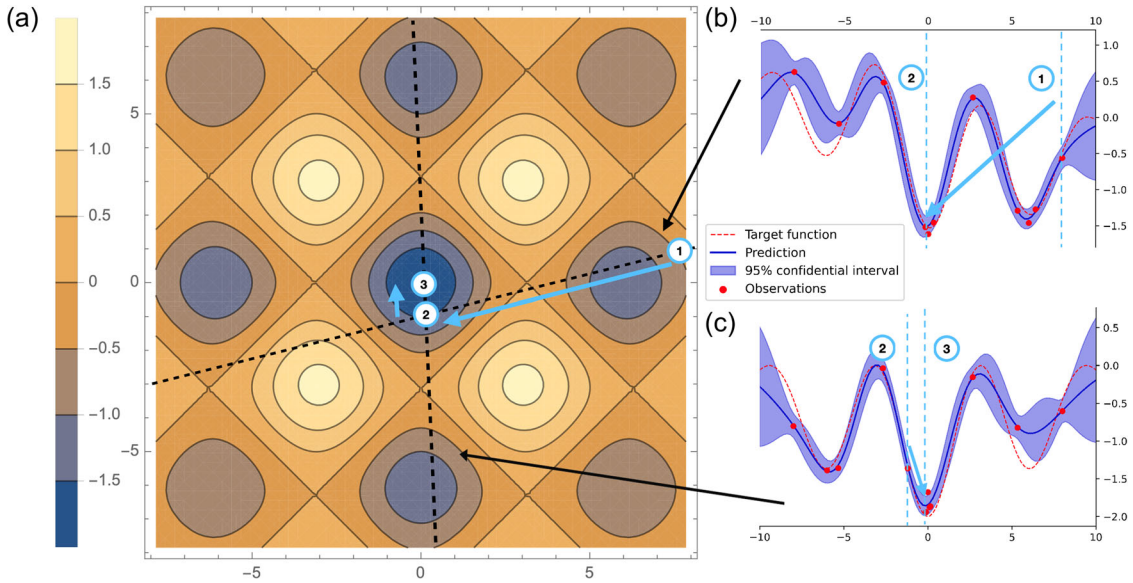

Fig. 1 An illustration of two iterations in SGLBO for minimizing a 2D cost function. a The figure represents the updating procedure of SGLBO on the landscape of the cost function. In particular, in the first iteration, at an initial point 1, we estimate a direction of the gradient of the cost function based on SGD and perform BO on the 1D subspace in this direction to estimate the optimal step size. **b** Then, we reach point 2 from the point 1 by moving in the estimated direction by the estimated optimal step size. **c** Next, at point 2, we perform the same procedure of estimating the gradient based on the SGD and estimating the optimal step size by the BO on the line of the 1D subspace, to move from point 2 to point 3. We iterate these procedures until SGLBO converges or consumes a preset number of measurement shots. After all these iterations, SGLBO returns a suffix average over the points visited in the iterations as an output.

necessarily using its gradient, which is especially suitable for optimizing imprecise and expensive-to-evaluate functions[38,39]. The BO has many successful applications such as computer vision, robotics, and experimental designs[40–43]. Owing to its robustness against noise in the imprecise evaluation of the functions[38,39], BO may also be useful for the optimization in VQAs[44,45]. However, it is known that BO becomes intractable in high-dimensional settings (typically $\geqq 10$)[46], and the number of parameters to be optimized in VQAs is usually too large to apply the BO directly.

To retain advantages of SGD and BO in VQAs while compensating for their shortcomings, we here construct the alternative framework for the optimization of parameterized circuits, stochastic gradient line Bayesian optimization (SGLBO), as illustrated in Fig. 1. The key idea of SGLBO is that we estimate an appropriate direction of updating the circuit parameters based on SGD, and also utilize BO to estimate the optimal step size in a 1D direction of the estimated gradient in each iteration. This idea aims at simultaneously resolving the problems of the step size in the SGD and of the infeasibility of high-dimensional optimization with the BO. To enhance the performance further, we combine the SGLBO with two noise-reducing techniques: adaptive shot strategy and suffix averaging. The adaptive shot strategy is a technique for dynamically determining the number of measurement shots to be used for the estimation of the gradient[34,47–53]. We here develop an adaptive shot strategy suitable for SGLBO, based on a technique of the norm test[48,49,51]. The norm test combined with SGD is known to provide faster convergence[49,51], and in the case of SGLBO, the norm test reduces not only the number of iterations but also the overall number of measurement shots. On the other hand, suffix averaging is a technique for achieving noise reduction. Instead of directly using the point of the final iteration in the optimization as an estimate of the minimizer of the cost function, the suffix averaging technique uses the average over a latter part of the sequence of points obtained from the iterations[54–56]. We utilize this technique to reduce the statistical noise in estimating the gradient and the optimal step size in SGLBO, and also reduce the effect of the hardware noise of the quantum device.

To show the significance of the SGLBO, we numerically demonstrate that the SGLBO can find an estimate of the minimizer of the cost function with a significantly small number of overall measurement shots compared to other state-of-art optimizers[23,34,57], in representative tasks for the VQAs, i.e., variational quantum eigensolver[5] and variational quantum compiling[58]. Thus, the reduction of the number of iterations achieved by finding the optimal step size by BO indeed contributes to the overall reduction of the number of measurement shots. We also discover that the SGLBO turns out to outperform the state-of-art optimizers not only in terms of the number of measurement shots but also the accuracy in estimating the minimum of the cost functions used in the simulation. Remarkably, we discover that even under a moderate amount of hardware noise, the SGLBO can estimate the minimum in a task with almost the same accuracy as noiseless cases, whereas the other state-of-the-art optimizers cannot in the same task. These results indicate that the SGLBO is a promising approach to reduce the number of measurement shots in the VQAs, and also to make the VQAs more feasible under unavoidable hardware noise in near-term quantum devices. Note that combination of SGD and BO has been previously studied only in a specific machine-learning setting[59], but its applicability and advantage for other tasks such as VQAs have been unknown; by contrast, our crucial contribution is to formulate SGLBO as the efficient and noise-robust framework for the task of optimizing parameterized quantum circuits and further develop the techniques of adaptive shot strategy and suffix averaging to demonstrate its advantage in this optimization task.

Consequently, the SGLBO establishes an alternative approach for efficient quantum-circuit optimizers, progressing beyond the existing state-of-the-art optimizers[23,34,57]; in particular, the novelty of SGLBO is to integrate two different optimization approaches, SGD and BO, to eliminate their shortcomings and take their advantages. Augmented with the further techniques of adaptive shot strategy and suffix averaging, the SGLBO is shown to have a significant advantage in the reduction of the cost of the number of measurement shots and also in the robustness against

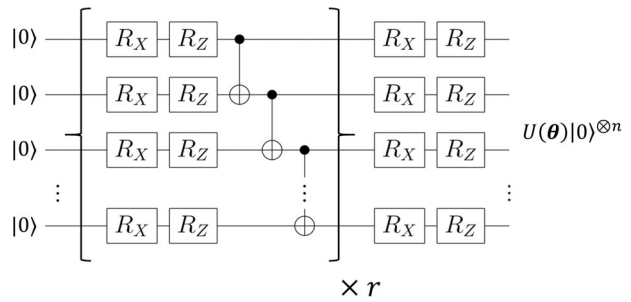

Fig. 2 An example of a parameterized quantum circuit used as an ansatz in VQAs. The circuit parameters $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_D]^\top \in \mathbb{R}^D$ with $D = 2n(r+1)$ elements are individually allocated as each rotation angle of Pauli rotation gates $R_X(\theta_i) := e^{-i\theta_i X}$ and $R_Z(\theta_j) := e^{-i\theta_j Z}$. The part of the circuit surrounded by the braces is repeated r times, where the repeated parts may have different parameters.

hardware noise, compared to the state-of-the-art optimizers for VQAs. These results open a way to practical algorithm designs for more efficient quantum-circuit optimization in terms of the overall cost of measurement shots, by avoiding both the precise estimation of expectation values and the many iterations of updating circuit parameters; at the same time, the approach developed for the SGLBO provides a fundamental insight into how VQAs can use classical information extracted from quantum states beyond estimating expectation values.

In the rest of this section, we describe the problem setting of optimization tasks in VQAs and review SGD and BO.

VQAs[2–4] are a class of algorithms that use a parameterized quantum circuit $U(\boldsymbol{\theta})$ to minimize a task-specific cost function $f(\boldsymbol{\theta})$. The vector $\boldsymbol{\theta} = [\theta_1, \cdots, \theta_D]^\top \in \mathbb{R}^D$ of D arguments of f is used as the circuit parameters of $U(\boldsymbol{\theta})$. The cost function $f(\boldsymbol{\theta})$ in VQAs is conventionally defined as an expectation-value of an observable O on n qubits, with respect to a quantum state output by the parameterized circuit, i.e.,

$$f(\boldsymbol{\theta}) = \mathrm{Tr}[OU(\boldsymbol{\theta})(|0\rangle\langle 0|)^{\otimes n}U^\dagger(\boldsymbol{\theta})], \tag{1}$$

where $|0\rangle$ is a standard-basis state used for initialization of each qubit, $U(\boldsymbol{\theta})|0\rangle^{\otimes n}$ is the output state of the n-qubit parameterized circuit, and $U^\dagger(\boldsymbol{\theta})$ is the complex conjugate of $U(\boldsymbol{\theta})$. The observable O is expanded as a sum of n-qubit tensor products of Pauli operators

$$O = \sum_k c_k P_k, \tag{2}$$

where c_k for each k is a real coefficient of the kth term, and P_k is a tensor product of n single-qubit Pauli operators $P_k = \bigotimes_{l=1}^n P_{k,l}$ with $P_{k,l} \in \{I, X, Y, Z\}$ being a Pauli (or identity) operator acting on the lth qubit. Here, the identity operator is denoted by $I := |0\rangle\langle 0| + |1\rangle\langle 1|$, and Pauli operators acting on a single qubit are $X := |0\rangle\langle 1| + |0\rangle\langle 1|$, $Y := -i|0\rangle\langle 1| + i|1\rangle\langle 0|$, and $Z := |0\rangle\langle 0| - |1\rangle\langle 1|$. In a usual setting of VQAs, $U(\boldsymbol{\theta})$ is composed of non-parametric gates such as CNOT gates, and parametric gates in the form of

$$U(\theta_i) = \exp(-iP_i\theta_i), \tag{3}$$

where P_i is also a tensor product of n single-qubit Pauli operators in the same way as P_k in Eq. (2). For example, Fig. 2 shows a representative choice of parameterized circuits used for VQAs[4]. Note that the parameter space of the circuit in Fig. 2 is a D-dimensional hypercube $\boldsymbol{\theta} \in [-\pi, \pi]^D$, i.e., a bounded subspace of $\mathbb{R}^D$, on which a uniform probability distribution is well defined.

The task in the VQAs is to obtain an estimate of the minimum of the cost function

$$\min_{\boldsymbol{\theta}\in\mathbb{R}^D} f(\boldsymbol{\theta}). \tag{4}$$

The minimizer is denoted by

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}\in\mathbb{R}^D} f(\boldsymbol{\theta}). \tag{5}$$

Note that the cost function $f(\boldsymbol{\theta})$, in general, can be non-convex, and it can be computationally hard in general to obtain the exact solution of the optimization problem in VQAs[60]. By contrast, this paper aims to provide a heuristic optimizer that approximately solves this optimization problem with a small number of measurement shots. In experiments using a quantum device, we can evaluate the cost function from the sum of the expectation values $\mathrm{Tr}[P_k U(\boldsymbol{\theta})(|0\rangle\langle 0|)^{\otimes n}U^\dagger(\boldsymbol{\theta})]$ for all k, each of which can be estimated by independently repeating the preparation of $U(\boldsymbol{\theta})|0\rangle^{\otimes n}$ by the parameterized circuit and the measurement of this state in the eigenbasis of the Pauli operator P_k. For each k, let $\overline{P}_k \in \mathbb{R}$ be a sample mean obtained from these measurements for P_k, and due to Eq. (2), we estimate $f(\boldsymbol{\theta})$ by

$$f(\boldsymbol{\theta}) \approx \sum_k c_k \overline{P}_k. \tag{6}$$

Each of these measurements is called a measurement shot. In this way, we evaluate f using a finite number of measurement shots; in this setting, we are only allowed imprecise queries to the cost function due to statistical errors with the finite number of measurement shots. Based on the central limit theorem[61], we may model each imprecise query to $f(\boldsymbol{\theta})$ as

$$y = f(\boldsymbol{\theta}) + \epsilon, \tag{7}$$

where y is an observed value, and $\epsilon \sim \mathcal{N}(0, \sigma^2)$ is independent and identically distributed (IID) Gaussian noise. From Hoeffding's inequality[62], to estimate $f(\boldsymbol{\theta})$ within an error ϵ with high probability, as large as $O(1/\epsilon^2)$ measurement shots may be required. In practice, it is prohibitively costly (i.e., an excessive number of measurement shots are needed) to evaluate a well-approximated value of the cost function (as well as its gradient), which leads to significant overhead in performing VQAs[16,17].

SGD aims to optimize a function $f(\boldsymbol{\theta})$ using an unbiased estimate of the gradient of f to update the parameters $\boldsymbol{\theta}$ iteratively toward the optimal point with high probability.

In the optimization of circuit parameters for VQAs, we may need to evaluate the gradient of the cost function $f(\boldsymbol{\theta})$. For $f(\boldsymbol{\theta})$ defined with parametric gates in the form of (3), we can utilize a parameter-shift rule[63,64] to calculate partial derivatives of the cost function from cost-function values at shifted circuit parameters, i.e.,

$$\frac{\partial f(\boldsymbol{\theta})}{\partial \theta_i} = \frac{f(\boldsymbol{\theta} + \frac{\pi}{2}\boldsymbol{e}_i) - f(\boldsymbol{\theta} - \frac{\pi}{2}\boldsymbol{e}_i)}{2}. \tag{8}$$

Here θ_i is a circuit parameter allocated to the rotation angle of the ith Pauli rotation gate $U(\theta_i) = \exp(-iP_i\theta_i)$, and $\boldsymbol{e}_i$ represents a unit vector along the coordinate of θ_i. Note that to obtain all the elements of the gradient of $f(\boldsymbol{\theta})$, we may need to evaluate each partial derivative independently.

However, as discussed above, we cannot exactly calculate the cost function and its gradient with a finite number of measurement shots, and the precise estimation of the gradient is costly in VQAs. In this setting, a standard method for solving Eq. (4) is stochastic gradient descent (SGD)[31,33], which updates the current point $\hat{\boldsymbol{\theta}}^{(t)}$ at iteration t according to

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \hat{\boldsymbol{\theta}}^{(t)} - \eta^{(t)}\hat{\boldsymbol{g}}^{(t)}(\hat{\boldsymbol{\theta}}^{(t)}), \tag{9}$$

where $\eta^{(t)}$ is the step size, and $\hat{\boldsymbol{g}}^{(t)}(\hat{\boldsymbol{\theta}}^{(t)}) := (\hat{g}_1^{(t)}(\hat{\boldsymbol{\theta}}^{(t)}), \ldots, \hat{g}_D^{(t)}(\hat{\boldsymbol{\theta}}^{(t)}))^\top$ is an unbiased estimator of the gradient $\nabla f(\hat{\boldsymbol{\theta}}^{(t)})$, i.e., $\mathbb{E}[\hat{\boldsymbol{g}}^{(t)}(\hat{\boldsymbol{\theta}}^{(t)})] = \nabla f(\hat{\boldsymbol{\theta}}^{(t)})$. Here $\hat{\boldsymbol{g}}^{(t)}$ is estimated with a finite number of measurement

shots, i.e., with a shot size

$$\boldsymbol{s}_{\mathrm{grad}}^{(t)} = (s_1^{(t)}, \ldots, s_D^{(t)})^\top. \tag{10}$$

The estimate of each partial derivative is individually computed as

$$\hat{g}_i^{(t)}(\boldsymbol{\theta}) = \frac{1}{s_i^{(t)}} \sum_{m=1}^{s_i^{(t)}} g_i^m(\boldsymbol{\theta}), \tag{11}$$

$$g_i^m(\boldsymbol{\theta}) = (O_+^m - O_-^m)/2, \tag{12}$$

where $O_\pm^m$ is a single-shot estimator of $f(\boldsymbol{\theta} \pm \frac{\pi}{2}\boldsymbol{e}_i)$. Each single-shot estimator of $f(\boldsymbol{\theta} \pm \frac{\pi}{2}\boldsymbol{e}_i)$ is constructed according to Eq. (6) by substituting $\boldsymbol{\theta}$ with $\boldsymbol{\theta} \pm \frac{\pi}{2}\boldsymbol{e}_i$, and the number of measurement shots used for estimating the kth term $c_k\overline{P}_k$ in Eq. (6) is denoted by $s_{i,k}^{(t)}$, which satisfies $\sum_k s_{i,k}^{(t)} = s_i^{(t)}$. Given the shot size $\boldsymbol{s}_{\mathrm{grad}}^{(t)}$, each $s_{i,k}^{(t)}$ is probabilistically determined using a multinomial distribution in such a way that the probability p_k of measuring the kth term should be proportional to the weight $|c_k|$, i.e., $p_k \propto |c_k|$ and $\sum_k p_k = 1$[22]; that is, it should hold that $\mathbb{E}[s_{i,k}^{(t)}] = p_k s_i^{(t)}$ for each k and i. Since the gradient is estimated from two values $f(\boldsymbol{\theta} \pm \frac{\pi}{2}\boldsymbol{e}_i)$ of the cost function, the number of measurement shots used for obtaining $\hat{\boldsymbol{g}}^{(t)}$ is

$$\sum_{i=1}^{D} 2s_i^{(t)} = 2s_{\mathrm{grad}}^{(t)}, \tag{13}$$

where we write $s_{\mathrm{grad}}^{(t)} := ||\boldsymbol{s}_{\mathrm{grad}}^{(t)}||_1$.

The estimator $\hat{\boldsymbol{g}}^{(t)}(\boldsymbol{\theta})$ in VQAs is unbiased for all $\boldsymbol{\theta} \in \mathbb{R}^D$, which is a preferable property to achieve convergence of SGD[33]. In addition, to guarantee convergence of SGD, we may require the step size to vanish as the estimated points approach a minimizer. In this case, the SGD achieves the optimization to accuracy ϵ within $O(1/\epsilon^4)$ iterations in general for non-convex functions[32], such as typical cost functions in VQAs. However, in practice, a user needs to designate a specific decay rate of step size to achieve good performance, whose optimization can be difficult.

BO is a gradient-free framework for optimization of an unknown function $f(\boldsymbol{\theta})$[38,39]. BO can be employed to optimize an expensive-to-evaluate cost function in settings where only noisy observations of the function are possible, and we try to seek a minimizer of $f(\boldsymbol{\theta})$ with as small a number of noisy observations as possible. One of the features of BO is to utilize an easy-to-compute surrogate model that approximates the unknown cost function based on observed data[65–67]. A popular surrogate model for BO is Gaussian process (GP)[68]. GP is a collection of random variables such that every finite subset of random variables obeys a multivariate normal distribution. In the BO, we put a GP prior over the true function $f(\boldsymbol{\theta})$ as $f(\boldsymbol{\theta}) \sim \mathcal{GP}(\mu(\boldsymbol{\theta}), k(\boldsymbol{\theta}, \boldsymbol{\theta}'))$, where $\mu(\boldsymbol{\theta}) = \mathbb{E}(f(\boldsymbol{\theta}))$ is a mean function, $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ is a covariance kernel function. In practice, if one has no prior knowledge about the mean of the function $\mu(\boldsymbol{\theta})$ that one tries to fit, $\mu(\boldsymbol{\theta})$ can be set to 0. A major choice of the kernel function is a Gaussian kernel

$$k(\boldsymbol{\theta}, \boldsymbol{\theta}') = \tau^2 \exp\left(\frac{-\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|^2}{2l^2}\right), \tag{14}$$

where τ^2 is called the signal variance that determines the average of the differences from the mean of the function, and l is called the length-scale that determines the length required for the values of the function to be uncorrelated[68]. For other conventional kernel functions, e.g., a Matérn kernel, see ref. [68].

Here we consider a situation where we have a set of N noisy observations of the cost function $\mathcal{D}_{1:N} = \{(\boldsymbol{\theta}^{(i)}, y^{(i)})\}_{i=1}^N$ at points $\boldsymbol{\theta}^{(1)}, \ldots, \boldsymbol{\theta}^{(N)}$, where each $y^{(i)} = f(\boldsymbol{\theta}^{(i)}) + \epsilon$ suffers from the IID Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Assuming that these observations are given according to GP, we calculate a GP posterior conditioned on these estimations, which is governed by hyperparameters, namely, the signal variance τ^2, the length-scale l, and the variance of Gaussian noise σ^2. These hyperparameters can be estimated by means of maximizing a log marginal likelihood[68]. Then, if we observe the cost function f at a new point $\boldsymbol{\theta}_*$, the value to be observed will obey a GP posterior expressed as

$$f_*|\boldsymbol{\theta}_*, \mathcal{D}_{1:N} \sim \mathcal{N}(\boldsymbol{k}_*^\top [K + \sigma^2 I]^{-1} \boldsymbol{y}, k_{**} - \boldsymbol{k}^\top [K + \sigma^2 I]^{-1} \boldsymbol{k}), \tag{15}$$

where $f_* = f(\boldsymbol{\theta}_*)$, $\boldsymbol{k}_* = [k(\boldsymbol{\theta}_*, \boldsymbol{\theta}^{(1)}), \cdots, k(\boldsymbol{\theta}_*, \boldsymbol{\theta}^{(N)})]^\top$, $k_{**} = k(\boldsymbol{\theta}_*, \boldsymbol{\theta}_*)$, and K is the covariance matrix $[k(\boldsymbol{\theta}^{(i)}, \boldsymbol{\theta}^{(j)})]_{i,j=1}^N$[68].

In BO, we construct an acquisition function $\varphi(\boldsymbol{\theta})$ from the posterior in Eq. (15) and determine the next query point according to

$$\boldsymbol{\theta}^{(N+1)} = \arg\min_{\boldsymbol{\theta} \in \mathbb{R}^D} \varphi(\boldsymbol{\theta}). \tag{16}$$

Several ways of constructing the acquisition function have been proposed, such as Thompson sampling[69], upper confidence bound[70], and expected improvement[71]. In particular, Thompson sampling estimates values of f at a given set of points by sampling according to the multivariate normal distributions obtained from Eq. (15), and use these sampled values as the values of the acquisition function at these points. Then, we take the minimum among the values of the acquisition function for the set of points and perform the next query to f at the minimum point in the set as shown in Eq. (16). The minimization of $\varphi(\boldsymbol{\theta})$ is performed by using efficient optimization heuristics[72,73]. BO proceeds with querying the cost function at the minimizer of $\varphi(\boldsymbol{\theta})$ and iteratively update the GP posterior according to Eq. (15) until a fixed number of queries to the cost function are performed[38].

This framework of BO has been shown to reduce the required number of queries to the cost function in achieving the minimization compared to other global optimization algorithms[38]. The performance of BO itself is governed by the ability to find the minimizer of $\varphi(\boldsymbol{\theta})$, which is also non-convex as well as the cost function. Thus, it is important to design the acquisition function suitably so that the computational cost is relatively low and optimization heuristics are tractable[46,74–76]. However, if the acquisition function is defined in a high-dimensional parameter space that typically appears in VQAs, it is excessively costly to use the BO.

RESULTS

In the following, we present the description of SGLBO and introduce the adaptive shot strategy and suffix averaging. Moreover, numerical experiments are provided to demonstrate the advantage of SGLBO compared to other state-of-the-art optimizers for VQAs.

Algorithm 1 Stochastic gradient line Bayesian optimization (SGLBO).

Require: Cost function $f(\boldsymbol{\theta})$ with D parameters in Eq. (1), the initial shot size $\boldsymbol{s}_{\mathrm{grad}}^{(0)}$ for evaluating the gradient in Eq. (10), a kernel $k(\boldsymbol{\theta}, \boldsymbol{\theta}')$ and an acquisition function $\varphi(\boldsymbol{\theta})$ used for GP in Eqs. (15) and (16), the initial point $\hat{\boldsymbol{\theta}}^{(0)}$ to be updated according to Eq. (17), the bound $\eta_{\max}$ of the 1D subspace $\mathcal{L}^{(t)}$ to perform the BO in Eq. (19), the initial number $s_{\mathrm{cost}}^{(0)}$ of measurement shots for evaluating the cost function in BO in Eq. (20), the number $N = N_{\mathrm{init}} + N_{\mathrm{eval}}$ of queries used for the BO in Eq. (21), the total number s_{tot} of measurement shots for the stopping condition (23), the precision κ in estimating the gradient according to Eq. (30), the description of the lower bound $G_{\mathrm{grad}}^{(t)}$ of the shot size in Eq. (30), the description of the lower bound $G_{\mathrm{cost}}^{(t)}$ of the number of measurement shots in estimating the cost function in Eq. (31), a parameter a for suffix averaging in Eq. (32).

1: **initialize:**
2: $t \leftarrow 0$, $s_{\mathrm{temp}}^{(t)} \leftarrow 0$
3: **while** $s_{\mathrm{temp}}^{(t)} < s_{\mathrm{tot}}$ ▷ Iterate until the stopping condition (23) is satisfied.
4: $\hat{\boldsymbol{g}}^{(t)}, S^{(t)} \leftarrow$ Estimate the gradient $\mathbb{E}[\hat{\boldsymbol{g}}^{(t)}] = \nabla f(\hat{\boldsymbol{\theta}}^{(t)})$ using $2 \times \boldsymbol{s}_{\mathrm{grad}}^{(t)}$ measurement shots according to Eq. (11), and calculate its empirical variance $S^{(t)}$ in Eq. (30).
5: $\mathcal{L}^{(t)} \leftarrow$ Take the 1D subspace $\mathcal{L}^{(t)}$ depending on $\hat{\boldsymbol{\theta}}^{(t)}, \hat{\boldsymbol{g}}^{(t)}, \eta_{\max}$ according to Eq. (19).
6: $\hat{\boldsymbol{\theta}}^{(t+1)} \leftarrow$ Determine $\hat{\boldsymbol{\theta}}^{(t+1)}$ by the BO on $\mathcal{L}^{(t)}$ with $k(\boldsymbol{\theta}, \boldsymbol{\theta}'), \varphi(\boldsymbol{\theta}), N_{\mathrm{init}}, N_{\mathrm{eval}}$ as described in the main text below Eq. (21).
7: $\boldsymbol{s}_{\mathrm{grad}}^{(t+1)} \leftarrow$ Determine the shot size for estimating the gradient, from $\kappa, \hat{\boldsymbol{g}}^{(t)}, S^{(t)}, D, G_{\mathrm{grad}}^{(t)}$ according to Eq. (30).
8: $s_{\mathrm{cost}}^{(t+1)} \leftarrow$ Determine the number of measurement shots for estimating the cost function in the BO, from $\boldsymbol{s}_{\mathrm{grad}}^{(t+1)}, G_{\mathrm{cost}}^{(t)}$ according to Eq. (31).
9: $s_{\mathrm{temp}}^{(t+1)} \leftarrow s_{\mathrm{temp}}^{(t)} + 2s_{\mathrm{grad}}^{(t)} + Ns_{\mathrm{cost}}^{(t)}$ due to Eq. (22).
10: $t \leftarrow t+1$
11: **end while**
12: $T \leftarrow t$
13: **return** $\overline{\boldsymbol{\theta}}_{\alpha,T} \leftarrow$ Take the suffix average according to Eq. (32).

Description of algorithm

We present a framework for the optimizer of parameterized quantum circuits in the VQAs, stochastic gradient descent line Bayesian optimization (SGLBO). The idea behind SGLBO is to estimate the direction of the gradient based on SGD and further to utilize BO to estimate the optimal step size within the one-dimensional subspace of parameters in this direction. This allows us to avoid the difficulty of choosing an appropriate step size in SGD, and also to achieve a feasible use of BO by limiting the domain to apply the BO to the one-dimensional space. In addition, we introduce two noise-reduction techniques, adaptive shot strategy and suffix averaging, to improve the speed and the accuracy of minimizing the cost function. Adaptive shot strategy and suffix averaging are crucial and characteristic components for the feasibility of SGLBO and will be explained in "Adaptive shot strategy" section and "Suffix averaging for SGLBO" section. Below, we will present the procedure of SGLBO (see also Algorithm 1).

The SGLBO achieves the minimization of the cost function by iteratively updating the points to estimate the minimizer of the cost function. Let T denote the total number of iterations in the SGLBO. For each iteration $t = 0, 1, \ldots, T-1$, let $\hat{\boldsymbol{\theta}}^{(t)}$ denote the point obtained in the $(t+1)$th iteration of the SGLBO, which is an estimator of the circuit parameters that minimize the cost function, and the initial point $\hat{\boldsymbol{\theta}}^{(0)}$ represents an initial guess of the minimizer. Note that we here take $\hat{\boldsymbol{\theta}}^{(0)}$ uniformly at random, but in case a better initial guess of the minimizer than the uniformly random point is available, $\hat{\boldsymbol{\theta}}^{(0)}$ could be chosen as the better guess[77,78]. Similarly to the SGD, the SGLBO computes an unbiased estimator $\hat{\boldsymbol{g}}^{(t)}$ of the gradient of the cost function at the point $\hat{\boldsymbol{\theta}}^{(t)}$, using $2s_{\mathrm{grad}}^{(t)}$ measurement shots due to Eq. (13). The shot size $\boldsymbol{s}_{\mathrm{grad}}^{(t)}$ is determined in each iteration t based on adaptive shot strategy, which will be explained in the following section. Using $\hat{\boldsymbol{g}}^{(t)}$, the SGLBO updates the point $\hat{\boldsymbol{\theta}}^{(t)}$ to the next point according to an update rule described by

$$\hat{\boldsymbol{\theta}}^{(t+1)} = \hat{\boldsymbol{\theta}}^{(t)} - \hat{\eta}^{*(t)}\hat{\boldsymbol{g}}^{(t)}, \tag{17}$$

where $\hat{\eta}^{*(t)}$ is an estimator of the optimal step size. The optimal step size $\eta^{*(t)}$ is defined as

$$\boldsymbol{\theta}^{*(t)} := \underset{\boldsymbol{\theta} \in \mathcal{L}^{(t)}}{\arg\min}\, f(\boldsymbol{\theta}) = \hat{\boldsymbol{\theta}}^{(t)} - \eta^{*(t)}\hat{\boldsymbol{g}}^{(t)}, \tag{18}$$

where $\mathcal{L}^{(t)}$ is the one-dimensional subspace for applying the BO, i.e.,

$$\mathcal{L}^{(t)} := \{\hat{\boldsymbol{\theta}}^{(t)} - \eta^{(t)}\hat{\boldsymbol{g}}^{(t)} | \eta^{(t)} \in [-\eta_{\max}, \eta_{\max}]\}, \tag{19}$$

and $\eta_{\max} > 0$ is a constant hyperparameter to bound the one-dimensional subspace that will be specified in "Example of choice of hyperparameters and implementation" section. We remark that we choose $\eta_{\max}$ as a constant independent of D so that the BO should be feasible even in the case of large D. A parameter region of D parameters of a circuit can be a D-dimensional hypercube, e.g., $\boldsymbol{\theta} \in [-\pi, \pi]^D$ for the circuit in Fig. 2, and thus, to cross the whole parameter region by $\mathcal{L}^{(t)}$, one may be tempted to choose $\eta_{\max}$ as the length of the diagonal of this D-dimensional hypercube, i.e., $\eta_{\max} \approx \sqrt{D}$; however, for the feasibility of the BO, it is indeed essential to keep $\eta_{\max}$ constant. Our approach can be considered an improvement over the SGD with a constant step size $\eta_{\max}$, where we use the BO to estimate the optimal step size $\hat{\eta}^{*(t)}$ instead of using the fixed step size $\eta_{\max}$.

To obtain an estimate of the optimal step size $\hat{\eta}^{*(t)}$ in Eq. (17), we perform the procedure of BO on $\mathcal{L}^{(t)}$ by using a fixed number of measurement shots

$$s_{\mathrm{cost}}^{(t)} \tag{20}$$

per query to the cost function, and querying these noisy observations of the cost function N times in total with

$$N = N_{\mathrm{init}} + N_{\mathrm{eval}}. \tag{21}$$

where N_{init} is the number of points used for initial evaluation for BO, and N_{eval} is the number of points evaluated during the BO in each step in addition to N_{init}. This procedure determines $\hat{\eta}^{*(t)}$ in such a way that $\hat{\boldsymbol{\theta}}^{(t+1)}$ in Eq. (17) should be given by $\boldsymbol{\theta}^{(N+1)}$ in Eq. (16). We will specify N_{init} and N_{eval} in "Example of choice of hyperparameters and implementation" section. In the BO, we use N_{init} points for the initial queries, which we take at equal intervals in the 1D subspace $\mathcal{L}^{(t)}$. Using the observed points, the BO iterates a cycle according to Eqs. (15) and (16) to decide an additional point to evaluate per cycle. Repeating N_{eval} cycles, we have N_{eval} points in addition to the N_{init} initial points, where the nth cycle for $n \in \{1, \ldots, N_{\mathrm{eval}}\}$ uses $(N_{\mathrm{init}} + n - 1)$ points to decide the $(N_{\mathrm{init}} + n)$th point. These N points are used for the update according to Eq. (17), i.e., the calculation of $\hat{\eta}^{*(t)}$.

In this way, the SGLBO updates the point $\hat{\boldsymbol{\theta}}^{(t)}$ according to Eq. (17) until we consume a preset total number of measurement shots s_{tot}, which we initially designate. In particular, in the $(t+1)$th iteration for each $t = 0, \ldots, T-1$, we use $2s_{\mathrm{grad}}^{(t)}$ measurement shots for estimating the gradient according to Eq. (13), and also use $s_{\mathrm{cost}}^{(t)}$ measurement shots for each of the N queries to the cost function in the BO; that is, the number of measurement shots that we use in the $(t+1)$th iteration is

$$2s_{\mathrm{grad}}^{(t)} + Ns_{\mathrm{cost}}^{(t)}. \tag{22}$$

In the SGLBO, if the total number of measurement shots used in the iterations exceeds the preset bound s_{tot}, i.e.,

$$\sum_{t=0}^{T-1}\left[2s_{\mathrm{grad}}^{(t)} + Ns_{\mathrm{cost}}^{(t)}\right] \geqq s_{\mathrm{tot}}, \tag{23}$$

then we stop the iterations. Note that T is given by the minimum number of iterations satisfying Eq. (23), determined during running the SGBLO depending on s_{tot}. We could also stop the iterations if we achieve the convergence of the cost function,

while we here use the stopping condition based on s_{tot} for simplicity of presentation. We remark that it would be too costly in VQAs to check the convergence of the values of the cost function $f(\hat{\boldsymbol{\theta}}^{(t)})$ itself, which we avoid here; instead, it would be possible, e.g., to use another stopping condition by checking the convergence of the sequence of parameters $(\hat{\boldsymbol{\theta}}^{(t)})_{t=0,\ldots,T-1}$.

Finally, after the last iteration, the optimizer calculates a suffix average[55] of the points $(\hat{\boldsymbol{\theta}}^{(t)})_{t=0,\ldots,T-1}$, i.e., an average of a subset of the points in a latter part of the iterations, which we will explain in "Suffix averaging for SGLBO" section. This suffix average is output as the estimate of the minimizer of the cost function.

The procedure of the SGLBO may require an additional cost of measurement shots for the BO compared to the SGD without using the BO, but this cost is negligible as explained in the following. To estimate the optimal step size by the BO, we may use an extra number of measurement shots to query the cost function, in addition to the gradient estimation based on the SGD. For simplicity, suppose that the shot size (10) and the number of measurement shots to evaluate the cost function in the BO are given by a constant s, i.e., $s_i^{(t)} = s$ $(i \in \{1, \ldots, D\})$ and $s_{\mathrm{cost}}^{(t)} = s$. Then, due to Eq. (22), the number of measurement shots to be used in each iteration of the SGLBO is $(2D + N)s$. In this case, the cost of estimating the optimal step size is the same as the cost of the gradient estimation for a parameterized quantum circuit with $N/2$ additional parameters. This cost can be negligibly low as the number of circuit parameters D gets large, and hence, we can indeed gain the benefit of estimating the optimal step size by the BO.

The foundation for why SGLBO can efficiently find a candidate of the minimum point, i.e., a stationary point, can be explained as follows. The constant step-size SGD with averaging converges to a stationary point even in a non-convex setting[79]. The SGLBO is designed to converge faster than this constant step-size SGD with averaging since we use the BO to find a step size that further reduces the value of the cost function compared to taking the deterministic constant step size. In particular, in each step $t \in \{0, \ldots, T-1\}$, BO aims to find the minimum point along a 1D subspace; that is, the cost function $f(\hat{\boldsymbol{\theta}}^{(t)})$ is reduced to $f(\hat{\boldsymbol{\theta}}^{(t+1)})$ satisfying $f(\hat{\boldsymbol{\theta}}^{(t+1)}) \leqq f(\hat{\boldsymbol{\theta}}^{(t)})$ with high probability, in the case where BO is performed with sufficiently good precision. In this case, as the iterations proceed, SGLBO improves the cost function according to $f(\hat{\boldsymbol{\theta}}^{(0)}) \geqq f(\hat{\boldsymbol{\theta}}^{(1)}) \geqq \cdots \geqq f(\hat{\boldsymbol{\theta}}^{(T-1)})$, which does not necessarily hold in SGD but should hold in the SGLBO with high probability, leading to an improvement compared to the mere use of the SGD. We remark that the optimization problems in VQAs are non-convex, and hence, a tight analysis of the convergence speed would be challenging in general. Some previous research such as refs. [33,37] performs convergence analyses of optimizers for VQAs with assumptions on convexity or strong convexity, but the performance for non-convex problems that typically appear in VQAs are unknown. In contrast, the above explanation of convergence does not require the convexity assumptions. However, to bound the speed of convergence of SGLBO, further assumptions may be needed since non-convex optimization problems are hard to solve by nature. We leave the tight analysis of the convergence speed of the SGLBO under an appropriate assumption for the setting of VQAs for further research; instead, we will use numerical simulation to show the fast convergence speed of the SGLBO in our numerical experiments.

Adaptive shot strategy

The number of measurement shots used for estimating values and gradients of the cost function is one of the crucial parameters in stochastic optimization algorithms. In such algorithms, we may have a trade-off between efficiency and accuracy. In particular, at the beginning of optimization, we can use an imprecise gradient estimated with few measurement shots to roughly move to points around the minimizer. On the other hand, at the end of optimization, the gradients with less noise are needed to further decrease the value of the cost function. This observation motivates us to establish a strategy to gradually increase the shot size (10) used for estimating the gradient in the SGLBO as the optimization proceeds.

Such adaptive shot strategies have been well studied in the field of machine learning[47–53], and one of them has been applied also in the context of VQAs[34,35]. However, the formula for estimating the next number of measurement shots given in refs. [34,35] depends on the step size and becomes invalid when the step size exceeds a certain range. Problematically, the step size in the SGLBO often exceeds the range. Thus, our algorithm utilizes a different approach, the norm test[48,49,51], which determines the number of measurement shots to maintain a constant signal-to-noise ratio of the estimate of the gradient.

In the norm test, we want to decide the shot size based on a condition that the estimated vector $-\hat{\boldsymbol{g}}^{(t)}$ should be appropriately in a descent direction[51], which ideally would be

$$\delta^{(t)} := ||\hat{\boldsymbol{g}}^{(t)} - \nabla f(\hat{\boldsymbol{\theta}}^{(t)})|| \leqq \kappa ||\hat{\boldsymbol{g}}^{(t)}||, \tag{24}$$

with a parameter κ satisfying $0 \leqq \kappa < 1$. Intuitively, as the optimization proceeds, the norm $||\hat{\boldsymbol{g}}^{(t)}||$ of the gradient becomes small, and the condition (24) requires that the estimate $\hat{\boldsymbol{g}}^{(t)}$ of the gradient should become precise as $||\hat{\boldsymbol{g}}^{(t)}||$ gets small. However, the exact evaluation of $\delta^{(t)}$ would be prohibitively costly in VQAs. Thus, we square both sides of the above inequality and then replace the left hand side with its expectation, i.e., $\mathbb{E}[(\delta^{(t)})^2] = \mathrm{Var}[\hat{\boldsymbol{g}}^{(t)}]$, where $\mathrm{Var}[\hat{\boldsymbol{g}}^{(t)}]$ is the variance of $\hat{\boldsymbol{g}}^{(t)}$. The exact value of this variance is still difficult to calculate, and hence, we make the approximation using a sample variance[80], i.e.,

$$\mathrm{Var}[\hat{\boldsymbol{g}}^{(t)}] \simeq \frac{\mathrm{Tr}(\Sigma^{(t)})}{s_{\mathrm{grad}}^{(t)}} \tag{25}$$

where $\Sigma_{ij}^{(t)} := \mathbb{E}[(g_i^{(t)} - \nabla f_i^{(t)})(g_j^{(t)} - \nabla f_j^{(t)})]$. Instead of Eq. (24), the norm test could check

$$\frac{\mathrm{Tr}(\Sigma^{(t)})}{s_{\mathrm{grad}}^{(t)}} \leqq \kappa^2 ||\hat{\boldsymbol{g}}^{(t)}||^2. \tag{26}$$

To adapt the condition (26) to the setting of VQAs, we consider the freedom of choosing the number of measurement shots for estimating each partial derivative of the cost function in Eq. (8). Since each partial derivative is estimated independently, Eq. (26) can be written as,

$$\sum_i \frac{\left(\sigma_i^{(t)}\right)^2}{s_i^{(t)}} \leqq \kappa^2 ||\hat{\boldsymbol{g}}^{(t)}||^2, \tag{27}$$

where $\sigma_i^{(t)} := \sqrt{\mathrm{Var}[g_i^{(t)}]}$. Now we impose a constraint on the number of measurement shots so that each estimate of the partial derivative should have an equal variance, i.e., $\left(\sigma_i^{(t)}\right)^2 / s_i^{(t)} = \left(\sigma_j^{(t)}\right)^2 / s_j^{(t)}$ for $i \neq j$. Then, we obtain a lower bound of $s_i^{(t)}$ for each i, i.e.,

$$s_i^{(t)} \geqq \frac{1}{\kappa^2} \frac{\left(\sigma_i^{(t)}\right)^2 D}{||\hat{\boldsymbol{g}}^{(t)}||^2}. \tag{28}$$

In practice, the true variance $\left(\sigma_i^{(t)}\right)^2$ is still too costly to evaluate, and thus, we replace it with the empirical variance $\left(S^{(t)}\right)^2$, which is accessible. Consequently, we forecast the number of measurement

shots so that it should satisfy

$$s_i^{(t+1)} \geqq \frac{1}{\kappa^2}\frac{\left(S_i^{(t)}\right)^2 D}{||\hat{\boldsymbol{g}}^{(t)}||^2}, \tag{29}$$

which we use to estimate the gradient in the next iteration. Since the SGLBO is intended to be applied to highly noisy cases, to avoid the cases where $s_i^{(t+1)}$ is too small to estimate the gradient appropriately, we here set a lower bound $G_{\mathrm{grad}}^{(t)}$ on the shot size and decide the next shot size according to

$$s_i^{(t+1)} = \max\left\{\left\lceil \frac{1}{\kappa^2}\frac{\left(S_i^{(t)}\right)^2 D}{||\hat{\boldsymbol{g}}^{(t)}||^2}\right\rceil, G_{\mathrm{grad}}^{(t)}\right\}, \tag{30}$$

where $\lceil \cdots \rceil$ is the ceiling funciton. The choice of $G_{\mathrm{grad}}^{(t)}$ will be specified in "Example of choice of hyperparameters and implementation" section.

Using the shot size specified by Eq. (30), we also decide the number of measurement shots used for observing values of the cost function in the BO according to

$$s_{\mathrm{cost}}^{(t+1)} = \max\left\{\frac{1}{D}\sum_{i=1}^{D} s_i^{(t)}, G_{\mathrm{cost}}^{(t)}\right\}, \tag{31}$$

where $G_{\mathrm{cost}}^{(t)} > 0$ is a constant for avoiding the cases where $s_{\mathrm{cost}}^{(t+1)}$ becomes too small to estimate the optimal step size appropriately. The choice of $G_{\mathrm{cost}}^{(t)}$ will also be specified in "Example of choice of hyperparameters and implementation" section.

Suffix averaging for SGLBO

In VQAs, one could use a point obtained from the final iteration as the result of the optimization. However, in SGLBO, we use BO to estimate the optimal step size in Eq. (18), and due to statistical error in the estimation, we suffer from the influence of the error between the estimate of the optimal step size obtained from the BO and the true optimal step size. Moreover, hardware noise also prevents steady update of the points, especially when we use near-term noisy quantum devices. Such errors or noises may lead to an oscillation of the points in the final part of the iterations around the minimizer. To suppress such oscillation, we take a suffix average of these points in the final part of the iterations, rather than using the single point of the final iteration itself.

Given the sequence of points obtained from T iterations $\hat{\boldsymbol{\theta}}^{(0)}, \ldots, \hat{\boldsymbol{\theta}}^{(T-1)}$, the a-suffix average is defined as the average of the last aT points[55]

$$\overline{\boldsymbol{\theta}}_{a,T} = \frac{1}{aT}\sum_{t=(1-a)T-1}^{T-1} \hat{\boldsymbol{\theta}}^{(t)}, \tag{32}$$

where $a \in (0, 1]$ is some constant, and a and T are taken here in such a way that aT should be an integer. During the optimization, we store the sequence of the points $(\hat{\boldsymbol{\theta}}^{(t)})_t$ in memory. At the end of optimization, we calculate the suffix average of these points according to the above formula and output the suffix average as the result of the SGLBO.

Importantly, to achieve the goal of suppressing the effect of noise at the points in the final part of the iterations, the suffix averaging here uses an equal weight in averaging out the noise in this part. To achieve this suppression with small overhead, the parameter a should be chosen appropriately, in such a way that the last aT points should be kept in a reasonably small fraction among all T points yet still large enough to suppress the noise effectively. We note that, instead of using the equal weight, averaging with a decaying sequence of weights would also work[56], which may have a merit in a case where one does not have enough memory to store all points and wants to average the points on the fly. Detailed comparison of suffix-averaging techniques using different sequences of weights in VQAs is left for future work.

The suffix averaging can accelerate the convergence of SGD in some cases; for example, for optimization of a strongly convex function, i.e., a function that is (roughly speaking) more convex than a quadratic function, the error of the point in the Tth iteration decreases at the speed of $O(\log(T)/T)$ with high probability, but the error of the suffix average of the points in the latter half of the T iterations reduces to $O(1/T)$, achieving the optimal speed[55]. In the case of VQAs, f may not be strongly convex. However, even in the SGLBO, we can suppress the oscillation around the minimizer in practice by taking the suffix average, which contributes to improving the results of the optimization.

Example of choice of hyperparameters and implementation

We show an example of the choice of hyperparameters in Algorithm 1. These hyperparameters will be used in numerical experiments. In the numerical experiments, we also consider the cases with and without hardware noise, referring to them as the noisy case and the noiseless case, respectively.

For estimating the gradient in the SGLBO, we take the initial shot size as

$$s_i^{(0)} = 2 \quad \text{for all} \quad i, \tag{33}$$

and initialize $\hat{\boldsymbol{\theta}}^{(0)}$ by sampling from the uniform probability distribution. We set the lower bound $G_{\mathrm{grad}}^{(t)}$ on the shot size by an average shot size in the last 10 iterations; i.e., for $t+1 \geqq 10$, according to Eq. (30), we take

$$\begin{aligned} G_{\mathrm{grad}}^{(t)} &= \tfrac{1}{10D}\sum_{i'=1}^{D}\sum_{t'=1}^{10} s_{i'}^{(t-10+t')}, \\ \text{i.e.,}\ s_i^{(t+1)} &= \max\left\{\left\lceil \frac{1}{\kappa^2}\frac{\left(S_i^{(t)}\right)^2 D}{||\hat{\boldsymbol{g}}^{(t)}||^2}\right\rceil, \frac{1}{10D}\sum_{i'=1}^{D}\sum_{t'=1}^{10} s_{i'}^{(t-10+t')}\right\}, \end{aligned} \tag{34}$$

and $G_{\mathrm{grad}}^{(t)} = 1$ for $t \leqq 10$. We set $\kappa = 0.99$ in Eq. (30).

In the BO that is used as a subroutine in the SGLBO, we use the Gaussian kernel in Eq. (14) with $\tau^2 = 0.2$, $l = 0.7$ as initial values. Before performing the GP regression to estimate values of a cost function, we optimize the hyperparameters, i.e., τ^2, l, and the variance of Gaussian noise σ^2, by maximizing the marginal likelihood of the hyperparameters. To avoid overfitting, we restrict the parameter region of these hyperparameters; in our numerical experiments, we set the parameter region as $10^{-3} \leqq \tau^2 \leqq 5$, $10^{-3} \leqq l \leqq 1$, and $10^{-5} \leqq \sigma^2 \leqq 5$. In addition, we perform this hyperparameter optimization 10 times from uniformly random starting points and take the best parameters to ensure that the hyperparameters are not a poor local optimum. As the acquisition function used in the BO, we choose Thompson sampling[68,69]. After performing the BO, we set the estimated optimal step size as the minimum point of the predictive mean of a GP posterior conditioned on N observed data points.

For the BO, we set $N_{\mathrm{init}} = 5$ and $N_{\mathrm{eval}} = 5$. The N_{init} points of the initial evaluation is randomly chosen according to the uniform probability distribution over the 1D subspace $\mathcal{L}^{(t)}$ in Eq. (19) with

$$\eta^{(t)} \in [-\eta_{\max}, \eta_{\max}],\ \eta_{\max} = \min\left\{\frac{\beta}{||H||}, \pi\right\}, \tag{35}$$

where $||H||$ is the operator norm, and $\beta > 0$ is a constant that we set depending on the problem later in "Advantage of SGLBO for various system sizes" section and "Robustness against hardware noise in SGLBO" section. Note that one of the initial evaluation points must be taken as $\eta^{(t)} = 0$, i.e., the current point $\hat{\boldsymbol{\theta}}^{(t)}$, for the stability of the BO. The number of measurement shots used for

evaluating each point in the BO is given by Eq. (31) with

$$G_{\mathrm{cost}}^{(t)} = \frac{\|H\|^2}{\epsilon^2} \text{ for all } t,$$
$$\text{i.e., } s_{\mathrm{cost}}^{(t)} = \max\left\{\frac{1}{D}\sum_{i=1}^{D} s_i^{(t)}, \frac{\|H\|^2}{\epsilon^2}\right\}, \tag{36}$$

where $\epsilon = 0.1$. Given the outcomes of these measurements, we perform GP regression using GPy[81].

For the suffix averaging, we set $a = 0.1$ in Eq. (32).

Numerical experiments

In the following, we numerically demonstrate the advantages of the SGLBO in comparison with state-of-the-art optimizers for VQAs. The optimizers to be compared with the SGLBO are summarized in "Optimizers for VQAs and their implementations" section. In particular, we investigate two situations: (1) when the size of a system scales up in "Advantage of SGLBO for various system sizes" section, and (2) when hardware noise and connectivity between qubits on hardware are taken into account in "Robustness against hardware noise in SGLBO" section. To this end, we simulate the performance of the optimizers in tasks of variational quantum eigensolver (VQE)[5] for (1) and variational quantum compilation (VQC)[58] for (2). Furthermore, we demonstrate in "Merits of noise-reducing techniques for general optimizers" section that the techniques of suffix averaging and adaptive shot strategy used in the SGLBO can also improve performance and noise robustness of a general class of optimizers, not only the SGLBO.

Optimizers for VQAs and their implementations

To compare the SGLBO with other existing optimizers, we consider the following three state-of-the-art optimizers: adaptive moment estimation (Adam)[57], individual coupled adaptive number of shots (iCANS)[34], and Nakanishi-Fujii-Todo method (NFT)[23]. Adam is a variant of SGD; although a number of different strategies for choosing step size in SGD have been proposed, Adam chooses the step size adaptively based on the accumulated information of estimates of the gradient used in previous iterations. The choice of step size in Adam is known to work well for many applications in the field of machine learning, but for VQAs, the required number of measurement shots for the optimization with Adam has been still prohibitively large[34]. We use Adam as a representative choice of a straightforward application of SGD to VQAs. The iCANS is also a variant of stochastic gradient optimizers in which the number of measurement shots at each iteration is chosen frugally based on the first and second moment of the gradient to improve performance in VQAs. While both of these optimizers are gradient-based optimizers, NFT is a sequential optimization method along an axis of the parameters using function fitting rather than the gradient.

For iCANS, we in particular use iCANS1[34], and for Adam, we used the same values of the hyperparameters as ref. [34]. In terms of the initial number of measurement shots used in iCANS, which is not mentioned in ref. [34], we set $s_i^{(0)} = 2$ for all i in our numerical experiments. Here we note that for iCANS1, the step size η_t is changed depending on the tasks of VQAs as specified in "Advantage of SGLBO for various system sizes" section and "Robustness against hardware noise in SGLBO" section, following ref. [34]. In addition, we used $s_i^{(t)} = 1000$ shots for each evaluation of the cost function in Eq. (8) in Adam and $s_{\mathrm{cost}}^{(t)} = 1000$ shots for each evaluation of the cost function to fit the function in NFT. Note that the values of the hyperparameters for which the optimizer works well are selected manually or by referring to the values of previous studies, and we did not perform an exhaustive hyperparameter search since such a search is computationally too costly to perform. After all, it may be infeasible to run such a hyperparameter search when we apply these optimizers to practical problems.

In these numerical experiments, we simulate quantum circuits by using Pennylane[82]. In "Advantage of SGLBO for varisou system sizes" section and "Robustness against hardware noise in SGLBO" section, the values of the cost function appearing in the figures are evaluated at the point of the final iterate in $(\hat{\boldsymbol{\theta}}^{(t)})_t$ (and the suffix averaged point in the SGLBO) by a noiseless simulator, where both the statistical noise and the hardware noise are ignored; in "Merits of noise-reducing techniques for general optimizers" section, these values are evaluated at the suffix averaged point by the noiseless simulator. For each optimizer, we repeated the overall optimization procedures fifteen times from uniformly random initial points, where each run from an initial point is repeated twice, and took the average over all the thirty runs. In the figures, we display the logarithm of the average as a thick line and each run as a thin line, using log-linear plots.

Advantage of SGLBO for various system sizes

In this section, we investigate the performance of SGLBO as we scale up the system size. We evaluate the performance of the optimizers in terms of the total number of measurement shots used during the optimization. In each iteration, we calculate the difference per site between the cost-function value at the current point of each optimizer and the minimum value of the cost function. In particular, we here consider a VQE task[5] for a 1D transverse field Ising model under open boundary conditions. The VQE is an algorithm to calculate the ground state energy of a given Hamiltonian, where the cost function is defined as the expectation-value of the Hamiltonian. The Hamiltonian here is given by

$$H = -J\left(\sum_{j=1}^{n-1} Z_j Z_{j+1} + g\sum_{j=1}^{n} X_j\right) \tag{37}$$

where Z_j and X_j are the Pauli Z and X matrices, respecitvely, at the jth site on a 1D chain of qubits, J represents the energy scale, and g is the relative strength of the external field compared to the nearest-neighbor couplings[83]. We choose $J = 1.0$ and $g = 1.5$. We use the ansatz circuit in Fig. 2 with $r = 4$ repetitions for $n = 4, 8, 12$ qubits. These sizes of the circuits are chosen based on the feasibility of classical simulation. We remark that we do not change the depth of the ansatz circuits in this setting and change only the system size, so that the gradient does not vanish exponentially for the large system size[84]; that is, it is expected that the problem of the barren plateau, which potentially make the optimization infeasible[84–86], is avoided in our setting. In this problem, for the SGLBO, we restrict the region for the line search $\mathcal{L}_i$ by $\beta = 3$, and for the iCANS, we set the step size $\eta_t = 1/||H||$, following ref. [34].

The result of the numerical simulation is shown in Fig. 3. Significantly, we discover that the SGLBO outperforms the other optimizers[23,34,57] in all the cases of $n = 4, 8, 12$ qubits, in terms of both the speed of convergence and the accuracy of estimating the minimum of the cost function. Thus, these advantages of the SGLBO can be obtained not only for the relatively small system size $n = 4$ but more broadly for the larger system sizes $n = 8, 12$. While NFT and Adam hit the limit of accuracy of the minimization in the early stage of the optimization, SGLBO and iCANS continue to improve the cost function even at the end of the optimization, which shows the advantage of deciding the number of measurement shots adaptively for each iteration in these algorithms. Moreover, owing to using the BO for estimating the optimal step size in each iteration, the SGLBO enjoys faster convergence with a fewer number of overall measurement shots. The additional cost of measurement shots in the BO in Eq. (22) turns out to be negligible even on a small scale $n = 4$, as well as the larger scales discussed in "Description of algorithm" section. Consequently, for the VQE tasks in Fig. 3, the SGLBO achieves the optimization of

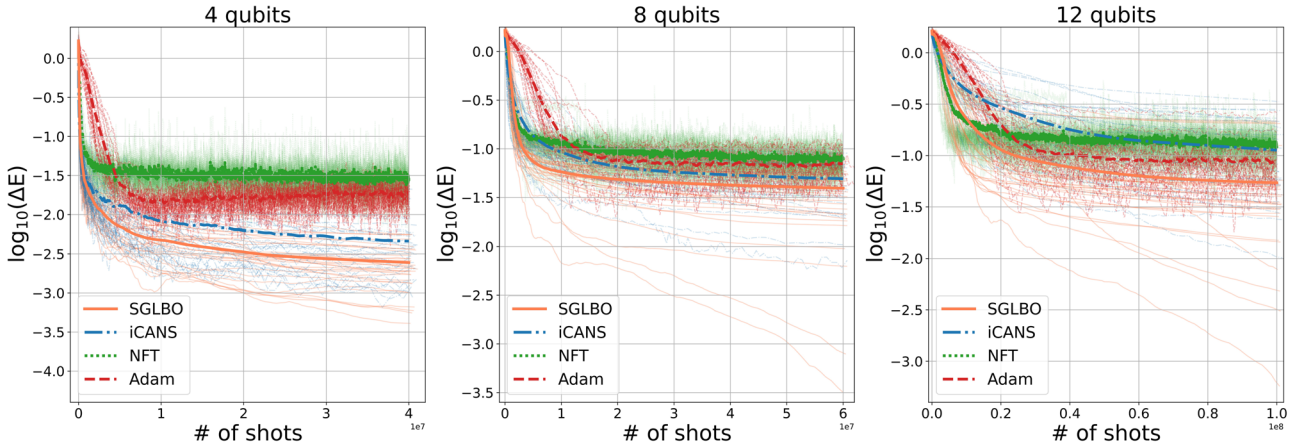

Fig. 3 Comparison of optimizers in terms of the performance on the VQE tasks. We optimize the cost function of the VQE for 1D transverse field Ising model with $n = 4, 8, 12$ qubits in the noiseless case. In all plots, the x-axis represents the total number of measurement shots used during the optimization, and the y-axis represents the difference ΔE per site between the true value of the cost function (i.e., not evaluated with finite measurement shots) at each iteration and the minimum value of the cost function under the ansatz described in Fig. 2 with $r = 4$. For each optimizer, the thin lines represent each run repeated twice from fifteen different initial points, and the thick line represents the average of these thirty runs. Significantly, the SGLBO outperforms the other state-of-the-art optimizers in terms of both the convergence speed and the achievable accuracy for, a broad region $n = 4, 8, 12$ of the number of qubits.

parameterized quantum circuits at the significantly faster convergence speed in terms of the number of measurement shots, and with better accuracy in minimizing the cost function than the other state-of-the-art optimizers.

Robustness against hardware noise in SGLBO

Next, we investigate the noise robustness of SGLBO. We consider VQC[58] with a fixed input state. The task of VQC is to find parameters of a parameterized circuit so that the unitary implemented by the circuit should act as equivalently as possible to a given target unitary when acting on a given input state. Following ref. [58], we define the cost function as

$$f(\boldsymbol{\theta}) = 1 - \frac{1}{n}\sum_{j=1}^{n} G_0^{(j)}, \tag{38}$$

where

$$G_0^{(j)} = \mathrm{Tr}[(|0\rangle\langle 0|_j \otimes \mathbb{1}_{\bar{j}})U^\dagger(\boldsymbol{\theta})U(\boldsymbol{\theta}^*)(|0\rangle\langle 0|)^{\otimes n}U^\dagger(\boldsymbol{\theta}^*)U(\boldsymbol{\theta})]. \tag{39}$$

Here $\mathbb{1}_{\bar{j}}$ is an identity operator acting on all qubits except the jth qubit, $G_0^{(j)}$ is the probability of getting the outcome 0 on the jth qubit, $\boldsymbol{\theta}$ is a vector of circuit parameters to be optimized, and $\boldsymbol{\theta}^*$ is a target vector of circuit parameters that are chosen here as $\boldsymbol{\theta}^* = (0, \ldots, 0)^\top \in \mathbb{R}^D$. The target unitary is $U(\boldsymbol{\theta}^*)$, and the input state is $(|0\rangle\langle 0|)^{\otimes n}$. The ansatz circuit $U(\boldsymbol{\theta})$ used here is the one in Fig. 2 with $n = 4$ and $r = 6$. In this case, the ansatz circuit can reach the optimal point at $\boldsymbol{\theta} = \boldsymbol{\theta}^*$ to output $(|0\rangle\langle 0|)^{\otimes n}$, where the value of the cost function is exactly zero at the optimal point, and y-axis shows the difference between the true optimal value (i.e., zero) and the value at the estimated optimal point. We note that this cost function is defined by local observables, so the gradient does not vanish in the shallow ansatz circuit used in this VQC task[58,85]. In VQC, we demonstrate the performance of the optimizers in both noiseless and noisy cases. To simulate noise in the noisy case, we used information about the gate-operation and readout errors and the connectivity of IBM's Bogota processor[87,88]. The detailed explanation on the parameters of the noise model is in Supplementary Information. We set $\beta = 6$ to limit the region $\mathcal{L}_i$ for SGLBO and choose the step size $\eta_t = 0.1$ for iCANS, following ref. [34].

The result of the numerical simulation is presented in Fig. 4. In the noiseless case, the SGLBO works better than the other state-of-the-art optimizers, which is consistent with the result of the VQE in Fig. 3. Even more remarkably, even in the presence of a moderate amount of hardware noise described above, the SGLBO can achieve almost the same accuracy in minimizing the cost function as that in the noiseless case, while the other optimizers converge to worse cost-function values. This result indicates a remarkable noise resilience of the SGLBO, owing to using the BO and also the technique of suffix averaging. In the SGLBO, the estimates of the minimizer of the cost function may be affected by hardware noise, and even if we use the BO that is relatively robust against the noise, these estimates may oscillate around the minimizer. However, the suffix averaging of these estimates makes it possible to obtain a point that is even nearer to the minimizer. In addition, the cost function in VQC has a preferable property that the minimizer is not susceptible to shifting caused by hardware noise[89], and this property also contributes to the noise resilience in this case; that is, in other tasks for the VQAs without this property, the same accuracy as noiseless cases would be hard to achieve in noisy cases. This result shows that the SGLBO can be more tolerant to hardware noise than the other state-of-the-art optimizers, which is crucial for the feasibility of performing VQAs on NISQ devices.

Merits of noise-reducing techniques for general optimizers

We here also show that the technique of suffix averaging and adaptive shot strategy that we use in SGLBO turns out to be advantageous even in improving performance and noise robustness of the other state-of-the-art optimizers, not only the SGLBO.

In particular, we here consider the same task of VQC as "Robustness against hardware noise in SGLBO" section, and we first apply the suffix averaging technique to all the optimizers, i.e., iCANS, Adam, and NFT as well as SGLBO. The result of the numerical simulation is shown in Fig. 5. In both the noiseless and noisy cases, the technique of suffix averaging can significantly improve the accuracy of the state-of-the-art optimizers, especially NFT and Adam, compared to the cases without suffix averaging in Fig. 4. For iCANS, suffix averaging may not be as effective as NFT and Adam, but can still achieve a comparable accuracy to the cases without suffix averaging. This result shows that the technique of suffix averaging that we apply in the SGLBO can indeed be useful as a general technique for improving a wide class of optimizers, not only for the SGLBO itself. At the same time, our numerical simulation shows that even if we improve the other optimizers by the suffix averaging, the SGLBO still outperforms these optimizers.

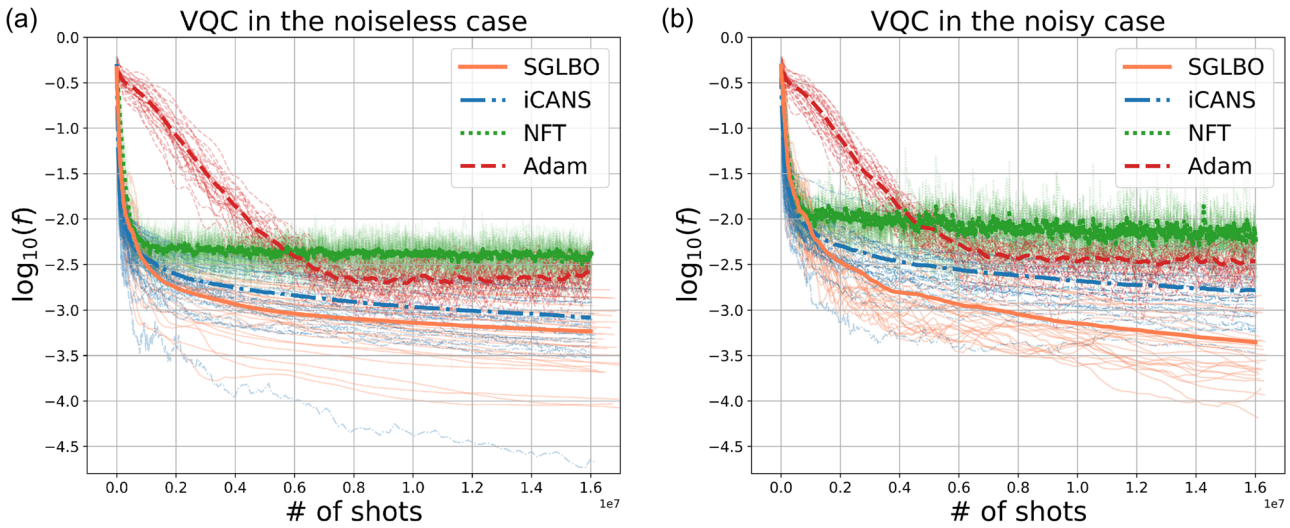

Fig. 4 Comparison of optimizers in terms of the performance on VQC tasks. We optimize the cost function of the VQC task **a** without hardware noise and **b** with hardware noise for the ansatz circuit in Fig. 2 with $n = 4$ qubits and $r = 6$ repetitions. In both plots, x-axis represents the total number of measurement shots used during the optimization, and y-axis represents the cost-function value. For each optimizer, the thin lines represent each run repeated twice from fifteen different initial points, and the thick line represents the average of these thirty runs. Remarkably, even under the moderate amount of the noise explained in the main text, the SGLBO can achieve almost the same accuracy as the noiseless case, whereas the achievable accuracy of the other state-of-the-art optimizers becomes worse in the noisy case.

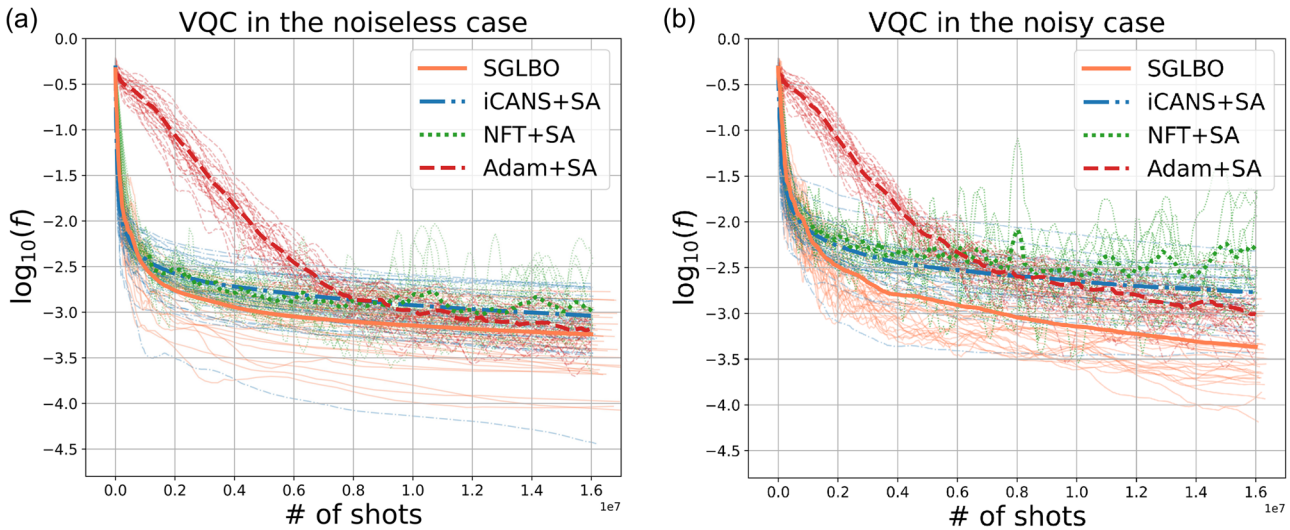

Fig. 5 Comparison of optimizers with the suffix averaging technique (SA), in the performance on the same VQC tasks as Fig. 4. The suffix averaging technique is not applied to iCANS, NFT, and Adam in Fig. 4 but is applied to all the optimizers in this figure. The x- and y-axes are the same as Fig. 4. For each optimizer, the thin lines represent each run repeated twice from fifteen different initial points, and the thick line represents the average of these thirty runs. In both the noiseless (**a**) and noisy (**b**) cases, the technique of suffix averaging can significantly improve the accuracy of state-of-the-art optimizers, especially NFT and Adam, while the SGLBO still outperforms the others. This shows that the suffix averaging technique developed here is not only a particular technique for improving the SGLBO but can be a broadly applicable technique for designing an efficient optimizer for VQAs.

Next, we apply the technique of adaptive shot strategy to Adam. Note that our technique of adaptive shot strategy cannot be applied directly to NFT since NFT does not use gradient; also, iCANS uses its own variant of adaptive shot strategies, and hence, our technique based on the norm test cannot be combined with iCANS either without changing its own strategy. Following the setting of SGLBO with (33), we set $s_i^{(0)} = 2$ for all i when we combine the adaptive shot strategy with Adam in these experiments. The results of the numerical experiments are shown in Fig. 6. In both noiseless and noisy cases, the adaptive shot strategy improves the performance of the original Adam. This indicates that the adaptive shot strategy based on the norm test is effectively applicable to the gradient-based optimizers and can improve the performance of the optimizers. In Fig. 6, we also demonstrate the combination of the suffix averaging and the adaptive shot strategy with Adam. In noiseless case, since Adam with the adaptive shot strategy has not yet hit the floor in the minimization and is still improving its accuracy, taking suffix averaging worsened the accuracy, as opposed to the case of averaging out the noise around the optimal points. On the other hand, in noisy case, the accuracy is improved. This result further confirms the effectiveness of the suffix averaging technique against hardware noise. The SGLBO still outperforms the other optimizers combined with these techniques.

In this way, the techniques that we develop for the SGLBO are also applicable broadly beyond the SGLBO itself, establishing a foundation for designing further efficient optimizers for VQAs in

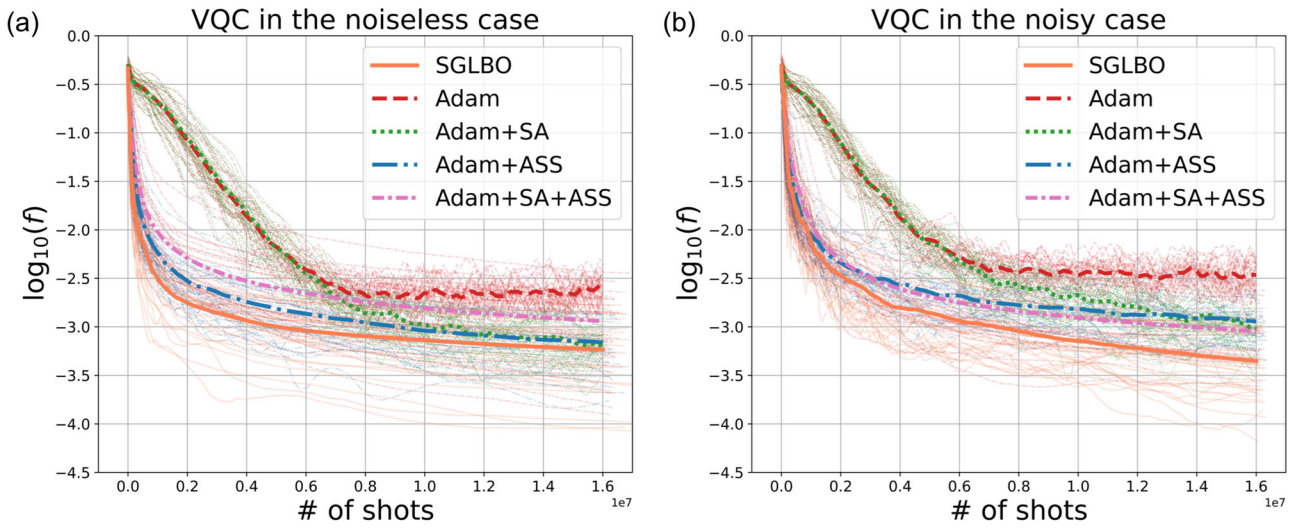

Fig. 6 Comparison of Adam with the suffix averaging technique (SA) and/or the adaptive shot strategy (ASS), in terms of the performance on the same VQC task as Fig. 4. The *x*- and *y*-axes are the same as Fig. 4. For each optimizer, the thin lines represent each run repeated twice from fifteen different initial points, and the thick line represents the average of these thirty runs. In both the noiseless (**a**) and noisy (**b**) cases, the adaptive shot strategy improves the performance of the original Adam, but the SGLBO outperforms the others. This shows that the adaptive shot strategy is also useful in improving the accuracy of Adam, rather than a specific technique for the SGLBO.

future research. At the same time, these results show that SGLBO is an effective combination of all the techniques, i.e., SGD, BO, the suffix averaging, and the adaptive shot strategy, to outperform the state-of-the-art optimizers.

DISCUSSION

In this work, we have developed an efficient framework, stochastic gradient line Bayesian optimization (SGLBO), for optimizing parameterized quantum circuits in variational quantum algorithms (VQAs). The core idea of the SGLBO is to estimate the direction of the gradient based on stochastic gradient descent (SGD), and also to use Bayesian optimization (BO) for estimating the optimal step size in this direction. The BO used for estimating the optimal step size in the SGLBO contributes to minimizing the cost function faster and more accurately, owing to the robustness of the BO against noise. To achieve the optimization feasibly within the fewer number of measurement shots, we also formulated an adaptive measurement-shot strategy based on the norm test to estimate the direction of the gradient efficiently. In addition, to suppress the effect of statistical error and hardware noise, we introduce the suffix averaging technique. The SGLBO with these techniques can save the cost of the number of measurement shots in optimizing the parameterized circuits, and also improve the accuracy in minimizing the cost function in the VQAs.

To compare the performance of the SGLBO with other state-of-the-art optimizers, we numerically investigated two situations: (1) when the system size increases and (2) when the hardware noise is present. For various system sizes, we discover that the SGLBO significantly improves the required number of measurement shots for achieving a desired accuracy in minimizing cost functions, and reaches an even better accuracy in minimizing the cost functions than other state-of-the-art optimizers, as shown in Fig. 3. Furthermore, we have shown that, even in the presence of a moderate amount of hardware noise, the SGLBO can achieve almost the same accuracy as that in the noiseless case, whereas the accuracy of the other state-of-the-art optimizers has got worse, in the task shown in Fig. 4. To suppress the noise, the suffix averaging technique as well as the use of the BO is crucial, and it turns out that the suffix averaging and the adaptive shot strategy developed for the SGLBO can also improve the accuracy and the noise robustness of other existing optimizers as demonstrated in Fig. 5.

Consequently, integrating two different optimization approaches, SGD and BO, our results on the SGLBO open an alternative way to drastically reduce the cost of measurement shots in the optimization of parameterized quantum circuits, and also to make VQAs more feasible under unavoidable hardware noise in near-term quantum devices. The techniques introduced here are versatile for problems with various system sizes, effective even in presence of noise, and widely applicable to a variety of algorithms for optimizing parameterized quantum circuits in the setting of VQAs, as demonstrated above. At the same time, the approach developed for the SGLBO provides a fundamental insight into how VQAs can use classical information extracted from quantum states, progressing beyond estimating expectation values. Moreover, the idea of the SGLBO indeed provides a general framework for optimizing noisy functions in the field of machine learning (ML), not specifically to VQAs. Thus, our results are expected to be of interest not only to users of noisy intermediate-scale quantum (NISQ) devices but to much broader communities of quantum information, such as those working on ML-assisted calibration of quantum devices in experiments, quantum tomography using an ansatz, and quantum metrology.

These results point toward various directions of future research. One possible direction is to investigate the difference in performance when the 1D subspace for the BO currently taken in the gradient descent direction (Eq. (19)) is chosen in another direction, such as natural gradient descent[28,30,90,91], negative curvature descent[92], and conjugate gradient[93]. Also, the development of a more efficient method for determining appropriate hyperparameter values in the SGLBO is also important for improving the accuracy. In our work, we have empirically found that the SGLBO with suffix averaging performs well in practice even if hardware noise is considered, but further research is needed to clarify of what class of hardware noise the suffix averaging can be tolerant, and how many iterations are needed to achieve comparable performance to the noiseless case. It would also be interesting to provide a theoretical guarantee on the performance of the SGLBO under appropriate assumptions, especially in the setting of non-convex optimization; after all, both empirical and theoretical studies are crucial for harnessing the potential for near-term applications of VQAs. Finally, since the SGLBO discovers a way to avoid the cost of precise estimation of expectation values in optimizing parameterized circuits for VQAs, it is even more advantageous to pursue applications of VQAs that do not require estimating the expectation values throughout running the entire

algorithm, i.e., even after the optimization; for example, state-of-the-art quantum algorithms for quantum machine learning avoid the expectation-value estimation by solving sampling problems so that the speedup should not be canceled out[94–96], and further research is needed to clarify how we can similarly avoid the expectation-value estimation in quantum machine learning with VQAs.

DATA AVAILABILITY

Data for the plots supporting the results in this work can be obtained from the corresponding author upon reasonable request.

CODE AVAILABILITY

Computer codes to perform the numerical experiments in this work are available from the corresponding author upon reasonable request.

Received: 17 December 2021; Accepted: 23 June 2022;
Published online: 27 July 2022

REFERENCES

1. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
2. Cerezo, M. et al. Variational quantum algorithms. *Nat. Rev. Phys.* **3**, 625–644 (2021).
3. Endo, S., Cai, Z., Benjamin, S. C. & Yuan, X. Hybrid quantum-classical algorithms and quantum error mitigation. *J. Phys. Soc. Jpn.* **90**, 032001 (2021).
4. Bharti, K. et al. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* **94**, 015004 (2022).
5. Peruzzo, A. et al. A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* **5**, 4213 (2014).
6. Kandala, A. et al. Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets. *Nature* **549**, 242–246 (2017).
7. McClean, J. R., Romero, J., Babbush, R. & Aspuru-Guzik, A. The theory of variational hybrid quantum-classical algorithms. *New J. Phys.* **18**, 023023 (2016).
8. McArdle, S., Endo, S., Aspuru-Guzik, A., Benjamin, S. C. & Yuan, X. Quantum computational chemistry. *Rev. Mod. Phys.* **92**, 015003 (2020).
9. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. Preprint at https://arxiv.org/abs/1411.4028 (2014).
10. Zhou, L., Wang, S.-T., Choi, S., Pichler, H. & Lukin, M. D. Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices. *Phys. Rev. X* **10**, 021067 (2020).
11. Harrigan, M. P. et al. Quantum approximate optimization of non-planar graph problems on a planar superconducting processor. *Nat. Phys.* **17**, 332–336 (2021).
12. Havlíček, V. et al. Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209–212 (2019).
13. Romero, J., Olson, J. P. & Aspuru-Guzik, A. Quantum autoencoders for efficient compression of quantum data. *Quantum Sci. Technol.* **2**, 045001 (2017).
14. Benedetti, M., Garcia-Pintos, D., Nam, Y. & Perdomo-Ortiz, A. A generative modeling approach for benchmarking and training shallow quantum circuits. *NPJ Quant. Inf.* **5**, 45 (2018).
15. Schuld, M. & Killoran, N. Quantum machine learning in feature hilbert spaces. *Phys. Rev. Lett.* **122**, 040504 (2019).
16. Wecker, D., Hastings, M. B. & Troyer, M. Progress towards practical quantum variational algorithms. *Phys. Rev. A* **92**, 042303 (2015).
17. Gonthier, J. F. et al. Identifying challenges towards practical quantum advantage through resource estimation: the measurement roadblock in the variational quantum eigensolver. Preprint at https://arxiv.org/abs/2012.04001 (2020).
18. Sung, K. J. et al. Using models to improve optimizers for variational quantum algorithms. *Quantum Sci. Technol.* **5**, 044008 (2020).
19. Huggins, W. J. et al. Efficient and noise resilient measurements for quantum chemistry on near-term quantum computers. *NPJ Quant. Inf.* **7**, 23 (2021).
20. Huang, H.-Y., Kueng, R. & Preskill, J. Predicting many properties of a quantum system from very few measurements. *Nat. Phys.* **16**, 1050–1057 (2020).
21. Huang, H.-Y., Kueng, R. & Preskill, J. Efficient estimation of pauli observables by derandomization. *Phys. Rev. Lett.* **127**, 030503 (2021).
22. Arrasmith, A., Cincio, L., Somma, R. D. & Coles, P. J. Operator sampling for shot-frugal optimization in variational algorithms. Preprint at https://arxiv.org/abs/2004.06252 (2020).
23. Nakanishi, K. M., Fujii, K. & Todo, S. Sequential minimal optimization for quantum-classical hybrid algorithms. *Phys. Rev. Res.* **2**, 043158 (2020).
24. Wilson, M. et al. Optimizing quantum heuristics with meta-learning. *Quantum Mach. Intell.* **3**, 13 (2021).
25. Koczor, B. & Benjamin, S. C. Quantum analytic descent. *Phys. Rev. Res.* **4**, 023017 (2022).
26. Ostaszewski, M., Grant, E. & Benedetti, M. Structure optimization for parameterized quantum circuits. *Quantum* **5**, 391 (2021).
27. Cervera-Lierta, A., Kottmann, J. S. & Aspuru-Guzik, A. Meta-variational quantum eigensolver: Learning energy profiles of parameterized hamiltonians for quantum simulation. *PRX Quantum* **2**, 020329 (2021).
28. Stokes, J., Izaac, J., Killoran, N. & Carleo, G. Quantum natural gradient. *Quantum* **4**, 269 (2020).
29. Self, C. N. et al. Variational quantum algorithm with information sharing. *NPJ Quant. Inf.* **7**, 116 (2021).
30. Haug, T. & Kim, M. S. Optimal training of variational quantum algorithms without barren plateaus. Preprint at https://arxiv.org/abs/2104.14543 (2021).
31. Robbins, H. & Monro, S. A stochastic approximation method. *Ann. Math. Stat* **22**, 400–407 (1951).
32. Bottou, L., Curtis, F. E. & Nocedal, J. Optimization methods for large-scale machine learning. *SIAM Rev.* **60**, 223–311 (2018).
33. Sweke, R. et al. Stochastic gradient descent for hybrid quantum-classical optimization. *Quantum* **4**, 314 (2020).
34. Kübler, J. M., Arrasmith, A., Cincio, L. & Coles, P. J. An adaptive optimizer for measurement-frugal variational algorithms. *Quantum* **4**, 263 (2020).
35. Gu, A., Lowe, A., Dub, P. A., Coles, P. J. & Arrasmith, A. Adaptive shot allocation for fast convergence in variational quantum algorithms. Preprint at https://arxiv.org/abs/2108.10434 (2021).
36. Lavrijsen, W., Tudor, A., Muller, J., Iancu, C. & de Jong, W. Classical optimizers for noisy intermediate-scale quantum devices. In *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 267–277 (IEEE, 2020).
37. Harrow, A. W. & Napp, J. C. Low-depth gradient measurements can improve convergence in variational hybrid quantum-classical algorithms. *Phys. Rev. Lett.* **126**, 140502 (2021).
38. Shahriari, B., Swersky, K., Wang, Z., Adams, R. P. & de Freitas, N. Taking the human out of the loop: a review of bayesian optimization. *Proc. IEEE* **104**, 148–175 (2016).
39. Snoek, J., Larochelle, H. & Adams, R. P. Practical Bayesian Optimization of Machine Learning Algorithms. In *Advances in Neural Information Processing Systems*, Vol. 25 (NIPS, 2012).
40. Bergstra, J., Yamins, D. & Cox, D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *Proceedings of the 30th International Conference on Machine Learning*, Vol. 28, 115–123 (PMLR, 2013).
41. Martinez-Cantin, R., Freitas, N., Brochu, E., Castellanos, J. & Doucet, A. A bayesian exploration-exploitation approach for optimal online sensing and planning with a visually guided mobile robot. *Auton. Robots* **27**, 93–103 (2009).
42. Lizotte, D. J., Wang, T., Bowling, M. H. & Schuurmans, D. Automatic gait optimization with gaussian process regression. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, 944–949 (Morgan Kaufmann Publishers Inc., 2007).
43. Azimi, J. et al. Myopic policies for budgeted optimization with constrained experiments. In *Proceedings of the National Conference on Artificial Intelligence* (AAAI, 2010).
44. Otterbach, J. S. et al. Unsupervised machine learning on a hybrid quantum computer. Preprint at https://arxiv.org/abs/1712.05771 (2017).
45. Zhu, D. et al. Training of quantum circuits on a hybrid quantum computer. *Sci. Adv.* **5**, eaaw9918 (2019).
46. Kandasamy, K., Schneider, J. & Poczos, B. High dimensional bayesian optimisation and bandits via additive models. In *Proceedings of the 32nd International Conference on Machine Learning*, Vol. 37, 295–304 (PMLR, 2015).
47. Friedlander, M. P. & Schmidt, M. Hybrid deterministic-stochastic methods for data fitting. *SIAM J. Sci. Comput.* **34**, A1380–A1405 (2012).
48. Bollapragada, R., Byrd, R. & Nocedal, J. Adaptive sampling strategies for stochastic optimization. *SIAM J. Optim.* **28**, 3312–3343 (2017).
49. Byrd, R., Chin, G., Nocedal, J. & Wu, Y. Sample size selection in optimization methods for machine learning. *Math. Program.* **134**, 127–155 (2012).
50. Pasupathy, R., Glynn, P., Ghosh, S. & Hashemi, F. On sampling rates in simulation-based recursions. *SIAM J. Optim.* **28**, 45–73 (2018).
51. De, S., Yadav, A., Jacobs, D. & Goldstein, T. Automated Inference with Adaptive Batches. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, Vol. 54, 1504–1513 (PMLR, 2017).
52. Balles, L., Romero, J. & Hennig, P. Coupling adaptive batch sizes with learning rates. In *Proceedings of the Thirty-Third Conference on Uncertainty in Artificial Intelligence, UAI*, 675–684 (Curran Associates, Inc., 2017).
53. Bollapragada, R., Nocedal, J., Mudigere, D., Shi, H.-J. & Tang, P. T. P. A progressive batching l-BFGS method for machine learning. In *Proceedings of the 35th International Conference on Machine Learning*, PMLR, Proceedings of Machine Learning Research, Vol. 80, 620–629 (2018).
54. Rakhlin, A., Shamir, O. & Sridharan, K. Making gradient descent optimal for strongly convex stochastic optimization. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, 1571–1578 (Omnipress, 2012).

55. Harvey, N. J. A., Liaw, C., Plan, Y. & Randhawa, S. Tight analyses for non-smooth stochastic gradient descent. In *Conference on Learning Theory*, (eds Beygelzimer, A. & Hsu, D.) 1579–1613 (2019).
56. Shamir, O. & Zhang, T. Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*, Vol. 28, 71–79 (JMLR.org, 2013).
57. Kingma, D. P. & Ba, J. L. Adam: a method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, (2015).
58. Khatri, S. et al. Quantum-assisted quantum compiling. *Quantum* **3**, 140 (2019).
59. Mahsereci, M. & Hennig, P. Probabilistic line searches for stochastic optimization. *J. Mach. Learn. Res.* **18**, 4262–4320 (2017).
60. Bittel, L. & Kliesch, M. Training variational quantum algorithms is np-hard. *Phys. Rev. Lett.* **127**, 120502 (2021).
61. Kwak, S. & Kim, J. Central limit theorem: the cornerstone of modern statistics. *Korean J. Anesthesiol.* **70**, 144 (2017).
62. Hoeffding, W. Probability inequalities for sums of bounded random variables. *J. Am. Stat. Assoc.* **58**, 13–30 (1963).
63. Mitarai, K., Negoro, M., Kitagawa, M. & Fujii, K. Quantum circuit learning. *Phys. Rev. A* **98**, 032309 (2018).
64. Schuld, M., Bergholm, V., Gogolin, C., Izaac, J. & Killoran, N. Evaluating analytic gradients on quantum hardware. *Phys. Rev. A* **99**, 032331 (2019).
65. Bodin, E. et al. Modulating surrogates for bayesian optimization. In *ICML 2020: 37th International Conference on Machine Learning*, Vol. 1, 970–979 (PMLR, 2020).
66. Springenberg, J. T., Klein, A., Falkner, S. & Hutter, F. Bayesian optimization with robust bayesian neural networks. In *Advances in Neural Information Processing Systems*, Vol. 29, 4134–4142 (2016).
67. Snoek, J. et al. Scalable bayesian optimization using deep neural networks. *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, Vol. 37, 2171–2180 (JMLR, 2015).
68. Rasmussen, C. E. & Williams, C. K. I.Gaussian Processes for Machine Learning (The MIT Press, 2005).
69. Basu, K. & Ghosh, S. Adaptive rate of convergence of thompson sampling for gaussian process optimization. Preprint at https://arxiv.org/abs/1705.06808 (2020).
70. Srinivas, N., Krause, A., Kakade, S. M. & Seeger, M. W. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Trans. Inf. Theory* **58**, 3250–3265 (2012).
71. Jones, D. R. A taxonomy of global optimization methods based on response surfaces. *J. Glob. Optim.* **21**, 345–383 (2001).
72. Spall, J. An overview of the simultaneous perturbation method for efficient optimization. *Johns Hopkins APL Tech. Dig.* **19**, 482–492 (1998).
73. Jones, D. R.Direct global optimization algorithm, 431–440 (Springer, 2001).
74. Rolland, P. T. Y., Scarlett, J., Bogunovic, I. & Cevher, V. High dimensional bayesian optimization via additive models with overlapping groups. In *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS) 2018*, 298–307 (PMLR, 2018).
75. Djolonga, J., Krause, A. & Cevher, V. High-dimensional gaussian process bandits. In *Advances in Neural Information Processing Systems*, Vol. 26, 1025–1033 (NIPS, 2013).
76. Kirschner, J., Mutny, M., Hiller, N., Ischebeck, R. & Krause, A. Adaptive and safe bayesian optimization in high dimensions via one-dimensional subspaces. In *Proceedings of the 36th International Conference on Machine Learning (ICML 2019)*, Vol. 97, 3429–3438 (PMLR, 2019).
77. Grant, E., Wossnig, L., Ostaszewski, M. & Benedetti, M. An initialization strategy for addressing barren plateaus in parametrized quantum circuits. *Quantum* **3**, 214 (2019).
78. Mitarai, K., Suzuki, Y., Mizukami, W., Nakagawa, Y. O. & Fujii, K. Quadratic clifford expansion for efficient benchmarking and initialization of variational quantum algorithms. *Phys. Rev. Res.* **4**, 033012 (2022).
79. Yu, L., Balasubramanian, K., Volgushev, S. & Erdogdu, M. A. An analysis of constant step size sgd in the non-convex regime: Asymptotic normality and bias. In *Advances in Neural Information Processing Systems*, Vol. 34, 4234–4248 (NeurIPS, 2021).
80. Freund, J. E. *Mathematical Statistics with Applications*, 8th edn. (Pearson, 2014).
81. GPy. GPy: Gaussian processes framework in python. https://github.com/SheffieldML/GPy (2021).
82. Bergholm, V. et al. Pennylane: Automatic differentiation of hybrid quantum-classical computations. Preprint at https://arxiv.org/abs/1811.04968 (2020).
83. Pfeuty, P. The one-dimensional ising model with a transverse field. *Ann. Phys.* **57**, 79–90 (1970).
84. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R. & Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **9**, 4812 (2018).
85. Cerezo, M., Sone, A., Volkoff, T., Cincio, L. & Coles, P. J. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nat. Commun.* **12**, 1791 (2021).
86. Ortiz Marrero, C., Kieferová, M. & Wiebe, N. Entanglement-induced barren plateaus. *PRX Quantum* **2**, 040316 (2021).
87. IBM Quantum Experience. https://quantum-computing.ibm.com/ (2021).
88. IBM Quantum Backends. https://github.com/Qiskit/qiskit-terra/tree/main/qiskit/test/mock/backends (2021).
89. Sharma, K., Khatri, S., Cerezo, M. & Coles, P. J. Noise resilience of variational quantum compiling. *New J. Phys.* **22**, 043006 (2020).
90. Wierichs, D., Gogolin, C. & Kastoryano, M. Avoiding local minima in variational quantum eigensolvers with the natural gradient optimizer. *Phys. Rev. Res.* **2**, 043246 (2020).
91. van Straaten, B. & Koczor, B. Measurement cost of metric-aware variational quantum algorithms. *PRX Quantum* **2**, 030324 (2021).
92. Liu, M., Li, Z., Wang, X., Yi, J. & Yang, T. Adaptive negative curvature descent with applications in non-convex optimization. In *Advances in Neural Information Processing Systems*, Vol. 31, 4854–4863 (NIPS, 2018).
93. Fletcher, R. & Reeves, C. M. Function minimization by conjugate gradients. *Comput. J.* **7**, 149–154 (1964).
94. Yamasaki, H., Subramanian, S., Sonoda, S. & Koashi, M. Learning with optimized random features: exponential speedup by quantum machine learning without sparsity and low-rank assumptions. In *Advances in Neural Information Processing Systems*, Vol. 33, 13674–13687 (NeurIPS, 2020).
95. Yamasaki, H. & Sonoda, S. Exponential error convergence in data classification with optimized random features: Acceleration by quantum machine learning. Preprint at https://arxiv.org/abs/2106.09028 (2021).
96. Kerenidis, I. & Prakash, A. Quantum Recommendation Systems. In *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, Vol. 67, 49:1–49:21 (ACM, 2017).

ACKNOWLEDGEMENTS

This work was supported by JST [Moonshot R&D][Grant Number JPMJMS2061], JSPS Overseas Research Fellowships, and JST PRESTO Grant Number JPMJPR201A.

AUTHOR CONTRIBUTIONS

S.T. and H.Y. contributed to the initial conception of the ideas, to the working out of details, and to the writing and editing of the manuscript.

COMPETING INTERESTS

The authors declare no competing interests.

ADDITIONAL INFORMATION

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41534-022-00592-6.

Correspondence and requests for materials should be addressed to Shiro Tamiya or Hayata Yamasaki.

Reprints and permission information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.