Minimizing Estimation Runtime on Noisy Quantum Computers

Guoming Wang,[*] Dax Enshan Koh,[†] Peter D. Johnson,[‡] and Yudong Cao[§]
Zapata Computing Inc., 100 Federal Street, Boston, Massachusetts 02110, USA

(Received 6 July 2020; revised 14 December 2020; accepted 18 February 2021; published 19 March 2021)

The number of measurements demanded by hybrid quantum-classical algorithms such as the variational quantum eigensolver (VQE) is prohibitively high for many problems of practical value. For such problems, realizing quantum advantage will require methods that dramatically reduce this cost. Previous quantum algorithms that reduce the measurement cost (e.g., quantum amplitude and phase estimation) require error rates that are too low for near-term implementation. Here we propose methods that take advantage of the available quantum coherence to maximally enhance the power of sampling on noisy quantum devices, reducing the measurement number and runtime compared to the standard sampling method of the VQE. Our scheme derives inspiration from quantum metrology, phase estimation, and the more recent "alpha-VQE" proposal, arriving at a general formulation that is robust to error and does not require ancilla qubits. The central object of this method is what we call the "engineered likelihood function" (ELF), used for carrying out Bayesian inference. We show how the ELF formalism enhances the rate of information gain in sampling as the physical hardware transitions from the regime of noisy intermediate-scale quantum computers to that of quantum error–corrected ones. This technique speeds up a central component of many quantum algorithms, with applications including chemistry, materials, finance, and beyond. Similar to the VQE, we expect small-scale implementations to be realizable on today's quantum devices.

DOI: 10.1103/PRXQuantum.2.010346

I. INTRODUCTION

Which quantum algorithms will deliver practical value first? A recent flurry of methods that cater to the limitations of near-term quantum devices have drawn significant attention. These methods include the variational quantum eigensolver (VQE) [1–4], the quantum approximate optimization algorithm [5] and variants [6], the variational quantum linear systems solver [7–9], other quantum algorithms leveraging the variational principles [10], and quantum machine learning algorithms [11–13]. In spite of such algorithmic innovations, many of these approaches have appeared to be impractical for commercially relevant problems owing to their high cost in terms of the number of measurements [2,14] and hence runtime of the expectation value estimation subroutine. Through an extensive benchmarking study of the VQE algorithm [15], it was shown that, for a set of molecules having industrial relevance, the VQE is very unlikely to yield an advantage over state-of-the-art quantum chemistry methods. Unfortunately, methods offering a quadratic speedup over the VQE in the runtime of the expectation value estimation subroutine, such as phase estimation, demand quantum resources that are far beyond the reach of near-term devices for moderately large problem instances [16].

Expectation value estimation is also a building block for many nonvariational quantum algorithms that have high-impact applications. Unfortunately, the standard versions of these algorithms lie out of reach for near-term quantum computers, in part due to the coherence requirements needed to implement estimation subroutines. Such techniques include quantum algorithms for Monte Carlo estimation [17] and quantum algorithms for solving linear systems of equations [18,19]. These algorithms find application in finance, engineering, and machine learning. Thus, there is strong motivation for developing estimation methods that make these techniques more amenable to near-term implementation.

Recently, the method of "α-VQE" [20] was proposed for interpolating between the VQE and phase estimation in terms of the asymptotic trade-off between sample count and quantum coherence. The basic idea is to start from the general framework of the VQE, namely the iterative optimization of the energy expectation that is a sum of

[*]Current address: 325 Front Street West, Suite 300, Toronto, ON M5V 2Y1, Canada. guoming.wang@zapatacomputing.com
[†]Current address: Institute of High Performance Computing, Agency for Science, Technology and Research (A*STAR), 1 Fusionopolis Way, #16-16 Connexis, Singapore 138632, Singapore. dax_koh@ihpc.a-star.edu.sg
[‡]peter@zapatacomputing.com
[§]yudong@zapatacomputing.com

Published by the American Physical Society under the terms of the Creative Commons Attribution 4.0 International license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

individual operator expectations, and proceed to estimate each individual operator with a Bayesian variant of the overlap estimation algorithm [21] that shares the same statistical inference backbone with known Bayesian parameter estimation schemes [22–25]. While phase estimation is commonly regarded as a quantum algorithm intended for fault-tolerant quantum computers, previous works [20,25] have demonstrated that in a noisy setting, Bayesian phase estimation can still yield quantum advantages in sampling. For instance, in Ref. [25, Sec. III A] it is shown that depolarizing noise reduces but does not eliminate the ability of the likelihood function to distinguish between different possible values of the parameter to be estimated.

This motivates the central question of our work: with realistic, *noisy* quantum computers, how do we maximize information gain from the coherence available to speed up expectation value estimation, and in doing so, speed up algorithms such as the VQE that rely on sampling? We note that this question is not only urgently relevant in the current era of noisy quantum devices without quantum error correction, but remains relevant for error-corrected quantum computation.

In this work, we investigate the impact of gate and measurement-readout error on the performance of quantum estimation tasks such as amplitude estimation and expectation value estimation. Note that the standard formulation of amplitude estimation [26] is equivalent to estimating the expectation value of an observable with eigenvalues ± 1 with respect to a quantum state generated by a given circuit. We introduce a simple noise model and show that the typical sample-generation scheme is hindered by a phenomenon we refer to as "dead spots" in the likelihood function. Motivated by these findings, we develop the framework of *engineered likelihood functions* (ELFs), in which signal processing techniques are used to boost the information gain per sample during the inference process of estimation. We develop several algorithms for the framework of engineered likelihood functions and investigate their performance with simulations. Finally, we develop a model for the runtime performance of these estimation algorithms and discuss the implications for near-term and far-term quantum computing.

The remainder of the paper is organized as follows. The remaining subsections of the Introduction review relevant prior work on quantum estimation and describe our main results in more detail. In Sec. II, we present a concrete example of our scheme for readers who wish to glean only a first impression of the key ideas. Subsequent sections then expand on the general formulation of our scheme. In Sec. III we describe in detail the general quantum circuit construction for realizing ELFs, and analyze the structure of the ELF in both noisy and noiseless settings. In addition to the quantum circuit scheme, our scheme also involves (1) tuning the circuit parameters to maximize information gain, and (2) Bayesian inference for updating the current belief about the distribution of the true expectation value. In Sec. IV we present heuristic algorithms for both. We then show numerical results in Sec. V, comparing our approach with existing methods based on Chebyshev likelihood functions (CLFs). In Sec. VI, we derive a mathematical model for the runtimes of our algorithms on noisy devices. We conclude in Sec. VII with implications of our results from a broad perspective of quantum computing.

A. Prior work

Evaluating the expectation value of an operator P with respect to a quantum state $|A\rangle$ is a fundamental element of quantum information processing. In the simplest setting where samples are drawn repeatedly by measuring the same operator P on the same quantum state $|A\rangle$, the measurement process is equivalent to a sequence of independent Bernoulli experiments. Yielding an estimate of the expectation value within error ε (with high probability) requires the number of samples to scale as $O(1/\varepsilon^2)$. We highlight the following key points regarding quantum estimation that are relevant to the context of this work.

1. *Cost scaling improvement using phase estimation*. Quantum effects introduce the opportunity to asymptotically accelerate the measurement process. In particular, there is a set of schemes [21] based on quantum phase estimation that is able to reduce the sample complexity to $O(\log(1/\varepsilon))$. This saturates the information-theoretical lower bound for the number of samples since in order to determine a bounded quantity to resolution ε one must be able to distinguish $O(1/\varepsilon)$ different values, requiring at least $\Omega(\log(1/\varepsilon))$ bits [24]. However, such optimal sample complexity comes at the cost of $O(1/\varepsilon)$ many coherent quantum operations. This trade-off between sample complexity and quantum coherence is also well understood in quantum metrology [27,28].
2. *Amplitude estimation and generalized reflections*. Phase estimation is closely related to the task of amplitude estimation [26] with many of the performance guarantees of the former applying to the latter. In its original definition, amplitude estimation is the problem of evaluating the quantity $\langle A|P_+|A\rangle$, where P_+ is a projection operator and $|A\rangle = A|0^n\rangle$ is an ansatz state. This problem is essentially equivalent to estimating the expectation value $\langle A|P|A\rangle$, where $P = 2P_+ - I$ is a reflection operator. Brassard *et al.* [26] showed that amplitude estimation can be solved by running phase estimation on the Grover iterate $U = (2|A\rangle\langle A| - I)P$ with initial state $|A\rangle$. Namely, the desired amplitude information is encoded in the eigenvalues of the unitary operator U. Subsequent works demonstrated [29] that using generalized reflection operators [i.e.,

those with a phase parameter φ such that $R_\varphi = (1 - e^{i\varphi})|A\rangle\langle A| - I$ or $R_\varphi = (1 - e^{i\varphi})P_+ - I$], one can realize a much larger set of SU(2) rotations in the subspace span$\{|A\rangle, P|A\rangle\}$ than with only common reflection operators. The set of SU(2) rotations realizable with such generalized construction has also been rigorously characterized [30] and later used for some of the most advanced Hamiltonian simulation algorithms such as qubitization [31] and signal processing [32].

3. *Bayesian inference perspective*. The problem of expectation value estimation can be framed as a parameter estimation problem, common in statistical inference. In fact, previous work (for example, Ref. [3, Sec. IV A]) has already pointed out a Bayesian perspective for considering the standard sampling process for VQE algorithms. The general setting is first to treat the operator expectation $\Pi = \langle A|P|A\rangle$ as the parameter to be estimated. Then, a parameterized quantum circuit $V(\vec{\theta})$ that may be related to $|A\rangle$ and P is constructed. The ability to execute the circuit and collect measurement outcome d translates to the ability to sample from a likelihood function $p(d|\vec{\theta}, \Pi)$. For a given prior $p(\Pi)$ representing the current belief of the true value of Π, Bayesian inference uses a measurement outcome d to produce (or update the prior to) a posterior distribution $p(\Pi|d) = p(d|\Pi, \vec{\theta})p(\Pi)/[\int p(d|\Pi, \vec{\theta})p(\Pi)d\Pi]$. For the settings considered in this paper, as well as in previous works [22–25], the prior and posterior distributions are maintained on the classical computer, while sampling from the likelihood function involves using a quantum device.

The combination of phase estimation and the Bayesian perspective gives rise to Bayesian phase estimation techniques [24,25,33] that are more suitable for noisy quantum devices capable of realizing limited-depth quantum circuits than earlier proposals [34]. The goal of Bayesian phase estimation is to estimate the phase $\theta = \arccos(\Pi)$ in an eigenvalue $e^{i\theta}$ of the unitary U. The quantum circuits used in this algorithm yield measurement outcomes with likelihoods given by

$$p(d|m, \Pi) = \frac{1 + (-1)^d \mathcal{T}_m(\Pi)}{2}, \tag{1}$$

where $d \in \{0, 1\}$ and $\mathcal{T}_m(\Pi) = \cos[m \arccos(\Pi)]$ is the mth-degree Chebyshev polynomial, found in many settings beyond Bayesian phase estimation (cf. Refs. [22, Eq. 2], [23, Eq. 1], [25, Eq. 2], and [20, Eq. 4]). In Ref. [23] the exponential advantage of Bayesian inference with a Gaussian prior over other nonadaptive sampling methods is established by showing that the expected posterior variance σ decays *exponentially* in the number of inference steps. Such exponential convergence is at a cost of $O(1/\sigma)$ amount of quantum coherence required at each inference step [23]. Such scaling is also confirmed in Ref. [25] in the context of Bayesian phase estimation.

Combining the above observations one may devise a Bayesian inference method for expectation value estimation that smoothly interpolates between the standard sampling regime and phase estimation regime. This is proposed in Ref. [35] as "α-VQE," where the asymptotic scaling is $O(1/\varepsilon^\alpha)$ with the extremal values of $\alpha = 2$ corresponding to the standard sampling regime (typically realized in the VQE) and $\alpha = 1$ corresponding to the quantum-enhanced regime where the scaling reaches the Heisenberg limit (typically realized with phase estimation). By varying the parameters for the Bayesian inference one can also achieve α values between 1 and 2. The lower the α value, the deeper the quantum circuit needed for Bayesian phase estimation. This accomplishes the tradeoff between quantum coherence and asymptotic speedup for the measurement process (point 1 above).

It is also worth noting that phase estimation is not the only paradigm that can reach the Heisenberg limit for amplitude estimation [36–38]. In Ref. [36] the authors considered the task of estimating the parameter θ of a quantum state ρ_θ. A parallel strategy is proposed where m copies of the parameterized circuit for generating ρ_θ, together with an entangled initial state and measurements in an entangled basis, are used to create states with the parameter θ amplified to $m\theta$. Such amplification can also give rise to likelihood functions that are similar to that in Eq. (1). In Ref. [37] it is shown that with randomized quantum operations and Bayesian inference one can extract information in fewer iterations than classical sampling even in the presence of noise. In Ref. [38] quantum amplitude estimation circuits with varying numbers m of iterations and numbers N of measurements are considered. A particularly chosen set of pairs (m, N) gives rise to a likelihood function that can be used for inferring the amplitude to be estimated. The Heisenberg limit is demonstrated for one particular likelihood function construction given by the authors. Both works [37,38] highlight the power of parameterized likelihood functions, making it tempting to investigate their performance under imperfect hardware conditions. As will become clear, although the methods we propose can achieve Heisenberg-limit scaling, they do not take the perspective of many previous works that consider interfering many copies of the same physical probe.

B. Main results

In this work we focus on estimating the expectation value $\Pi = \langle A|P|A\rangle$ where the state $|A\rangle$ can be prepared by a circuit A such that $|A\rangle = A|0^n\rangle$ for some integer $n \geq 1$. We consider a family of quantum circuits such that, as the circuit deepens with more repetitions of A, it allows

TABLE I. Comparison of our scheme with relevant proposals that appear in the literature. Here the list of features include whether the quantum circuit used in the scheme requires ancilla qubits in addition to qubits holding the state for amplitude estimation or phase estimation, whether the scheme uses Bayesian inference, whether any noise resilience is considered, whether the initial state is required to be an eigenstate, and whether the likelihood function (LF) is fully tunable like ELFs proposed here or restricted to CLFs.

Scheme	Bayesian inference	Noise consideration	Fully tunable LFs	Requires ancilla	Requires eigenstate
Knill *et al.* [21]	No	No	No	Yes	No
Svore *et al.* [24]	No	No	No	Yes	Yes
Wiebe and Grenade [25]	Yes	Yes	No	Yes	Yes
Wang *et al.* [20]	Yes	Yes	No	Yes	Yes
O'Brien *et al.* [33]	Yes	Yes	No	Yes	No
Zintchenko and Wiebe [37]	No	Yes	No	No	No
Suzuki *et al.* [38]	No	No	No	No	No
This work (Sec. III)	Yes	Yes	Yes	No	No
This work (Appendix D)	Yes	Yes	Yes	Yes	No

for likelihood functions that are polynomial in Π of ever higher degree. As we demonstrate in the next section with a concrete example, a direct consequence of this increase in polynomial degree is an increase in the power of inference, which can be quantified by Fisher information gain at each inference step. After establishing this "enhanced sampling" technique, we further introduce parameters into the quantum circuit and render the resulting likelihood function tunable. We then optimize the parameters for maximal information gain during each step of inference. See Table I for a comparison of our scheme with relevant proposals that appear in previous literature. The following lines of insight emerge from our efforts.

1. *The role of noise and error in amplitude estimation*. Previous works [20,25,33,37] have revealed the impact of noise on the likelihood function and the estimation of the Hamiltonian spectrum. Here we investigate the same for our scheme of amplitude estimation. Our findings show that, while noise and error do increase the runtime needed for producing an output that is within a specific statistical error tolerance, they do not necessarily introduce systematic bias in the output of the estimation algorithm. Systematic bias in the estimate can be suppressed by using active noise-tailoring techniques [39] and calibrating the effect of noise.

   We have performed simulation using realistic error parameters for near-term devices and discovered that the enhanced sampling scheme can outperform the VQE in terms of sampling efficiency. Our results have also revealed a perspective on tolerating error in quantum algorithm implementation where higher fidelity does not necessarily lead to better algorithmic performance. For fixed gate fidelity, there appears to be an optimal circuit fidelity around the range of 0.5–0.7 at which the enhanced scheme yields the maximum amount of quantum speedup.

2. *The role of likelihood function tunability*. Parameterized likelihood functions are centerpieces of phase estimation or amplitude estimation routines. To our knowledge, all of the current methods focus on likelihood functions of the Chebyshev form [Eq. (1)]. For these CLFs, we observe that in the presence of noise there are specific values of the parameter Π (the "dead spots") for which the CLFs are significantly less effective for inference than other values of Π. We remove such dead spots by engineering the form of the likelihood function with generalized reflection operators (point 2 in Sec. I) whose angle parameters are made tunable.

3. *Runtime model for estimation as error rates decrease*. Previous works [20,38] have demonstrated smooth transitions in the asymptotic cost scaling from the $O(1/\varepsilon^2)$ of the VQE to $O(1/\varepsilon)$ of phase estimation. We advance this line of thinking by developing a model for estimating the runtime t_ε to target accuracy ε using devices with degree of noise $\lambda \in [0, \infty)$ (cf. Sec. VI):

$$t_\varepsilon \sim O\left(\frac{\lambda}{\varepsilon^2} + \frac{1}{\sqrt{2}\varepsilon} + \sqrt{\left(\frac{\lambda}{\varepsilon^2}\right)^2 + \left(\frac{2\sqrt{2}}{\varepsilon}\right)^2}\right). \tag{2}$$

   The model interpolates between the $O(1/\varepsilon)$ scaling and $O(1/\varepsilon^2)$ scaling as a function of λ. Such bounds also allow us to make concrete statements about the extent of quantum speedup as a function of hardware specifications such as the number of qubits and two-qubit gate fidelity, and therefore estimate runtimes using realistic parameters for current and future hardware.

II. A FIRST EXAMPLE

There are two main strategies for estimating the expectation value $\langle A|P|A\rangle$ of some operator P with respect to a quantum state $|A\rangle$. The method of *quantum amplitude estimation* [40] provides a provable quantum speedup with respect to certain computational models. However, to achieve precision ε in the estimate, the circuit depth needed in this method scales as $O(1/\varepsilon)$, making it impractical for near-term quantum computers. The variational quantum eigensolver uses *standard sampling* to carry out amplitude estimation. Standard sampling allows for low-depth quantum circuits, making it more amenable to implementation on near-term quantum computers. However, in practice, the inefficiency of this method makes the VQE impractical for many problems of interest [2]. In this section we introduce the method of *enhanced sampling* for amplitude estimation. This technique draws inspiration from quantum-enhanced metrology [41] and seeks to maximize the statistical power of noisy quantum devices. We motivate this method by starting from a simple analysis of standard sampling as used in the VQE. We note that, although the subroutine of estimation is a critical bottleneck, other aspects of the VQE algorithm also must be improved, including the optimization of the parameters in parameterized quantum circuits [42–45].

The energy estimation subroutine of the VQE estimates amplitudes with respect to Pauli strings. For a Hamiltonian decomposed into a linear combination of Pauli strings $H = \sum_j \mu_j P_j$ and "ansatz state" $|A\rangle$, the energy expectation value is estimated as a linear combination of Pauli expectation value estimates

$$\hat{E} = \sum_j \mu_j \hat{\Pi}_j, \tag{3}$$

where $\hat{\Pi}_j$ is the (amplitude) estimate of $\langle A|P_j|A\rangle$. The VQE uses the standard sampling method to build up Pauli expectation value estimates with respect to the ansatz state, which can be summarized as follows. Prepare $|A\rangle$ and measure operator P receiving outcome $d \in \{0, 1\}$. Repeat this M times, receiving k outcomes labeled 0 and $M - k$ outcomes labeled 1. Estimate $\Pi = \langle A|P|A\rangle$ as $\hat{\Pi} = [k - (M - k)]/M$.

We can quantify the performance of this estimation strategy using the mean squared error of the estimator as a function of time $t = TM$, where T is the time cost of each measurement. Because the estimator is unbiased, the mean squared error (MSE) is simply the variance in the estimator,

$$\mathrm{MSE}(\hat{\Pi}) = \frac{1 - \Pi^2}{M}. \tag{4}$$

For a specific mean squared error $\mathrm{MSE}(\hat{\Pi}) = \varepsilon^2$, the runtime of the algorithm needed to ensure mean squared error ε^2 is

$$t_\varepsilon = T\frac{1 - \Pi^2}{\varepsilon^2}. \tag{5}$$

The total runtime of energy estimation in the VQE is the sum of the runtimes of the individual Pauli expectation value estimation runtimes. For problems of interest, this runtime can be far too costly, even when certain parallelization techniques are used [46]. The source of this cost is the insensitivity of the standard sampling estimation process to small deviations in Π: the expected information gain about Π contained in the standard-sampling measurement outcome data is low.

Generally, we can measure the information gain of an estimation process of M repetitions of standard sampling with the Fisher information

$$\begin{aligned} I_M(\Pi) &= \mathbb{E}_D\left[\left(\frac{\partial}{\partial \Pi}\log \mathbb{P}(D|\Pi)\right)^2\right] \\ &= -\mathbb{E}_D\left[\frac{\partial^2}{\partial \Pi^2}\log \mathbb{P}(D|\Pi)\right] \\ &= \sum_D \frac{1}{\mathbb{P}(D|\Pi)}\left(\frac{\partial}{\partial \Pi}\mathbb{P}(D|\Pi)\right)^2, \end{aligned} \tag{6}$$

where $D = \{d_1, d_2, \ldots, d_M\}$ is the set of outcomes from M repetitions of the standard sampling. The Fisher information identifies the likelihood function $\mathbb{P}(D|\Pi)$ as being responsible for information gain. We can lower bound the mean squared error of an (unbiased) estimator with the Cramer-Rao bound

$$\mathrm{MSE}(\hat{\Pi}) \geq \frac{1}{I_M(\Pi)}. \tag{7}$$

Using the fact that the Fisher information is additive in the number of samples, we have $I_M(\Pi) = MI_1(\Pi)$, where $I_1(\Pi) = 1/(1 - \Pi^2)$ is the Fisher information of a single sample drawn from likelihood function $\mathbb{P}(d|\Pi) = [1 + (-1)^d\Pi]/2$. Using the Cramer-Rao bound, we can find a lower bound for the runtime of the estimation process as

$$t_\varepsilon \geq \frac{T}{I_1(\Pi)\varepsilon^2}, \tag{8}$$

which shows that in order to reduce the runtime of an estimation algorithm, we should aim to increase the Fisher information.

The purpose of enhanced sampling is to reduce the runtime of amplitude estimation by engineering likelihood functions that increase the rate of information gain. We consider the simplest case of enhanced sampling, which is illustrated in Fig. 1. To generate data, we prepare the ansatz state $|A\rangle$, apply the operation P, apply a phase flip

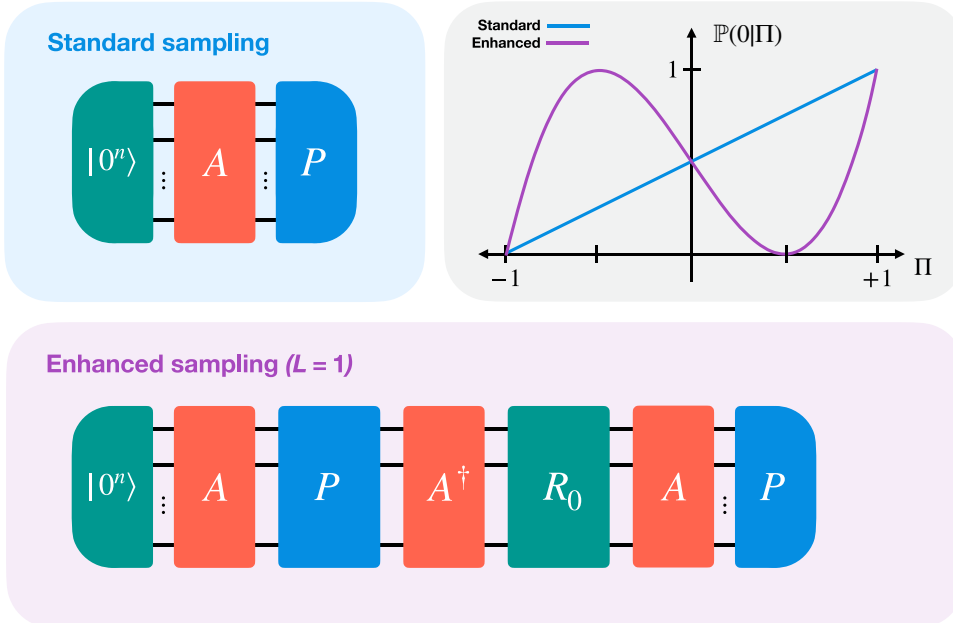


FIG. 1. Quantum circuits for standard sampling (used in the VQE) and for the simplest nontrivial version of enhanced sampling, along with their corresponding likelihood functions. The blocks represent unitary transformations, while the caps at the left and right indicate state preparation and measurement, respectively. Enhanced sampling can yield a larger statistical power in this estimation of $\Pi = \langle A|P|A\rangle$. The likelihoods of the outcome data in enhanced sampling can depend more sensitively on the unknown value Π than they do in standard sampling. This increased sensitivity can reduce the runtime needed to achieve a target average error in the estimate of Π.

about the ansatz state, and then measure P. The phase flip about the ansatz state can be achieved by applying the inverse of the ansatz circuit A^{-1}, applying a phase flip about the initial state $R_0 = 2|0^n\rangle\langle 0^n| - I$, and then reapplying the ansatz circuit A. In this case, the likelihood function becomes

$$\mathbb{P}(d|\Pi) = \frac{1 + (-1)^d \cos[3\arccos(\Pi)]}{2} = \frac{1 + (-1)^d (4\Pi^3 - 3\Pi)}{2}. \tag{9}$$

The bias is a degree-3 Chebyshev polynomial in Π. We refer to such likelihood functions as *Chebyshev likelihood functions*.

In order to compare the Chebyshev likelihood function of enhanced sampling to that of standard sampling, we consider the case of $\Pi = 0$. Here, $\mathbb{P}(0|\Pi = 0) = \mathbb{P}(1|\Pi = 0)$ and so the Fisher information is proportional to the square of the slope of the likelihood function

$$I_1(\Pi = 0) = 4\left(\frac{\partial \mathbb{P}(d = 0|\Pi)}{\partial \Pi}\right)^2. \tag{10}$$

As seen in Fig. 1, the slope of the Chebyshev likelihood function at $\Pi = 0$ is steeper than that of the standard sampling likelihood function. The single-sample Fisher information in each case evaluates to

$$I_1(\Pi = 0) = \begin{cases} 1 & \text{standard,} \\ 9 & \text{enhanced,} \end{cases} \tag{11}$$

demonstrating how a simple variant of the quantum circuit can enhance information gain. In this example, using the simplest case of enhanced sampling can reduce the number of measurements needed to achieve a target error by at least a factor of 9. As we discuss later, we can further increase the Fisher information by applying L layers of $P \circ A^\dagger \circ R_0 \circ A$ before measuring P. In fact, the Fisher information $I_1(\Pi) = (2L + 1)^2/(1 - \Pi^2) = O(L^2)$ grows quadratically in L.

We have yet to propose an estimation scheme that converts enhanced sampling measurement data into an estimation. One intricacy that enhanced sampling introduces is the option to vary L as we are collecting measurement data. In this case, given a set of measurement outcomes from circuits with varying L, the sample mean of the 0 and 1 counts loses its meaning. Instead of using the sample mean, we use *Bayesian inference* to process the measurement outcomes into information about Π. In Sec. III B we describe the use of Bayesian inference for estimation.

At this point, one may be tempted to point out that the comparison between standard sampling and enhanced

sampling is unfair because only one query to A is used in the standard sampling case while the enhanced sampling scheme uses three queries of A. It seems that if one considers a likelihood function that arises from *three* standard sampling steps, one could also yield a cubic polynomial form in the likelihood function. Indeed, suppose one performs three independent standard sampling steps, yielding results $x_1, x_2, x_3 \in \{0, 1\}$, and produces a binary outcome $z \in \{0, 1\}$ classically by sampling from a distribution $\mathbb{P}(z|x_1, x_2, x_3)$. Then the likelihood function takes the form

$$\mathbb{P}(z|\Pi) = \sum_{x_1,x_2,x_3} \mathbb{P}(z|x_1,x_2,x_3)\mathbb{P}(x_1,x_2,x_3|\Pi) = \sum_{i=0}^{3} \alpha_i \binom{3}{i} \left(\frac{1+\Pi}{2}\right)^i \left(\frac{1-\Pi}{2}\right)^{3-i}, \quad (12)$$

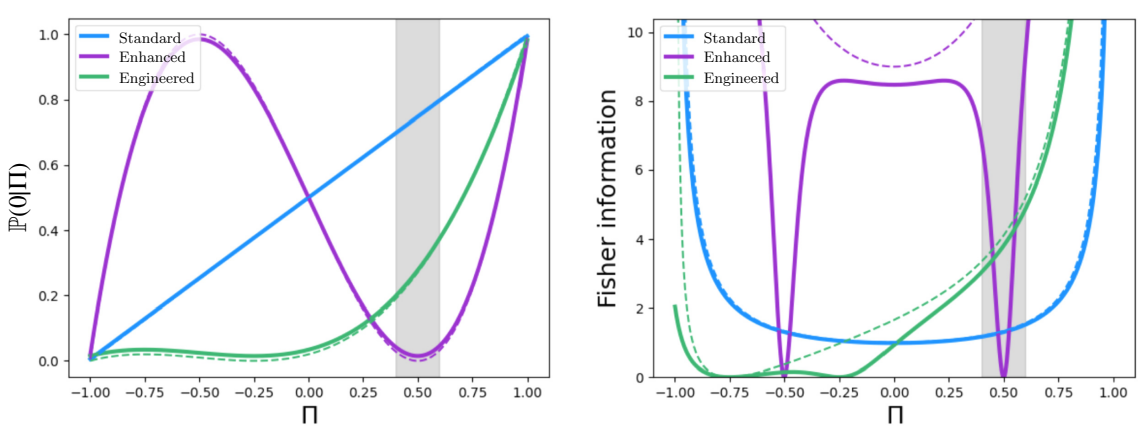
where each $\alpha_i \in [0, 1]$ is a parameter that can be tuned classically through changing the distribution $\mathbb{P}(z|x_1, x_2, x_3)$. More specifically, $\alpha_i = \sum_{x_1x_2x_3:h(x_1x_2x_3)=i} \mathbb{P}(z|x_1, x_2, x_3)$, where $h(x_1x_2x_3)$ is the Hamming weight of the bit string $x_1x_2x_3$. Suppose that we want $\mathbb{P}(z = 0|\Pi)$ to be equal to $\mathbb{P}(d = 0|\Pi)$ in Eq. (9). This implies that $\alpha_0 = 1, \alpha_1 = -2$, $\alpha_2 = 3$, and $\alpha_3 = 0$, which is clearly beyond the classical tunability of the likelihood function in Eq. (12). This evidence suggests that the likelihood function arising from the quantum scheme in Eq. (9) is beyond classical means.

As the number of circuit layers L is increased, the time per sample T grows linearly in L. This linear growth in the circuit layer number, along with the quadratic growth in Fisher information leads to a lower bound on the expected runtime,

$$t_\varepsilon \in \Omega\left(\frac{1}{L\varepsilon^2}\right), \quad (13)$$

assuming a fixed-L estimation strategy with an unbiased estimator. In practice, the operations implemented on the quantum computer are subject to error. Fortunately, Bayesian inference can incorporate such errors into the estimation process. As long as the influence of errors on the form of the likelihood function is accurately modeled, the principal effect of such errors is only to slow the rate of information gain. Error in the quantum circuit accumulates as we increase the number of circuit layers L. Consequently, beyond a certain number of circuit layers, we receive diminishing returns with respect to gains in Fisher information (or the reduction in runtime). The estimation algorithm should then seek to balance these competing factors in order to optimize the overall performance.

The introduction of error poses another issue for estimation. Without error, the Fisher information gain per sample in the enhanced sampling case with $L = 1$ is greater than or equal to 9 for all Π. As shown in Fig. 2, with the introduction of even a small degree of error, the values of Π where the likelihood function is flat incur a dramatic drop in Fisher information. We refer to such regions as estimation *dead spots*. This observation motivates the concept of ELFs to increase their statistical power. By promoting the P and R_0 operations to generalized reflections $U(x) = \exp(-ixP)$ and $R_0(y) = \exp(-iyR_0)$, we can choose rotation angles such that the information gain is



Likelihood function comparison

Fisher information comparison

FIG. 2. These plots demonstrate improvement in information gain if the likelihood function is engineered. Dotted curves are the noiseless case, and solid curves incorporate a 1% error per ansatz application [using the notation of Eq. (23), $\bar{p} = 0.99$, $p = 0.99^2 = 0.9801$, and $L = 1$]. The likelihood function of enhanced sampling yields a large Fisher information for many values of Π relative to that of standard sampling. The introduction of even a small degree of error in the quantum circuits causes the Fisher information to become zero where the enhanced sampling likelihood function is flat (indicated by the gray bands). This can be verified by evaluating Eq. (6) using the expression for the model of the noisy likelihood function in Eq. (23). By tuning the generalized reflection angles $[(x_1, x_2) = (-0.6847, 0.6847)]$, we can engineer a likelihood function to boost the information gain in the estimation "dead spot" (gray region).

boosted around such dead spots. We find that, even for deeper enhanced sampling circuits, engineering likelihood functions allow us to mitigate the effect of estimation dead spots.

III. ENGINEERED LIKELIHOOD FUNCTIONS

In this section, we propose the methodology of engineering likelihood functions for amplitude estimation. We first introduce the quantum circuits for drawing samples that correspond to engineered likelihood functions, and then describe how to tune the circuit parameters and carry out Bayesian inference with the resultant likelihood functions.

A. Quantum circuits for engineered likelihood functions

Our objective is to design a procedure for estimating the expectation value

$$\Pi = \cos(\theta) = \langle A|P|A\rangle, \tag{14}$$

where $|A\rangle = A|0^n\rangle$ in which A is an n-qubit unitary operator, P is an n-qubit Hermitian operator with eigenvalues ± 1, and $\theta = \arccos(\Pi)$ is introduced to facilitate Bayesian inference later on. In constructing our estimation algorithms, we assume that we are able to perform the following primitive operations. First, we can prepare the computational basis state $|0^n\rangle$ and apply an ansatz circuit A to it, obtaining $|A\rangle = A|0^n\rangle$. Second, we can implement the unitary operator $U(x) = \exp(-ixP)$ for any angle $x \in \mathbb{R}$. Finally, we can perform the measurement of P that is modeled as a projection-valued measure $\{(I+P)/2, (I-P)/2\}$ with respective outcome labels $\{0, 1\}$. We also make use of the unitary operator $V(y) = AR_0(y)A^\dagger$, where $R_0(y) = \exp[-iy(2|0^n\rangle\langle 0^n| - I)]$ and $y \in \mathbb{R}$. Following the convention (see, e.g., Ref. [30]), we call $U(x)$ and $V(y)$ the *generalized reflections* about the $+1$ eigenspace of P and the state $|A\rangle$, respectively, where x and y are the *angles* of these generalized reflections, respectively.

We use the ancilla-free [47] quantum circuit in Fig. 3 to generate the ELF, that is, the probability distribution of the outcome $d \in \{0, 1\}$ given the unknown quantity θ to be estimated. The circuit consists of a sequence of generalized reflections. Specifically, after preparing the ansatz state $|A\rangle = A|0^n\rangle$, we apply $2L$ generalized reflections $U(x_1), V(x_2), \ldots, U(x_{2L-1}), V(x_{2L})$ to it, varying the rotation angle x_j in each operation. For convenience, we call $V(x_{2j})U(x_{2j-1})$ the jth layer of the circuit for $j = 1, 2, \ldots, L$. The output state of this circuit is

$$Q(\vec{x})|A\rangle = V(x_{2L})U(x_{2L-1})\cdots V(x_2)U(x_1)|A\rangle, \tag{15}$$

where $\vec{x} = (x_1, x_2, \ldots, x_{2L-1}, x_{2L}) \in \mathbb{R}^{2L}$ is the vector of tunable parameters. Finally, we perform the projective measurement $\{(I+P)/2, (I-P)/2\}$ on this state, receiving an outcome $d \in \{0, 1\}$.

As in Grover's search algorithm, the generalized reflections $U(x_{2j-1})$ and $V(x_{2j})$ ensure that the quantum state remains in the two-dimensional subspace $S := \mathrm{span}\{|A\rangle, P|A\rangle\}$ [48] for any j. Let $|A^\perp\rangle$ be the state (unique, up to a phase) in S that is orthogonal to $|A\rangle$, i.e.,

$$|A^\perp\rangle = \frac{P\,|A\rangle - \langle A|\,P\,|A\rangle\,|A\rangle}{\sqrt{1 - \langle A|\,P\,|A\rangle^2}}. \tag{16}$$

To help the analysis, we view this two-dimensional subspace as a qubit, writing $|A\rangle$ and $|A^\perp\rangle$ as $|\bar{0}\rangle$ and $|\bar{1}\rangle$,

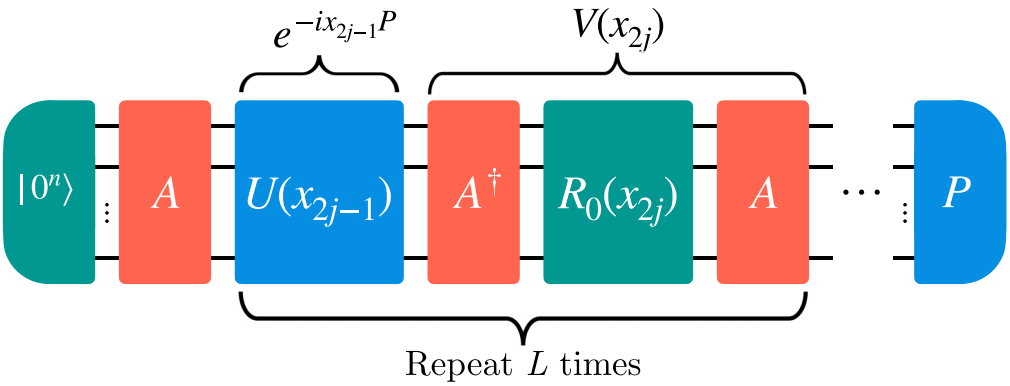


FIG. 3. An illustration of the operations used for generating samples that correspond to an engineered likelihood function. Here A is the state preparation circuit, P is the observable of interest, and $R_0(x_{i+1})$ is a generalized reflection about the state $|0^n\rangle$. The blocks represent unitary transformations, while the caps at the left and right indicate state preparation and measurement, respectively. The outcomes of measurement of P yield information about the expectation value $\Pi = \langle A|P|A\rangle$. The case of $L = 0$ simply prepares $|A\rangle$ and measures P. This corresponds to the standard sampling method used in the VQE. Even with an error-prone implementation, we can enhance the information gain rate by applying a sequence of generalized reflections before the measurement. In such *enhanced sampling*, the likelihood of outcomes depends more sensitively on Π. These circuit elements are color coded to highlight the commonalities in the way the features P (blue), A (red), and $|0^n\rangle$ (green) enter in the likelihood function.

respectively. Let $\bar{X}$, $\bar{Y}$, $\bar{Z}$, and $\bar{I}$ be the Pauli operators and identity operator on this virtual qubit, respectively. Then, focusing on the subspace $S = \text{span}\{|\bar{0}\rangle, |\bar{1}\rangle\}$, we can rewrite P as

$$P(\theta) = \cos(\theta)\bar{Z} + \sin(\theta)\bar{X}, \tag{17}$$

and rewrite the generalized reflections $U(x_{2j-1})$ and $V(x_{2j})$ as

$$U(\theta; x_{2j-1}) = \cos(x_{2j-1})\bar{I} - i\sin(x_{2j-1})[\cos(\theta)\bar{Z} + \sin(\theta)\bar{X}] \tag{18}$$

and

$$V(x_{2j}) = \cos(x_{2j})\bar{I} - i\sin(x_{2j})\bar{Z}, \tag{19}$$

where $x_{2j-1}, x_{2j} \in \mathbb{R}$ are tunable parameters. Then the unitary operator $Q(\vec{x})$ implemented by the L-layer circuit becomes

$$Q(\theta; \vec{x}) = V(x_{2L})U(\theta; x_{2L-1}) \cdots V(x_2)U(\theta; x_1). \tag{20}$$

Note that in this picture, $|A\rangle = |\bar{0}\rangle$ is fixed, while $P = P(\theta)$, $U(x) = U(\theta; x)$, and $Q(\vec{x}) = Q(\theta; \vec{x})$ depend on the unknown quantity θ. It turns out to be more convenient to design and analyze the estimation algorithms in this "logical" picture than in the original "physical" picture. Therefore, we stick to this picture for the remainder of this paper.

The engineered likelihood function (i.e., the probability distribution of measurement outcome $d \in \{0, 1\}$) depends on the output state $\rho(\theta; \vec{x})$ of the circuit and the observable $P(\theta)$. Precisely, it is

$$\mathbb{P}(d|\theta; \vec{x}) = \frac{1 + (-1)^d \Delta(\theta; \vec{x})}{2}, \tag{21}$$

where

$$\Delta(\theta; \vec{x}) = \langle\bar{0}|Q^\dagger(\theta; \vec{x})P(\theta)Q(\theta; \vec{x})|\bar{0}\rangle \tag{22}$$

is the *bias* of the likelihood function [from now on, we use $\mathbb{P}'(d|\theta; \vec{x})$ and $\Delta'(\theta; \vec{x})$ to denote the partial derivatives of $\mathbb{P}(d|\theta; \vec{x})$ and $\Delta(\theta; \vec{x})$ with respect to θ, respectively]. In particular, if $\vec{x} = (\pi/2, \pi/2, \ldots, \pi/2, \pi/2)$ then we have $\Delta(\theta; \vec{x}) = \cos[(2L + 1)\theta]$. Namely, the bias of the likelihood function for this $\vec{x}$ is the Chebyshev polynomial of degree $2L + 1$ (of the first kind) of Π. For this reason, we call the likelihood function for this $\vec{x}$ the *Chebyshev likelihood function*. In Sec. V we explore the performance gap between CLFs and general ELFs.

In reality, quantum devices are subject to noise. To make the estimation process robust against errors, we incorporate the following exponential decay noise model into the likelihood function [25,49]. Recently, this exponential decay noise model of the likelihood function has been used in several related works [50,51] and validated in small-scale experiments [52]. It has also been used in an 18-qubit experiment on spectrum estimation [53]. Furthermore, it is closely related to the noise analysis carried out in Ref. [54]. Validation at large scales remains an important line of research, which we emphasize in Sec. VII and leave to future work. Letting p be the exponential decay factor, we have

$$\mathbb{P}(d|\theta; \bar{p}, p, \vec{x}) = \frac{1}{2}[1 + (-1)^d \bar{p}p^L \Delta(\theta; \vec{x})], \tag{23}$$

where $\bar{p}$ accounts for state preparation and measurement error (cf. Appendix A) and $\Delta(\theta, \vec{x})$ is the bias of the ideal likelihood function as defined in Eq. (22). From now on, we use $f = \bar{p}p^L$ as the *fidelity* of the whole process for generating the ELF, and call p the layer fidelity. Moreover, for convenience, we write $\mathbb{P}(d|\theta; \bar{p}, p, \vec{x})$ simply as $\mathbb{P}(d|\theta; f, \vec{x})$ [we also use $\mathbb{P}'(d|\theta; f, \vec{x})$ to denote the partial derivative of $\mathbb{P}(d|\theta; f, \vec{x})$ with respect to θ]. Note that the effect of noise on the ELF is that it rescales the bias by a factor of f. This implies that the less errored the generation process, the steeper the resultant ELF, as one would expect.

Before moving on to the discussion of Bayesian inference with ELFs, it is worth mentioning the following property of engineered likelihood functions, as it will play a pivotal role in Sec. IV. In Ref. [55], we introduced the concepts of *trigono-multilinear* and *trigono-multiquadratic* functions. Basically, a multivariable function $f : \mathbb{R}^k \to \mathbb{C}$ is trigono-multilinear if, for any $j \in \{1, 2, \ldots, k\}$, we can write $f(x_1, x_2, \ldots, x_k)$ as

$$f(x_1, x_2, \ldots, x_k) = C_j(\vec{x}_{\neg j})\cos(x_j) + S_j(\vec{x}_{\neg j})\sin(x_j) \tag{24}$$

for some (complex-valued) functions C_j and S_j of $\vec{x}_{\neg j} := (x_1, \ldots, x_{j-1}, x_{j+1}, x_k)$, and we call C_j and S_j the *cosine-sine-decomposition* (CSD) coefficient functions of f with respect to x_j. Similarly, a multivariable function $f : \mathbb{R}^k \to \mathbb{C}$ is trigono-multiquadratic if, for any $j \in \{1, 2, \ldots, k\}$, we can write $f(x_1, x_2, \ldots, x_k)$ as

$$f(x_1, x_2, \ldots, x_k) = C_j(\vec{x}_{\neg j})\cos(2x_j) + S_j(\vec{x}_{\neg j})\sin(2x_j) + B_j(\vec{x}_{\neg j}) \tag{25}$$

for some (complex-valued) functions C_j, S_j, and B_j of $\vec{x}_{\neg j} := (x_1, \ldots, x_{j-1}, x_{j+1}, x_k)$, and we call C_j, S_j, and B_j the *cosine-sine-bias-decomposition* (CSBD) coefficient functions of f with respect to x_j. The concepts of trigono-multilinearity and trigono-multiquadraticity can be naturally generalized to linear operators. Namely, a linear operator is trigono-multilinear (or trigono-multiquadratic) in a set of variables if each entry of this operator (written in an arbitrary basis) is trigono-multilinear (or trigono-multiquadratic) in the same variables. Now Eqs. (18)–(20)

imply that $Q(\theta;\vec{x})$ is a trigono-multilinear operator of $\vec{x}$. Then it follows from Eq. (22) that $\Delta(\theta;\vec{x})$ is a trigono-multiquadratic function of $\vec{x}$. Furthermore, we show in Sec. IV A that the CSBD coefficient functions of $\Delta(\theta;\vec{x})$ with respect to any x_j can be evaluated in $O(L)$ time, and this greatly facilitates the construction of the algorithms in Sec. IV A for tuning the circuit parameters $\vec{x} = (x_1, x_2, \ldots, x_{2L-1}, x_{2L})$.

B. Bayesian inference with engineered likelihood functions

With the model of (noisy) engineered likelihood functions in place, we are now ready to describe our methodology for tuning the circuit parameters $\vec{x}$ and performing Bayesian inference with the resultant likelihood functions for amplitude estimation.

Let us begin with a high-level overview of our algorithm for estimating $\Pi = \cos(\theta) = \langle A|P|A\rangle$. For convenience, our algorithm mainly works with $\theta = \arccos(\Pi)$ instead of Π. We use a Gaussian distribution to represent our knowledge of θ and make this distribution gradually concentrate around the true value of θ as the inference process proceeds. We start with an initial distribution of Π (which can be generated by standard sampling or domain knowledge) and convert it to the initial distribution of θ. Then we repeat the following procedure until a convergence criterion is satisfied. At each round, we first find the circuit parameters $\vec{x}$ that maximize the information gain from the measurement outcome d in a certain sense (based on our current knowledge of θ). Then we run the quantum circuit in Fig. 3 with the optimized parameters $\vec{x}$ and receive a measurement outcome $d \in \{0, 1\}$. Finally, we update the distribution of θ by using Bayes' rule, conditioned on the received outcome d. Once this loop is finished, we convert the final distribution of θ to the final distribution of Π, and set the mean of this distribution as the final estimate of Π. See Fig. 4 for the conceptual diagram of this algorithm.

Next, we describe each component of the above algorithm in more detail. Throughout the inference process, we use a Gaussian distribution to keep track of our belief of the value of θ. Namely, at each round, θ has prior distribution

$$p(\theta) = p(\theta;\mu,\sigma) := \frac{1}{\sqrt{2\pi}\sigma} e^{-(\theta-\mu)^2/2\sigma^2} \tag{26}$$

for some prior mean $\mu \in \mathbb{R}$ and prior variance $\sigma^2 \in \mathbb{R}^+$. After receiving the measurement outcome d, we compute the posterior distribution of θ by using Bayes' rule

$$p(\theta|d;f,\vec{x}) = \frac{\mathbb{P}(d|\theta;f,\vec{x})p(\theta)}{\mathbb{P}(d;f,\vec{x})}, \tag{27}$$

where the normalization factor, or model evidence, is defined as $\mathbb{P}(d;f,\vec{x}) = \int \mathbb{P}(d|\theta;f,\vec{x})p(\theta)d\theta$ (recall that

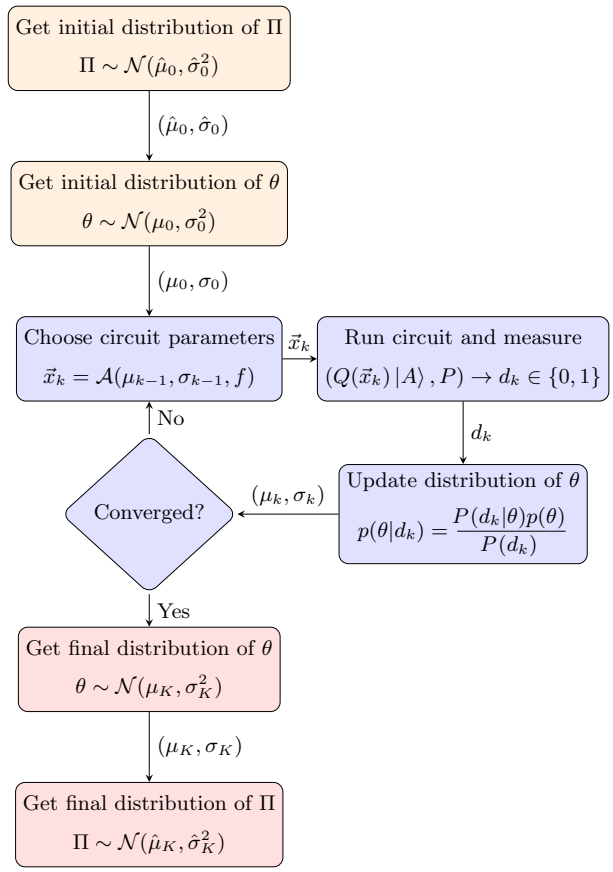


FIG. 4. High-level flowchart of the algorithm for estimating $\Pi = \cos(\theta) = \langle A|P|A\rangle$. Here f is the fidelity of the process for generating the ELF. This algorithm mainly works with θ instead of Π, and there are conversions between the distributions of θ and Π at the beginning and end of the algorithm. The final estimate of Π is $\hat{\mu}_K$. Note that only the "Run circuit and measure" step involves a quantum device.

f is the fidelity of the process for generating the ELF). Although the true posterior distribution will not be a Gaussian, we approximate it as such. Following the methodology in Ref. [56], we replace the true posterior with a Gaussian distribution of the same mean and variance [57], and set it as the prior of θ for the next round. We repeat this measurement-and-Bayesian-update procedure until the distribution of θ is sufficiently concentrated around a single value.

Since the algorithm mainly works with θ and we are eventually interested in Π, we need to make conversions between the estimators of θ and Π. This is done as follows. Suppose that at round k the prior distribution of θ is $\mathcal{N}(\mu_k, \sigma_k^2)$ and that the prior distribution of Π is $\mathcal{N}(\hat{\mu}_k, \hat{\sigma}_k^2)$ (note that μ_k, σ_k, $\hat{\mu}_k$, and $\hat{\sigma}_k$ are random variables as they depend on the history of random measurement outcomes up to round k). The estimators of θ and Π at this round are μ_k and $\hat{\mu}_k$, respectively. Given the distribution $\mathcal{N}(\mu_k, \sigma_k^2)$ of θ, we compute the mean $\hat{\mu}_k$ and variance $\hat{\sigma}_k^2$ of $\cos(\theta)$, and set $\mathcal{N}(\hat{\mu}_k, \hat{\sigma}_k^2)$ as the distribution of Π.

This step can be done analytically, as if $X \sim \mathcal{N}(\mu, \sigma^2)$. Then

$$\mathbb{E}[\cos(X)] = e^{-\sigma^2/2}\cos(\mu), \tag{28}$$

$$\mathrm{Var}[\cos(X)] = \frac{(1-e^{-\sigma^2})[1-e^{-\sigma^2}\cos(2\mu)]}{2}. \tag{29}$$

Conversely, given the distribution $\mathcal{N}(\hat{\mu}_k, \hat{\sigma}_k^2)$ of Π, we compute the mean μ_k and variance σ_k^2 of $\arccos(\Pi)$ (clipping Π to $[-1, 1]$), and set $\mathcal{N}(\mu_k, \sigma_k^2)$ as the distribution of θ. This step is done numerically. Even though the cos or arccos function of a Gaussian variable is not truly Gaussian, we approximate it as such and find that this has negligible impact on the performance of the algorithm.

Our method for tuning the circuit parameters $\vec{x}$ is as follows. Ideally, we want to choose them carefully so that the MSE of the estimator μ_k of θ decreases as fast as possible as k grows. In practice, however, it is hard to compute this quantity directly, and we must resort to a proxy of its value. The MSE of an estimator is a sum of the variance of the estimator and the squared bias of the estimator. We find that, for large enough k, the squared bias of μ_k is smaller than its variance, i.e., $\mathrm{Bias}(\mu_k)^2 = (\mathbb{E}[\mu_k] - \theta^*)^2 < \mathrm{Var}(\mu_k)$, where θ^* is the true value of θ. We also find that, for large enough k, the variance σ_k^2 of θ is often close to the variance of μ_k, i.e., $\sigma_k^2 \approx \mathrm{Var}(\mu_k)$ with high probability (see Appendix E for evidences of these claims). Combining these facts, we know that, for large enough k, $\mathrm{MSE}(\mu_k) = \mathbb{E}[(\mu_k - \theta^*)^2] \le 2\sigma_k^2$ with high probability. So we find the parameters $\vec{x}$ that minimize the variance σ_k^2 of θ instead.

Specifically, suppose that θ has prior distribution $\mathcal{N}(\mu, \sigma^2)$. Upon receiving the measurement outcome $d \in \{0, 1\}$, the expected posterior variance [55] of θ is

$$\mathbb{E}_d[\mathrm{Var}(\theta|d; f, \vec{x})] = \sigma^2\left(1 - \sigma^2\frac{f^2[\partial_\mu b(\mu,\sigma;\vec{x})]^2}{1 - f^2[b(\mu,\sigma;\vec{x})]^2}\right), \tag{30}$$

where

$$b(\mu,\sigma;\vec{x}) = \int_{-\infty}^{\infty} p(\theta;\mu,\sigma)\Delta(\theta;\vec{x})d\theta \tag{31}$$

in which $\Delta(\theta;\vec{x})$ is the bias of the ideal likelihood function as defined in Eq. (22), and f is the fidelity of the process for generating the likelihood function. We introduce an important quantity for engineering likelihood functions that we refer to as the *variance reduction factor*,

$$\mathcal{V}(\mu,\sigma;f,\vec{x}) := \frac{f^2[\partial_\mu b(\mu,\sigma;\vec{x})]^2}{1 - f^2[b(\mu,\sigma;\vec{x})]^2}. \tag{32}$$

Then we have

$$\mathbb{E}_d[\mathrm{Var}(\theta|d; f, \vec{x})] = \sigma^2[1 - \sigma^2\mathcal{V}(\mu,\sigma;f,\vec{x})]. \tag{33}$$

The larger $\mathcal{V}$ is, the faster the variance of θ decreases on average. Furthermore, to quantify the growth rate (per time step) of the inverse variance of θ, we introduce the quantity

$$\begin{aligned} R(\mu,\sigma;f,\vec{x}) &:= \frac{1}{T(L)}\left(\frac{1}{\mathbb{E}_d[\mathrm{Var}(\theta|d;f,\vec{x})]} - \frac{1}{\sigma^2}\right) \\ &:= \frac{1}{T(L)}\frac{\mathcal{V}(\mu,\sigma;f,\vec{x})}{1-\sigma^2\mathcal{V}(\mu,\sigma;f,\vec{x})}, \end{aligned} \tag{34}$$

where $T(L)$ is the duration of the L-layer circuit in Fig. 3 (recall that $\vec{x} \in \mathbb{R}^{2L}$). Note that R is a monotonic function of $\mathcal{V}$ for $\mathcal{V} \in (0, 1)$. Therefore, when L is fixed, we can maximize R (with respect to $\vec{x}$) by maximizing $\mathcal{V}$. In addition, when σ is small, R is approximately proportional to $\mathcal{V}$, i.e., $R \approx \mathcal{V}/T(L)$. For the remainder of this work, we assume that the ansatz circuit A and its inverse $A^\dagger$ contribute most significantly to the duration of the overall circuit. So we take $T(L)$ to be proportional to the number of times A or $A^\dagger$ is invoked in the circuit, setting $T(L) = 2L + 1$, where time is in units of A's duration.

So now we need to find the parameters $\vec{x} = (x_1, x_2, \ldots, x_{2L}) \in \mathbb{R}^{2L}$ that maximize the variance reduction factor $\mathcal{V}(\mu,\sigma;f,\vec{x})$ for given $\mu \in \mathbb{R}$, $\sigma \in \mathbb{R}^+$, and $f \in [0, 1]$. This optimization problem turns out to be difficult to solve in general. Fortunately, in practice, we may assume that the prior variance σ^2 of θ is small (e.g., at most 0.01), and in this case, $\mathcal{V}(\mu,\sigma;f,\vec{x})$ can be approximated by the Fisher information of the likelihood function $\mathbb{P}(d|\theta;f,\vec{x})$ at $\theta = \mu$, as shown in Appendix F, i.e.,

$$\mathcal{V}(\mu,\sigma;f,\vec{x}) \approx \mathcal{I}(\mu;f,\vec{x}) \quad \text{when } \sigma \text{ is small}, \tag{35}$$

where

$$\begin{aligned} \mathcal{I}(\theta;f,\vec{x}) &= \mathbb{E}_d\left[\left(\frac{\partial}{\partial\theta}\log\mathbb{P}(d|\theta;f,\vec{x})\right)^2\right] \\ &= \frac{f^2[\Delta'(\theta;\vec{x})]^2}{1 - f^2[\Delta(\theta;\vec{x})]^2} \end{aligned} \tag{36}$$

is the Fisher information of the two-outcome likelihood function $\mathbb{P}(d|\theta;f,\vec{x})$ as defined in Eq. (23). Therefore, rather than directly optimizing the variance reduction factor $\mathcal{V}(\mu,\sigma;f,\vec{x})$, we optimize the Fisher information $\mathcal{I}(\mu;f,\vec{x})$, which can be done efficiently by the algorithms in Sec. IV A 1. Furthermore, when the fidelity f of the process for generating the ELF is low, we have $\mathcal{I}(\theta;f,\vec{x}) \approx f^2[\Delta'(\theta;\vec{x})]^2$. It follows that

$$\mathcal{V}(\mu,\sigma;f,\vec{x}) \approx f^2[\Delta'(\mu;\vec{x})]^2 \text{ when } \sigma \text{ and } f \text{ are small}. \tag{37}$$

So in this case, we can simply optimize $|\Delta'(\mu;\vec{x})|$, which is proportional to the slope of the likelihood function

$\mathbb{P}(d|\theta;f,\vec{x})$ at $\theta=\mu$, and this task can be accomplished efficiently by the algorithms in Sec. IV A 2.

Finally, we make a prediction on how fast the MSE of the estimator $\hat{\mu}_k$ of Π decreases as k grows, under the assumption that the number L of circuit layers is fixed during the inference process. Note that $\mathrm{MSE}(\hat{\mu}_k)=\Theta(1/k)$ as $k\to\infty$ in this case. Let θ^* and Π^* be the true values of θ and Π, respectively. Then, as $k\to\infty$, we have $\mu_k\to\theta^*$, $\sigma_k\to 0$, $\hat{\mu}_k\to\Pi^*$, and $\hat{\sigma}_k\to 0$ with high probability. When this event happens, we obtain, for large k,

$$\frac{1}{\sigma_{k+1}^2}-\frac{1}{\sigma_k^2}\approx\mathcal{I}(\mu_k;f,\vec{x}_k). \tag{38}$$

Consequently, by Eq. (29), we know that, for large k,

$$\frac{1}{\hat{\sigma}_{k+1}^2}-\frac{1}{\hat{\sigma}_k^2}\approx\frac{\mathcal{I}(\mu_k;f,\vec{x}_k)}{\sin^2(\mu_k)}, \tag{39}$$

where $\mu_k\approx\arccos(\hat{\mu}_k)$. Since $\mathrm{Bias}(\hat{\mu}_k)^2\ll\mathrm{Var}(\hat{\mu}_k)\approx\hat{\sigma}_k^2$ for large k, we predict that

$$\mathrm{MSE}(\hat{\mu}_k)\approx\frac{1-\hat{\mu}_k^2}{k\mathcal{I}[\arccos(\hat{\mu}_k);f,\vec{x}_k]}. \tag{40}$$

This means that the asymptotic growth rate (per time step) of the inverse MSE of $\hat{\mu}_k$ should be roughly

$$\hat{R}_0(\Pi^*;f,\vec{x}):=\frac{\mathcal{I}[\arccos(\Pi^*);f,\vec{x}]}{(2L+1)[1-(\Pi^*)^2]}, \tag{41}$$

where $\vec{x}\in\mathbb{R}^{2L}$ is chosen such that $\mathcal{I}[\arccos(\Pi^*);f,\vec{x}]$ is maximized. We compare this rate with the empirical growth rate (per time step) of the inverse MSE of $\hat{\mu}_k$ in Sec. V.

IV. EFFICIENT HEURISTIC ALGORITHMS FOR CIRCUIT PARAMETER TUNING AND BAYESIAN INFERENCE

In this section, we present heuristic algorithms for tuning the parameters $\vec{x}$ of the circuit in Fig. 3 and describe how to efficiently carry out Bayesian inference with the resultant likelihood functions.

A. Efficient maximization of proxies of the variance reduction factor

Our algorithms for tuning the circuit parameters $\vec{x}$ are based on maximizing two proxies of the variance reduction factor $\mathcal{V}$—the Fisher information and slope of the likelihood function $\mathbb{P}(d|\theta;f,\vec{x})$. All of these algorithms require efficient procedures for evaluating the CSBD coefficient functions of $\Delta(\theta;\vec{x})$ and $\Delta'(\theta;\vec{x})$ with respect to x_j for $j=1,2,\ldots,2L$. Recall that we have shown in Sec. III A that $\Delta(\theta;\vec{x})$ is trigono-multiquadratic in $\vec{x}$. Namely, for any $j\in\{1,2,\ldots,2L\}$, there exist functions $C_j(\theta;\vec{x}_{\neg j})$, $S_j(\theta;\vec{x}_{\neg j})$, and $B_j(\theta;\vec{x}_{\neg j})$ of $\vec{x}_{\neg j}:=(x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_{2L})$ such that

$$\Delta(\theta;\vec{x})=C_j(\theta;\vec{x}_{\neg j})\cos(2x_j)+S_j(\theta;\vec{x}_{\neg j})\sin(2x_j)+B_j(\theta;\vec{x}_{\neg j}). \tag{42}$$

It follows that

$$\Delta'(\theta;\vec{x})=C_j'(\theta;\vec{x}_{\neg j})\cos(2x_j)+S_j'(\theta;\vec{x}_{\neg j})\sin(2x_j)+B_j'(\theta;\vec{x}_{\neg j}) \tag{43}$$

is also trigono-multiquadratic in $\vec{x}$, where $C_j'(\theta;\vec{x}_{\neg j})=\partial_\theta C_j(\theta;\vec{x}_{\neg j})$, $S_j'(\theta;\vec{x}_{\neg j})=\partial_\theta S_j(\theta;\vec{x}_{\neg j})$, $B_j'(\theta;\vec{x}_{\neg j})=\partial_\theta B_j(\theta;\vec{x}_{\neg j})$ are the partial derivatives of $C_j(\theta;\vec{x}_{\neg j})$, $S_j(\theta;\vec{x}_{\neg j})$, $B_j(\theta;\vec{x}_{\neg j})$ with respect to θ, respectively. It turns out that, given θ and $\vec{x}_{\neg j}$, each of $C_j(\theta;\vec{x}_{\neg j})$, $S_j(\theta;\vec{x}_{\neg j})$, $B_j(\theta;\vec{x}_{\neg j})$, $C_j'(\theta;\vec{x}_{\neg j})$, $S_j'(\theta;\vec{x}_{\neg j})$, and $B_j'(\theta;\vec{x}_{\neg j})$ can be computed in $O(L)$ time.

Lemma 1. *Given θ and $\vec{x}_{\neg j}$, each of $C_j(\theta;\vec{x}_{\neg j})$, $S_j(\theta;\vec{x}_{\neg j})$, $B_j(\theta;\vec{x}_{\neg j})$, $C_j'(\theta;\vec{x}_{\neg j})$, $S_j'(\theta;\vec{x}_{\neg j})$ and $B_j'(\theta;\vec{x}_{\neg j})$ can be computed in $O(L)$ time.*

Proof. See Appendix B. ■

1. Maximizing the Fisher information of the likelihood function

We propose two algorithms for maximizing the Fisher information of the likelihood function $\mathbb{P}(d|\theta;f,\vec{x})$ at a given point $\theta=\mu$ (i.e., the prior mean of θ). Namely, our goal is to find $\vec{x}\in\mathbb{R}^{2L}$ that maximizes

$$\mathcal{I}(\mu;f,\vec{x})=\frac{f^2[\Delta'(\mu;\vec{x})]^2}{1-f^2\Delta(\mu;\vec{x})^2}. \tag{44}$$

The first algorithm is based on *gradient ascent*. Namely, it starts with a random initial point, and keeps taking steps proportional to the gradient of $\mathcal{I}$ at the current point, until a convergence criterion is satisfied. Specifically, let $\vec{x}^{(t)}$ be the parameter vector at iteration t. We update it as

$$\vec{x}^{(t+1)}=\vec{x}^{(t)}+\delta(t)\nabla\mathcal{I}(\mu;f,\vec{x})\big|_{\vec{x}=\vec{x}^{(t)}}, \tag{45}$$

where $\delta:\mathbb{Z}^{\geq 0}\to\mathbb{R}^+$ is the step size schedule [58]. This requires the calculation of the partial derivative of $\mathcal{I}(\mu;f,\vec{x})$ with respect to each x_j, which can be done as follows. We first use the procedures in Lemma 1 to compute $C_j:=C_j(\mu;\vec{x}_{\neg j})$, $S_j:=S_j(\mu;\vec{x}_{\neg j})$, $B_j:=B_j(\mu;\vec{x}_{\neg j})$,

$C'_j := C'_j(\mu;\vec{x}_{\neg j})$, $S'_j := S'_j(\mu;\vec{x}_{\neg j})$, and $B'_j := B'_j(\mu;\vec{x}_{\neg j})$ for each j. Then we get

$$\begin{aligned}\Delta &:= \Delta(\mu;\vec{x}) \\ &:= C_j\cos(2x_j) + S_j\sin(2x_j) + B_j, \qquad (46)\\ \Delta' &:= \Delta'(\mu;\vec{x}) \\ &:= C'_j\cos(2x_j) + S'_j\sin(2x_j) + B'_j, \qquad (47)\\ \chi_j &:= \frac{\partial\Delta(\mu;\vec{x})}{\partial x_j} \\ &:= 2[-C_j\sin(2x_j) + S_j\cos(2x_j)], \qquad (48)\\ \chi'_j &:= \frac{\partial\Delta'(\mu;\vec{x})}{\partial x_j} \\ &:= 2[-C'_j\sin(2x_j) + S'_j\cos(2x_j)]. \qquad (49)\end{aligned}$$

Knowing these quantities, we can compute the partial derivative of $\mathcal{I}(\mu;f,\vec{x})$ with respect to x_j as

$$\begin{aligned}\gamma_j &:= \frac{\partial\mathcal{I}(\mu;f,\vec{x})}{\partial x_j} \\ &:= \frac{2f^2[(1-f^2\Delta^2)\Delta'\chi'_j + f^2\Delta\chi_j(\Delta')^2]}{(1-f^2\Delta^2)^2}. \qquad (50)\end{aligned}$$

Repeat this procedure for $j = 1,2,\ldots,2L$. Then we obtain $\nabla\mathcal{I}(\mu;f,\vec{x}) = (\gamma_1,\gamma_2,\ldots,\gamma_{2L})$. Each iteration of the algorithm takes $O(L^2)$ time. The number of iterations in the algorithm depends on the initial point, the termination criterion, and the step size schedule δ. See Algorithm 1 for more details.

The second algorithm is based on *coordinate ascent*. Unlike gradient ascent, this algorithm does not require step sizes, and allows each variable to change dramatically in a single step. As a consequence, it may converge faster than the previous algorithm. Specifically, this algorithm starts with a random initial point, and successively maximizes the objective function $\mathcal{I}(\mu;f,\vec{x})$ along coordinate directions, until a convergence criterion is satisfied. At the jth step of each round, it solves the following single-variable optimization problem for a coordinate x_j:

$$\operatorname*{argmax}_z \frac{f^2[C'_j\cos(2z) + S'_j\sin(2z) + B'_j]^2}{1-f^2[C_j\cos(2z) + S_j\sin(2z) + B_j]^2}. \qquad (51)$$

Here $C_j = C_j(\mu;\vec{x}_{\neg j})$, $S_j = S_j(\mu;\vec{x}_{\neg j})$, $B_j = B_j(\mu;\vec{x}_{\neg j})$, $C'_j = C'_j(\mu;\vec{x}_{\neg j})$, $S'_j = S'_j(\mu;\vec{x}_{\neg j})$, $B'_j = B'_j(\mu;\vec{x}_{\neg j})$ can be computed in $O(L)$ time by the procedures in Lemma 1. This single-variable optimization problem can be tackled by standard gradient-based methods, and we set x_j to be its solution. Repeat this procedure for $j = 1,2,\ldots,2L$. This algorithm produces a sequence $\vec{x}^{(0)}$, $\vec{x}^{(1)}$, $\vec{x}^{(2)}$, $\ldots$, such that $\mathcal{I}(\mu;f,\vec{x}^{(0)}) \le \mathcal{I}(\mu;f,\vec{x}^{(1)}) \le \mathcal{I}(\mu;f,\vec{x}^{(2)}) \le \cdots$. Namely, the value of $\mathcal{I}(\mu;f,\vec{x}^{(t)})$ increases monotonically as t grows. Each round of the algorithm takes $O(L^2)$ time. The number of rounds in the algorithm depends on the initial point and the termination criterion. See Algorithm 2 for more details.

We have used Algorithms 1 and 2 to find the parameters $\vec{x}_\theta \in \mathbb{R}^{2L}$ that maximize $\mathcal{I}(\theta;f,\vec{x})$ [59] for various $\theta \in (0,\pi)$ (fixing f) and obtained Fig. 5. This figure indicates that the Fisher information of the ELF is larger than that of the CLF for the majority of $\theta \in (0,\pi)$. Consequently, the estimation algorithm based on the ELF is more efficient than that based on the CLF, as will be demonstrated in Sec. V.

2. Maximizing the slope of the likelihood function

We also propose two algorithms for maximizing the slope of the likelihood function $\mathbb{P}(d|\theta;f,\vec{x})$ at a given point $\theta = \mu$ (i.e., the prior mean of θ). Namely, our goal is to find $\vec{x} \in \mathbb{R}^{2L}$ that maximizes $|\mathbb{P}'(\mu;f,\vec{x})| = f|\Delta'(\mu;\vec{x})|/2$. These algorithms are similar to Algorithms 1 and 2 for Fisher information maximization, in the sense that they are also based on gradient ascent and coordinate ascent, respectively. They are formally described in Algorithms 3 and 4 in Appendix C, respectively. We have used them to find the parameters $\vec{x}_\theta \in \mathbb{R}^{2L}$ that maximize $|\Delta'(\theta;\vec{x})|$ for various $\theta \in (0,\pi)$ and obtained Fig. 6. This figure implies that the slope-based ELF is steeper than the CLF for the majority of $\theta \in (0,\pi)$ and hence has more statistical power than the CLF (at least) in the low-fidelity setting by Eq. (37).

B. Approximate Bayesian inference with engineered likelihood functions

With the algorithms for tuning the circuit parameters $\vec{x}$ in place, we now describe how to efficiently carry out Bayesian inference with the resultant likelihood functions. In principle, we can compute the posterior mean and variance of θ directly after receiving a measurement outcome d. But this approach is time consuming, as it involves numerical integration. By taking advantage of certain properties of the engineered likelihood functions, we can greatly accelerate this process.

Suppose that θ has prior distribution $\mathcal{N}(\mu,\sigma^2)$, where $\sigma \ll 1/L$, and that the fidelity of the process for generating the ELF is f. We find that the parameters $\vec{x} = (x_1,x_2,\ldots,x_{2L})$ that maximize $\mathcal{I}(\mu;f,\vec{x})$ [or $|\Delta'(\mu;\vec{x})|$] satisfy the following property: when θ is close to μ, i.e., $\theta \in [\mu - O(\sigma), \mu + O(\sigma)]$, we have

$$\mathbb{P}(d|\theta;f,\vec{x}) \approx \frac{1+(-1)^d f\,\sin(r\theta+b)}{2} \qquad (57)$$

for some $r,b \in \mathbb{R}$. Namely, $\Delta(\theta;\vec{x})$ can be approximated by a sinusoidal function in this region of θ. Figure 7 illustrates one such example.

Algorithm 1: Gradient ascent for Fisher information maximization in the ancilla-free case

Input: The prior mean μ of θ, the number L of circuit layers, the fidelity f of the process for generating the ELF, the step size schedule $\delta : \mathbb{Z}^{\geq 0} \to \mathbb{R}^+$, the error tolerance ϵ for termination.

Output: A set of parameters $\vec{x} = (x_1, x_2, \ldots, x_{2L}) \in \mathbb{R}^{2L}$ that are a local maximum point of the function $\mathcal{I}(\mu; f, \vec{x})$.

Choose random initial point $\vec{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_{2L}^{(0)}) \in (-\pi, \pi]^{2L}$;
$t \leftarrow 0$;
while *True* **do**
  for $j \leftarrow 1$ **to** $2L$ **do**
    Let $\vec{x}_{\neg j}^{(t)} = (x_1^{(t)}, \ldots, x_{j-1}^{(t)}, x_{j+1}^{(t)}, \ldots, x_{2L}^{(t)})$;
    Compute $C_j^{(t)} := C_j(\mu; \vec{x}_{\neg j}^{(t)})$, $S_j^{(t)} := S_j(\mu; \vec{x}_{\neg j}^{(t)})$, $B_j^{(t)} := B_j(\mu; \vec{x}_{\neg j}^{(t)})$, $C_j'^{(t)} := C_j'(\mu; \vec{x}_{\neg j}^{(t)})$, $S_j'^{(t)} := S_j'(\mu; \vec{x}_{\neg j}^{(t)})$, $B_j'^{(t)} := B_j'(\mu; \vec{x}_{\neg j}^{(t)})$ by using the procedures in Lemma 1;
    Compute $\Delta(\mu; \vec{x})$, $\Delta'(\mu; \vec{x})$ and their partial derivatives with respect to x_j at $\vec{x} = \vec{x}^{(t)}$ as follows:

$$\Delta^{(t)} := \Delta(\mu; \vec{x}^{(t)}) = C_j^{(t)} \cos(2x_j) + S_j^{(t)} \sin(2x_j) + B_j^{(t)}, \tag{52}$$

$$\Delta'^{(t)} := \Delta'(\mu; \vec{x}^{(t)}) = C_j'^{(t)} \cos(2x_j) + S_j'^{(t)} \sin(2x_j) + B_j'^{(t)}, \tag{53}$$

$$\chi_j^{(t)} := \frac{\partial \Delta(\mu; \vec{x})}{\partial x_j}|_{\vec{x}=\vec{x}^{(t)}} = 2\left(-C_j^{(t)} \sin(2x_j) + S_j^{(t)} \cos(2x_j)\right), \tag{54}$$

$$\chi_j'^{(t)} := \frac{\partial \Delta'(\mu; \vec{x})}{\partial x_j}|_{\vec{x}=\vec{x}^{(t)}} = 2\left(-C_j'^{(t)} \sin(2x_j) + S_j'^{(t)} \cos(2x_j)\right); \tag{55}$$

    Compute the partial derivative of $\mathcal{I}(\mu; f, \vec{x})$ with respect to x_j at $\vec{x} = \vec{x}^{(t)}$ as follows:

$$\gamma_j^{(t)} := \frac{\partial \mathcal{I}(\mu; f, \vec{x})}{\partial x_j}|_{\vec{x}=\vec{x}^{(t)}} = \frac{2f^2\left[(1 - f^2(\Delta^{(t)})^2)\Delta'^{(t)}\chi_j'^{(t)} + f^2\Delta^{(t)}\chi_j^{(t)}(\Delta'^{(t)})^2\right]}{[1 - f^2(\Delta^{(t)})^2]^2} \tag{56}$$

  end
  Set $\vec{x}^{(t+1)} = \vec{x}^{(t)} + \delta(t)\nabla\mathcal{I}(\mu; f, \vec{x})|_{\vec{x}=\vec{x}^{(t)}}$, where $\nabla\mathcal{I}(\mu; f, \vec{x})|_{\vec{x}=\vec{x}^{(t)}} = (\gamma_1^{(t)}, \gamma_2^{(t)}, \ldots, \gamma_{2L}^{(t)})$;
  if $|\mathcal{I}(\mu; f, \vec{x}^{(t+1)}) - \mathcal{I}(\mu; f, \vec{x}^{(t)})| < \epsilon$ **then**
    break;
  end
  $t \leftarrow t + 1$;
end
Return $\vec{x}^{(t+1)} = (x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_{2L}^{(t+1)})$ as the optimal parameters.

We can find the best fitting r and b by solving the least-squares problem

$$(r^*, b^*) = \underset{r,b}{\operatorname{argmin}} \sum_{\theta \in \Theta} |\arcsin[\Delta(\theta; \vec{x})] - r\theta - b|^2, \tag{58}$$

where $\Theta = \{\theta_1, \theta_2, \ldots, \theta_k\} \subseteq [\mu - O(\sigma), \mu + O(\sigma)]$. This least-squares problem has the analytical solution

$$\begin{pmatrix} r^* \\ b^* \end{pmatrix} = A^+ z = (A^T A)^{-1} A^T z, \tag{59}$$

where

$$A = \begin{pmatrix} \theta_1 & 1 \\ \theta_2 & 1 \\ \vdots & \vdots \\ \theta_k & 1 \end{pmatrix}, \qquad z = \begin{pmatrix} \arcsin[\Delta(\theta_1; \vec{x})] \\ \arcsin[\Delta(\theta_2; \vec{x})] \\ \vdots \\ \arcsin[\Delta(\theta_k; \vec{x})] \end{pmatrix}. \tag{60}$$

In Fig. 7 we present an example of the true and fitted likelihood functions.

Once we obtain the optimal r and b, we can approximate the posterior mean and variance of θ by those for

$$\mathbb{P}(d|\theta; f) = \frac{1 + (-1)^d f \sin(r\theta + b)}{2}, \tag{61}$$

which have analytical formulas. Specifically, suppose that θ has prior distribution $\mathcal{N}(\mu_k, \sigma_k^2)$ at round k. Let d_k be the measurement outcome, and let (r_k, b_k) be the best-fitting parameters at this round. Then we approximate the posterior mean and variance of θ by

$$\mu_{k+1} = \mu_k + \frac{(-1)^{d_k} f e^{-r_k^2\sigma_k^2/2} r_k \sigma_k^2 \cos(r_k\mu_k + b_k)}{1 + (-1)^{d_k} f e^{-r_k^2\sigma_k^2/2} \sin(r_k\mu_k + b_k)}, \tag{62}$$

$$\sigma_{k+1}^2 = \sigma_k^2 \times \left(1 - \frac{f r_k^2 \sigma_k^2 e^{-r_k^2\sigma_k^2/2}[f e^{-r_k^2\sigma_k^2/2} + (-1)^{d_k} \sin(r_k\mu_k + b_k)]}{[1 + (-1)^{d_k} f e^{-r_k^2\sigma_k^2/2} \sin(r_k\mu_k + b_k)]^2}\right). \tag{63}$$

Algorithm 2: Coordinate ascent for Fisher information maximization in the ancilla-free case

Input: The prior mean μ of θ, the number L of circuit layers, the fidelity f of the process for generating the ELF, the error tolerance ϵ for termination.

Output: A set of parameters $\vec{x} = (x_1, x_2, \ldots, x_{2L}) \in (-\pi, \pi]^{2L}$ that are a local maximum point of the function $\mathcal{I}(\mu; f, \vec{x})$.

Choose random initial point $\vec{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_{2L}^{(0)}) \in (-\pi, \pi]^{2L}$;
$t \leftarrow 1$;
while *True* **do**
  for $j \leftarrow 1$ **to** $2L$ **do**
    Let $\vec{x}_{\neg j}^{(t)} = (x_1^{(t)}, \ldots, x_{j-1}^{(t)}, x_{j+1}^{(t-1)}, \ldots, x_{2L}^{(t-1)})$;
    Compute $C_j^{(t)} := C_j(\mu; \vec{x}_{\neg j}^{(t)})$, $S_j^{(t)} := S_j(\mu; \vec{x}_{\neg j}^{(t)})$, $B_j^{(t)} := B_j(\mu; \vec{x}_{\neg j}^{(t)})$, $C_j'^{(t)} := C_j'(\mu; \vec{x}_{\neg j}^{(t)})$, $S_j'^{(t)} := S_j'(\mu; \vec{x}_{\neg j}^{(t)})$, $B_j'^{(t)} := B_j'(\mu; \vec{x}_{\neg j}^{(t)})$ by using the procedures in Lemma 1;
    Solve the single-variable optimization problem

$$\arg\max_z \frac{f^2 \left(C_j'^{(t)} \cos(2z) + S_j'^{(t)} \sin(2z) + B_j'^{(t)}\right)^2}{1 - f^2 \left(C_j^{(t)} \cos(2z) + S_j^{(t)} \sin(2z) + B_j^{(t)}\right)^2}$$

    by standard gradient-based methods and set $x_j^{(t)}$ to be its solution;
  end
  if $|\mathcal{I}(\mu; f, \vec{x}^{(t)}) - \mathcal{I}(\mu; f, \vec{x}^{(t-1)})| < \epsilon$ **then**
    break;
  end
  $t \leftarrow t + 1$;
end
Return $\vec{x}^{(t)} = (x_1^{(t)}, x_2^{(t)}, \ldots, x_{2L}^{(t)})$ as the optimal parameters.

After that, we proceed to the next round, setting $\mathcal{N}(\mu_{k+1}, \sigma_{k+1}^2)$ as the prior distribution of θ for that round.

Note that, as Fig. 7 illustrates, the difference between the true and fitted likelihood functions can be large when θ is far from μ, i.e., $|\theta - \mu| \gg \sigma$. But, since the prior distribution $p(\theta) = e^{-(\theta-\mu)^2/(2\sigma^2)}/(\sqrt{2\pi}\sigma)$ decays exponentially in $|\theta - \mu|$, such θ have little contribution to the computation of the posterior mean and variance of θ. So Eqs. (62) and (63) give highly accurate estimates of the posterior mean and variance of θ, and their errors have negligible impact on the performance of the whole algorithm.

V. SIMULATION RESULTS

In this section, we present the simulation results of Bayesian inference with engineered likelihood functions for amplitude estimation. These results demonstrate the advantages of engineered likelihood functions over unengineered ones, as well as the impacts of circuit depth and fidelity on their performance.

A. Experimental details

In our experiments, we assume that the ansatz circuit A and its inverse $A^\dagger$ contribute most significantly to the duration of the circuit (in Fig. 3 or 21) for generating the likelihood function. So, when the number of circuit layers is L, the time cost of an inference round is roughly $2L + 1$, where time is in units of A's duration. Moreover, we assume that there is no state preparation and measurement error, i.e., $\bar{p} = 1$, in the experiments.

Suppose that we aim to estimate the expectation value $\Pi = \cos(\theta) = \langle A|P|A\rangle$. Let $\hat{\mu}_t$ be the estimator of Π at time t. Note that $\hat{\mu}_t$ is a random variable, since it depends on the history of random measurement outcomes up to time t. We measure the performance of a scheme by the root-mean-squared error (RMSE) of $\hat{\mu}_t$, which is given by

$$\mathrm{RMSE}_t := \sqrt{\mathrm{MSE}_t} = \sqrt{\mathbb{E}[(\hat{\mu}_t - \Pi)^2]}. \tag{64}$$

We investigate how fast RMSE_t decreases as t grows for various schemes, including the ancilla-based Chebyshev likelihood function (AB CLF), ancilla-based engineered likelihood function (AB ELF), ancilla-free Chebyshev likelihood function (AF CLF), and ancilla-free engineered likelihood function (AF ELF).

In general, the distribution of $\hat{\mu}_t$ is difficult to characterize, and there is no analytical formula for RMSE_t. To estimate this quantity, we simulate the inference process M times, and collect M samples $\hat{\mu}_t^{(1)}, \hat{\mu}_t^{(2)}, \ldots, \hat{\mu}_t^{(M)}$ of $\hat{\mu}_t$, where $\hat{\mu}_t^{(i)}$ is the estimate of Π at time t in the ith run for $i = 1, 2, \ldots, M$. Then we use the quantity

$$\overline{\mathrm{RMSE}}_t := \sqrt{\frac{1}{M} \sum_{i=1}^{M} (\hat{\mu}_t^{(i)} - \Pi)^2} \tag{65}$$

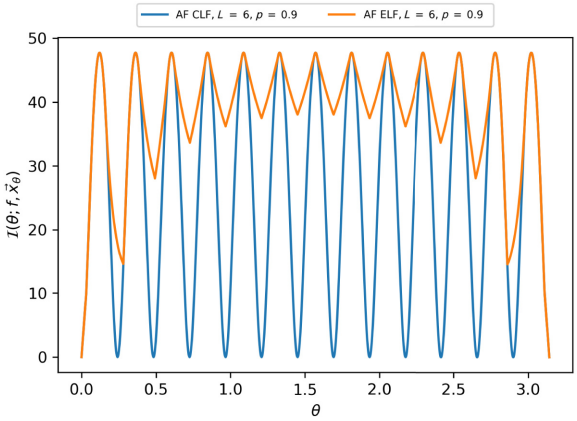


FIG. 5. Comparison of the Fisher information of the ELF and CLF for various $\theta \in (0, \pi)$, when the number of circuit layers is $L = 6$, the fidelity of each layer is $p = 0.9$, and there is no state preparation and measurement error (i.e., $\bar{p} = 1$). For the ELF, $\vec{x}_\theta$ is a global maximum point of $\mathcal{I}(\theta; f, \vec{x})$ for given θ and $f = \bar{p}p^L = 0.531\,441$. For the CLF, $\vec{x}_\theta = (\pi/2, \pi/2, \ldots, \pi/2)$ is fixed. One can see that the Fisher information of the ELF is larger than that of the CLF for the majority of $\theta \in (0, \pi)$. Furthermore, the Fisher information of the CLF changes dramatically for different θ [in fact, it is exactly 0 when $\theta = j\pi/(2L+1)$ for $j = 0, 1, \ldots, 2L+1$], whereas the Fisher information of the ELF is less sensitive to the value of θ.

to approximate the true RMSE_t. In our experiments, we set $M = 300$ and find that this leads to a low enough empirical variance in the estimate to yield meaningful results.

In each experiment, we set the true value of Π and choose a prior distribution of Π, and run the algorithm

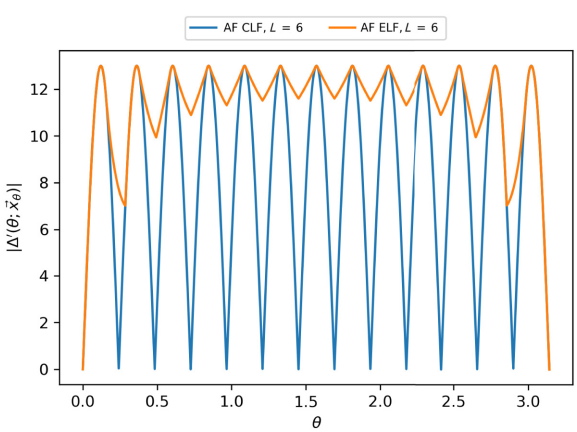


FIG. 6. Comparison of the values of $|\Delta'(\theta; \vec{x}_\theta)|$ for the slope-based ELF and CLF for various $\theta \in (0, \pi)$, when the number of circuit layers is $L = 6$. For the ELF, $\vec{x}_\theta$ is a global maximum point of $|\Delta'(\theta; \vec{x})|$ for given θ. For the CLF, $\vec{x}_\theta = (\pi/2, \pi/2, \ldots, \pi/2)$ is fixed. This figure implies that the slope-based ELF is steeper than the CLF for the majority of $\theta \in (0, \pi)$. Furthermore, the slope of the CLF changes dramatically for different θ [in fact, it is exactly 0 when $\theta = j\pi/(2L+1)$ for $j = 0, 1, \ldots, 2L+1$], whereas the slope of the ELF is less sensitive to the value of θ.

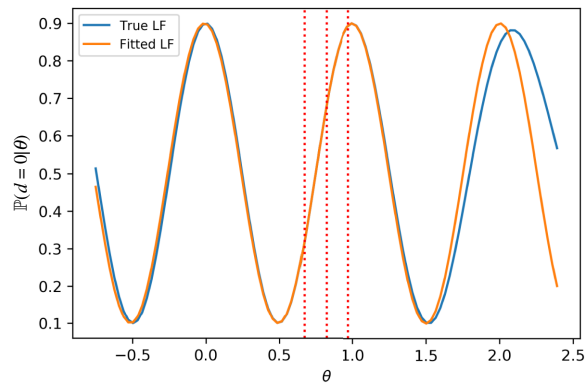


FIG. 7. The true and fitted likelihood functions when $L = 3$, $f = 0.8$, and θ has prior distribution $\mathcal{N}(0.82, 0.0009)$. The true likelihood function is generated by Algorithm 2. During the sinusoidal fitting of this function, we set $\Theta = \{\mu - \sigma, \mu - 0.8\sigma, \ldots, \mu + 0.8\sigma, \mu + \sigma\}$ (i.e., Θ contains 11 uniformly distributed points in $[\mu - \sigma, \mu + \sigma]$) in Eq. (51). The fitted likelihood function is $\mathbb{P}(d|\theta) = [1 + (-1)^d f \sin(r\theta + b)]/2$, where $r = 6.24$ and $b = -4.65$. Note that the true and fitted likelihood functions are close for $\theta \in [0.67, 0.97]$.

in Fig. 4 to test its performance for estimating this quantity. The only quantum component of this algorithm, i.e., executing the circuit in Fig. 3 and measuring the outcome $d \in \{0, 1\}$, is simulated by an efficient classical procedure, since the probability distribution of the outcome d is characterized by Eq. (23) or (D3), depending on whether the scheme is ancilla-free or ancilla-based. Namely, we synthesize the outcome d by a classical pseudorandom number generator [based on Eq. (23) or (D3)] instead of simulating the noisy quantum circuit. This not only greatly accelerates our simulation, but also makes our results applicable to a wide range of scenarios (i.e., they are not specific to a certain ansatz circuit A or observable P).

For engineering the likelihood function, we use Algorithm 2 or 6 to optimize the circuit parameters $\vec{x}$, depending on whether the scheme is ancilla-free or ancilla based. We find that Algorithms 1 and 2 generate the same likelihood function (as they both find the optimal parameters within a reasonable number of trials), and the same holds for Algorithms 5 and 6. So the simulation results in Sec. V will not change if we have used Algorithms 1 and 5 instead, as the AF ELF and AB ELF will remain intact. Meanwhile, we do not use Algorithm 3, 4, 7, or 8 in our experiments, because these algorithms are used to maximize the slope of the AF or AB likelihood function, which is a good proxy of the variance reduction factor $\mathcal{V}$ only when the fidelity f is close to zero [see Eqs. (37) and (D5)]. In our experiments, f is mostly between 0.5 and 0.9, and in such a case, the slope is not a good proxy of the variance reduction factor and we need to maximize the Fisher information by Algorithm 1, 2, 5, or 6 instead.

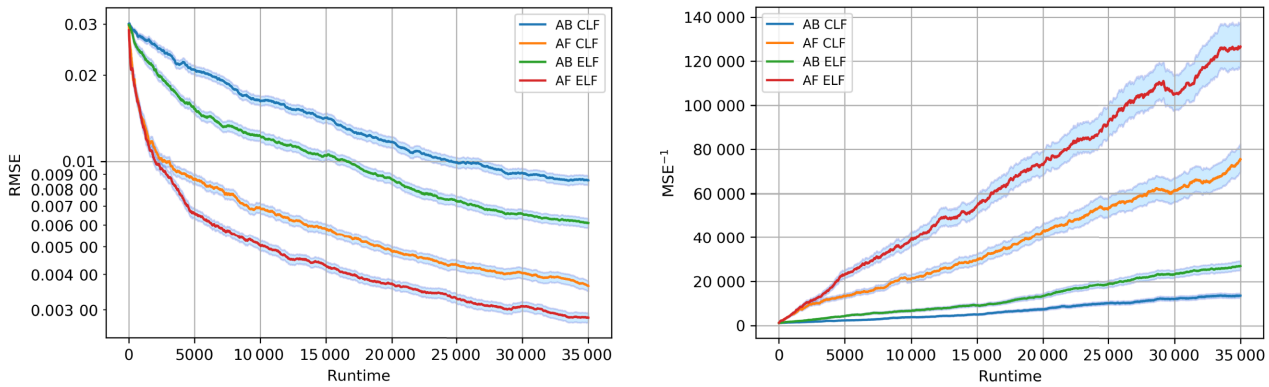


FIG. 8. Comparison of the performances of the AB CLF, AB ELF, AF CLF, and AF ELF when the expectation value Π has true value -0.4 and prior distribution $\mathcal{N}(-0.43, 0.0009)$, the number of circuit layers is $L = 6$, and the layer fidelity is $p = 0.9$. Note that the ELF outperforms the CLF in both the ancilla-free and ancilla-based cases, and the ancilla-free schemes outperform the ancilla-based ones.

Our estimation algorithm needs to tune the circuit parameters at each round, and since it can have tens of thousands of rounds, this component can be quite time consuming. We use the following method to mitigate this issue. Given the number L of circuit layers and the fidelity f of the process for generating the ELF, we first construct a "lookup table" that consists of the optimal parameters $\vec{x}_\Pi$ for all the Π in a discrete set $\mathcal{S} = \{\Pi_1, \Pi_2, \ldots, \Pi_N\} \subset [-1, 1]$. Then at any stage of the estimation algorithm, we find the $\Pi_j \in \mathcal{S}$ that is closest to the current estimate of Π, and use the optimal parameters for Π_j to approximate the optimal parameters for Π. In our experiments, we set $\mathcal{S} = \{-1.0, -0.9995, -0.999, \ldots, 0.999, 0.9995, 1.0\}$ (i.e., $\mathcal{S}$ contains 4001 uniformly distributed points in $[-1, 1]$) and find that this greatly accelerates the estimation algorithm without deteriorating its performance much.

For Bayesian update with an ELF, we use the method in Sec. IV B or Appendix D 2 to compute the posterior mean and variance of θ, depending on whether the scheme is ancilla-free or ancilla-based. In particular, during the sinusoidal fitting of the ELF, we set $\Theta = \{\mu - \sigma, \mu - 0.8\sigma, \ldots, \mu + 0.8\sigma, \mu + \sigma\}$ (i.e., Θ contains 11 uniformly distributed points in $[\mu - \sigma, \mu + \sigma]$) in Eq. (58) or (D41). We find that this is sufficient for obtaining a high-quality sinusoidal fit of the true likelihood function.

B. Comparing the performance of various schemes

To compare the performance of various quantum-generated likelihood functions, including the AB CLF, AB ELF, AF CLF, and AF ELF, we run Bayesian inference with each of them, fixing the number of circuit layers $L = 6$ and layer fidelity $p = 0.9$ (note that this layer fidelity corresponds to a 12-qubit experiment with a two-qubit gate depth of 12 and a two-qubit gate fidelity of 99.92%, which is almost within reach for today's quantum devices). In Figs. 8–12 we illustrate the performances of different schemes with respect to various true values of Π. These results suggest the following.

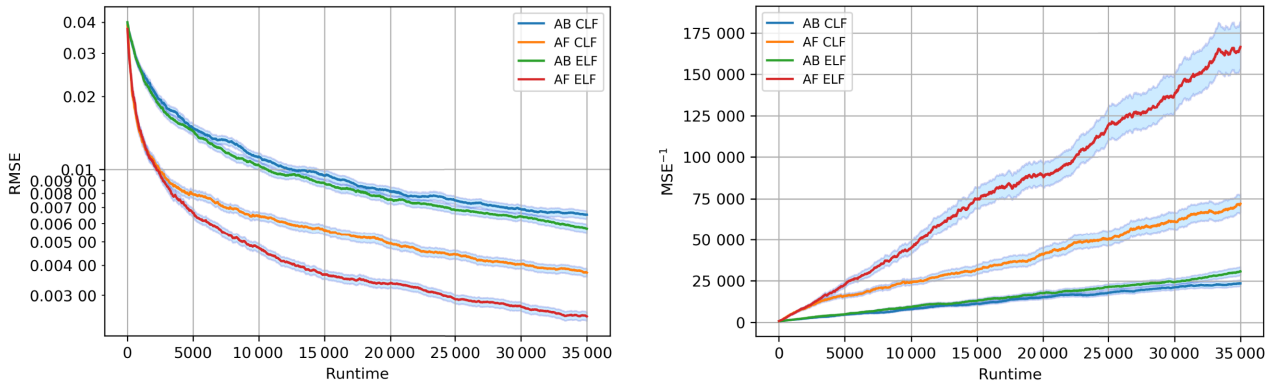


FIG. 9. Comparison of the performances of the AB CLF, AB ELF, AF CLF, and AF ELF when the expectation value Π has true value 0.6 and prior distribution $\mathcal{N}(0.64, 0.0009)$, the number of circuit layers is $L = 6$, and the layer fidelity is $p = 0.9$. Note that the AB ELF slightly outperforms the AB CLF, while the AF ELF outperforms the AF CLF to a larger extent.

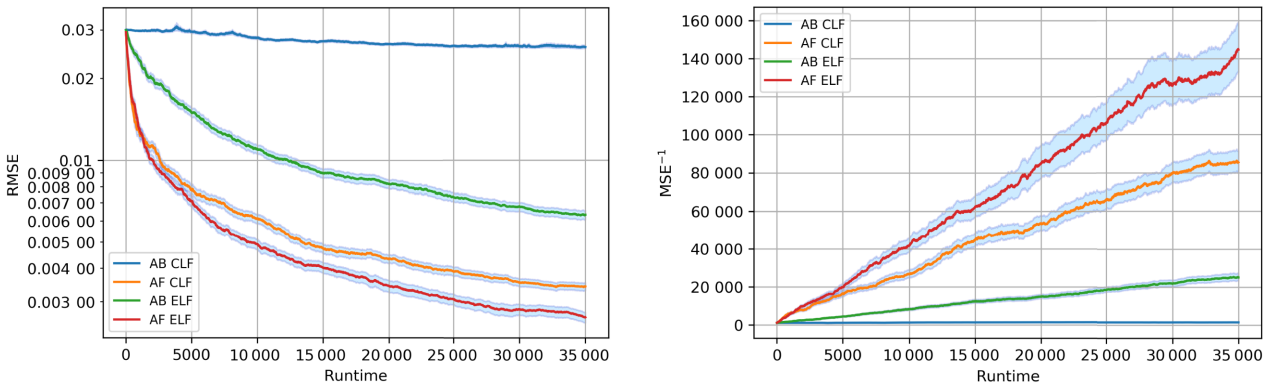


FIG. 10. Comparison of the performances of the AB CLF, AB ELF, AF CLF, and AF ELF when the expectation value Π has true value 0.52 and prior distribution $\mathcal{N}(0.49, 0.0009)$, the number of circuit layers is $L = 6$, and the layer fidelity is $p = 0.9$. Note that the RMSE of the estimator of Π converges to 0 for all schemes except the AB CLF. The AF ELF achieves the best performance.

(a) In both the ancilla-based and ancilla-free cases, the ELF performs better than (or as well as) the CLF. This means that by tuning the generalized reflection angles, we do enhance the rate of information gain and make the estimation of Π more efficient.
(b) The AF ELF always performs better than the AB ELF, whereas the AF CLF may perform better or worse than the AB CLF, depending on the true value of Π, but on average, the AF CLF outperforms the AB CLF. So overall the ancilla-free schemes are superior to the ancilla-based ones.
(c) While $\mathrm{RMSE}_t \to 0$ as $t \to \infty$ for the AB ELF and AF ELF, the same is not always true for the AB CLF and AF CLF. In fact, the performances of the AB CLF and AF CLF depend heavily on the true value of Π, while the performances of the AB ELF and AF ELF are not much affected by this value.

We may compare the above results with those of Fig. 13 in which we illustrate the $\hat{R}_0$ factors [as defined in Eq. (41)] of the AB CLF, AB ELF, AF CLF, and AF ELF in the same setting. One can observe the following from this figure.

(a) The $\hat{R}_0$ factor of the AB ELF is equal to or larger than that of the AB CLF, and the same is true for the AF ELF versus the AB CLF. This explains why the ELF outperforms the CLF in both the ancilla-based and ancilla-free cases.
(b) The $\hat{R}_0$ factor of the AF ELF is larger than that of the AB ELF. Meanwhile, the $\hat{R}_0$ factor of the AF CLF can be larger or smaller than that of the AB CLF, depending on the value of Π, but on average the AF CLF has larger $\hat{R}_0$ factor than the AB CLF. This explains the superiority of the ancilla-free schemes over the ancilla-based ones.
(c) The $\hat{R}_0$ factors of the AB ELF and AF ELF are bounded away from 0 for all $\Pi \in [-1, 1]$ [60]. This explains why their performances are stable regardless of the true value of Π. On the other hand, the $\hat{R}_0$ factors of the AB CLF and AF CLF change dramatically for different Π. In fact, the $\hat{R}_0$ factor

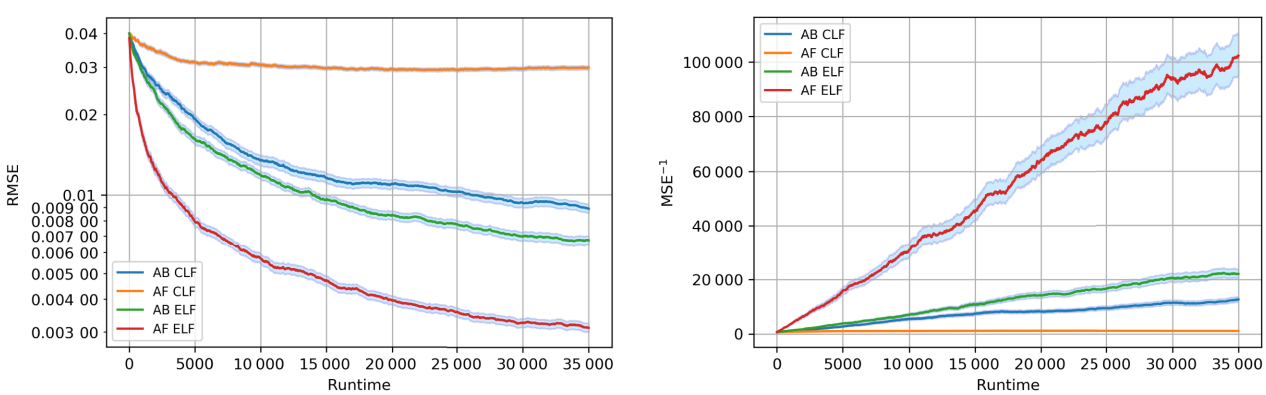


FIG. 11. Comparison of the performances of the AB CLF, AB ELF, AF CLF, and AF ELF when the expectation value Π has true value -0.1 and prior distribution $\mathcal{N}(-0.14, 0.0009)$, the number of circuit layers is $L = 6$, and the layer fidelity is $p = 0.9$. Note that the RMSE of the estimator of Π converges to 0 for all schemes except the AF CLF. The AF ELF achieves the best performance.

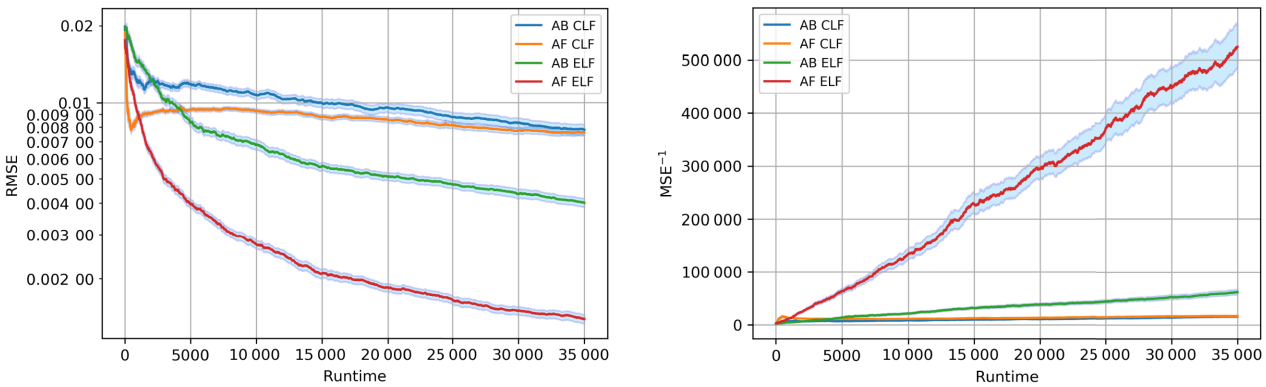


FIG. 12. Comparison of the performances of the AB CLF, AB ELF, AF CLF, and AF ELF when the expectation value Π has true value 0.9 and prior distribution $\mathcal{N}(0.92, 0.0009)$, the number of circuit layers is $L = 6$, and the layer fidelity is $p = 0.9$. Note that the RMSE of the estimator of Π fails to converge to 0 for the AF CLF and decreases slowly for the AB CLF. Both the AB CLF and AF CLF are outperformed by the AB ELF and AF ELF, with the AF ELF achieving the best performance.

of the AB CLF is 0 when $\Pi = \cos(j\pi/L)$ for $j = 0, 1, \ldots, L$, and the $\hat{R}_0$ factor of the AF CLF is 0 when $\Pi = \cos[j\pi/(2L+1)]$ for $j = 0, 1, \ldots, 2L+1$. This means that if the true value of Π is close to one of these "dead spots" then its estimator will struggle to improve and hence the performance of the AB or AF CLF will suffer (see Figs. 10–12 for examples.). The AB ELF and AF ELF, on the other hand, do not have this weakness.

C. Understanding the performance of Bayesian inference with ELFs

Having established the improved performance of ELFs over CLFs, we now analyze the performance of Bayesian inference with ELFs in more detail. Note that Figs. 8–12 suggest that the inverse MSEs of both AB-ELF-based and AF-ELF-based estimators of Π grow linearly in time when the circuit depth is fixed. By fitting a linear model to the data, we obtain the empirical growth rates of these quantities, which are shown in Tables II and III, respectively. We also compare these rates with the $\hat{R}_0$ factors of the AB ELF and AF ELF in the same setting. It turns out that the $\hat{R}_0$ factor is a rough estimate of the true growth rate of the inverse MSE of an ELF-based estimator of Π, but it can be unreliable sometimes. We leave it as an open question to give a more precise characterization of the decay of the RMSEs of ELF-based estimators of Π during the inference process.

1. Analyzing the impact of layer fidelity on the performance of estimation

To investigate the influence of layer fidelity on the performance of estimation, we run Bayesian inference with an AB or AF ELF for fixed circuit depth but vary layer fidelity. Specifically, we set the number L of circuit layers to be 6, and vary the layer fidelity as $p = 0.75, 0.8, 0.85, 0.9, 0.95$. In Figs. 14 and 15 we illustrate the simulation results in the ancilla-based and ancilla-free cases, respectively. As expected, higher layer fidelity leads to better performance of the algorithm. Namely, the less noisy the circuit, the faster the RMSE of the estimator of Π decreases. This is consistent with the fact that the $\hat{R}_0$ factors of the AB ELF and AF ELF are monotonically increasing functions of $f = p^L$, as demonstrated in Fig. 16.

TABLE II. The predicted and empirical growth rates of the inverse MSEs of AB-ELF-based estimators of Π in the five experiments in Sec. V B. In all of these experiments, the number of circuit layers is $L = 6$ and the layer fidelity is $p = 0.9$.

True value of Π	Prior distribution of Π	Predicated growth rate of MSE_t^{-1}	Empirical growth rate of MSE_t^{-1}
−0.4	$\mathcal{N}(-0.43, 0.0009)$	0.71	0.75
0.6	$\mathcal{N}(0.64, 0.0009)$	0.91	0.86
0.52	$\mathcal{N}(0.49, 0.0009)$	0.70	0.75
−0.1	$\mathcal{N}(-0.14, 0.0009)$	0.60	0.68
0.9	$\mathcal{N}(0.92, 0.0009)$	2.03	1.84

TABLE III. The predicted and empirical growth rates of the inverse MSEs of AF-ELF-based estimators of Π in the five experiments in Sec. V B. In all of these experiments, the number of circuit layers is $L = 6$ and the layer fidelity is $p = 0.9$.

True value of Π	Prior distribution of Π	Predicated growth rate of MSE_t^{-1}	Empirical growth rate of MSE_t^{-1}
−0.4	$\mathcal{N}(-0.43, 0.0009)$	3.77	3.70
0.6	$\mathcal{N}(0.64, 0.0009)$	4.73	4.68
0.52	$\mathcal{N}(0.49, 0.0009)$	4.36	4.19
−0.1	$\mathcal{N}(-0.14, 0.0009)$	3.07	3.08
0.9	$\mathcal{N}(0.92, 0.0009)$	13.22	14.74

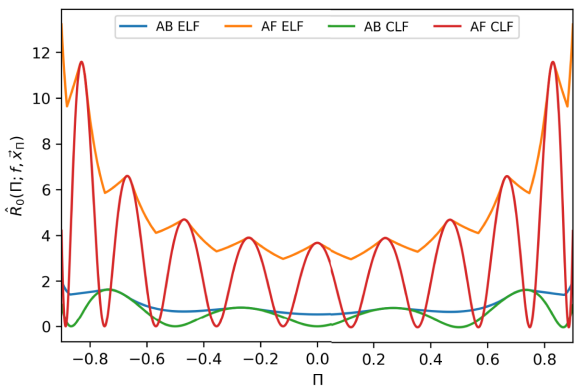


FIG. 13. Comparison of the $\hat{R}_0$ factors of the AB CLF, AB ELF, AF CLF, and AF ELF for $\Pi \in [-0.9, 0.9]$ when the number of circuit layers is $L = 6$ and the layer fidelity is $p = 0.9$. Here $\vec{x}_\Pi \in \mathbb{R}^{2L}$ is a global maximum point of $\hat{R}_0(\Pi; f, \vec{x})$ for given Π and $f = p^L = 0.531\,441$. All four curves are plotted on a fine grid with interval length 0.0005 (i.e., there are 3601 equally spaced grid points in $[-0.9, 0.9]$). Note that the $\hat{R}_0$ factors of the AB CLF and AF CLF change dramatically for different Π. In fact, they can be close to 0 for certain Π. By contrast, the $\hat{R}_0$ factors of the AB ELF and AF ELF are bounded away from 0 for all Π.

2. Analyzing the impact of circuit depth on the performance of estimation

To investigate the influence of circuit depth on the performance of estimation, we run Bayesian inference with an AB or AF ELF for fixed layer fidelity but vary circuit depth. Specifically, we set the layer fidelity p to be 0.9, and vary the number L of circuit layers from 1 to 5. In Figs. 17 and 18 we illustrate the simulation results in the ancilla-based and ancilla-free cases, respectively. These results indicate that a larger L (i.e., deeper circuit) does not necessarily lead to better performance. The optimal choice of L is indeed a subtle issue. This can be intuitively understood as follows. As L increases, the likelihood function becomes steeper [61] and hence gains more statistical power, if the circuit for generating it is noiseless. But, on the other hand, the true fidelity of the circuit decreases exponentially in L and the implementation cost of this circuit grows linearly in L. So one must find a perfect balance among these factors in order to maximize the performance of the estimation algorithm.

The above results are consistent with those of Fig. 19 in which we illustrate the $\hat{R}_0$ factors of the AB ELF and AF ELF in the same setting. Note that a larger L does not necessarily lead to a larger $\hat{R}_0$ factor of the AB or AF ELF. One can evaluate this factor for different L and choose the one that maximizes this factor. This often enables us to find a satisfactory (albeit not necessarily optimal) L. It remains an open question to devise an efficient strategy for determining the L that optimizes the performance of estimation given the layer fidelity p and a prior distribution of Π.

VI. A MODEL FOR NOISY ALGORITHM PERFORMANCE

Our aim is to build a model for the runtime needed to achieve a target mean squared error in the estimate of Π as it is scaled to larger systems and run on devices with better gate fidelities. This model will be built on two main assumptions. The first is that the growth rate of the inverse mean squared error is well described by half the inverse variance rate expression [cf. Eq. (34)]. The variance contribution to the MSE is the variance in the estimator, which is not necessarily the same as the posterior variance. In Appendix E we show that the variance closely tracks the variance of the estimator (cf. Fig. 26). We use the posterior variance σ_{i+1}^2 in place of the estimator variance. The half is due to the conservative estimate that the variance and squared bias contribute equally to the mean squared error. In Appendix E we show evidence supporting this assumption of a small bias (cf. Fig. 25).

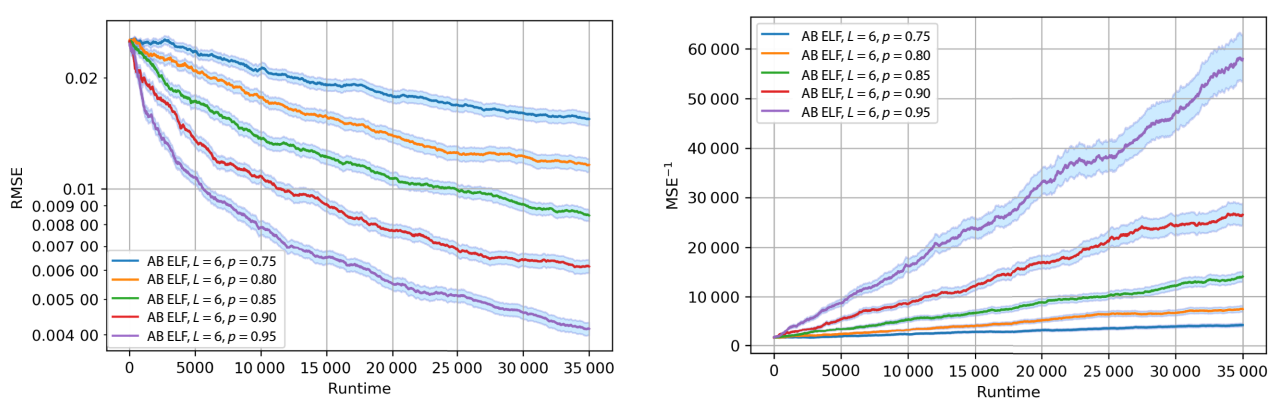


FIG. 14. The impact of layer fidelity on the performance of the AB ELF. Here Π has true value 0.18 and prior distribution $\mathcal{N}(0.205, 0.0009)$, the number L of circuit layers is 6, and the layer fidelity p is varied as $p = 0.75, 0.8, 0.85, 0.9, 0.95$. Note that higher layer fidelity leads to better performance of estimation.

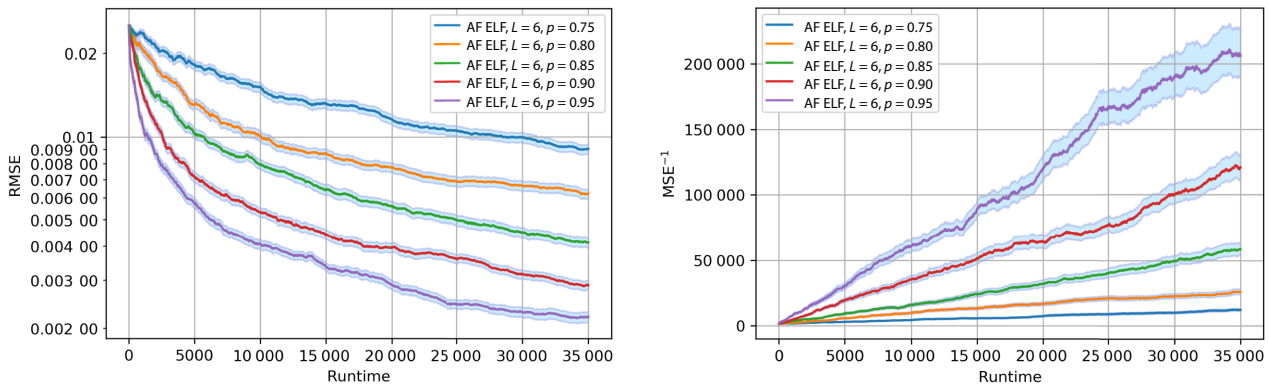


FIG. 15. The impact of layer fidelity on the performance of the AF ELF. Here Π has true value 0.18 and prior distribution $\mathcal{N}(0.205, 0.0009)$, the number L of circuit layers is 6, and the layer fidelity p is varied as $p = 0.75, 0.8, 0.85, 0.9, 0.95$. Note that higher layer fidelity leads to better performance of estimation.

The second assumption is an empirical lower bound on the variance reduction factor, which is motivated by numerical investigations of the Chebyshev likelihood function.

We carry out analysis for the MSE with respect to the estimate of θ. We then convert the MSE of this estimate to an estimate of MSE with respect to Π. Our strategy will be to integrate upper and lower bounds for the rate expression $R(\mu, \sigma; f, m)$ in Eq. (34) to arrive at bounds for the inverse MSE as a function of time. To help our analysis, we make the substitution $m = T(L) = 2L + 1$ and reparameterize the way noise is incorporated by introducing λ and α such that $f^2 = \bar{p}^2 p^{2L} = e^{-\lambda(2L+1)-\alpha} = e^{-\lambda m - \alpha}$.

The upper and lower bounds on this rate expression are based on findings for the Chebyshev likelihood functions, where $\vec{x} = (\pi/2)^{\oplus 2L} := (\pi/2, \pi/2, \ldots, \pi/2) \in \mathbb{R}^{2L}$. Since the Chebyshev likelihood functions are a subset of the engineered likelihood functions, a lower bound on the Chebyshev performance gives a lower bound on the ELF performance. We leave as a conjecture that the upper bound for this rate in the case of the ELF is a small multiple (e.g., 1.5) of the upper bound we have established for the Chebyshev rate.

The Chebyshev upper bound is established as follows. For fixed σ, λ, and m, one can show [62] that the variance reduction factor achieves a maximum value of $\mathcal{V} = m^2 \exp(-m^2\sigma^2 - \lambda m - \alpha)$, occurring at $\mu = \pi/2$. This expression is less than $m^2 e^{-m^2\sigma^2}$, which achieves a maximum of $(e\sigma^2)^{-1}$ at $m = 1/\sigma$. Thus, the factor $1/(1 - \sigma^2\mathcal{V})$ cannot exceed $1/(1 - e^{-1}) \approx 1.582$. Putting this all together, for fixed σ, λ, and m, the maximum rate is upper bounded as $R(\mu, \sigma; \lambda, \alpha, m) \leq em \exp(-m^2\sigma^2 - \lambda m - \alpha)/(e - 1)$. This follows from the fact that R is monotonic in $\mathcal{V}$ and that $\mathcal{V}$ is maximized at $\mu = \pi/2$. In practice, we aim to choose a value of L that maximizes the inverse variance rate. The rate achieved by discrete L cannot exceed the value we obtain when optimizing the above upper bound over continuous values of m. This optimal value is realized for $1/m = \frac{1}{2}(\sqrt{\lambda^2 + 8\sigma^2} + \lambda)$. We

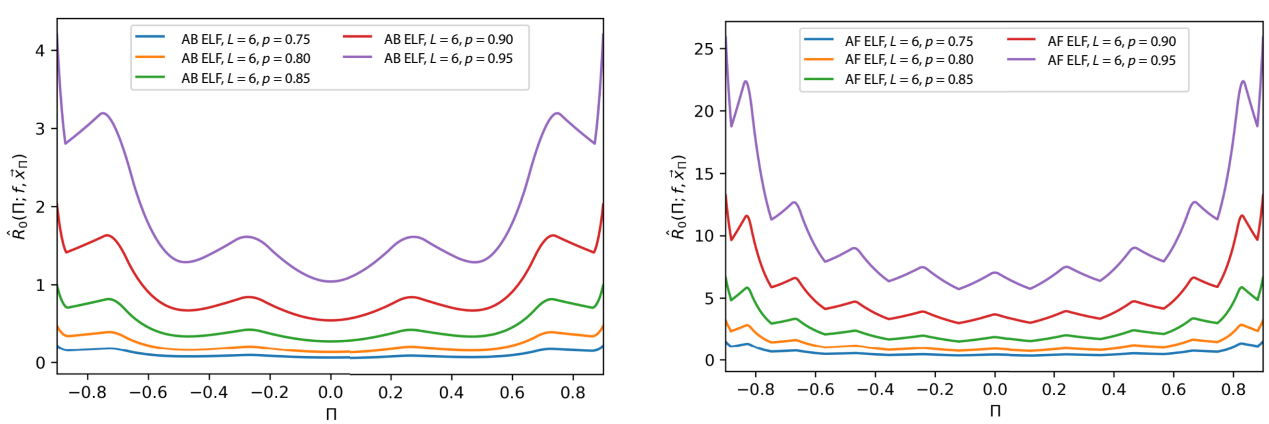


FIG. 16. The $\hat{R}_0$ factors of the AB ELF and AF ELF for $\Pi \in [-0.9, 0.9]$ when the number L of circuit layers is fixed to 6 and the layer fidelity p is varied as $p = 0.75, 0.8, 0.85, 0.9, 0.95$. Here $\vec{x}_\Pi \in \mathbb{R}^{2L}$ is a global maximum point of $\hat{R}_0(\Pi; f, \vec{x})$ for given Π and $f = p^L$. Note that higher layer fidelity leads to larger $\hat{R}_0$ factors of the AB ELF and AF ELF.

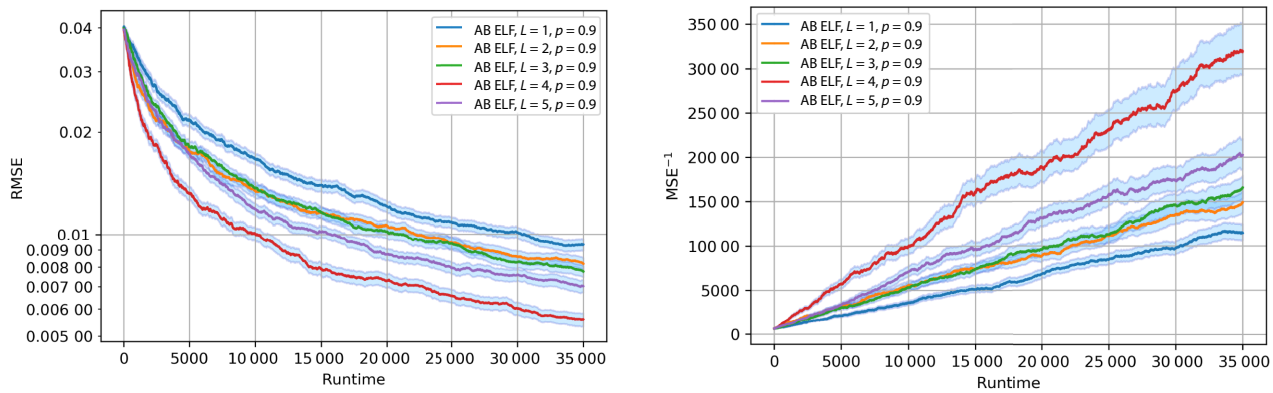


FIG. 17. The impact of circuit depth on the performance of the AB ELF. Here Π has true value 0.35 and prior distribution $\mathcal{N}(0.39, 0.0016)$, the layer fidelity p is 0.9, and the number L of layers is varied from 1 to 5. Note that the best performance is achieved by $L = 4$ instead of $L = 5$.

define $\bar{R}(\sigma;\lambda,\alpha)$ by evaluating $R(\pi/2,\sigma;\lambda,\alpha,m)$ at this optimum value,

$$\bar{R}(\sigma;\lambda,\alpha) = \frac{2e^{-\alpha-1}}{\sqrt{\lambda^2+8\sigma^2}+\lambda} \times \exp\Big(\frac{2\sigma^2}{4\sigma^2+\lambda^2+\lambda^2\sqrt{8\sigma^2/\lambda^2+1}}\Big), \tag{66}$$

which gives the upper bound on the Chebyshev rate

$$R_C^*(\mu,\sigma;\lambda,\alpha) = \max_L R(\mu,\sigma;\lambda,\alpha,m) \le \frac{e}{e-1}\bar{R}(\sigma;\lambda,\alpha). \tag{67}$$

We do not have an analytic lower bound on the Chebyshev likelihood performance. We can establish an empirical lower bound based on numerical checks. For any fixed L, the inverse variance rate is zero at the $2L+2$ points $\mu \in \{0, \pi/(2L+1), 2\pi/(2L+1), \ldots, 2L\pi/(2L+1), \pi\}$. Since the rate is zero at these end points for all L, the global lower bound on R_C^* is zero. However, we are not concerned with the poor performance of the inverse variance rate near these end points. When we convert the estimator from $\hat{\theta}$ to $\hat{\Pi} = \cos\hat{\theta}$, the information gain near these end points actually tends to a large value. For the purpose of establishing useful bounds, we restrict μ to be in the range $[0.1\pi, 0.9\pi]$. In the numerical tests [63] we find that, for all $\mu \in [0.1\pi, 0.9\pi]$, there is always a choice of L for which the inverse variance rate is above $(e-1)^2/e^2 \approx 0.40$ times the upper bound. Putting these together, we have

$$\frac{e-1}{e}\bar{R}(\sigma;\lambda,\alpha) \le R_C^*(\mu,\sigma;\lambda,\alpha) \le \frac{e}{e-1}\bar{R}(\sigma;\lambda,\alpha). \tag{68}$$

It is important to note that, by letting m be continuous, certain values of σ and λ can lead to an optimal m for which $L = (m-1)/2$ is negative. Therefore, these results apply only in the case that $\lambda \le 1$, which ensures that $m \ge 1$. We

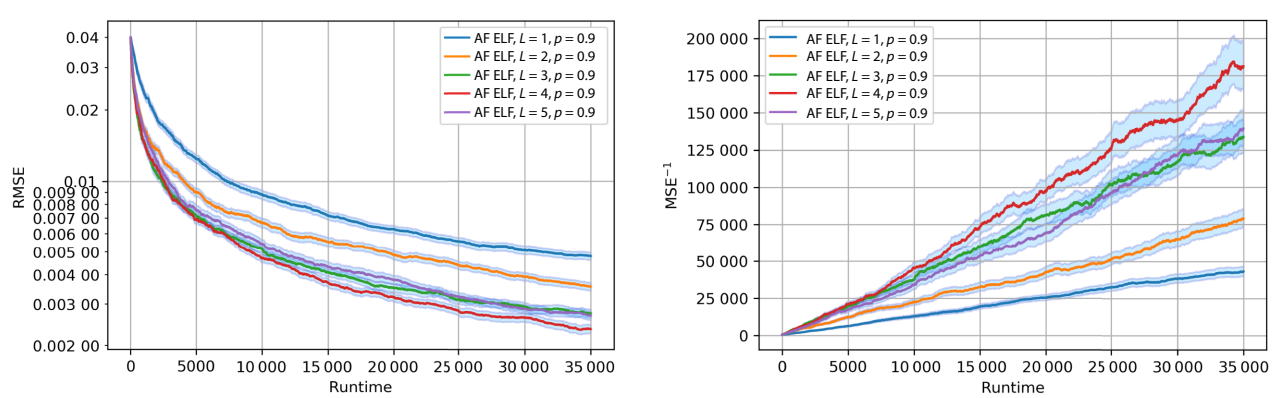


FIG. 18. The impact of circuit depth on the performance of the AF ELF. Here Π has true value 0.35 and prior distribution $\mathcal{N}(0.39, 0.0016)$, the layer fidelity p is 0.9, and the number L of layers is varied from 1 to 5. Note that the best performance is achieved by $L = 4$ instead of $L = 5$.

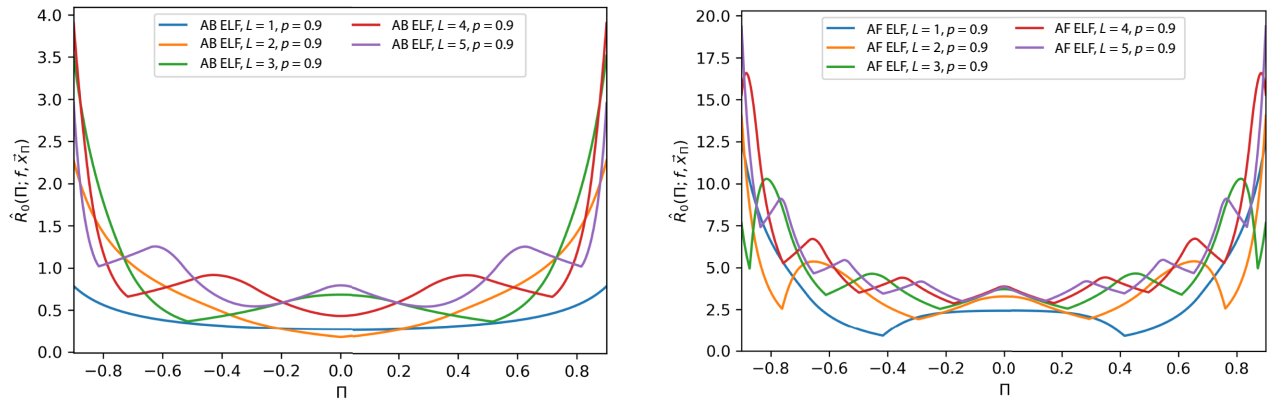


FIG. 19. The $\hat{R}_0$ factors of the AB ELF and AF ELF for $\Pi \in [-0.9, 0.9]$ when the number L of circuit layers is varied from 1 to 5 and the layer fidelity p is fixed to 0.9. Here $\vec{x}_\Pi \in \mathbb{R}^{2L}$ is a global maximum point of $\hat{R}_0(\Pi; f, \vec{x})$ for given Π and $f = p^L$. Note that the L that maximizes the $\hat{R}_0$ factor depends heavily on the value of Π in both the ancilla-based and ancilla-free cases.

expect this model to break down in the large-noise regime (i.e., $\lambda \geq 1$).

For now, we assume that the rate tracks the geometric mean of these two bounds, i.e., $R^*_C(\sigma, \lambda, \mu) = \bar{R}(\sigma, \lambda)$, keeping in mind that the upper and lower bounds are small constant factors off of this. We assume that the inverse variance grows continuously in time at a rate given by the difference quotient expression captured by the inverse-variance rate, $R^* = (d/dt)(1/\sigma^2)$. Letting $F = 1/\sigma^2$ denote this inverse variance, the rate equation above can be recast as a differential equation for F,

$$\frac{dF}{dt} = \frac{2e^{-\alpha-1}}{\lambda\sqrt{1 + 8/(F\lambda^2)} + \lambda} \times \exp\left(\frac{2}{4 + \lambda^2 F + \lambda^2 F\sqrt{1 + 8/(F\lambda^2)}}\right). \quad (69)$$

Through this expression, we can identify both the Heisenberg limit behavior and shot-noise limit behavior. For $F \ll 1/\lambda^2$, the differential equation becomes

$$\frac{dF}{dt} = \frac{e^{-\alpha-1/2}}{\sqrt{2}}\sqrt{F}, \quad (70)$$

which integrates to a quadratic growth of the inverse squared error $F(t) \sim t^2$. This is the signature of the Heisenberg limit regime. For $F \gg 1/\lambda^2$, the rate approaches a constant,

$$\frac{dF}{dt} = \frac{e^{-\alpha-1}}{\lambda}. \quad (71)$$

This regime yields a linear growth in the inverse squared error $F(t) \sim t$, indicative of the shot-noise limit regime.

In order to make the integral tractable, we can replace the rate expression with integrable upper and lower bound expressions (to be used in tandem with our previous bounds). Letting $x = \lambda^2 F$, these bounds are reexpressed as

$$\frac{2e^{-\alpha-1}\lambda}{1 + 1/\sqrt{12x} + (x+4)/(\sqrt{x^2+8x})} \geq \frac{dx}{dt} \geq \frac{2e^{-\alpha-1}\lambda}{1 + 1/\sqrt{4x} + (x+4)/(\sqrt{x^2+8x})}. \quad (72)$$

From the upper bound we can establish a lower bound on the runtime, by treating time as a function of x and integrating,

$$\int_0^t dt \geq \int_{x_0}^{x_f} dx \frac{e^{\alpha+1}}{2\lambda}\left(1 + \frac{1}{\sqrt{12x}} + \frac{x+4}{\sqrt{x^2+8x}}\right) = \frac{e^{\alpha+1}}{2\lambda}\left(x_f + \sqrt{\frac{x_f}{3}} + \frac{1}{2}\sqrt{x_f^2 + 8x_f} - x_0 - \sqrt{\frac{x_0}{3}} - \frac{1}{2}\sqrt{x_0^2 + 8x_0}\right). \quad (73)$$

Similarly, we can use the lower bound to establish an upper bound on the runtime. Here we introduce our assumption that, in the worst case, the MSE of the phase estimate ε_θ^2 is twice the variance (i.e., the variance equals the squared bias), so the variance must reach half the MSE: $\sigma^2 = \varepsilon_\theta^2/2 = \lambda^2/x$. In the best case, we assume that the bias in the estimate is zero and set $\varepsilon_\theta^2 = \lambda^2/x$. We combine these bounds with the upper and lower bounds of Eq. (68) to arrive at the bounds on the estimation runtime as a function

of target MSE,

$$(e-1)\frac{e^{-\lambda}}{2\bar{p}^2}\left[\frac{\lambda}{\varepsilon_\theta^2}+\frac{1}{\sqrt{3}\varepsilon_\theta}+\sqrt{\left(\frac{\lambda}{\varepsilon_\theta^2}\right)^2+\left(\frac{2\sqrt{2}}{\varepsilon_\theta}\right)^2}\right] \le t_{\varepsilon_\theta} \le \frac{e^2}{e-1}\frac{e^{-\lambda}}{\bar{p}^2}\left[\frac{\lambda}{\varepsilon_\theta^2}+\frac{1}{\sqrt{2}\varepsilon_\theta}+\sqrt{\left(\frac{\lambda}{\varepsilon_\theta^2}\right)^2+\left(\frac{2\sqrt{2}}{\varepsilon_\theta}\right)^2}\right], \tag{74}$$

where $\theta \in [0.1\pi, 0.9\pi]$.

At this point, we can convert our phase estimate $\hat{\theta}$ back into the amplitude estimate $\hat{\Pi}$. The MSE with respect to the amplitude estimate ε_Π^2 can be approximated in terms of the phase estimate MSE as

$$\begin{aligned}\varepsilon_\Pi^2 &= \mathbb{E}(\hat{\Pi}-\Pi)^2\\ &= \mathbb{E}(\cos\hat{\theta}-\cos\theta)^2\\ &\approx \mathbb{E}\left((\hat{\theta}-\theta)\frac{d\cos\theta}{d\theta}\right)^2\\ &= \varepsilon_\theta^2\sin^2\theta,\end{aligned} \tag{75}$$

where we have assumed that the distribution of the estimator is sufficiently peaked about θ to ignore higher-order terms. This leads to $\varepsilon_\theta^2 = \varepsilon_\Pi^2/(1-\Pi^2)$, which can be substituted into the above expressions for the bounds, which hold for $\Pi \in [\cos 0.9\pi, \cos 0.1\pi] \approx [-0.95, 0.95]$. Dropping the estimator subscripts (as they only contribute constant factors), we can establish the runtime scaling in the low-noise and high-noise limits,

$$t_\varepsilon = \begin{cases} O(e^\alpha/\varepsilon), & \lambda \ll \varepsilon,\\ O(e^\alpha\lambda/\varepsilon^2), & \lambda \gg \varepsilon,\end{cases} \tag{76}$$

observing that the Heisenberg-limit scaling and shot-noise limit scaling are each recovered.

We arrived at these bounds using properties of Chebyshev likelihood functions. As we have shown in the previous section, by engineering likelihood functions, in many cases we can reduce estimation runtimes. Motivated by our numerical findings of the variance reduction factors of engineered likelihood functions (see, e.g., Fig. 13), we conjecture that using engineered likelihood functions increases the worst-case inverse-variance rate in Eq. (68) to $\bar{R}(\sigma;\lambda,\alpha) \le R_C^*(\mu,\sigma;\lambda,\alpha)$.

In order to give more meaning to this model, we refine it to be in terms of the number of qubits n and two-qubit gate fidelities f_{2Q}. We consider the task of estimating the expectation value of a Pauli string P with respect to state $|A\rangle$. Assume that $\Pi = \langle A|P|A\rangle$ is very near zero so that $\varepsilon^2 = \varepsilon_\Pi^2 \approx \varepsilon_\theta^2$. Let the two-qubit gate depth of each of the L layers be D. We model the total layer fidelity as $p = f_{2Q}^{nD/2}$, where we have ignored errors due to single-qubit gates. From this, we have $\lambda = \frac{1}{2}nD\ln(1/f_{2Q})$ and $\alpha = 2\ln(1/\bar{p}) - \frac{1}{2}nD\ln(1/f_{2Q})$. Putting these together and using the lower bound expression in Eq. (74), we arrive at the runtime expression

$$t_\varepsilon = e\frac{f_{2Q}^{nD/2}}{2\bar{p}^2}\left[\frac{nD\ln(1/f_{2Q})}{2\varepsilon^2}+\frac{1}{\sqrt{3}\varepsilon} + \sqrt{\left(\frac{nD\ln(1/f_{2Q})}{2\varepsilon^2}\right)^2+\left(\frac{2\sqrt{2}}{\varepsilon}\right)^2}\right]. \tag{77}$$

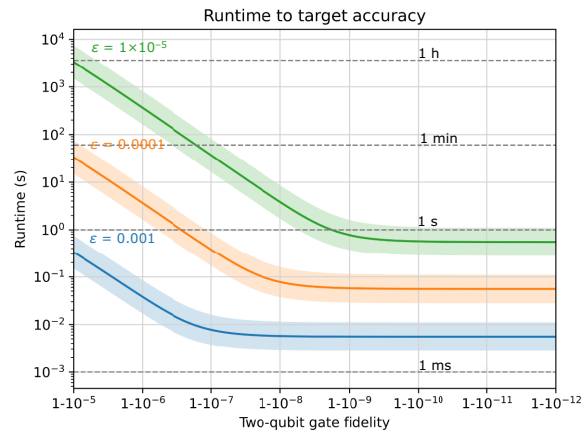


FIG. 20. As two-qubit gate fidelities are improved, deeper enhanced sampling circuits warrant being implemented, yielding shorter estimation runtimes. Here, we consider the case of $n = 100$ qubits, $D = 200$ two-qubit gate depth per layer, and target accuracies of $\varepsilon = 10^{-3}$, $\varepsilon = 10^{-4}$, and $\varepsilon = 10^{-5}$. The bands indicate the upper and lower bounds of Eq. (74).

Finally, we put some meaningful numbers in this expression and estimate the required runtime in seconds as a function of two-qubit gate fidelities. To achieve quantum advantage, we expect that the problem instance will require of the order of $n = 100$ logical qubits and that the two-qubit gate depth is of the order of the number of qubits, $D = 200$. Furthermore, we expect that target accuracies ε will need to be of the order of $\varepsilon = 10^{-3}$ to 10^{-5}. The runtime model measures time in terms of ansatz circuit durations. To convert this into seconds, we assume that each layer of the two-qubit gates will take time $G = 10^{-8}$ s, which is an optimistic assumption for today's superconducting qubit hardware. In Fig. 20 we show this estimated runtime as a function of the two-qubit gate fidelity.

The two-qubit gate fidelities required to reduce runtimes into a practical region will most likely require error correction. Performing quantum error correction requires an overhead that increases these runtimes. In designing quantum error-correction protocols, it is essential that the improvement in gate fidelities is not outweighed by the increase in estimation runtime. The proposed model gives a means of quantifying this trade-off: the product of gate infidelity and (error-corrected) gate time should decrease as useful error correction is incorporated. In practice, there are many subtleties that should be accounted for to make a more rigorous statement. These include considering the variation in gate fidelities among gates in the circuit and the varying time costs of different types of gates. Nevertheless, the cost analyses afforded by this simple model may be a useful tool in the design of quantum gates, quantum chips, error-correcting schemes, and noise mitigation schemes.

VII. OUTLOOK

This work is motivated by the impractical runtimes required by many near-term quantum algorithms. We aim to improve the performance of estimation subroutines that have relied on standard sampling, as used in the VQE. Drawing on the recent alpha-VQE [20] and quantum metrology [41], we investigate the technique of enhanced sampling to explore the continuum between standard sampling and quantum amplitude (or phase) estimation. In this continuum, we can make optimal use of the quantum coherence available on a given device to speed up estimation. Similar to standard sampling in the VQE, enhanced sampling does not require ancilla qubits. Quantum advantage for tasks relying on estimation will likely occur within this continuum rather than at one of the extremes.

Our central object of study is the quantum-generated likelihood function, relating measurement outcome data to a parameter of interest encoded in a quantum circuit. We explore engineering likelihood functions to optimize their statistical power. This leads to several insights for improving estimation. First, we should incorporate a well-calibrated noise model directly in the likelihood function to make inference robust to certain error. Second, we should choose a circuit depth (reflected in the number of enhanced sampling circuit layers) that balances gain in statistical power with accrual of error. Finally, we should tune generalized reflection angles to mitigate the effect of "dead spots" during the inference process.

We use engineered likelihood functions to carry out adaptive approximate Bayesian inference for parameter estimation. Carrying out this process in simulation requires us to build mathematical and algorithmic infrastructures. We develop mathematical tools for analyzing a class of quantum-generated likelihood functions. From this analysis, we propose several optimization algorithms for tuning circuit parameters to engineer likelihood functions. We investigate the performance of estimation using engineered likelihood functions and compare this to estimation using fixed likelihood functions. Finally, we propose a model for predicting the performance of enhanced sampling estimation algorithms as the quality of quantum devices is improved.

These simulations and the model lead to several insights. As highlighted in Sec. V B, for the degree of device error expected in the near term (two-qubit gate fidelities of approximately 99.92%), we show that enhanced sampling and engineered likelihood functions can be used to outperform standard sampling used in the VQE. Furthermore, these simulations suggest a nonstandard perspective on the tolerance of error in quantum algorithm implementations. We find that, for fixed gate fidelities, the best performance is achieved when we push circuit depths to a point where circuit fidelities are around the range of 0.5–0.7. This suggests that, compared to the logical circuit fidelities suggested in other works (e.g., 0.99 in Ref. [16]), we can afford a 50-fold increase in circuit depth. We arrive at this balance between fidelity and statistical power by taking estimation runtime to be the cost to minimize.

The runtime model developed in Sec. VI sheds light on the trade-off between gate times and gate fidelity for estimation. For gate times that are one-thousand times slower, the gate fidelities must have three more nines to achieve the same estimation runtimes. The runtime model gives insight on the role of quantum error correction in estimation algorithms. Roughly, we find that, for quantum error correction to be useful for estimation, the factor of increase in runtime from error-correction overhead must be less than the factor of decrease in logical gate error rates. Additionally, the runtime model predicts that, for a given estimation task, there is a level of logical gate fidelity beyond which further improvements effectively do not reduce runtimes (especially if time overhead is taken into account). For the 100-qubit example considered, seven nines in two-qubit gate fidelities sufficed.

We leave a number of questions for future investigation. In the VQE a set of techniques referred to as "grouping" are used to reduce the measurement count [46,64–67]. These grouping techniques allow sampling of multiple operators at once, providing a type of measurement parallelization. The grouping method introduced in Refs. [65,67] decomposes a Pauli Hamiltonian into sets of mutually anticommuting Pauli strings, which ensures that the sum within each set is a Hermitian reflection. This method of grouping is compatible with enhanced sampling, as the Hermitian reflections can be both measured and implemented as generalized reflections (i.e., an example of operator P). However, it remains to explore if the additional circuit depth incurred by implementing these reflections is worth the variance reduction in the resulting estimators. Beyond existing grouping techniques,

we anticipate opportunities for parallelizing measurements that are dedicated to enhanced sampling.

Our work emphasizes the importance of developing accurate error models at the algorithmic level. Efficiently learning the "nuisance parameters" of these models, or *likelihood function calibration*, will be an essential ingredient to realizing the performance gain of any enhanced sampling methods in the near term. The motivation is similar to that of randomized benchmarking [68], where measurement data are fit to models of gate noise. An important problem that we leave for future work is to improve methods for likelihood function calibration. Miscalibration can introduce a systematic bias in parameter estimates. A back-of-the-envelope calculation predicts that the relative error in estimation due to miscalibration bias is inversely proportional to the number of Grover iterates L (which is set proportionally to λ) and proportional to the absolute error in the likelihood function. Assuming that this absolute error grows sublinearly in L, we would expect a degree of robustness in the estimation procedure as L is increased. In future work we will explore in more detail to what precision we must calibrate the likelihood function so that the bias introduced by miscalibration (or model error) is negligible. Finally, we leave investigations into analytical upper and lower bounds on estimation performance to future work.

We have aimed to present a viable solution to the "measurement problem" [15] that plagues the VQE. It is likely that these methods will be needed to achieve quantum advantage for problems in quantum chemistry and materials. Furthermore, such amplitude estimation techniques may help to achieve quantum advantage for applications in finance and machine learning tasks as well. We hope that our model for estimation performance as a function of device metrics is useful in assessing the relative importance of a variety of quantum resources including qubit number, two-qubit gate fidelity, gate times, qubit stability, error-correction cycle time, readout error rate, and others.

ACKNOWLEDGMENTS

We would like to thank Pierre-Luc Dallaire-Demers, Amara Katabarwa, Jhonathan Romero, Max Radin, Peter Love, Yihui Quek, Jonny Olson, Hannah Sim, and Jens Eisert for insightful conversations and valuable feedback, and Christopher Savoie for grammatical tidying.

APPENDIX A: A SUFFICIENT MODEL OF CIRCUIT NOISE

Here we describe one circuit noise model that yields the likelihood function noise model of Eq. (23). We expect that this circuit noise model is too simplistic to describe the output density matrix of the circuit. However, we also expect that physically realistic variants of this circuit noise model will also lead to an exponentially decaying likelihood function bias. In other words, this circuit noise model is sufficient, but not necessary for yielding the likelihood function noise model used in this work. The circuit noise model assumes that the noisy version of each circuit layer $V(x_{2j})U(\theta;x_{2j-1})$ implements a mixture of the target operation and the completely depolarizing channel acting on the same input state, i.e.,

$$\mathcal{U}_j(\rho) = pV(x_{2j})U(\theta;x_{2j-1})\rho U^\dagger(\theta;x_{2j-1})V^\dagger(x_{2j}) + (1-p)\frac{I}{2^n}, \tag{A1}$$

where p is the *fidelity* of this layer. Under composition of such imperfect operations, the output state of the L-layer circuit becomes

$$\rho_L = p^L Q(\theta;\vec{x})|A\rangle\langle A|Q^\dagger(\theta;\vec{x}) + (1-p^L)\frac{I}{2^n}. \tag{A2}$$

This imperfect circuit is preceded by an imperfect preparation of $|A\rangle$ and followed by an imperfect measurement of P. In the context of randomized benchmarking, such errors are referred to as *state preparation and measurement* (SPAM) errors [69]. We also model SPAM error with a depolarizing model, taking the noisy preparation of $|A\rangle$ to be $p_{\mathrm{SP}}|A\rangle\langle A| + (1-p_{\mathrm{SP}})I/2^n$ and taking the noisy measurement of P to be the positive operator-valued measure $\{p_M(I+P)/2 + (1-p_M)I/2, p_M(I-P)/2 + (1-p_M)I/2\}$. Combining the SPAM error parameters into $\bar{p} = p_{\mathrm{SP}}p_M$, we arrive at a model for the noisy likelihood function

$$\mathbb{P}(d|\theta;f,\vec{x}) = \frac{1}{2}[1 + (-1)^d\bar{p}p^L\Delta(\theta;\vec{x})], \tag{A3}$$

where $f = \bar{p}p^L$ is the *fidelity* of the whole process for generating the ELF, and $\Delta(\theta,\vec{x})$ is the bias of the ideal likelihood function as defined in Eq. (22).

APPENDIX B: PROOF OF LEMMA 1

In this appendix, we prove Lemma 1, which states that the CSBD coefficient functions of $\Delta(\theta;\vec{x})$ and $\Delta'(\theta;\vec{x})$ with respect to x_j can be evaluated in $O(L)$ time for any $j \in \{1,2,\ldots,2L\}$.

For convenience, we introduce the following notation. Let $W_{2i} = U^\dagger(\theta;x_{2i+1}) = U(\theta;-x_{2i+1})$, $W_{2i+1} = V^\dagger(x_{2i+2}) = V(-x_{2i+2})$, $W_{4L-2i} = U(\theta;x_{2i+1})$, and $W_{4L-2i-1} = V(x_{2i+2})$, for $i = 0,1,\ldots,L-1$, and $W_{2L} = P(\theta)$. Furthermore, let $W_j' = \partial_\theta W_j$ for $j = 0,1,\ldots,4L$. Note that $W_j' = 0$ if j is odd. Then we define $P_{a,b} = W_aW_{a+1}\cdots W_b$ if $0 \le a \le b \le 4L$, and $P_{a,b} = I$ otherwise.

With this notation, Eq. (20) implies that

$$Q^{\dagger}(\theta;\vec{x}) = P_{0,a-1}W_aP_{a+1,2L-1} \quad \text{for all } 0 \le a \le 2L-1, \tag{B1}$$

$$Q(\theta;\vec{x}) = P_{2L+1,b-1}W_bP_{b+1,4L} \quad \text{for all } 2L+1 \le b \le 4L, \tag{B2}$$

$$Q^{\dagger}(\theta;\vec{x})P(\theta)Q(\theta;\vec{x}) = P_{0,a-1}W_aP_{a+1,b-1}W_bP_{b+1,4L} \quad \text{for all } 0 \le a < b \le 4L. \tag{B3}$$

Moreover, taking the partial derivative of Eq. (20) with respect to θ yields

$$\begin{aligned} Q'(\theta;\vec{x}) &= \frac{\partial Q(\theta;\vec{x})}{\partial\theta} \\ &= V(x_{2L})U'(\theta;x_{2L-1})V(x_{2L-2})U(\theta;x_{2L-3})\cdots V(x_4)U(\theta;x_3)V(x_2)U(\theta;x_1) \\ &\quad + V(x_{2L})U(\theta;x_{2L-1})V(x_{2L-2})U'(\theta;x_{2L-3})\cdots V(x_4)U(\theta;x_3)V(x_2)U(\theta;x_1) \\ &\quad + \cdots \\ &\quad + V(x_{2L})U(\theta;x_{2L-1})V(x_{2L-2})U(\theta;x_{2L-3})\cdots V(x_4)U'(\theta;x_3)V(x_2)U(\theta;x_1) \\ &\quad + V(x_{2L})U(\theta;x_{2L-1})V(x_{2L-2})U(\theta;x_{2L-3})\cdots V(x_4)U(\theta;x_3)V(x_2)U'(\theta;x_1), \end{aligned} \tag{B4}$$

where

$$U'(\theta;\alpha) = \frac{\partial U(\theta;\alpha)}{\partial\theta} = -i\sin(\alpha)P'(\theta) = i\sin(\alpha)[\sin(\theta)\bar{Z} - \cos(\theta)\bar{X}] \tag{B5}$$

is the partial derivative of $U(\theta;\alpha)$ with respect to θ, in which

$$P'(\theta) = -\sin(\theta)\bar{Z} + \cos(\theta)\bar{X} \tag{B6}$$

is the derivative of $P(\theta)$ with respect to θ. It follows that

$$\begin{aligned} Q'(\theta;\vec{x}) &= P_{2L+1,2L+1}W'_{2L+2}P_{2L+3,4L} + P_{2L+1,2L+3}W'_{2L+4}P_{2L+5,4L} + \cdots \\ &\quad + P_{2L+1,4L-3}W'_{4L-2}P_{4L-1,4L} + P_{2L+1,4L-1}W'_{4L}. \end{aligned} \tag{B7}$$

The following facts will be useful. Suppose that A, B, and C are arbitrary linear operators on the Hilbert space $\mathcal{H} = \text{span}\{|\bar{0}\rangle, |\bar{1}\rangle\}$. Then, by direct calculation, one can verify that

$$\begin{aligned} \langle\bar{0}|AV(-x)BV(x)C|\bar{0}\rangle &= \langle\bar{0}|A[\cos(x)\bar{I} + i\sin(x)\bar{Z}]B[\cos(x)\bar{I} - i\sin(x)\bar{Z}]C|\bar{0}\rangle \\ &= \frac{1}{2}[\cos(2x)\langle\bar{0}|A(B - \bar{Z}B\bar{Z})C|\bar{0}\rangle \\ &\quad - i\sin(2x)\langle\bar{0}|A(B\bar{Z} - \bar{Z}B)C|\bar{0}\rangle \\ &\quad + \langle\bar{0}|A(B + \bar{Z}B\bar{Z})C|\bar{0}\rangle], \end{aligned} \tag{B8}$$

$$\begin{aligned} \langle\bar{0}|AU(\theta;-x)BU(\theta;x)C|\bar{0}\rangle &= \langle\bar{0}|A[\cos(x)\bar{I} + i\sin(x)P(\theta)]B[\cos(x)\bar{I} - i\sin(x)P(\theta)]C|\bar{0}\rangle \\ &= \frac{1}{2}\{\cos(2x)\langle\bar{0}|A[B - P(\theta)BP(\theta)]C|\bar{0}\rangle \\ &\quad - i\sin(2x)\langle\bar{0}|A[BP(\theta) - P(\theta)B]C|\bar{0}\rangle \\ &\quad + \langle\bar{0}|A[B + P(\theta)BP(\theta)]C|\bar{0}\rangle\}, \end{aligned} \tag{B9}$$

and

$$\begin{aligned} \langle\bar{0}|AU(\theta;-x)BU'(\theta;x)C|\bar{0}\rangle &= \langle\bar{0}|A[\cos(x)\bar{I} + i\sin(x)P(\theta)]B[-i\sin(x)P'(\theta)]C|\bar{0}\rangle \\ &= \frac{1}{2}[-\cos(2x)\langle\bar{0}|AP(\theta)BP'(\theta)C|\bar{0}\rangle \\ &\quad - i\sin(2x)\langle\bar{0}|ABP'(\theta)C|\bar{0}\rangle + \langle\bar{0}|AP(\theta)BP'(\theta)C|\bar{0}\rangle]. \end{aligned} \tag{B10}$$

The following fact will also be useful. Taking the partial derivative of Eq. (22) with respect to θ yields

$$\begin{aligned}\Delta'(\theta;\vec{x}) &= \langle\bar{0}|Q^\dagger(\theta;\vec{x})P(\theta)Q'(\theta;\vec{x})|\bar{0}\rangle + \langle\bar{0}|Q^\dagger(\theta;\vec{x})P'(\theta)Q(\theta;\vec{x})|\bar{0}\rangle \\ &\quad + \langle\bar{0}|[Q'(\theta;\vec{x})]^\dagger P(\theta)Q(\theta;\vec{x})|\bar{0}\rangle \\ &= 2\mathrm{Re}[\langle\bar{0}|Q^\dagger(\theta;\vec{x})P(\theta)Q'(\theta;\vec{x})|\bar{0}\rangle] + \langle\bar{0}|Q^\dagger(\theta;\vec{x})P'(\theta)Q(\theta;\vec{x})|\bar{0}\rangle. \end{aligned} \tag{B11}$$

In order to evaluate $C_j(\theta;\vec{x}_{\neg j})$, $S_j(\theta;\vec{x}_{\neg j})$, $B_j(\theta;\vec{x}_{\neg j})$, $C'_j(\theta;\vec{x}_{\neg j})$, $S'_j(\theta;\vec{x}_{\neg j})$, and $B'_j(\theta;\vec{x}_{\neg j})$ for given θ and $\vec{x}_{\neg j}$, we consider the case j is even and the case j is odd separately.

Case 1: $j = 2(t+1)$ is even, where $0 \le t \le L-1$. In this case, $W_{2t+1} = V(-x_j)$ and $W_{4L-2t-1} = V(x_j)$. Then, by Eqs. (22), (B3), and (B8), we obtain

$$\begin{aligned}\Delta(\theta;\vec{x}) &= \langle\bar{0}|P_{0,2t}V(-x_j)P_{2t+2,4L-2t-2}V(x_j)P_{4L-2t,4L}|\bar{0}\rangle \\ &= C_j(\theta;\vec{x}_{\neg j})\cos(2x_j) + S_j(\theta;\vec{x}_{\neg j})\sin(2x_j) + B_j(\theta;\vec{x}_{\neg j}), \end{aligned} \tag{B12}$$

where

$$C_j(\theta;\vec{x}_{\neg j}) = \frac{1}{2}\langle\bar{0}|P_{0,2t}(P_{2t+2,4L-2t-2} - \bar{Z}P_{2t+2,4L-2t-2}\bar{Z})P_{4L-2t,4L}|\bar{0}\rangle, \tag{B13}$$

$$S_j(\theta;\vec{x}_{\neg j}) = -\frac{i}{2}\langle\bar{0}|P_{0,2t}(P_{2t+2,4L-2t-2}\bar{Z} - \bar{Z}P_{2t+2,4L-2t-2})P_{4L-2t,4L}|\bar{0}\rangle, \tag{B14}$$

$$B_j(\theta;\vec{x}_{\neg j}) = \frac{1}{2}\langle\bar{0}|P_{0,2t}(P_{2t+2,4L-2t-2} + \bar{Z}P_{2t+2,4L-2t-2}\bar{Z})P_{4L-2t,4L}|\bar{0}\rangle. \tag{B15}$$

Given θ and $\vec{x}_{\neg j}$, we first compute $P_{0,2t}$, $P_{2t+2,4L-2t-2}$, and $P_{4L-2t,4L}$ in $O(L)$ time. Then we calculate $C_j(\theta;\vec{x}_{\neg j})$, $S_j(\theta;\vec{x}_{\neg j})$, and $B_j(\theta;\vec{x}_{\neg j})$ by Eqs. (B13)–(B15). This procedure takes only $O(L)$ time.

Next, we show how to compute $C'_j(\theta;\vec{x}_{\neg j})$, $S'_j(\theta;\vec{x}_{\neg j})$, and $B'_j(\theta;\vec{x}_{\neg j})$. Using Eq. (B7) and the fact that $P_{a,b} = P_{a,4L-2t-2}W_{4L-2t-1}P_{4L-2t,b}$ for any $a \le 4L-2t-1 \le b$, we obtain

$$\begin{aligned}Q'(\theta;\vec{x}) &= P_{2L+1,2L+1}W'_{2L+2}P_{2L+3,4L-2t-2}W_{4L-2t-1}P_{4L-2t,4L} \\ &\quad + P_{2L+1,2L+3}W'_{2L+4}P_{2L+5,4L-2t-2}W_{4L-2t-1}P_{4L-2t,4L} \\ &\quad + \cdots \\ &\quad + P_{2L+1,4L-2t-3}W'_{4L-2t-2}W_{4L-2t-1}P_{4L-2t,4L} \\ &\quad + P_{2L+1,4L-2t-2}W_{4L-2t-1}W'_{4L-2t}P_{4L-2t+1,4L} \\ &\quad + \cdots \\ &\quad + P_{2L+1,4L-2t-2}W_{4L-2t-1}P_{4L-2t,4L-3}W'_{4L-2}P_{4L-1,4L} \\ &\quad + P_{2L+1,4L-2t-2}W_{4L-2t-1}P_{4L-2t,4L-1}W'_{4L}. \end{aligned} \tag{B16}$$

Then it follows from Eqs. (B1) and (B16) that

$$\begin{aligned}Q^\dagger(\theta;\vec{x})P(\theta)Q'(\theta;\vec{x}) &= A_t^{(1)}W_{2t+1}B_t^{(1)}W_{4L-2t-1}C_t^{(1)} + A_t^{(2)}W_{2t+1}B_t^{(2)}W_{4L-2t-1}C_t^{(2)} \\ &= A_t^{(1)}V(-x_j)B_t^{(1)}V(x_j)C_t^{(1)} + A_t^{(2)}V(-x_j)B_t^{(2)}V(x_j)C_t^{(2)}, \end{aligned} \tag{B17}$$

where

$$A_t^{(1)} = P_{0,2t}, \tag{B18}$$

$$B_t^{(1)} = P_{2t+2,4L-2t-2}, \tag{B19}$$

$$C_t^{(1)} = \sum_{k=0}^{t} P_{4L-2t,4L-2k-1} W'_{4L-2k} P_{4L-2k+1,4L}$$
$$= \sum_{k=0}^{t} P_{4L-2t,4L-2k-1} U'(\theta; x_{2k+1}) P_{4L-2k+1,4L}, \tag{B20}$$
$$A_t^{(2)} = P_{0,2t}, \tag{B21}$$
$$B_t^{(2)} = \sum_{k=t+1}^{L-1} P_{2t+2,4L-2k-1} W'_{4L-2k} P_{4L-2k+1,4L-2t-2}$$
$$= \sum_{k=t+1}^{L-1} P_{2t+2,4L-2k-1} U'(\theta; x_{2k+1}) P_{4L-2k+1,4L-2t-2}, \tag{B22}$$
$$C_t^{(2)} = P_{4L-2t,4L}. \tag{B23}$$

Meanwhile, we have

$$Q^\dagger(\theta;\vec{x}) P'(\theta) Q(\theta;\vec{x}) = A_t^{(3)} W_{2t+1} B_t^{(3)} W_{4L-2t-1} C_t^{(3)} = A_t^{(3)} V(-x_j) B_t^{(3)} V(x_j) C_t^{(3)}, \tag{B24}$$

where

$$A_t^{(3)} = P_{0,2t}, \tag{B25}$$
$$B_t^{(3)} = P_{2t+2,2L-1} P'(\theta) P_{2L+1,4L-2t-2}, \tag{B26}$$
$$C_t^{(3)} = P_{4L-2t,4L}. \tag{B27}$$

Combining the above facts with Eqs. (B8) and (B11) yields

$$\Delta'(\theta;\vec{x}) = C'_j(\theta;\vec{x}_{\neg j}) \cos(2x_j) + S'_j(\theta;\vec{x}_{\neg j}) \sin(2x_j) + B'_j(\theta;\vec{x}_{\neg j}), \tag{B28}$$

where

$$C'_j(\theta;\vec{x}_{\neg j}) = \mathrm{Re}[\langle\bar{0}|A_t^{(1)}(B_t^{(1)} - \bar{Z}B_t^{(1)}\bar{Z})C_t^{(1)}|\bar{0}\rangle] + \mathrm{Re}[\langle\bar{0}|A_t^{(2)}(B_t^{(2)} - \bar{Z}B_t^{(2)}\bar{Z})C_t^{(2)}|\bar{0}\rangle]$$
$$+ \frac{1}{2}\langle\bar{0}|A_t^{(3)}(B_t^{(3)} - \bar{Z}B_t^{(3)}\bar{Z})C_t^{(3)}|\bar{0}\rangle, \tag{B29}$$

$$S'_j(\theta;\vec{x}_{\neg j}) = \mathrm{Im}[\langle\bar{0}|A_t^{(1)}(B_t^{(1)}\bar{Z} - \bar{Z}B_t^{(1)})C_t^{(1)}|\bar{0}\rangle] + \mathrm{Im}[\langle\bar{0}|A_t^{(2)}(B_t^{(2)}\bar{Z} - \bar{Z}B_t^{(2)})C_t^{(2)}|\bar{0}\rangle]$$
$$- \frac{i}{2}\langle\bar{0}|[A_t^{(3)}(B_t^{(3)}\bar{Z} - \bar{Z}B_t^{(3)})C_t^{(3)}]|\bar{0}\rangle \tag{B30}$$

$$B'_j(\theta;\vec{x}_{\neg j}) = \mathrm{Re}[\langle\bar{0}|A_t^{(1)}(B_t^{(1)} + \bar{Z}B_t^{(1)}\bar{Z})C_t^{(1)}|\bar{0}\rangle] + \mathrm{Re}[\langle\bar{0}|A_t^{(2)}(B_t^{(2)} + \bar{Z}B_t^{(2)}\bar{Z})C_t^{(2)}|\bar{0}\rangle]$$
$$+ \frac{1}{2}\langle\bar{0}|A_t^{(3)}(B_t^{(3)} + \bar{Z}B_t^{(3)}\bar{Z})C_t^{(3)}|\bar{0}\rangle. \tag{B31}$$

Given θ and $\vec{x}_{\neg j}$, we first compute the following matrices in a total of $O(L)$ time by standard dynamic programming techniques:

(1) $P_{0,2t}, P_{2t+2,4L-2t-2}, P_{4L-2t,4L}, P_{2t+2,2L-1}, P_{2L+1,4L-2t-2}$;
(2) $P_{4L-2t,4L-2k-1}$ and $P_{4L-2k+1,4L}$ for $k = 0, 1, \ldots, t$;
(3) $P_{2t+2,4L-2k-1}$ and $P_{4L-2k+1,4L-2t-2}$ for $k = t+1, t+2, \ldots, L-1$.

Then we compute $A_t^{(i)}$, $B_t^{(i)}$, and $C_t^{(i)}$ for $i = 1, 2, 3$ by Eqs. (B18)–(B23) and (B25)–(B27). After that, we calculate $C_j'(\theta;\vec{x}_{\neg j})$, $S_j'(\theta;\vec{x}_{\neg j})$, and $B_j'(\theta;\vec{x}_{\neg j})$ by Eqs. (B29)–(B31). Overall, this procedure takes $O(L)$ time.

Case 2: $j = 2t+1$ is odd, where $0 \le t \le L-1$. In this case, $W_{2t} = U(\theta;-x_j)$ and $W_{4L-2t} = U(\theta;x_j)$. Then, by Eqs. (22), (B3), and (B9), we obtain

$$\begin{aligned}\Delta(\theta;\vec{x}) &= \langle\bar{0}|P_{0,2t-1}U(\theta;-x_j)P_{2t+1,4L-2t-1}U(\theta;x_j)P_{4L-2t+1,4L}|\bar{0}\rangle \\ &= C_j(\theta;\vec{x}_{\neg j})\cos(2x_j) + S_j(\theta;\vec{x}_{\neg j})\sin(2x_j) + B_j(\theta;\vec{x}_{\neg j}),\end{aligned} \tag{B32}$$

where

$$C_j(\theta;\vec{x}_{\neg j}) = \frac{1}{2}\langle\bar{0}|P_{0,2t-1}[P_{2t+1,4L-2t-2} - P(\theta)P_{2t+1,4L-2t-1}P(\theta)]P_{4L-2t+1,4L}|\bar{0}\rangle, \tag{B33}$$

$$S_j(\theta;\vec{x}_{\neg j}) = -\frac{i}{2}\langle\bar{0}|P_{0,2t-1}[P_{2t+1,4L-2t-1}P(\theta) - P(\theta)P_{2t+1,4L-2t-1}]P_{4L-2t+1,4L}|\bar{0}\rangle, \tag{B34}$$

$$B_j(\theta;\vec{x}_{\neg j}) = \frac{1}{2}\langle\bar{0}|P_{0,2t-1}[P_{2t+1,4L-2t-1} + P(\theta)P_{2t+1,4L-2t-1}P(\theta)]P_{4L-2t+1,4L}|\bar{0}\rangle. \tag{B35}$$

Given θ and $\vec{x}_{\neg j}$, we first compute $P_{0,2t-1}$, $P_{2t+1,4L-2t-1}$, and $P_{4L-2t+1,4L}$ in $O(L)$ time. Then we calculate $C_j(\theta;\vec{x}_{\neg j})$, $S_j(\theta;\vec{x}_{\neg j})$, and $B_j(\theta;\vec{x}_{\neg j})$ by Eqs. (B33)–(B35). This procedure takes only $O(L)$ time.

Next, we describe how to compute $C_j'(\theta;\vec{x}_{\neg j})$, $S_j'(\theta;\vec{x}_{\neg j})$, and $B_j'(\theta;\vec{x}_{\neg j})$. Using Eq. (B7) and the fact that $P_{a,b} = P_{a,4L-2t-1}W_{4L-2t}P_{4L-2t+1,b}$ for any $a \le 4L-2t \le b$, we obtain

$$\begin{aligned}Q'(\theta;\vec{x}) = {} & P_{2L+1,2L+1}W'_{2L+2}P_{2L+3,4L-2t-1}W_{4L-2t}P_{4L-2t+1,4L} \\ & + P_{2L+1,2L+3}W'_{2L+4}P_{2L+5,4L-2t-1}W_{4L-2t}P_{4L-2t+1,4L} \\ & + \cdots \\ & + P_{2L+1,4L-2t-3}W'_{4L-2t-2}P_{4L-2t-1,4L-2t-1}W_{4L-2t}P_{4L-2t+1,4L} \\ & + P_{2L+1,4L-2t-1}W'_{4L-2t}P_{4L-2t+1,4L} \\ & + P_{2L+1,4L-2t-1}W_{4L-2t}P_{4L-2t+1,4L-2t+1}W'_{4L-2t+2}P_{4L-2t+3,4L} \\ & + \cdots \\ & + P_{2L+1,4L-2t-1}W_{4L-2t}P_{4L-2t+1,4L-3}W'_{4L-2}P_{4L-1,4L} \\ & + P_{2L+1,4L-2t-1}W_{4L-2t}P_{4L-2t+1,4L-1}W'_{4L}.\end{aligned} \tag{B36}$$

Then it follows from Eqs. (B1) and (B16) that

$$\begin{aligned}Q^\dagger(\theta;\vec{x})P(\theta)Q'(\theta;\vec{x}) &= A_t^{(1)}W_{2t}B_t^{(1)}W_{4L-2t}C_t^{(1)} + A_t^{(2)}W_{2t}B_t^{(2)}W'_{4L-2t}C_t^{(2)} \\ &\quad + A_t^{(3)}W_{2t}B_t^{(3)}W_{4L-2t}C_t^{(3)} \\ &= A_t^{(1)}U(\theta;-x_j)B_t^{(1)}U(\theta;x_j)C_t^{(1)} + A_t^{(2)}U(\theta;-x_j)B_t^{(2)}U'(\theta;x_j)C_t^{(2)} \\ &\quad + A_t^{(3)}U(\theta;-x_j)B_t^{(3)}U(\theta;x_j)C_t^{(3)},\end{aligned} \tag{B37}$$

where

$$A_t^{(1)} = P_{0,2t-1}, \tag{B38}$$

$$B_t^{(1)} = P_{2t+1,4L-2t-1}, \tag{B39}$$

$$C_t^{(1)} = \sum_{k=0}^{t-1} P_{4L-2t+1,4L-2k-1} W'_{4L-2k} P_{4L-2k+1,4L}$$
$$= \sum_{k=0}^{t-1} P_{4L-2t+1,4L-2k-1} U'(\theta; x_{2k+1}) P_{4L-2k+1,4L}, \tag{B40}$$
$$A_t^{(2)} = P_{0,2t-1}, \tag{B41}$$
$$B_t^{(2)} = P_{2t+1,4L-2t-1}, \tag{B42}$$
$$C_t^{(2)} = P_{4L-2t+1,4L}, \tag{B43}$$
$$A_t^{(3)} = P_{0,2t-1}, \tag{B44}$$
$$B_t^{(3)} = \sum_{k=t+1}^{L-1} P_{2t+1,4L-2k-1} W'_{4L-2k} P_{4L-2k+1,4L-2t-1}$$
$$= \sum_{k=t+1}^{L-1} P_{2t+1,4L-2k-1} U'(\theta; x_{2k+1}) P_{4L-2k+1,4L-2t-1}, \tag{B45}$$
$$C_t^{(3)} = P_{4L-2t+1,4L}. \tag{B46}$$

Meanwhile, we have

$$Q^\dagger(\theta;\vec{x}) P'(\theta) Q(\theta;\vec{x}) = A_t^{(4)} W_{2t} B_t^{(4)} W_{4L-2t} C_t^{(4)} = A_t^{(4)} U(\theta; -x_j) B_t^{(4)} U(\theta; x_j) C_t^{(4)}, \tag{B47}$$

where

$$A_t^{(4)} = P_{0,2t-1}, \tag{B48}$$
$$B_t^{(4)} = P_{2t+1,2L-1} P'(\theta) P_{2L+1,4L-2t-1}, \tag{B49}$$
$$C_t^{(4)} = P_{4L-2t+1,4L}. \tag{B50}$$

Combining the above facts with Eqs. (B9), (B10), and (B11) yields

$$\Delta'(\theta;\vec{x}) = C'_j(\theta;\vec{x}_{\neg j}) \cos(2x_j) + S'_j(\theta;\vec{x}_{\neg j}) \sin(2x_j) + B'_j(\theta;\vec{x}_{\neg j}), \tag{B51}$$

where

$$C'_j(\theta;\vec{x}_{\neg j}) = \mathrm{Re}\{\langle\bar{0}|A_t^{(1)}[B_t^{(1)} - P(\theta)B_t^{(1)}P(\theta)]C_t^{(1)}|\bar{0}\rangle\}$$
$$- \mathrm{Re}[\langle\bar{0}|A_t^{(2)}P(\theta)B_t^{(2)}P'(\theta)C_t^{(2)}|\bar{0}\rangle]$$
$$+ \mathrm{Re}\{\langle\bar{0}|A_t^{(3)}[B_t^{(3)} - P(\theta)B_t^{(3)}P(\theta)]C_t^{(3)}|\bar{0}\rangle\}$$
$$+ \frac{1}{2}\langle\bar{0}|A_t^{(4)}[B_t^{(4)} - P(\theta)B_t^{(4)}P(\theta)]C_t^{(4)}|\bar{0}\rangle, \tag{B52}$$
$$S'_j(\theta;\vec{x}_{\neg j}) = \mathrm{Im}\{\langle\bar{0}|A_t^{(1)}[B_t^{(1)}P(\theta) - P(\theta)B_t^{(1)}]C_t^{(1)}|\bar{0}\rangle\}$$
$$+ \mathrm{Im}[\langle\bar{0}|A_t^{(2)}B_t^{(2)}P'(\theta)C_t^{(2)}|\bar{0}\rangle]$$
$$+ \mathrm{Im}\{\langle\bar{0}|A_t^{(3)}[B_t^{(3)}P(\theta) - P(\theta)B_t^{(3)}]C_t^{(3)}|\bar{0}\rangle\}$$
$$- \frac{i}{2}\langle\bar{0}|\{A_t^{(4)}[B_t^{(4)}P(\theta) - P(\theta)B_t^{(4)}]C_t^{(4)}\}|\bar{0}\rangle, \tag{B53}$$

$$\begin{aligned} B'_j(\theta;\vec{x}_{\neg j}) = {} & \mathrm{Re}\{\langle\bar{0}|A_t^{(1)}[B_t^{(1)} + P(\theta)B_t^{(1)}P(\theta)]C_t^{(1)}|\bar{0}\rangle\} \\ & + \mathrm{Re}[\langle\bar{0}|A_t^{(2)}P(\theta)B_t^{(2)}P'(\theta)C_t^{(2)}|\bar{0}\rangle] \\ & + \mathrm{Re}\{\langle\bar{0}|A_t^{(3)}[B_t^{(3)} + P(\theta)B_t^{(3)}P(\theta)]C_t^{(3)}|\bar{0}\rangle\} \\ & + \frac{1}{2}\langle\bar{0}|A_t^{(4)}[B_t^{(4)} + P(\theta)B_t^{(4)}P(\theta)]C_t^{(4)}|\bar{0}\rangle. \end{aligned} \tag{B54}$$

Given θ and $\vec{x}_{\neg j}$, we first compute the following matrices in a total of $O(L)$ time by standard dynamic programming techniques:

(1) $P_{0,2t-1}, P_{2t+1,4L-2t-1}, P_{4L-2t+1,4L}, P_{2t+1,2L-1}, P_{2L+1,4L-2t-1}$;
(2) $P_{4L-2t+1,4L-2k-1}$ and $P_{4L-2k+1,4L}$ for $k = 0, 1, \ldots, t-1$;
(3) $P_{2t+1,4L-2k-1}$ and $P_{4L-2k+1,4L-2t-1}$ for $k = t+1, t+2, \ldots, L-1$.

Then we compute $A_t^{(i)}$, $B_t^{(i)}$, and $C_t^{(i)}$ for $i = 1, 2, 3, 4$ by Eqs. (B38)–(B46) and (B48)–(B50). After that, we calculate $C'_j(\theta;\vec{x}_{\neg j})$, $S'_j(\theta;\vec{x}_{\neg j})$, and $B'_j(\theta;\vec{x}_{\neg j})$ by Eqs. (B52)–(B54). Overall, this procedure takes $O(L)$ time.

APPENDIX C: ALGORITHMS FOR MAXIMIZING THE SLOPE OF THE ANCILLA-FREE LIKELIHOOD FUNCTION

In this appendix, we present two algorithms for maximizing the slope of the ancilla-free likelihood function $\mathbb{P}(d|\theta;f,\vec{x})$ at a given point $\theta = \mu$ (i.e., the prior mean of θ). Namely, our goal is to find $\vec{x} \in \mathbb{R}^{2L}$ that maximizes $|\mathbb{P}'(\mu;f,\vec{x})| = f\,|\Delta'(\mu;\vec{x})|/2$.

Similar to Algorithms 1 and 2 for Fisher information maximization, our algorithms for slope maximization are also based on gradient ascent and coordinate ascent, respectively. They both need to call the procedures in Lemma 1 to evaluate $C'(\mu;\vec{x}_{\neg j})$, $S'(\mu;\vec{x}_{\neg j})$, and $B'(\mu;\vec{x}_{\neg j})$ for given μ and $\vec{x}_{\neg j}$. However, the gradient-ascent-based algorithm uses the above quantities to compute the partial derivative of $[\Delta'(\mu;\vec{x})]^2$ with respect to x_j, while the coordinate-ascent-based algorithm uses them to directly update the value of x_j. These algorithms are formally described in Algorithms 3 and 4, respectively.

APPENDIX D: ANCILLA-BASED SCHEME

In this appendix, we present an alternative scheme, called the *ancilla-based* scheme. In this scheme, the ELF is generated by the quantum circuit in Fig. 21, where $U(\theta;x_{2j-1})$, $V(x_{2j})$, and $Q(\theta;\vec{x})$ are the same as in the ancilla-free scheme [i.e., they satisfy Eqs. (18)–(20)], in which $\vec{x} = (x_1, x_2, \ldots, x_{2L-1}, x_{2L}) \in \mathbb{R}^{2L}$ are tunable parameters.

Assuming the circuit in Fig. 21 is noiseless, the AB ELF is given by

$$\mathbb{P}(d|\theta;\vec{x}) = \frac{1}{2}[1 + (-1)^d\Lambda(\theta;\vec{x})] \quad \text{for all } d \in \{0,1\}, \tag{D1}$$

where

$$\Lambda(\theta;\vec{x}) = \mathrm{Re}[\langle A|Q(\theta;\vec{x})|A\rangle] \tag{D2}$$

is the bias of the likelihood function. In particular, if $\vec{x} = (\pi/2, \pi/2, \ldots, \pi/2)$, we get $\Lambda(\theta;\vec{x}) = (-1)^L\cos(L\theta)$ and call the corresponding likelihood function the ancilla-based Chebyshev likelihood function.

It turns out that most of the argument in Sec. III A still holds in the ancilla-based case, except that we need to replace $\Delta(\theta;\vec{x})$ with $\Lambda(\theta;\vec{x})$. So we use the same notation (e.g., $|\bar{0}\rangle$, $|\bar{1}\rangle$, $\bar{X}$, $\bar{Y}$, $\bar{Z}$, $\bar{I}$) as before, unless otherwise stated. In particular, when we take the errors in the circuit in Fig. 21 into account, the noisy likelihood function is given by

$$\mathbb{P}(d|\theta;f,\vec{x}) = \frac{1}{2}[1 + (-1)^d f\,\Lambda(\theta;\vec{x})] \quad \text{for all } d \in \{0,1\}, \tag{D3}$$

where f is the fidelity of the process for generating the ELF. Note, however, that there does exist a difference between $\Delta(\theta;\vec{x})$ and $\Lambda(\theta;\vec{x})$, as the former is trigono-multiquadratic in $\vec{x}$, while the latter is trigono-multilinear in $\vec{x}$.

We tune the circuit parameters $\vec{x}$ and perform Bayesian inference with the resultant ELFs in the same way as in Sec. III B. In fact, the argument in Sec. III B still holds in the ancilla-based case, except that we need to replace $\Delta(\theta;\vec{x})$ with $\Lambda(\theta;\vec{x})$. So we use the same notation as before, unless otherwise stated. In particular, we also define the variance reduction factor $\mathcal{V}(\mu,\sigma;f,\vec{x})$ asin Eqs.

Algorithm 3: Gradient ascent for slope maximization in the ancilla-free case

Input: The prior mean μ of θ, the number L of circuit layers, the step size schedule $\delta : \mathbb{Z}^{\geq 0} \to \mathbb{R}^+$, the error tolerance ϵ for termination.
Output: A set of parameters $\vec{x} = (x_1, x_2, \ldots, x_{2L}) \in \mathbb{R}^{2L}$ that are a local maximum point of the function $|\Delta'(\mu;\vec{x})|$.
Choose random initial point $\vec{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_{2L}^{(0)}) \in (-\pi, \pi]^{2L}$;
$t \leftarrow 0$;
while *True* **do**
  for $j \leftarrow 1$ **to** $2L$ **do**
    Let $\vec{x}_{\neg j}^{(t)} = (x_1^{(t)}, \ldots, x_{j-1}^{(t)}, x_{j+1}^{(t)}, \ldots, x_{2L}^{(t)})$;
    Compute $C_j'^{(t)} := C_j'(\mu;\vec{x}_{\neg j}^{(t)})$, $S_j'^{(t)} := S_j'(\mu;\vec{x}_{\neg j}^{(t)})$ and $B_j'^{(t)} := B_j'(\mu;\vec{x}_{\neg j}^{(t)})$ by using the procedures in Lemma 1;
    Compute $\Delta'(\mu;\vec{x})$ at $\vec{x} = \vec{x}^{(t)}$ as follows:

$$\Delta'^{(t)} := \Delta'(\mu;\vec{x}^{(t)}) = C_j'^{(t)} \cos\left(2x_j^{(t)}\right) + S_j'^{(t)} \sin\left(2x_j^{(t)}\right) + B_j'^{(t)}; \tag{C1}$$

    Compute the partial derivative of $\Delta'(\mu;\vec{x})$ with respect to x_j as follows:

$$\gamma_j^{(t)} := \frac{\partial \Delta'(\mu;\vec{x})}{\partial x_j}|_{\vec{x}=\vec{x}^{(t)}} = 2\left(-C_j'^{(t)} \sin\left(2x_j^{(t)}\right) + S_j'^{(t)} \cos\left(2x_j^{(t)}\right)\right); \tag{C2}$$

  end
  Set $\vec{x}^{(t+1)} = \vec{x}^{(t)} + \delta(t)\nabla^{(t)}$, where $\nabla^{(t)} := (2\Delta'^{(t)}\gamma_1^{(t)}, 2\Delta'^{(t)}\gamma_2^{(t)}, \ldots, 2\Delta'^{(t)}\gamma_{2L}^{(t)})$ is the gradient of $(\Delta'(\mu;\vec{x}))^2$ at $\vec{x} = \vec{x}^{(t)}$;
  if $|\Delta'(\mu;\vec{x}^{(t+1)}) - \Delta'(\mu;\vec{x}^{(t)})| < \epsilon$ **then**
    break;
  end
  $t \leftarrow t + 1$;
end
Return $\vec{x}^{(t+1)} = (x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_{2L}^{(t+1)})$ as the optimal parameters.

(31) and (32), replacing $\Delta(\theta;\vec{x})$ with $\Lambda(\theta;\vec{x})$. As shown in Appendix F,

$$\mathcal{V}(\mu,\sigma;f,\vec{x}) \approx \mathcal{I}(\mu;f,\vec{x}) = \frac{f^2(\Lambda'(\theta;\vec{x}))^2}{1 - f^2[\Lambda(\theta;\vec{x})]^2} \quad \text{when } \sigma \text{ is small} \tag{D4}$$

and

$$\mathcal{V}(\mu,\sigma;f,\vec{x}) \approx f^2[\Lambda'(\mu;\vec{x})]^2 \quad \text{when both } \sigma \text{ and f are small.} \tag{D5}$$

Namely, the Fisher information and slope of the likelihood function $\mathbb{P}(d|\theta;f,\vec{x})$ at $\theta = \mu$ are two proxies of the variance reduction factor $\mathcal{V}(\mu,\sigma;f,\vec{x})$ under reasonable

Algorithm 4: Coordinate ascent for slope maximization in the ancilla-free case

Input: The prior mean μ of θ, the number L of circuit layers, the error tolerance ϵ for termination.
Output: A set of parameters $\vec{x} = (x_1, x_2, \ldots, x_{2L}) \in (-\pi, \pi]^{2L}$ that are a local maximum point of the function $|\Delta'(\mu;\vec{x})|$.
Choose random initial point $\vec{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_{2L}^{(0)}) \in (-\pi, \pi]^{2L}$;
$t \leftarrow 1$;
while *True* **do**
  for $j \leftarrow 1$ **to** $2L$ **do**
    Let $\vec{x}_{\neg j}^{(t)} = (x_1^{(t)}, \ldots, x_{j-1}^{(t)}, x_{j+1}^{(t-1)}, \ldots, x_{2L}^{(t-1)})$;
    Compute $C_j'^{(t)} := C_j'(\mu;\vec{x}_{\neg j}^{(t)})$, $S_j'^{(t)} := S_j'(\mu;\vec{x}_{\neg j}^{(t)})$, $B_j'^{(t)} := B_j'(\mu;\vec{x}_{\neg j}^{(t)})$ by using the procedures in Lemma 1;
    Set $x_j^{(t)} = \mathrm{Arg}\left(\mathrm{sgn}\left(B_j'^{(t)}\right)(C_j'^{(t)} + iS_j'^{(t)})\right)/2$, where $\mathrm{sgn}(z) = 1$ if $z \geq 0$ and -1 otherwise;
  end
  if $|\Delta'(\mu;\vec{x}^{(t)}) - \Delta'(\mu;\vec{x}^{(t-1)})| < \epsilon$ **then**
    break;
  end
  $t \leftarrow t + 1$;
end
Return $\vec{x}^{(t)} = (x_1^{(t)}, x_2^{(t)}, \ldots, x_{2L}^{(t)})$ as the optimal parameters.

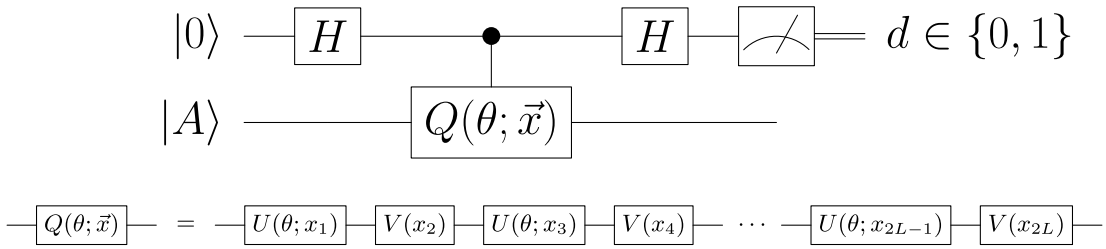


FIG. 21. Quantum circuit for the ancilla-based engineered likelihood functions.

assumptions. Since the direct optimization of $\mathcal{V}$ is hard in general, we tune the parameters $\vec{x}$ by optimizing these proxies instead.

1. Efficient maximization of proxies of the variance reduction factor

Now we present efficient heuristic algorithms for maximizing two proxies of the variance reduction factor $\mathcal{V}$—the Fisher information and slope of the likelihood function $\mathbb{P}(d|\theta; f, \vec{x})$. All of these algorithms make use of the following procedures for evaluating the CSD coefficient functions of the bias $\Lambda(\theta; \vec{x})$ and its derivative $\Lambda'(\theta; \vec{x})$ with respect to x_j for $j = 1, 2, \ldots, 2L$.

a. Evaluating the CSD coefficient functions of the bias and its derivative

Since $\Lambda(\theta; \vec{x})$ is trigono-multilinear in $\vec{x}$, for any $j \in \{1, 2, \ldots, 2L\}$, there exist functions $C_j(\theta; \vec{x}_{\neg j})$ and $S_j(\theta; \vec{x}_{\neg j})$, that are trigono-multilinear in $\vec{x}_{\neg j} = (x_1, \ldots,$

Algorithm 5: Gradient ascent for Fisher information maximization in the ancilla-based case

Input: The prior mean μ of θ, the number L of circuit layers, the fidelity f of the process for generating the ELF, the step size schedule $\delta : \mathbb{Z}^{\geq 0} \to \mathbb{R}^+$, the error tolerance ϵ for termination.

Output: A set of parameters $\vec{x} = (x_1, x_2, \ldots, x_{2L}) \in \mathbb{R}^{2L}$ that are a local maximum point of the function $\mathcal{I}(\mu; f, \vec{x})$.

Choose random initial point $\vec{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_{2L}^{(0)}) \in (-\pi, \pi]^{2L}$;
$t \leftarrow 0$;
while *True* **do**
  for $j \leftarrow 1$ **to** $2L$ **do**
    Let $\vec{x}_{\neg j}^{(t)} = (x_1^{(t)}, \ldots, x_{j-1}^{(t)}, x_{j+1}^{(t)}, \ldots, x_{2L}^{(t)})$;
    Compute $C_j^{(t)} := C_j(\mu; \vec{x}_{\neg j}^{(t)})$, $S_j^{(t)} := S_j(\mu; \vec{x}_{\neg j}^{(t)})$, $C_j'^{(t)} := C_j'(\mu; \vec{x}_{\neg j}^{(t)})$ and $S_j'^{(t)} := S_j'(\mu; \vec{x}_{\neg j}^{(t)})$ by using the procedures in Lemma 2;
    Compute $\Lambda(\mu; \vec{x})$, $\Lambda'(\mu; \vec{x})$ and their partial derivatives with respect to x_j at $\vec{x} = \vec{x}^{(t)}$ as follows:

$$\Lambda^{(t)} := \Lambda(\mu; \vec{x}^{(t)}) = C_j^{(t)} \cos(x_j) + S_j^{(t)} \sin(x_j), \tag{D33}$$

$$\Lambda'^{(t)} := \Lambda'(\mu; \vec{x}^{(t)}) = C_j'^{(t)} \cos(x_j) + S_j'^{(t)} \sin(x_j), \tag{D34}$$

$$\chi_j^{(t)} := \frac{\partial \Lambda(\mu; \vec{x})}{\partial x_j}|_{\vec{x}=\vec{x}^{(t)}} = -C_j^{(t)} \sin(x_j) + S_j^{(t)} \cos(x_j), \tag{D35}$$

$$\chi_j'^{(t)} := \frac{\partial \Lambda'(\mu; \vec{x})}{\partial x_j}|_{\vec{x}=\vec{x}^{(t)}} = -C_j'^{(t)} \sin(x_j) + S_j'^{(t)} \cos(x_j); \tag{D36}$$

    Compute the partial derivative of $\mathcal{I}(\mu; f, \vec{x})$ with respect to x_j at $\vec{x} = \vec{x}^{(t)}$ as follows:

$$\gamma_j^{(t)} := \frac{\partial \mathcal{I}(\mu; f, \vec{x})}{\partial x_j}|_{\vec{x}=\vec{x}^{(t)}} = \frac{2f^2\left[(1 - f^2(\Lambda^{(t)})^2)\Lambda'^{(t)}\chi_j'^{(t)} + f^2\Lambda^{(t)}\chi_j^{(t)}(\Lambda'^{(t)})^2\right]}{[1 - f^2(\Lambda^{(t)})^2]^2} \tag{D37}$$

  end
  Set $\vec{x}^{(t+1)} = \vec{x}^{(t)} + \delta(t)\nabla\mathcal{I}(\mu; f, \vec{x})|_{\vec{x}=\vec{x}^{(t)}}$, where $\nabla\mathcal{I}(\mu; f, \vec{x})|_{\vec{x}=\vec{x}^{(t)}} = (\gamma_1^{(t)}, \gamma_2^{(t)}, \ldots, \gamma_{2L}^{(t)})$;
  if $|\mathcal{I}(\mu; f, \vec{x}^{(t+1)}) - \mathcal{I}(\mu; f, \vec{x}^{(t)})| < \epsilon$ **then**
    break;
  end
  $t \leftarrow t + 1$;
end
Return $\vec{x}^{(t+1)} = (x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_{2L}^{(t+1)})$ as the optimal parameters.

Algorithm 6: Coordinate ascent for Fisher information maximization in the ancilla-based case

Input: The prior mean μ of θ, the number L of circuit layers, the fidelity f of the process for generating the ELF, the error tolerance ϵ for termination.

Output: A set of parameters $\vec{x} = (x_1, x_2, \ldots, x_{2L}) \in (-\pi, \pi]^{2L}$ that are a local maximum point of the function $\mathcal{I}(\mu; f, \vec{x})$.

Choose random initial point $\vec{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_{2L}^{(0)}) \in (-\pi, \pi]^{2L}$;
$t \leftarrow 1$;
while *True* **do**
  for $j \leftarrow 1$ **to** $2L$ **do**
    Let $\vec{x}_{\neg j}^{(t)} = (x_1^{(t)}, \ldots, x_{j-1}^{(t)}, x_{j+1}^{(t-1)}, \ldots, x_{2L}^{(t-1)})$;
    Compute $C_j^{(t)} := C_j(\mu; \vec{x}_{\neg j}^{(t)})$, $S_j^{(t)} := S_j(\mu; \vec{x}_{\neg j}^{(t)})$, $C_j'^{(t)} := C_j'(\mu; \vec{x}_{\neg j}^{(t)})$ and $S_j'^{(t)} := S_j'(\mu; \vec{x}_{\neg j}^{(t)})$ by using the procedures in Lemma 2;
    Solve the single-variable optimization problem

$$\arg\max_z \frac{f^2 \left(C_j'^{(t)} \cos(z) + S_j'^{(t)} \sin(z)\right)^2}{1 - f^2 \left(C_j^{(t)} \cos(z) + S_j^{(t)} \sin(z)\right)^2}$$

    by standard gradient-based methods and set $x_j^{(t)}$ to be its solution;
  end
  if $|\mathcal{I}(\mu; f, \vec{x}^{(t)}) - \mathcal{I}(\mu; f, \vec{x}^{(t-1)})| < \epsilon$ **then**
    break;
  end
  $t \leftarrow t + 1$;
end
Return $\vec{x}^{(t)} = (x_1^{(t)}, x_2^{(t)}, \ldots, x_{2L}^{(t)})$ as the optimal parameters.

$x_{j-1}, x_{j+1}, \ldots, x_{2L})$, such that

$$\Lambda(\theta; \vec{x}) = C_j(\theta; \vec{x}_{\neg j}) \cos(x_j) + S_j(\theta; \vec{x}_{\neg j}) \sin(x_j). \quad \text{(D6)}$$

It follows that

$$\Lambda'(\theta; \vec{x}) = C_j'(\theta; \vec{x}_{\neg j}) \cos(x_j) + S_j'(\theta; \vec{x}_{\neg j}) \sin(x_j) \quad \text{(D7)}$$

is also trigono-multilinear in $\vec{x}$, where $C_j'(\theta; \vec{x}_{\neg j}) = \partial_\theta C_j(\theta; \vec{x}_{\neg j})$ and $S_j'(\theta; \vec{x}_{\neg j}) = \partial_\theta S_j(\theta; \vec{x}_{\neg j})$ are the partial derivatives of $C_j(\theta; \vec{x}_{\neg j})$ and $S_j(\theta; \vec{x}_{\neg j})$ with respect to θ, respectively.

Our optimization algorithms require efficient procedures for evaluating $C_j(\theta; \vec{x}_{\neg j})$, $S_j(\theta; \vec{x}_{\neg j})$, $C_j'(\theta; \vec{x}_{\neg j})$, and $S_j'(\theta; \vec{x}_{\neg j})$ for given θ and $\vec{x}_{\neg j}$. It turns out that these tasks can be accomplished in $O(L)$ time.

Lemma 2. *Given θ and $\vec{x}_{\neg j}$, each of $C_j(\theta; \vec{x}_{\neg j})$, $S_j(\theta; \vec{x}_{\neg j})$, $C_j'(\theta; \vec{x}_{\neg j})$, and $S_j'(\theta; \vec{x}_{\neg j})$ can be computed in $O(L)$ time.*

Proof. For convenience, we introduce the following notation. Let $W_{2i} = V(x_{2L-2i})$ and $W_{2i+1} = U(\theta; x_{2L-2i-1})$ for $i = 0, 1, \ldots, L-1$. Furthermore, let $W_j' = \partial_\theta W_j$ for $j = 0, 1, \ldots, 2L-1$. Note that $W_j' = 0$ if j is even. Then we define $P_{a,b} = W_a W_{a+1} \cdots W_b$ if $0 \le a \le b \le 2L-1$, and $P_{a,b} = I$ otherwise.

With this notation, Eqs. (20) and (B4) imply that

$$Q(\theta; \vec{x}) = P_{0,a-1} W_a P_{a+1,2L-1} \quad \text{for all } 0 \le a \le 2L-1 \quad \text{(D8)}$$

and

$$Q'(\theta; \vec{x}) = P_{0,0} W_1' P_{2,2L-1} + P_{0,2} W_3' P_{4,2L-1} + \cdots + P_{0,2L-4} W_{2L-3}' P_{2L-2,2L-1} + P_{0,2L-2} W_{2L-1}'. \quad \text{(D9)}$$

In order to evaluate $C_j(\theta; \vec{x}_{\neg j})$, $S_j(\theta; \vec{x}_{\neg j})$, $C_j'(\theta; \vec{x}_{\neg j})$, and $S_j'(\theta; \vec{x}_{\neg j})$ for given θ and $\vec{x}_{\neg j}$, we consider the case j is even and the case j is odd separately.

Case 1: $j = 2(L-t)$ is even, where $0 \le t \le L-1$. In this case, $W_{2t} = V(x_j)$. Using the fact that

$$\begin{aligned} Q(\theta; \vec{x}) &= P_{0,2t-1} W_{2t} P_{2t+1,2L-1} \\ &= P_{0,2t-1}[\cos(x_j)\bar{I} - i\sin(x_j)\bar{Z}]P_{2t+1,2L-1} \\ &= \cos(x_j) P_{0,2t-1} P_{2t+1,2L-1} \\ &\quad - i\sin(x_j) P_{0,2t-1} \bar{Z} P_{2t+1,2L-1}, \end{aligned} \quad \text{(D10)}$$

we obtain

$$\Lambda(\theta; \vec{x}) = C_j(\theta; \vec{x}_{\neg j}) \cos(x_j) + S_j(\theta; \vec{x}_{\neg j}) \sin(x_j), \quad \text{(D11)}$$

where

$$C_j(\theta; \vec{x}_{\neg j}) = \mathrm{Re}(\langle \bar{0} | P_{0,2t-1} P_{2t+1,2L-1} | \bar{0} \rangle), \quad \text{(D12)}$$

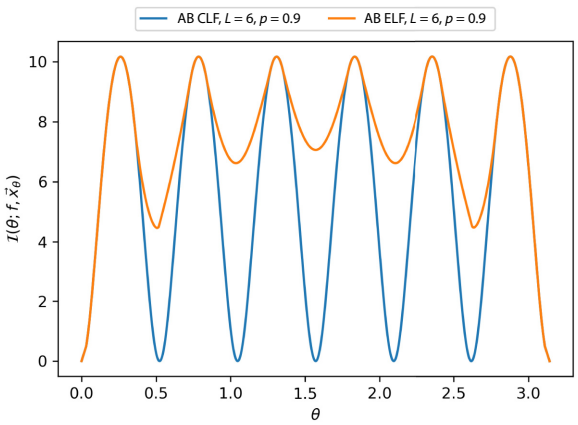


FIG. 22. Comparison of the Fisher information of the AB ELF and AB CLF for various $\theta \in (0, \pi)$ when the number of circuit layers is $L = 6$, the fidelity of each layer is $p = 0.9$, and there is no SPAM error (i.e., $\bar{p} = 1$). For the AB ELF, $\vec{x}_\theta$ is a global maximum point of $\mathcal{I}(\theta; f, \vec{x})$ for given θ and $f = \bar{p}p^L = 0.531\,441$. For the AB CLF, $\vec{x}_\theta = (\pi/2, \pi/2, \ldots, \pi/2)$ is fixed. One can see that the Fisher information of the AB ELF is larger than that of the AB CLF for the majority of $\theta \in (0, \pi)$. Furthermore, the Fisher information of the AB CLF changes dramatically for different θ (in fact, it is exactly 0 when $\theta = j\pi/L$ for $j = 0, 1, \ldots, L$), whereas the Fisher information of the AB ELF is less sensitive to the value of θ.

$$S_j(\theta; \vec{x}_{\neg j}) = \mathrm{Im}(\langle \bar{0} | P_{0,2t-1} \bar{Z} P_{2t+1,2L-1} | \bar{0} \rangle). \tag{D13}$$

Given θ and $\vec{x}_{\neg j}$, we first compute $P_{0,2t-1}$ and $P_{2t+1,2L-1}$ in $O(L)$ time. Then we calculate $C_j(\theta; \vec{x}_{\neg j})$ and $S_j(\theta; \vec{x}_{\neg j})$ by Eqs. (D12) and (D13). This procedure takes only $O(L)$ time.

Next, we describe how to compute $C'_j(\theta; \vec{x}_{\neg j})$ and $S'_j(\theta; \vec{x}_{\neg j})$. Using Eq. (D9) and the fact that $P_{a,b} = P_{a,2t-1} W_{2t} P_{2t+1,b}$ for any $a \le 2t \le b$, we obtain

$$\begin{aligned} Q'(\theta; \vec{x}) = {} & P_{0,0} W'_1 P_{2,2t-1} W_{2t} P_{2t+1,2L-1} \\ & + P_{0,2} W'_3 P_{4,2t-1} W_{2t} P_{2t+1,2L-1} \\ & + \cdots \\ & + P_{0,2t-2} W'_{2t-1} W_{2t} P_{2t+1,2L-1} \\ & + P_{0,2t-1} W_{2t} W'_{2t+1} P_{2t+2,2L-1} \\ & + \cdots \\ & + P_{0,2t-1} W_{2t} P_{2t+1,2L-4} W'_{2L-3} P_{2L-2,2L-1} \\ & + P_{0,2t-1} W_{2t} P_{2t+1,2L-2} W'_{2L-1}. \end{aligned} \tag{D14}$$

Let

$$\begin{aligned} A_t &= \sum_{s=1}^{t} P_{0,2s-2} W'_{2s-1} P_{2s,2t-1} \\ &= \sum_{s=1}^{t} P_{0,2s-2} U'(\theta; x_{2L-2s+1}) P_{2s,2t-1}, \end{aligned} \tag{D15}$$

$$\begin{aligned} B_t &= \sum_{s=t+1}^{L} P_{2t+1,2s-2} W'_{2s-1} P_{2s,2L-1} \\ &= \sum_{s=t+1}^{L} P_{2t+1,2s-2} U'(\theta; x_{2L-2s+1}) P_{2s,2L-1}. \end{aligned} \tag{D16}$$

Then Eq. (D14) yields

$$\begin{aligned} Q'(\theta; \vec{x}) &= A_t W_{2t} P_{2t+1,2L-1} + P_{0,2t-1} W_{2t} B_t \\ &= A_t [\cos(x_j) \bar{I} - i \sin(x_j) \bar{Z}] P_{2t+1,2L-1} \\ &\quad + P_{0,2t-1} [\cos(x_j) \bar{I} - i \sin(x_j) \bar{Z}] B_t \\ &= \cos(x_j)(A_t P_{2t+1,2L-1} + P_{0,2t-1} B_t) \\ &\quad - i \sin(x_j)(A_t \bar{Z} P_{2t+1,2L-1} + P_{0,2t-1} \bar{Z} B_t), \end{aligned} \tag{D17}$$

which leads to

$$\Lambda'(\theta; \vec{x}) = C'_j(\theta; \vec{x}_{\neg j}) \cos(x_j) + S'_j(\theta; \vec{x}_{\neg j}) \sin(x_j), \tag{D18}$$

where

$$C'_j(\theta; \vec{x}_{\neg j}) = \mathrm{Re}[\langle \bar{0} | (A_t P_{2t+1,2L-1} + P_{0,2t-1} B_t) | \bar{0} \rangle], \tag{D19}$$

$$S'_j(\theta; \vec{x}_{\neg j}) = \mathrm{Im}[\langle \bar{0} | (A_t \bar{Z} P_{2t+1,2L-1} + P_{0,2t-1} \bar{Z} B_t) | \bar{0} \rangle]. \tag{D20}$$

Given θ and $\vec{x}_{\neg j}$, we first compute the following matrices in a total of $O(L)$ time by standard dynamic programming techniques:

(1) $P_{0,2s-2}$ and $P_{2s,2t-1}$ for $s = 1, 2, \ldots, t$;
(2) $P_{2t+1,2s-2}$ and $P_{2s,2L-1}$ for $s = t+1, t+2, \ldots, L$;
(3) $P_{0,2t-1}$ and $P_{2t+1,2L-1}$.

Then we compute A_t and B_t by Eqs. (D15) and (D16). After that, we calculate $C'_j(\theta; \vec{x}_{\neg j})$ and $S'_j(\theta; \vec{x}_{\neg j})$ by Eqs. (D19) and (D20). Overall, this procedure takes $O(L)$ time.

Algorithm 7: Gradient ascent for slope maximization in the ancilla-based case

Input: The prior mean μ of θ, the number L of circuit layers, the step size schedule $\delta : \mathbb{Z}^{\geq 0} \to \mathbb{R}^+$, the error tolerance ϵ for termination.

Output: A set of parameters $\vec{x} = (x_1, x_2, \ldots, x_{2L}) \in \mathbb{R}^{2L}$ that are a local maximum point of the function $|\Lambda'(\mu; \vec{x})|$.

Choose random initial point $\vec{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_{2L}^{(0)}) \in (-\pi, \pi]^{2L}$;
$t \leftarrow 0$;
while *True* **do**
  for $j \leftarrow 1$ **to** $2L$ **do**
    Let $\vec{x}_{\neg j}^{(t)} = (x_1^{(t)}, \ldots, x_{j-1}^{(t)}, x_{j+1}^{(t)}, \ldots, x_{2L}^{(t)})$;
    Compute $C_j'^{(t)} := C_j'(\mu; \vec{x}_{\neg j}^{(t)})$ and $S_j'^{(t)} := S_j'(\mu; \vec{x}_{\neg j}^{(t)})$ by using the procedures in Lemma 2;
    Compute $\Lambda'(\mu; \vec{x})$ at $\vec{x} = \vec{x}^{(t)}$ as follows:

$$\Lambda'^{(t)} := \Lambda'(\mu; \vec{x}^{(t)}) = C_j'^{(t)} \cos\left(x_j^{(t)}\right) + S_j'^{(t)} \sin\left(x_j^{(t)}\right); \tag{D38}$$

    Compute the partial derivative of $\Lambda'(\mu; \vec{x})$ with respect to x_j as follows:

$$\gamma_j^{(t)} := \frac{\partial \Lambda'(\mu; \vec{x})}{\partial x_j}|_{\vec{x}=\vec{x}^{(t)}} = -C_j'^{(t)} \sin\left(x_j^{(t)}\right) + S_j'^{(t)} \cos\left(x_j^{(t)}\right); \tag{D39}$$

  end
  Set $\vec{x}^{(t+1)} = \vec{x}^{(t)} + \delta(t)\nabla^{(t)}$, where $\nabla^{(t)} := (2\Lambda'^{(t)}\gamma_1^{(t)}, 2\Lambda'^{(t)}\gamma_2^{(t)}, \ldots, 2\Lambda'^{(t)}\gamma_{2L}^{(t)})$ is the gradient of $(\Lambda'(\mu; \vec{x}))^2$ at $\vec{x} = \vec{x}^{(t)}$;
  if $|\Lambda'(\mu; \vec{x}^{(t+1)}) - \Lambda'(\mu; \vec{x}^{(t)})| < \epsilon$ **then**
    break;
  end
  $t \leftarrow t + 1$;
end
Return $\vec{x}^{(t+1)} = (x_1^{(t+1)}, x_2^{(t+1)}, \ldots, x_{2L}^{(t+1)})$ as the optimal parameters.

Case 2: $j = 2(L-t) - 1$ is odd, where $0 \leq t \leq L-1$. In this case, $W_{2t+1} = U(\theta; x_j)$. Using the fact that

$$\begin{aligned} Q(\theta; \vec{x}) &= P_{0,2t} W_{2t+1} P_{2t+2,2L-1} \\ &= P_{0,2t}[\cos(x_j)\bar{I} - i\sin(x_j)P(\theta)]P_{2t+2,2L-1} \\ &= \cos(x_j) P_{0,2t} P_{2t+2,2L-1} \\ &\quad - i\sin(x_j) P_{0,2t} P(\theta) P_{2t+2,2L-1}, \end{aligned} \tag{D21}$$

we obtain

$$\Lambda(\theta; \vec{x}) = C_j(\theta; \vec{x}_{\neg j})\cos(x_j) + S_j(\theta; \vec{x}_{\neg j})\sin(x_j), \tag{D22}$$

where

$$C_j(\theta; \vec{x}_{\neg j}) = \mathrm{Re}(\langle \bar{0}| P_{0,2t} P_{2t+2,2L-1} |\bar{0}\rangle), \tag{D23}$$

$$S_j(\theta; \vec{x}_{\neg j}) = \mathrm{Im}[\langle \bar{0}| P_{0,2t} P(\theta) P_{2t+2,2L-1} |\bar{0}\rangle]. \tag{D24}$$

Given θ and $\vec{x}_{\neg j}$, we first compute $P_{0,2t}$ and $P_{2t+2,2L-1}$ in $O(L)$ time. Then we calculate $C_j(\theta; \vec{x}_{\neg j})$ and $S_j(\theta; \vec{x}_{\neg j})$ by Eqs. (D23) and (D24). This procedure takes only $O(L)$ time.

Next, we describe how to compute $C_j'(\theta; \vec{x}_{\neg j})$ and $S_j'(\theta; \vec{x}_{\neg j})$. Using Eq. (D9) and the fact that $P_{a,b} = P_{a,2t} W_{2t+1} P_{2t+2,b}$ for any $a \leq 2t+1 \leq b$, we get

$$\begin{aligned} Q'(\theta; \vec{x}) = {} & P_{0,0} W_1' P_{2,2t} W_{2t+1} P_{2t+2,2L-1} \\ & + P_{0,2} W_3' P_{4,2t} W_{2t+1} P_{2t+2,2L-1} \\ & + \cdots \\ & + P_{0,2t-2} W_{2t-1}' P_{2t,2t} W_{2t+1} P_{2t+2,2L-1} \\ & + P_{0,2t} W_{2t+1}' P_{2t+2,2L-1} \\ & + P_{0,2t} W_{2t+1} P_{2t+2,2t+2} W_{2t+3}' P_{2t+4,2L-1} \\ & + \cdots \\ & + P_{0,2t} W_{2t+1} P_{2t+2,2L-4} W_{2L-3}' P_{2L-2,2L-1} \\ & + P_{0,2t} W_{2t+1} P_{2t+2,2L-2} W_{2L-1}'. \end{aligned} \tag{D25}$$

Let

$$\begin{aligned} A_t &= \sum_{s=1}^{t} P_{0,2s-2} W_{2s-1}' P_{2s,2t} \\ &= \sum_{s=1}^{t} P_{0,2s-2} U'(\theta; x_{2L-2s+1}) P_{2s,2t}, \end{aligned} \tag{D26}$$

$$B_t = \sum_{s=t+2}^{L} P_{2t+2,2s-2} W_{2s-1}' P_{2s,2L-1}$$

Algorithm 8: Coordinate ascent for slope optimization in the ancilla-based case

Input: The prior mean μ of θ, the number L of circuit layers, the error tolerance ϵ for termination.
Output: A set of parameters $\vec{x} = (x_1, x_2, \ldots, x_{2L}) \in (-\pi, \pi]^{2L}$ that are a local maximum point of the function $|\Lambda'(\mu; \vec{x})|$.

```
Choose random initial point x⃗^(0) = (x_1^(0), x_2^(0), ..., x_2L^(0)) ∈ (−π, π]^2L;
t ← 1;
while True do
    for j ← 1 to 2L do
        Let x⃗_¬j^(t) = (x_1^(t), ..., x_{j−1}^(t), x_{j+1}^(t−1), ..., x_2L^(t−1));
        Compute C'_j^(t) := C'_j(μ; x⃗_¬j^(t)) and S'_j^(t) := S'_j(μ; x⃗_¬j^(t)) by using the procedures in Lemma 2;
        Set x_j^(t) = Arg(C'_j^(t) + i S'_j^(t));
    end
    if |Λ'(μ; x⃗^(t)) − Λ'(μ; x⃗^(t−1))| < ε then
        break;
    end
    t ← t + 1;
end
Return x⃗^(t) = (x_1^(t), x_2^(t), ..., x_2L^(t)) as the optimal parameters.
```

$$= \sum_{s=t+2}^{L} P_{2t+2,2s-2} U'(\theta; x_{2L-2s+1}) P_{2s,2L-1}. \tag{D27}$$

Then Eq. (D25) yields

$$\begin{aligned} Q'(\theta;\vec{x}) &= A_t W_{2t+1} P_{2t+2,2L-1} + P_{0,2t} W'_{2t+1} P_{2t+2,2L-1} \\ &\quad + P_{0,2t} W_{2t+1} B_t \\ &= A_t[\cos(x_j)\bar{I} - i\sin(x_j)P(\theta)]P_{2t+2,2L-1} \\ &\quad - i\sin(x_j)P_{0,2t}P'(\theta)P_{2t+2,2L-1} \\ &\quad + P_{0,2t}[\cos(x_j)\bar{I} - i\sin(x_j)P(\theta)]B_t \\ &= \cos(x_j)(A_t P_{2t+2,2L-1} + P_{0,2t}B_t) \\ &\quad - i\sin(x_j)[A_t P(\theta)P_{2t+2,2L-1} \\ &\quad + P_{0,2t}P'(\theta)P_{2t+2,2L-1} + P_{0,2t}P(\theta)B_t], \end{aligned} \tag{D28}$$

which leads to

$$\Lambda'(\theta;\vec{x}) = C'_j(\theta;\vec{x}_{\neg j})\cos(x_j) + S'_j(\theta;\vec{x}_{\neg j})\sin(x_j), \tag{D29}$$

where

$$C'_j(\theta;\vec{x}_{\neg j}) = \mathrm{Re}[\langle\bar{0}|(A_t P_{2t+2,2L-1} + P_{0,2t}B_t)|\bar{0}\rangle], \tag{D30}$$

$$\begin{aligned} S'_j(\theta;\vec{x}_{\neg j}) &= \mathrm{Im}\{\langle\bar{0}|[A_t P(\theta)P_{2t+2,2L-1} \\ &\quad + P_{0,2t}P'(\theta)P_{2t+2,2L-1} + P_{0,2t}P(\theta)B_t]|\bar{0}\rangle\}. \end{aligned} \tag{D31}$$

Given θ and $\vec{x}_{\neg j}$, we first compute the following matrices in a total of $O(L)$ time by standard dynamic programming techniques:

(1) $P_{0,2s-2}$ and $P_{2s,2t}$ for $s = 1, 2, \ldots, t$;
(2) $P_{2t+2,2s-2}$ and $P_{2s,2L-1}$ for $s = t+2, t+3, \ldots, L$;
(3) $P_{0,2t}$ and $P_{2t+2,2L-1}$.

Then we compute A_t and B_t by Eqs. (D26) and (D27). After that, we calculate $C'_j(\theta;\vec{x}_{\neg j})$ and $S'_j(\theta;\vec{x}_{\neg j})$ by Eqs. (D30) and (D31). Overall, this procedure takes $O(L)$ time. ■

b. Maximizing the Fisher information of the likelihood function

We propose two algorithms for maximizing the Fisher information of the likelihood function $\mathbb{P}(d|\theta; f, \vec{x})$ at a given point $\theta = \mu$ (i.e., the prior mean of θ). Namely, our goal is to find $\vec{x} \in \mathbb{R}^{2L}$ that maximizes

$$\mathcal{I}(\mu; f, \vec{x}) = \frac{f^2[\Lambda'(\mu;\vec{x})]^2}{1 - f^2[\Lambda(\mu;\vec{x})]^2}. \tag{D32}$$

These algorithms are similar to Algorithms 1 and 2 for Fisher information maximization in the ancilla-free case, in the sense that they are also based on gradient ascent and coordinate ascent, respectively. The main difference is that now we invoke the procedures in Lemma 2 to evaluate $C(\mu;\vec{x}_{\neg j})$, $S(\mu;\vec{x}_{\neg j})$, $C'(\mu;\vec{x}_{\neg j})$, and $S'(\mu;\vec{x}_{\neg j})$ for given μ and $\vec{x}_{\neg j}$, and then use them to either compute the partial derivative of $\mathcal{I}(\mu; f, \vec{x})$ with respect to x_j (in gradient ascent) or define a single-variable optimization problem for x_j (in coordinate ascent). These algorithms are formally described in Algorithms 5 and 6.

We have used Algorithms 5 and 6 to find the parameters $\vec{x}_\theta \in \mathbb{R}^{2L}$ that maximize $\mathcal{I}(\theta; f, \vec{x})$ for various $\theta \in (0, \pi)$ (fixing f) and obtained Fig. 22. This figure indicates that the Fisher information of the AB ELF is larger than that of the AB CLF for the majority of $\theta \in (0, \pi)$. Consequently,

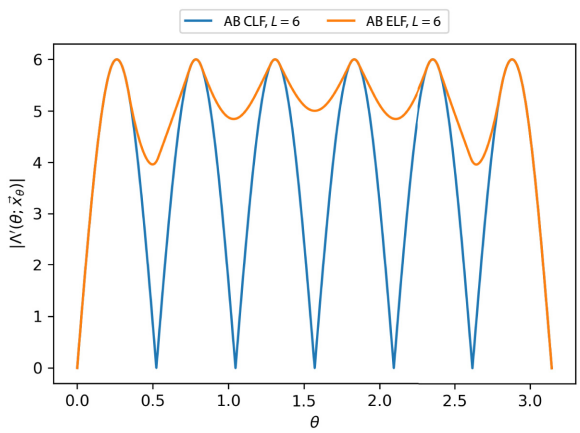


FIG. 23. Comparison of the values of $|\Lambda'(\theta;\vec{x}_\theta)|$ for the slope-based AB ELF and AB CLF for various $\theta \in (0,\pi)$ when the number of circuit layers is $L = 6$. For the AB ELF, $\vec{x}_\theta$ is a global maximum point of $|\Lambda'(\theta;\vec{x})|$ for given θ. For the AB CLF, $\vec{x}_\theta = (\pi/2, \pi/2, \ldots, \pi/2)$ is fixed. This figure implies that the slope-based AB ELF is steeper than the AB CLF for the majority of $\theta \in (0,\pi)$. Furthermore, the slope of the AB CLF changes dramatically for different θ (in fact, it is exactly 0 when $\theta = j\pi/L$ for $j = 0, 1, \ldots, L$), whereas the slope of the AB ELF is less sensitive to the value of θ.

the estimation algorithm based on the AB ELF is more efficient than that based on the AB CLF, as demonstrated in Sec. V.

c. Maximizing the slope of the likelihood function

We also propose two algorithms for maximizing the slope of the likelihood function $\mathbb{P}(d|\theta;f,\vec{x})$ at a given point $\theta = \mu$ (i.e., the prior mean of θ). Namely, our goal is to find $\vec{x} \in \mathbb{R}^{2L}$ that maximizes $|\mathbb{P}'(d|\mu;f,\vec{x})| = f\,|\Lambda'(\mu;\vec{x})|/2$.

These algorithms are similar to Algorithms 3 and 4 for slope maximization in the ancilla-free case, in the sense that they are also based on gradient ascent and coordinate ascent, respectively. The main difference is that now we invoke the procedures in Lemma 2 to evaluate $C'(\mu;\vec{x}_{\neg j})$ and $S'(\mu;\vec{x}_{\neg j})$ for given μ and $\vec{x}_{\neg j}$. Then we use these quantities to either compute the partial derivative of $[\Lambda(\mu;\vec{x})]^2$ with respect to x_j (in gradient ascent) or directly update the value of x_j (in coordinate ascent). These algorithms are formally described in Algorithms 7 and 8.

We have also used Algorithms 7 and 8 to find the parameters $\vec{x}_\theta$ that maximize $|\Lambda'(\theta;\vec{x})|$ for various $\theta \in (0,\pi)$ and obtained Fig. 23. This figure implies that the slope-based AB ELF is steeper than the AB CLF for the majority of $\theta \in (0,\pi)$ and hence has more statistical power than the AB CLF (at least) in the low-fidelity setting by Eq. (D5).

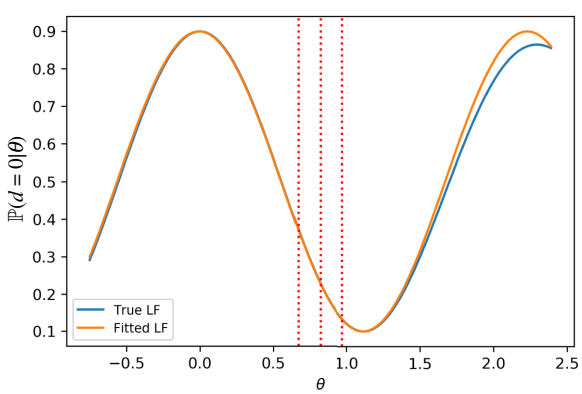


FIG. 24. The true and fitted likelihood functions when $L = 3$, $f = 0.8$, and θ has prior distribution $\mathcal{N}(0.82, 0.0009)$. The true likelihood function is generated by Algorithm 6. During the sinusoidal fitting of this function, we set $\Theta = \{\mu-\sigma, \mu-0.8\sigma, \ldots, \mu+0.8\sigma, \mu+\sigma\}$ (i.e., Θ contains 11 uniformly distributed points in $[\mu-\sigma, \mu+\sigma]$) in Eq. (D41). The fitted likelihood function is $\mathbb{P}(d|\theta) = [1+(-1)^d f\,\sin(r\theta+b)]/2$, where $r = -2.81$ and $b = 1.55$. Note that the true and fitted likelihood functions are close for $\theta \in [0.67, 0.97]$.

2. Approximate Bayesian inference with engineered likelihood functions

With the algorithms for tuning the circuit parameters $\vec{x}$ in place, we now briefly describe how to perform Bayesian inference efficiently with the resultant likelihood functions. The idea is similar to that in Sec. IV B for the ancilla-free scheme.

Suppose that θ has prior distribution $\mathcal{N}(\mu,\sigma^2)$, where $\sigma \ll 1/L$, and that the fidelity of the process for generating the ELF is f. We find that the parameters $\vec{x} = (x_1, x_2, \ldots, x_{2L})$ that maximize $\mathcal{I}(\mu;f,\vec{x})$ (or $|\Lambda'(\mu;\vec{x})|$) satisfy the following property: when θ is close to μ, i.e., $\theta \in [\mu - O(\sigma), \mu + O(\sigma)]$, we have

$$\mathbb{P}(d|\theta;f,\vec{x}) \approx \frac{1+(-1)^d f\,\sin(r\theta+b)}{2} \tag{D33}$$

for some $r, b \in \mathbb{R}$. We find the best fitting r and b by solving the least-squares problem

$$(r^*, b^*) = \underset{r,b}{\operatorname{argmin}} \sum_{\theta\in\Theta} |\arcsin[\Lambda(\theta;\vec{x})] - r\theta - b|^2, \tag{D34}$$

where $\Theta = \{\theta_1, \theta_2, \ldots, \theta_k\} \subseteq [\mu - O(\sigma), \mu + O(\sigma)]$. This least-squares problem has the analytical solution

$$\begin{pmatrix} r^* \\ b^* \end{pmatrix} = A^+ z = (A^T A)^{-1} A^T z, \tag{D35}$$

where

$$A = \begin{pmatrix} \theta_1 & 1 \\ \theta_2 & 1 \\ \vdots & \vdots \\ \theta_k & 1 \end{pmatrix}, \qquad z = \begin{pmatrix} \arcsin[\Lambda(\theta_1;\vec{x})] \\ \arcsin[\Lambda(\theta_2;\vec{x})] \\ \vdots \\ \arcsin[\Lambda(\theta_k;\vec{x})] \end{pmatrix}. \tag{D36}$$

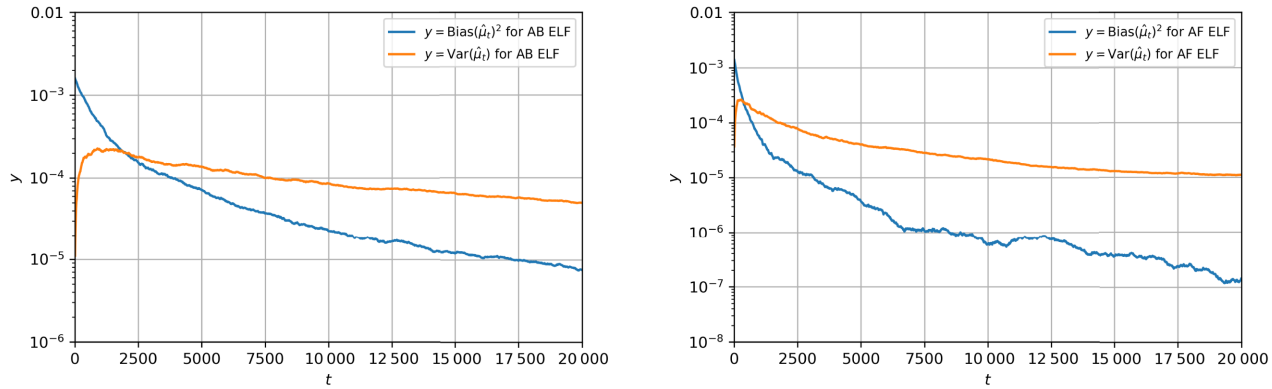


FIG. 25. Comparison of $\mathrm{Bias}(\hat{\mu}_t)^2$ and $\mathrm{Var}(\hat{\mu}_t)$ during the inference process for the AB ELF and AF ELF. Here Π has true value 0.6 and prior distribution $\mathcal{N}(0.64, 0.0009)$, the number of circuit layers is $L = 6$, each layer has fidelity $p = 0.9$, and there is no SPAM error (i.e., $\bar{p} = 1$). Each plot is generated by simulating the inference process 300 times. Note that $\mathrm{Bias}(\hat{\mu}_t)^2 < \mathrm{Var}(\hat{\mu}_t)$ as soon as t becomes sufficiently large. The variance of $\hat{\mu}_t$ increases at the early stage of the algorithm, because we always start with the same $\hat{\mu}_0$ and it takes a certain number of Bayesian updates for $\hat{\mu}_t$ to become sufficiently dispersed.

In Fig. 24 we present an example of the true and fitted likelihood functions.

Once we obtain the optimal r and b, we approximate the posterior mean and variance of θ with those for

$$\mathbb{P}(d|\theta; f) = \frac{1 + (-1)^d f \sin(r\theta + b)}{2}, \tag{D37}$$

which have analytical formulas. Specifically, suppose that θ has prior distribution $\mathcal{N}(\mu_k, \sigma_k^2)$ at round k. Let d_k be the measurement outcome, and let (r_k, b_k) be the best-fitting parameters at this round. Then we approximate the posterior mean and variance of θ by

$$\mu_{k+1} = \mu_k + \frac{(-1)^{d_k} f e^{-r_k^2\sigma_k^2/2} r_k \sigma_k^2 \cos(r_k\mu_k + b_k)}{1 + (-1)^{d_k} f e^{-r_k^2\sigma_k^2/2} \sin(r_k\mu_k + b_k)}, \tag{D38}$$

$$\sigma_{k+1}^2 = \sigma_k^2 \left(1 - \frac{f r_k^2 \sigma_k^2 e^{-r_k^2\sigma_k^2/2}[f e^{-r_k^2\sigma_k^2/2} + (-1)^{d_k} \sin(r_k\mu_k + b_k)]}{[1 + (-1)^{d_k} f e^{-r_k^2\sigma_k^2/2} \sin(r_k\mu_k + b_k)]^2}\right). \tag{D39}$$

After that, we proceed to the next round, setting $\mathcal{N}(\mu_{k+1}, \sigma_{k+1}^2)$ as the prior distribution of θ for that round. The approximation errors incurred by Eqs. (D45) and (D46) are small and have negligible impact on the performance of the whole algorithm for the same reason as in the ancilla-free case.

APPENDIX E: BIAS AND VARIANCE OF THE ELF-BASED ESTIMATOR OF Π

In this appendix, we state the following two facts about the ELF-based estimator $\hat{\mu}_t$ of Π during the inference process [recall that we use the Gaussian distribution $\mathcal{N}(\hat{\mu}_t, \hat{\sigma}_t^2)$ to represent our knowledge of Π at time t, where $\hat{\mu}_t$ and $\hat{\sigma}_t$ are random variables depending on the history of random measurement outcomes up to time t].

(a) In both the ancilla-based and ancilla-free cases, the squared bias of $\hat{\mu}_t$ is smaller than its variance, i.e., $\mathrm{Bias}(\hat{\mu}_t)^2 = (\mathbb{E}[\hat{\mu}_t] - \Pi)^2 < \mathrm{Var}(\hat{\mu}_t)$ for large enough t. See Fig. 25 for examples.

(b) In both the ancilla-based and ancilla-free cases, the perceived variance $\hat{\sigma}_t^2$ of Π is often close to the variance of $\hat{\mu}_t$, i.e., $\hat{\sigma}_t^2 \approx \mathrm{Var}(\hat{\mu}_t)$ with high probability for large enough t. See Fig. 26 for examples.

Combining these facts, we know that, for large enough t, $\mathrm{MSE}(\hat{\mu}_t) = \mathbb{E}[(\hat{\mu}_t - \Pi)^2] \le 2\hat{\sigma}_t^2$ with high probability. Similar statements hold for the ELF-based estimator μ_t of θ during the inference process.

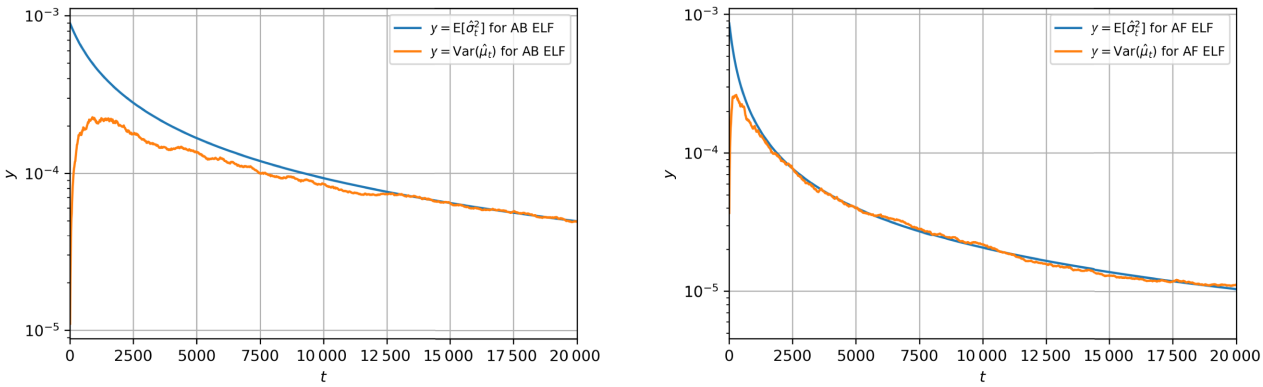


FIG. 26. Comparison of $\mathbb{E}[\hat{\sigma}_t^2]$ and $\mathrm{Var}(\hat{\mu}_t)$ during the inference process for the AB ELF and AF ELF. Here Π has true value 0.6 and prior distribution $\mathcal{N}(0.64, 0.0009)$, the number of circuit layers is $L = 6$, each layer has fidelity $p = 0.9$, and there is no SPAM error (i.e., $\bar{p} = 1$). Each plot is generated by simulating the inference process 300 times.Note that $\mathbb{E}[\hat{\sigma}_t^2] \approx \mathrm{Var}(\hat{\mu}_t)$ once t becomes sufficiently large. Furthermore, we discover that $\hat{\sigma}_t^2$ does not vary much among different runs of the algorithm. Namely, $\hat{\sigma}_t^2 \approx \mathbb{E}[\hat{\sigma}_t^2]$ with high probability. So $\hat{\sigma}_t^2 \approx \mathrm{Var}(\hat{\mu}_t)$ with high probability for large enough t. Note that the variance of $\hat{\mu}_t$ increases at the early stage of the algorithm, because we always start with the same $\hat{\mu}_0$ and it takes a certain number of Bayesian updates for $\hat{\mu}_t$ to become sufficiently dispersed.

APPENDIX F: COMPARISON OF EXACT OPTIMIZATION WITH OPTIMIZATION OF PROXIES

In this appendix, we compare the maximization of the variance reduction factor with the maximization of the proxies used in Sec. IV. We start with motivating the use of these proxies by studying the limiting behavior of $\mathcal{V}(\mu, \sigma; f, \vec{x})$ as $\sigma \to 0$ or as $f \to 0$ or 1.

1. Limiting behavior of the variance reduction factor

Consider the following three limiting situations: (i) when the variance vanishes, i.e., $\sigma^2 \to 0$; (ii) when the fidelity is close to zero, i.e., $f \approx 0$; and (iii) when the fidelity is equal to one, i.e., $f = 1$. We derive expressions for the variance reduction factor in these conditions (see Fig. 27 for a flowchart summarizing the results below). For simplicity, we assume in this section that either (i) $0 \leq f < 1$ or (ii) $f = 1$ and $|\Delta(\mu; \vec{x})| \neq 1$. While the results here are stated in terms of the ancilla-free bias (22), they also hold for the ancilla-based bias (D2) (just replace Δ with Λ).

(a) In case (i), as $\sigma \to 0$, the expected bias (31) behaves as

$$\begin{aligned} b(\mu, \sigma; \vec{x}) &\to \Delta(\mu; \vec{x}), \\ \partial_\mu b(\mu, \sigma; \vec{x}) &\to \Delta'(\mu; \vec{x}), \end{aligned} \tag{F1}$$

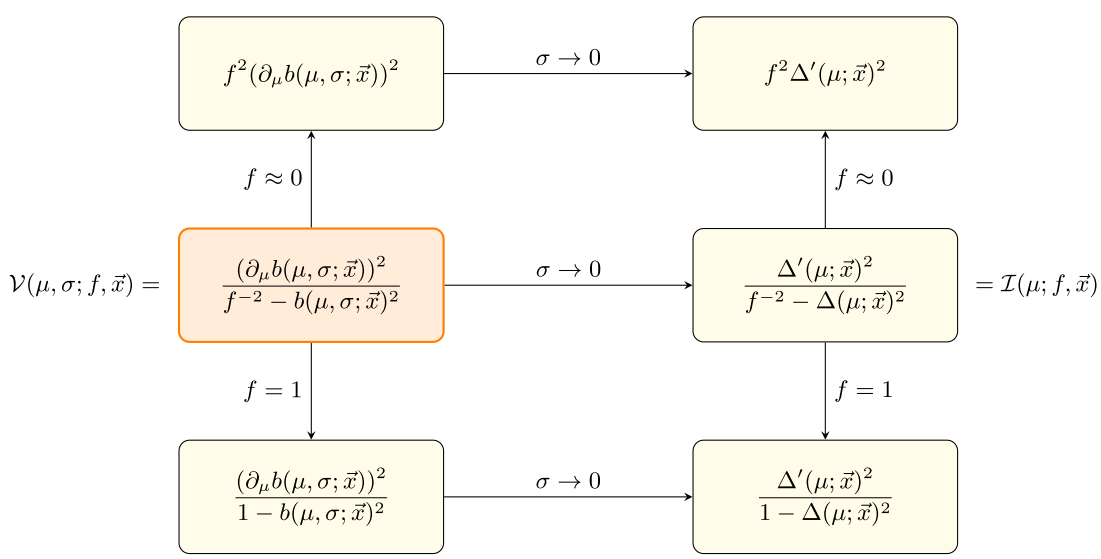


FIG. 27. Flowchart showing the behavior of the expected posterior variance $\mathcal{V}(\mu, \sigma; f, \vec{x})$ (in the orange box) as (i) $\sigma \to 0$, (ii) $f \approx 0$, (iii) $f = 1$. For ease of notation, we use $f = \bar{p}p^L$, where p is the fidelity of each circuit layer and $\bar{p}$ captures the SPAM error.

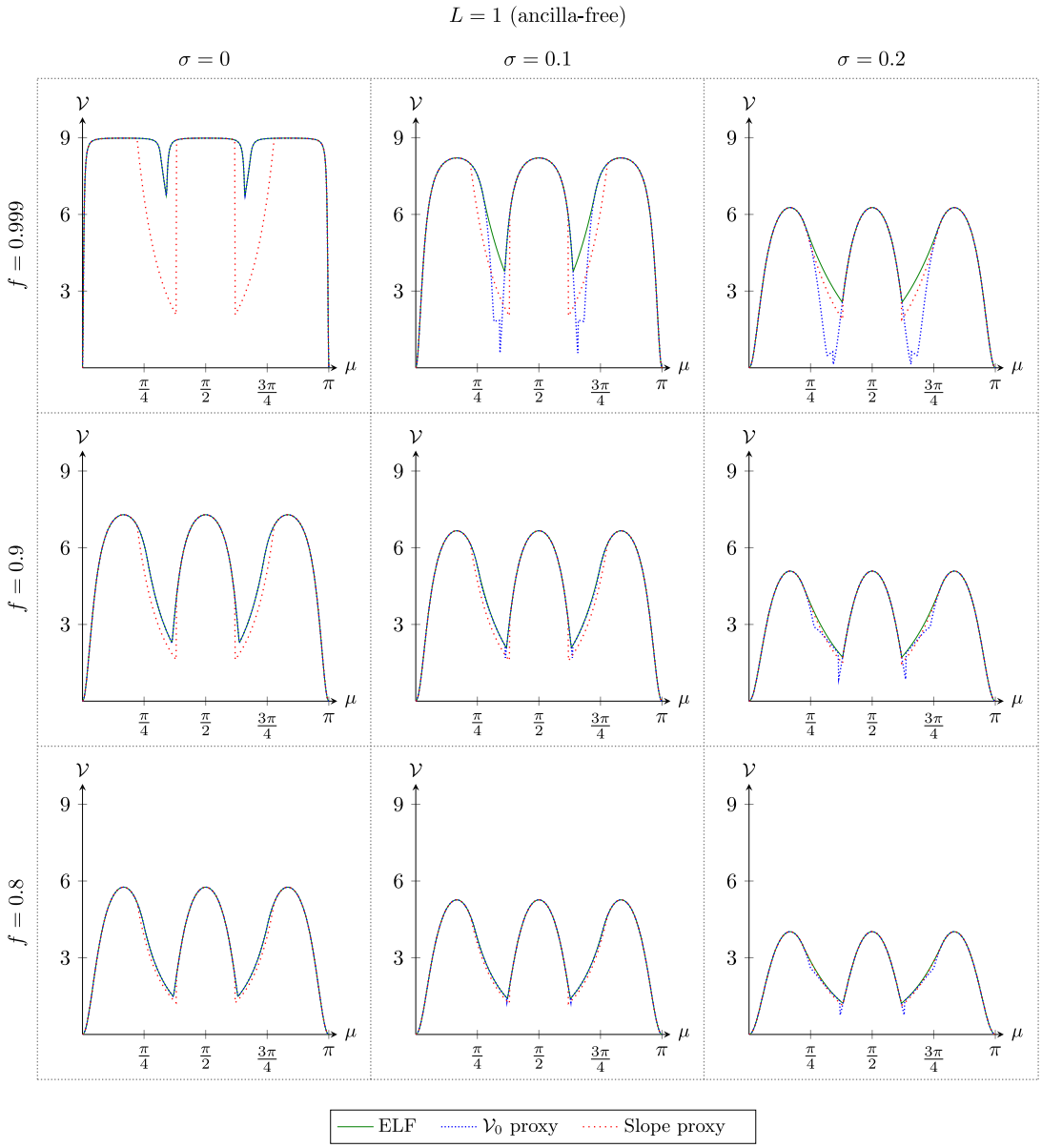


FIG. 28. Plots of the variance reduction factor $\mathcal{V}$ versus the prior mean μ for $L = 1$ for the ancilla-free scheme. The proxies work well when f and σ are not too large (say $f \leq 0.9$ and $\sigma \leq 0.1$). When f and σ are both large (for example, $f = 0.999$ and $\sigma = 0.2$), the proxies fail to be good approximations. The plots are obtained with the `NMaximize` RandomSearch method in Wolfram's MATHEMATICA with 120 search points.

which implies that the limit of the variance reduction factor as $\sigma \to 0$ is

$$\mathcal{V}_0(\mu;f,\vec{x}) := \lim_{\sigma\to 0} \mathcal{V}(\mu,\sigma;f,\vec{x}) = \mathcal{I}(\mu;f,\vec{x}), \tag{F2}$$

where

$$\mathcal{I}(\mu;f,\vec{x}) := \frac{f^2\Delta'(\mu;\vec{x})^2}{1-f^2\Delta(\mu;\vec{x})^2} \tag{F3}$$

is the Fisher information corresponding to the noisy version of the engineered likelihood function given by Eq. (21).

(b) In case (ii), we get

$$\mathcal{V}(\mu,\sigma;f,\vec{x}) \approx f^2[\partial_\mu b(\mu,\sigma;\vec{x})]^2 \quad \text{for } f \approx 0. \tag{F4}$$

(c) In case (iii), we get

$$\mathcal{V}(\mu,\sigma;f,\vec{x}) = \frac{[\partial_\mu b(\mu,\sigma;\vec{x})]^2}{1-b(\mu,\sigma;\vec{x})^2} \quad \text{for } f = 1. \tag{F5}$$

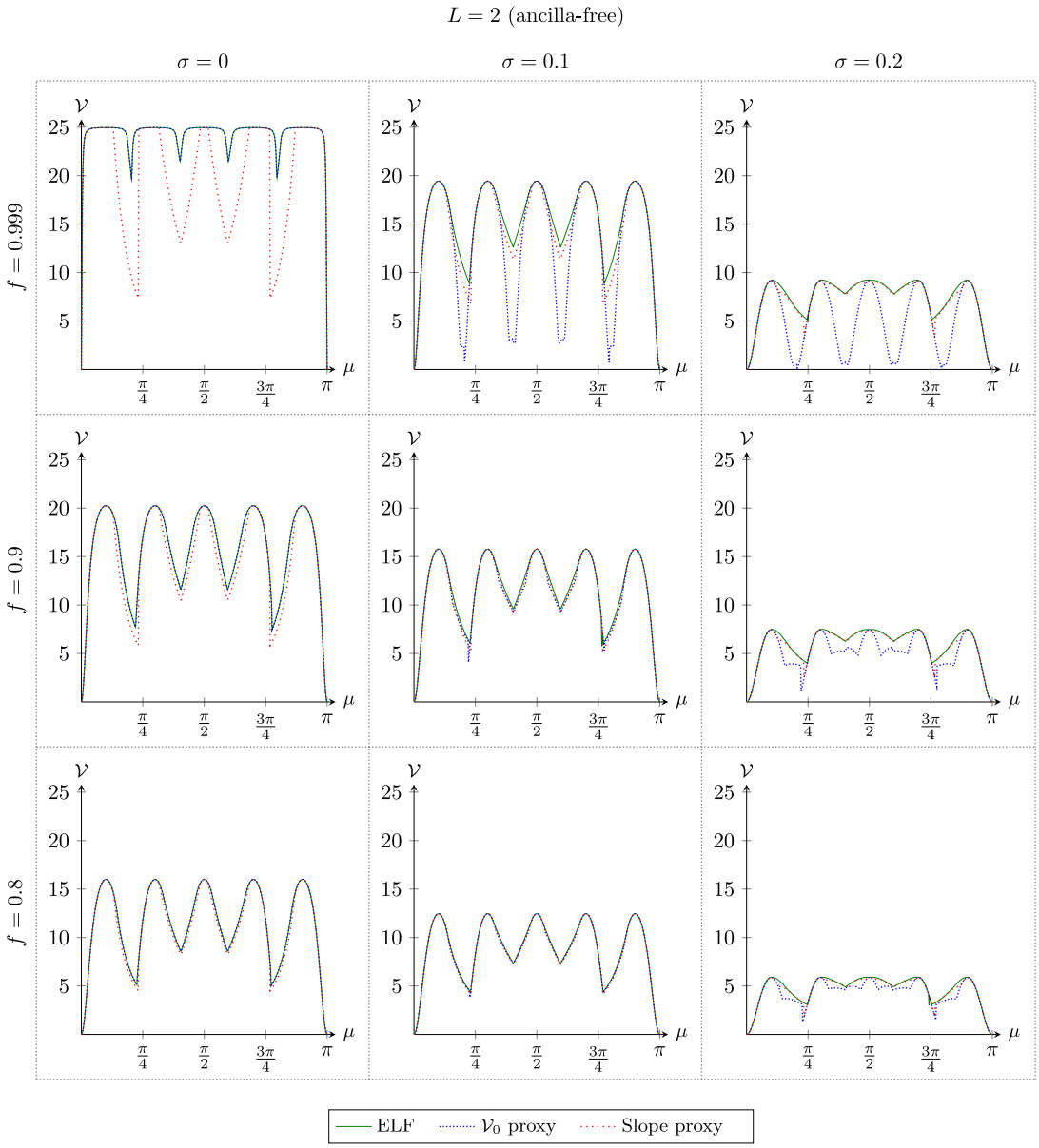


FIG. 29. Plots of the variance reduction factor $\mathcal{V}$ versus the prior mean μ for $L = 2$ for the ancilla-free scheme. The proxies work well when f and σ are not too large (say $f \leq 0.9$ and $\sigma \leq 0.1$). When f and σ are both large (for example, $f = 0.999$ and $\sigma = 0.2$), the proxies fail to be good approximations. The plots are obtained with the `NMaximize` RandomSearch method in Wolfram's MATHEMATICA with 220 search points.

(d) Combining cases (i) and (ii), we get

$$\mathcal{V}(\mu,\sigma;f,\vec{x}) \approx f^2\Delta'(\mu;\vec{x})^2 \quad \text{for } f \approx 0, \sigma \approx 0. \tag{F6}$$

(e) Combining cases (i) and (iii), we get

$$\lim_{\sigma\to 0}\mathcal{V}(\mu,\sigma;f,\vec{x}) = \frac{\Delta'(\mu;\vec{x})^2}{1-\Delta(\mu;\vec{x})^2} = \mathcal{I}(\mu;1,\vec{x}) \quad \text{for } f = 1. \tag{F7}$$

In the next part, we show how these approximations give us good proxies for maximizing the variance reduction factor.

2. Implementing the exact variance reduction factor optimization and comparison with proxies

Consider the optimization problems

$$\texttt{input}: (\mu, f), \text{ where } \mu \in \mathbb{R},\ f \in [0,1],$$

$$\texttt{output}: \operatorname*{argmax}_{\vec{x}\in(-\pi,\pi]^{2L}} \mathcal{I}(\mu;f,\vec{x}) = \operatorname*{argmax}_{\vec{x}\in(-\pi,\pi]^{2L}} \frac{\Delta'(\mu;\vec{x})^2}{f^{-2}-\Delta(\mu;\vec{x})^2} \tag{F8}$$

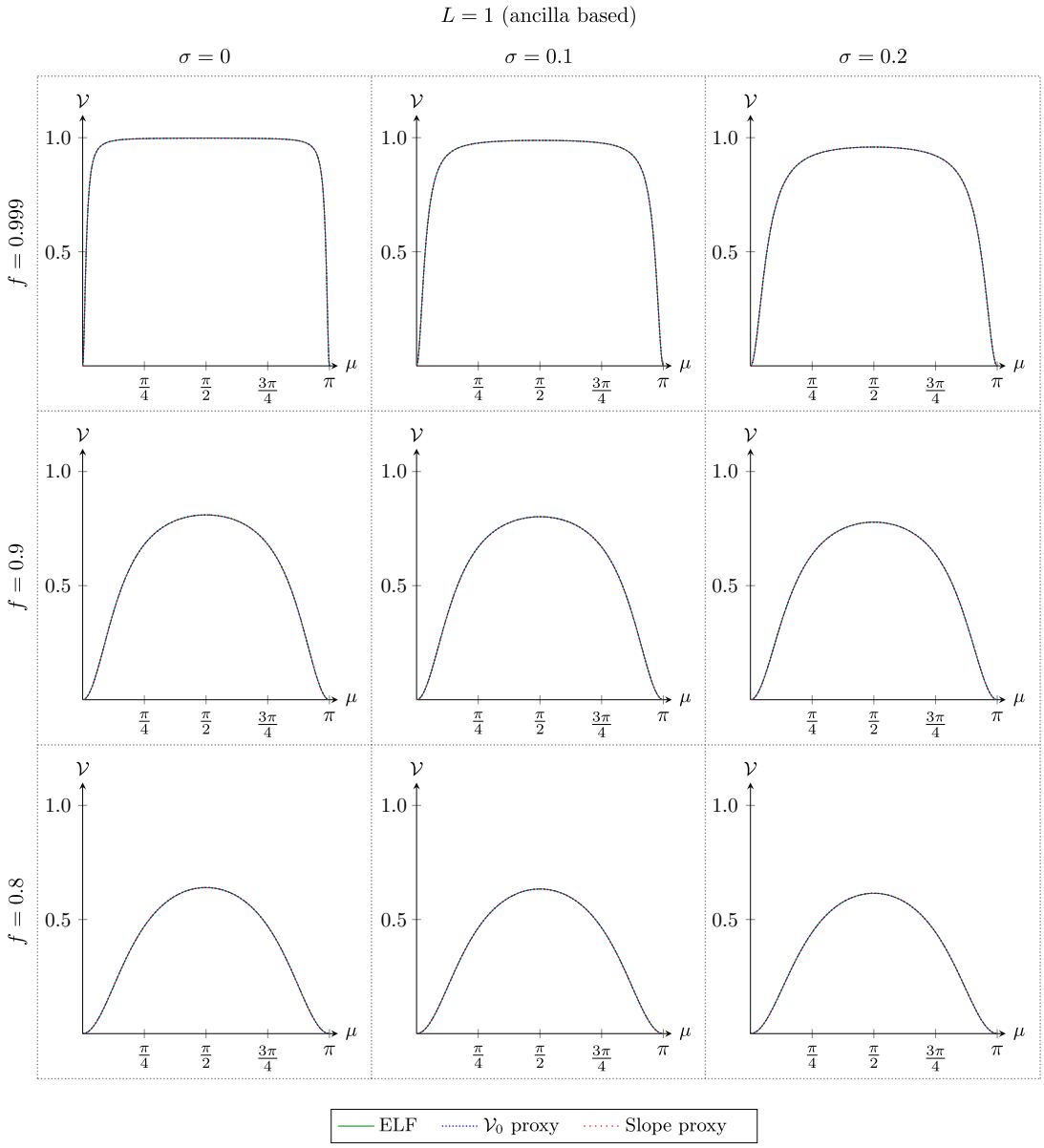


FIG. 30. Plots of the variance reduction factor $\mathcal{V}$ versus the prior mean μ for $L = 1$ for the ancilla-based scheme. The proxies give identical results to the exact optimization of the variance reduction factor. The plots are obtained with the `NMaximize` RandomSearch method in Wolfram's MATHEMATICA with 120 search points.

and

$$\begin{aligned} &\texttt{input}: \mu \in \mathbb{R}, \\ &\texttt{output}: \operatorname*{argmax}_{\vec{x}\in(-\pi,\pi]^{2L}} |\Delta'(\mu;\vec{x})|. \end{aligned} \tag{F9}$$

By Eq. (F2), we expect that a solution to optimization problem (F8) would be a good proxy for maximizing the expected posterior variance when σ is small, i.e., if $\hat{x}_{\mu,f}$ is a solution to optimization problem (F8) on input (μ, f), then we expect that

$$\max_{\vec{x}\in(-\pi,\pi]^{2L}} \mathcal{V}(\mu,\sigma;f,\vec{x}) \approx \frac{\Delta'(\mu;\hat{x}_{\mu,f})^2}{f^{-2} - \Delta(\mu;\hat{x}_{\mu,f})^2} \quad \text{when } \sigma \text{ is small.} \tag{F10}$$

Similarly, by Eq. (F6), we expect that a solution to optimization problem (F9) would be a good proxy for maximizing the expected posterior variance when both f and σ are small, i.e., if $\hat{x}_\mu$ is a solution to optimization problem

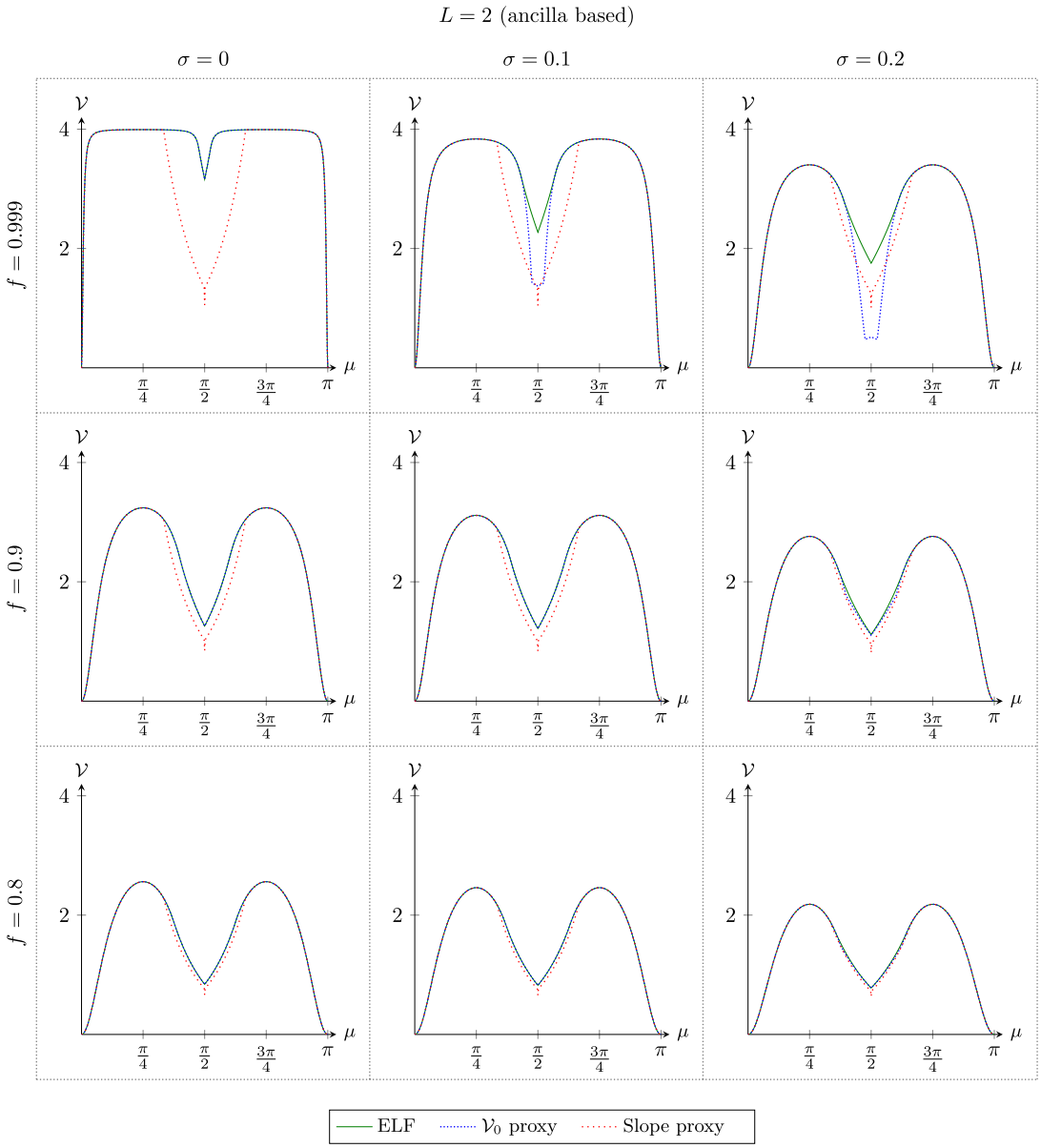


FIG. 31. Plots of the variance reduction factor $\mathcal{V}$ versus the prior mean μ for $L = 2$ for the ancilla-based scheme. The proxies work well when f and σ are not too large (say $f \leq 0.9$ and $\sigma \leq 0.1$). When f and σ are both large (for example, $f = 0.999$ and $\sigma = 0.2$), the proxies fail to be good approximations. The plots are obtained with the `NMaximize` RandomSearch method in Wolfram's MATHEMATICA with 120 search points.

(F9) on input μ, then we expect that

$$\max_{\vec{x}\in(-\pi,\pi]^{2L}} \mathcal{V}(\mu,\sigma;f,\vec{x}) \approx f^2\Delta'(\mu;\hat{x}_\mu)^2 \quad \text{when } \sigma \text{ and } f \text{ are small.} \tag{F11}$$

To investigate the performance of the proxies (F10) and (F11), we numerically maximize the expected posterior variance and the proxies (F8) and (F9) for small problem sizes L. The results of this optimization are presented in Figures 28–31. For $L = 1$, it turns out that the optimization problem (F9) for the ancilla-free bias can be solved analytically. We present this analytical solution in Appendix F 3.

3. Analytical expressions for an $L = 1$ slope proxy

In this appendix, we present an analytical solution to optimization problem (F9) for the ancilla-free bias when $L = 1$. In this case, the bias (22) can be written as the

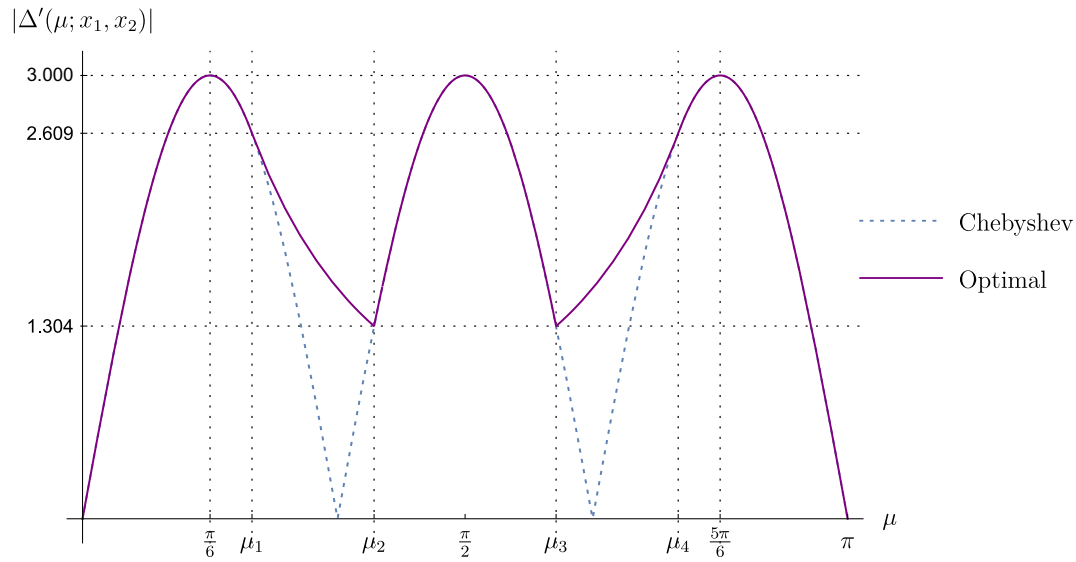


FIG. 32. Graph of the optimal slope, with angles μ_1, μ_2, μ_3, and μ_4 given by Eqs. (F18)–(F21), respectively.

Fourier series

$$\Delta(\theta; x_1, x_2) = \sum_{l=0}^{3} \mu_l(x_1, x_2) \cos(l\theta), \tag{F12}$$

where

$$\begin{aligned}
\mu_0(x_1, x_2) &= 2\cos(x_1)\sin(x_2)\sin(x_1)\cos(x_2),\\
\mu_1(x_1, x_2) &= \cos^2(x_1)\cos^2(x_2) + \cos^2(x_1)\sin^2(x_2)\\
&\quad + \cos^2(x_1)\sin^2(x_2),\\
\mu_2(x_1, x_2) &= -2\cos(x_1)\cos(x_2)\sin(x_1)\sin(x_2),\\
\mu_3(x_1, x_2) &= \sin^2(x_1)\sin^2(x_2).
\end{aligned}$$

Optimization problem (F9) may be stated as

$$\begin{aligned}
&\text{maximize} && |\Delta'(\mu; x_1, x_2)|\\
&\text{subject to} && x_1, x_2 \in (-\pi, \pi],
\end{aligned} \tag{F13}$$

where

$$\begin{aligned}
\Delta'(\mu; x_1, x_2) = &-[\cos^2(x_1)\cos^2(x_2) + \cos^2(x_1)\sin^2(x_2)\\
&+ \cos^2(x_1)\sin^2(x_2)]\sin(\mu)\\
&+ 4\cos(x_1)\cos(x_2)\sin(x_1)\sin(x_2)\sin(2\mu)\\
&- 3\sin^2(x_1)\sin^2(x_2)\sin(3\mu).
\end{aligned} \tag{F14}$$

Solving Eq. (F13) gives the following solution.

Proposition 1. *The maximum magnitude of the slope of the $L = 1$ likelihood function is*

$$\max_{(x_1,x_2)\in[0,\pi]^2} |\Delta'(\mu; x_1, x_2)| = \begin{cases} 3\sin(3\mu), & \mu \in [0, \mu_1] \cup [\mu_4, \pi],\\ \dfrac{4\cos^4(\mu/2)\cot(\mu/2)}{1 + 3\cos(\mu)}, & \mu \in (\mu_1, \mu_2),\\ -3\sin(3\mu), & \mu \in [\mu_2, \mu_3],\\ \dfrac{4\sin^4(\mu/2)\tan(\mu/2)}{1 - 3\cos(\mu)}, & \mu \in (\mu_3, \mu_4), \end{cases} \tag{F15}$$

and an example of a pair of angles that achieves this maximum is

$$(\gamma_1, \gamma_2) \in \operatorname*{argmax}_{(x_1,x_2)\in[0,\pi]^2} |\Delta'(\mu; x_1, x_2)|, \tag{F16}$$

where, for $i = 1, 2$,

$$\gamma_i = \begin{cases} \frac{\pi}{2}, & \mu \in [0,\mu_1] \cup [\mu_2,\mu_3] \cup [\mu_4,\pi], \\ (-1)^i \operatorname{arccot}\left[\sqrt{1-3\cos(\mu)+\sec(\mu)}\right], & \mu \in (\mu_1,\mu_2), \\ \operatorname{arccot}\left[\sqrt{1+3\cos(\mu)-\sec(\mu)}\right], & \mu \in (\mu_3,\mu_4), \end{cases} \tag{F17}$$

where $\mu_1, \mu_2, \mu_3, \mu_4$ are given by

$$\mu_1 = 2\arctan\left[\sqrt{\tfrac{1}{3}(4-\sqrt{13})}\right] \approx 0.6957, \tag{F18}$$

$$\mu_2 = 4\arctan\left[\sqrt{\mathrm{root}_3(p_2)}\right] \approx 1.1971, \tag{F19}$$

$$\mu_3 = 4\arctan\left[\sqrt{\mathrm{root}_3(p_3)}\right] \approx 1.9445, \tag{F20}$$

$$\mu_4 = 2\arctan\left[\sqrt{4+\sqrt{13}}\right] \approx 2.4459, \tag{F21}$$

where p_2 and p_3 are octic polynomials given by

$$p_2(x) = 1 + 72x - 1540x^2 + 8568x^3 - 16506x^4 + 8568x^5 - 1540x^6 + 72x^7 + x^8, \tag{F22}$$

$$p_3(x) = 9 - 264x + 2492x^2 - 9016x^3 + 13302x^4 - 9016x^5 + 2492x^6 - 264x^7 + 9x^8. \tag{F23}$$

The notation $\mathrm{root}_a(p)$ *refers to the* ath *smallest (with starting index* 1*) real root of the polynomial* p.

A plot of Eq. (F15) is shown in Fig. 32.

[1] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien, A variational eigenvalue solver on a photonic quantum processor, Nat. Commun. **5**, 4213 (2014).

[2] Dave Wecker, Matthew B. Hastings, and Matthias Troyer, Progress towards practical quantum variational algorithms, Phys. Rev. A **92**, 042303 (2015).

[3] Jarrod R. McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik, The theory of variational hybrid quantum-classical algorithms, New J. Phys. **18**, 023023 (2016).

[4] Jonathan Romero, Ryan Babbush, Jarrod R. McClean, Cornelius Hempel, Peter J. Love, and Alán Aspuru-Guzik, Strategies for quantum computing molecular energies using the unitary coupled cluster ansatz, Quantum Sci. Technol. **4**, 014008 (2018).

[5] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann, A quantum approximate optimization algorithm, arXiv:1411.4028 (2014).

[6] Stuart Hadfield, Zhihui Wang, Bryan O'Gorman, Eleanor Rieffel, Davide Venturelli, and Rupak Biswas, From the quantum approximate optimization algorithm to a quantum alternating operator ansatz, Algorithms **12**, 34 (2019).

[7] Carlos Bravo-Prieto, Ryan LaRose, Marco Cerezo, Yigit Subasi, Lukasz Cincio, and Patrick J. Coles, Variational quantum linear solver, arXiv:1909.05820 (2019).

[8] Dong An and Lin Lin, Quantum linear system solver based on time-optimal adiabatic quantum computing and quantum approximate optimization algorithm, arXiv:1909.05500 (2019).

[9] Xiaosi Xu, Jinzhao Sun, Suguru Endo, Ying Li, Simon C. Benjamin, and Xiao Yuan, Variational algorithms for linear algebra, arXiv:1909.03898 (2019).

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

[11] Jonathan Romero, Jonathan P. Olson, and Alan Aspuru-Guzik, Quantum autoencoders for efficient compression of quantum data, Quantum Sci. Technol. **2**, 045001 (2017).

[12] Maria Schuld and Nathan Killoran, Quantum Machine Learning in Feature Hilbert Spaces, Phys. Rev. Lett. **122**, 040504 (2019).

[13] D. Zhu, N. M. Linke, M. Benedetti, K. A. Landsman, N. H. Nguyen, C. H. Alderete, A. Perdomo-Ortiz, N. Korda, A. Garfoot, C. Brecque, L. Egan, O. Perdomo, and C. Monroe, Training of quantum circuits on a hybrid quantum computer, Sci. Adv. **5**, eaaw9918 (2019).

[14] William J. Huggins, Jarrod McClean, Nicholas Rubin, Zhang Jiang, Nathan Wiebe, K. Birgitta Whaley, and Ryan Babbush, Efficient and noise resilient measurements for quantum chemistry on near-term quantum computers, arXiv:1907.13117 (2019).

[15] Jerome F. Gonthier, Maxwell D. Radin, Corneliu Buda, Eric J. Doskocil, Clena M. Abuan, and Jhonathan Romero, Identifying challenges towards practical quantum advantage through resource estimation: The measurement roadblock in the variational quantum eigensolver, arXiv:2012.04001 (2020).

[16] Ryan Babbush, Craig Gidney, Dominic W. Berry, Nathan Wiebe, Jarrod McClean, Alexandru Paler, Austin Fowler, and Hartmut Neven, Encoding Electronic Spectra in Quantum Circuits with Linear T Complexity, Phys. Rev. X **8**, 041015 (2018).

[17] Ashley Montanaro, Quantum speedup of monte carlo methods, Proc. R. Soc. A: Math., Phys. Eng. Sci. 471, 20150301 (2015).

[18] Aram W. Harrow, Avinatan Hassidim, and Seth Lloyd, Quantum Algorithm for Linear Systems of Equations, Phys. Rev. Lett. **103**, 150502 (2009).

[19] B. D. Clader, B. C. Jacobs, and C. R. Sprouse, Preconditioned Quantum Linear System Algorithm, Phys. Rev. Lett. **110**, 250504 (2013).

[20] Daochen Wang, Oscar Higgott, and Stephen Brierley, Accelerated Variational Quantum Eigensolver, Phys. Rev. Lett. **122**, 140504 (2019).

[21] Emanuel Knill, Gerardo Ortiz, and Rolando D. Somma, Optimal quantum measurements of expectation values of observables, Phys. Rev. A **75**, 012328 (2007).

[22] Alexandr Sergeevich, Anushya Chandran, Joshua Combes, Stephen D. Bartlett, and Howard M. Wiseman, Characterization of a qubit Hamiltonian using adaptive measurements in a fixed basis, Phys. Rev. A **84**, 052315 (2011).

[23] Christopher Ferrie, Christopher E. Granade, and D. G. Cory, How to best sample a periodic probability distribution, or on the accuracy of Hamiltonian finding strategies, Quantum Inf. Process. **12**, 611 (2012).

[24] Krysta M. Svore, Matthew B. Hastings, and Michael Freedman, Faster phase estimation, Quantum Inf. Comput. **14**, 306 (2013).

[25] Nathan Wiebe and Chris Granade, Efficient Bayesian Phase Estimation, Phys. Rev. Lett. **117**, 010503 (2016).

[26] Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp, Quantum amplitude amplification and estimation, Contemp. Math. **305**, 53 (2002).

[27] V. Giovannetti, Quantum-enhanced measurements: Beating the standard quantum limit, Science **306**, 1330 (2004).

[28] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone, Advances in quantum metrology, Nat. Photonics **5**, 222 (2011).

[29] Theodore J. Yoder, Guang Hao Low, and Isaac L. Chuang, Fixed-Point Quantum Search with an Optimal Number of Queries, Phys. Rev. Lett. **113**, 210501 (2014).

[30] Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang, Methodology of Resonant Equiangular Composite Quantum Gates, Phys. Rev. X **6**, 041067 (2016).

[31] Guang Hao Low and Isaac L. Chuang, Hamiltonian simulation by qubitization, Quantum **3**, 163 (2019).

[32] Guang Hao Low and Isaac L. Chuang, Optimal Hamiltonian Simulation by Quantum Signal Processing, Phys. Rev. Lett. **118**, 010501 (2017).

[33] Thomas E. O'Brien, Brian Tarasinski, and Barbara M. Terhal, Quantum phase estimation of multiple eigenvalues for small-scale (noisy) experiments, New J. Phys. **21**, 023022 (2019).

[34] A. Yu. Kitaev, Quantum measurements and the Abelian stabilizer problem, quant-ph/9511026 (1995).

[35] Zhihui Wang, Stuart Hadfield, Zhang Jiang, and Eleanor G. Rieffel, Quantum approximate optimization algorithm for maxCut: A fermionic view, Phys. Rev. A **97**, 022304 (2018).

[36] Z. Ji, G. Wang, R. Duan, Y. Feng, and M. Ying, Parameter estimation of quantum channels, IEEE Trans. Inf. Theory **54**, 5172 (2008).

[37] Ilia Zintchenko and Nathan Wiebe, Randomized gap and amplitude estimation, Phys. Rev. A **93**, 062306 (2016).

[38] Yohichi Suzuki, Shumpei Uno, Rudy Raymond, Tomoki Tanaka, Tamiya Onodera, and Naoki Yamamoto, Amplitude estimation without phase estimation, Quantum Inf. Process. **19**, 75 (2020).

[39] Joel J. Wallman and Joseph Emerson, Noise tailoring for scalable quantum computation via randomized compiling, Phys. Rev. A **94**, 052325 (2016).

[40] Gilles Brassard, Peter Høyer, and Alain Tapp, in *International Colloquium on Automata, Languages, and Programming* (Springer, 1998), p. 820.

[41] Vittorio Giovannetti, Seth Lloyd, and Lorenzo Maccone, Quantum Metrology, Phys. Rev. Lett. **96**, 010401 (2006).

[42] Jonas M. Kübler, Andrew Arrasmith, Lukasz Cincio, and Patrick J. Coles, An adaptive optimizer for measurement-frugal variational algorithms, Quantum **4**, 263 (2020).

[43] Ryan Sweke, Frederik Wilde, Johannes Meyer, Maria Schuld, Paul K. Faehrmann, Barthélémy Meynard-Piganeau, and Jens Eisert, Stochastic gradient descent for hybrid quantum-classical optimization, Quantum **4**, 314 (2020).

[44] Andrew Arrasmith, Lukasz Cincio, Rolando D. Somma, and Patrick J. Coles, Operator sampling for shot-frugal optimization in variational algorithms, arXiv:2004.06252 (2020).

[45] Kevin J. Sung, Jiahao Yao, Matthew P. Harrigan, Nicholas C. Rubin, Zhang Jiang, Lin Lin, Ryan Babbush, and Jarrod R. McClean, Using models to improve optimizers for variational quantum algorithms, Quantum Sci. Technol. **5**, 044008 (2020).

[46] Abhinav Kandala, Antonio Mezzacapo, Kristan Temme, Maika Takita, Markus Brink, Jerry M. Chow, and Jay M. Gambetta, Hardware-efficient variational quantum eigensolver for small molecules and quantum magnets, Nature **549**, 242 (2017).

[47] We call this scheme "ancilla-free" since it does not involve any ancilla qubits. In Appendix D, we consider a different scheme named the "ancilla-based" scheme that involves one ancilla qubit.

[48] To ensure that S is two dimensional, we assume that $\Pi \neq \pm 1$, i.e., $\theta \neq 0$ or π.

[49] In practice, the establishment of the noise model requires a procedure for *calibrating the likelihood function* for the specific device being used. With respect to Bayesian inference, the parameters of this model are known as *nuisance parameters* [70,71]; the target parameter does not depend directly on them, but they determine how the data relate to the target parameter and, hence, should be incorporated into the inference process. We explore likelihood function calibration in future work. For the remainder of this article, we assume that the noise model has been calibrated to sufficient precision so as to render the effect of model error negligible. In Sec. VII we touch upon the relationship between model error and estimation error.

[50] Eric G. Brown, Oktay Goktas, and W. K. Tham, Quantum amplitude estimation in the presence of noise, arXiv:2006.14145 (2020).

[51] Shumpei Uno, Yohichi Suzuki, Keigo Hisanaga, Rudy Raymond, Tomoki Tanaka, Tamiya Onodera, and Naoki Yamamoto, Modified Grover operator for amplitude estimation, arXiv:2010.11656 (2020).

[52] Tomoki Tanaka, Yohichi Suzuki, Shumpei Uno, Rudy Raymond, Tamiya Onodera, and Naoki Yamamoto, Amplitude estimation via maximum likelihood on noisy quantum computer, arXiv:2006.16223 (2020).

[53] Google Quantum AI, Accurately computing electronic properties of materials using eigenenergies, arXiv:2012.00921 (2020).

[54] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, *et al.*, Quantum supremacy using a programmable superconducting processor, Nature **574**, 505 (2019).

[55] Dax Enshan Koh, Guoming Wang, Peter D. Johnson, and Yudong Cao, A framework for engineering quantum likelihood functions for expectation estimation, arXiv:2006.09349 (2020).

[56] Christopher E. Granade, Christopher Ferrie, Nathan Wiebe, and David G. Cory, Robust online Hamiltonian learning, New J. Phys. **14**, 103013 (2012).

[57] Although we can compute the mean and variance of the posterior distribution $p(\theta|d;f,\vec{x})$ directly by definition, this approach is time consuming, as it involves numerical integration. Instead, we accelerate this process by taking advantage of certain properties of engineered likelihood functions. See Sec. IV B for more details.

[58] In the simplest case, $\delta(t) = \delta$ is constant. But in order to achieve better performance, we might want $\delta(t) \to 0$ as $t \to \infty$.

[59] Since Algorithms 1 and 2 only output a local maximum point of $\mathcal{I}(\theta;f,\vec{x})$ for given θ and f, we need to run them multiple times with random initial points to find a global maximum point of the same function. We find that this does not require many trials. For example, for $L \leq 6$, a global-optimal solution can be found in ten trials with high probability. Similar statements hold for the other algorithms for parameter tuning in this paper.

[60] Though not shown in Fig. 13, the $\hat{R}_0$ factors of the AB ELF and AF ELF actually diverge to $+\infty$ as $\Pi \to \pm 1$, and this is true for any $L \in \mathbb{Z}^+$.

[61] More precisely, the slopes of the AB ELF and AF ELF scale linearly in the number L of circuit layers.

[62] For the Chebyshev likelihood functions, we can express the variance reduction factor as $\mathcal{V}[\mu,\sigma;f,(\pi/2)^{\oplus 2L}] = m_L^2/[1 + (f^{-2}e^{m_L^2\sigma^2} - 1)\csc^2(m_L\mu)]$ whenever $\sin(m_L\mu) \neq 0$. Then, $\csc^2(m_L\mu) \geq 1$ implies that $\mathcal{V}[\mu,\sigma;f,(\pi/2)^{\oplus 2L}] \leq f^2 m_L^2 e^{-m_L^2\sigma^2}$. Here $(\pi/2)^{\oplus 2L} = (\pi/2, \pi/2, \ldots, \pi/2) \in \mathbb{R}^{2L}$.

[63] We search over a uniform grid of 50 000 values of θ, L values from $L^*/3$ to $3L^*$, where L^* is to the optimized value used to arrive at Eq. (59), and σ and λ ranging over $10^{-1}, 10^{-2}, \ldots, 10^{-5}$. For each (σ,λ) pair, we find the θ for which the maximum inverse variance rate (over L) is a minimum. For all (σ,λ) pairs checked, this worst-case rate is always between 0.4 and 0.5, with the smallest value found being $R = 0.417\,003\,68 \geq (e-1)^2/e^2$.

[64] Vladyslav Verteletskyi, Tzu-Ching Yen, and Artur F. Izmaylov, Measurement optimization in the variational quantum eigensolver using a minimum clique cover, J. Chem. Phys. **152**, 124114 (2020).

[65] Artur F. Izmaylov, Tzu-Ching Yen, Robert A. Lang, and Vladyslav Verteletskyi, Unitary partitioning approach to the measurement problem in the variational quantum eigensolver method, J. Chem. Theory Comput. **16**, 190 (2020).

[66] Ophelia Crawford, Barnaby van Straaten, Daochen Wang, Thomas Parks, Earl Campbell, and Stephen Brierley, Efficient quantum measurement of Pauli operators in the presence of finite sampling error, Quantum **5**, 385 (2021).

[67] Andrew Zhao, Andrew Tranter, William M. Kirby, Shu Fay Ung, Akimasa Miyake, and Peter J. Love, Measurement reduction in variational quantum algorithms, Phys. Rev. A **101**, 062322 (2020).

[68] Ian Hincks, Joel J. Wallman, Chris Ferrie, Chris Granade, and David G. Cory, Bayesian inference for randomized benchmarking protocols, arXiv:1802.00401 (2018).

[69] J. M. Gambetta, A. D. Córcoles, S. T. Merkel, B. R. Johnson, J. A. Smolin, J. M. Chow, C. A. Ryan, C. Rigetti, S. Poletto, T. A. Ohki, M. B. Ketchen, and M. Steffen, Characterization of Addressability by Simultaneous Randomized Benchmarking, Phys. Rev. Lett. **109**, 240504 (2012).

[70] Edwin T. Jaynes, *Probability Theory: The Logic of Science* (Cambridge University Press, Cambridge, 2003).

[71] Richard Royall, On the probability of observing misleading statistical evidence, J. Am. Stat. Assoc. **95**, 760 (2000).