Low-depth amplitude estimation on a trapped-ion quantum computer

Tudor Giurgica-Tiron,[1,2,*] Sonika Johri,[3,†] Iordanis Kerenidis,[4,5,‡] Jason Nguyen,[3,§] Neal Pisenti,[3,‖] Anupam Prakash,[4,¶] Ksenia Sosnova,[3,**] Ken Wright,[3,††] and William Zeng[1]

[1]*Goldman, Sachs & Co., New York, New York 10282, USA*
[2]*Stanford University, Palo Alto, California 94305, USA*
[3]*IonQ Inc, 4505 Campus Dr, College Park, Maryland 20740, USA*
[4]*QC Ware, Palo Alto 94306, USA and Paris 75003, France*
[5]*IRIF, CNRS–University of Paris 75013, France*

(Received 22 November 2021; accepted 30 May 2022; published 13 July 2022)

Amplitude estimation (AE) is a fundamental quantum algorithmic primitive that enables quantum computers to achieve quadratic speedups for a large class of statistical estimation problems, including Monte Carlo methods. Recent works have succeeded in somewhat reducing the necessary resources for AE by trading off some of the speedup for lower depth circuits, but high quality qubits are still needed for demonstrating such algorithms. Here, we report the results of an experimental demonstration of AE on a state-of-the-art trapped ion quantum computer. AE was used to estimate the inner product of randomly chosen four-dimensional unit vectors, and the low-depth AE algorithms were based on the maximum likelihood estimation (MLE) and the Chinese remainder theorem (CRT) techniques. Significant improvements in accuracy were observed for the MLE based approach when deeper quantum circuits were taken into account, including circuits with more than 90 two-qubit gates and depth 60, achieving a mean additive estimation error on the order of 10^{-2}. The CRT based approach was found to provided accurate estimates for many of the data points but was less robust against noise on average. Last, we analyze two more AE algorithms that take into account the specifics of the hardware noise to further improve the results.

DOI: 10.1103/PhysRevResearch.4.033034

I. INTRODUCTION

Amplitude estimation (AE) [1] is a fundamental quantum algorithmic primitive that enables quantum computers to achieve quadratic speedups for a large class of statistical estimation problems. Amplitude estimation is of particular interest for quantum algorithms in computational finance as it underlies the quantum speedup for Monte Carlo methods [3].

In a typical quantum Monte Carlo setting, the quantum algorithm has access to an evaluation oracle for $f : X \to \mathbb{R}$ and the goal is to estimate the mean value of f to additive error ϵ. Montanaro [3] gave an algorithm to estimate the mean for arbitrary distributions on X in the setting where the variance of f is upper bounded by σ. It requires $O(\sigma/\epsilon)$ calls to the oracle for f, achieving a quadratic speedup over classical algorithms that require $O(\sigma^2/\epsilon^2)$ calls. Quantum Monte Carlo methods have also been developed for settings where the distribution on X is uniform [4–6] and for settings with known bounds on the mean [7,8].

The main drawback of quantum Monte Carlo methods from the perspective of near-term hardware implementations is that the amplitude estimation algorithm requires the evaluation oracle for f to be invoked sequentially $O(\sigma/\epsilon)$ times. The depth of the quantum circuit for Monte Carlo is therefore $O(\sigma D(f)/\epsilon)$, where $D(f)$ is the depth of the evaluation oracle. Reducing the resource requirements for quantum Monte Carlo methods to bring them closer to implementation on noisy intermediate scale quantum (NISQ) architectures has been the focus of a number of recent works.

Results on reducing resource requirements for amplitude estimation were given by Suzuki *et al.* [9] who used maximum likelihood estimation (MLE) to eliminate the quantum Fourier transform (QFT) step in the AE algorithm. Aaronson and Rall [10] gave a QFT-free algorithm with provable guarantees, and further work by Grinko *et al.* [11] on iterative AE reduced the constant overheads for the QFT-free AE. Eliminating the QFT does not significantly lower the depth of the AE algorithm as the QFT circuit has depth $O(\log_2 \log_2(1/\epsilon))$ but it is an important step toward making AE nearer term.

A different line of research was initiated in Ref. [12] where two AE algorithms that trade off the maximum depth for the AE algorithm against the total number of oracle calls were proposed. The first algorithm used MLE with power law schedules while the second algorithm has provable guarantees and is based on boosting lower precision AE estimates using the Chinese remainder theorem (CRT). Supporting results from the lower bound literature indicate the optimality of the depth versus oracle call tradeoff [13–15].

*tgt@stanford.edu
†johri@ionq.co
‡iordanis.kerenidis@qcware.com
§nguyen@ionq.co
‖pisenti@ionq.co
¶anupam.prakash@qcware.com
**sosnova@ionq.co
††wright@ionq.co

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.

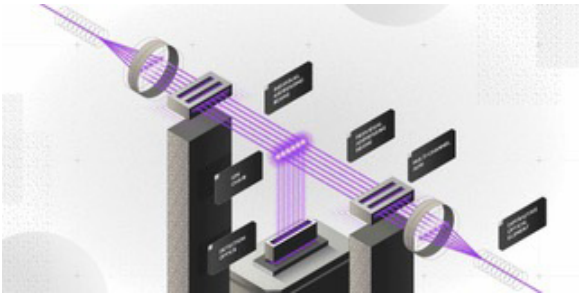


FIG. 1. Schematic of next-generation IonQ hardware, where two fans of individually addressable beams illuminate a chain of individually imaged ions.

Quantum Monte Carlo methods for computational finance have been explored in recent works on pricing simple options and for credit-risk calculations [17–21]. Some of these works have an experimental component, for example [19,21], reporting proof-of-concept results of running their models on three and four qubit quantum devices. Experimental results are limited both by the number of qubits available and the circuit depth of the AE procedure, so a quantum advantage over classical Monte Carlo methods has not been established so far due to hardware limitations. The costs for implementing the evaluation oracle for these applications are significant—some works have focused on resource estimation [21,22] while others have tried to reduce the cost of implementing the oracles [23–26].

Experiments on near-term AE using the iterative AE [11] and MLE [9] were recently carried out on IBM quantum computers [27]. AE was used for Monte Carlo estimation of the integral $\int_0^{\pi/2} \sin^2(x)dx$. The evaluation oracle circuit consisted of three qubits and two two-qubit gates, the maximum oracle depth that was demonstrated was 2, and the hardware noise led to significant loss of performance for the AE algorithm compared to simulator runs. The additive error for estimating the integral was also fairly large compared to the magnitude of the integral, thus attesting to the limitations of the hardware.

In this paper, we report the results of experiments demonstrating low-depth AE algorithms [9,12] on IonQ's state-of-the-art trapped ion quantum computer. We compare AE performance against sampling directly from the evaluation oracle. The results show that AE has a significantly less estimation error for the same number of samples. On a noiseless quantum computer, the advantage from the AE approach would be quadratic in the depth of the quantum algorithm run. On the noisy hardware used for the experiments, we expect to see the AE advantage increase to some maximum depth after which noise from longer circuits overwhelms the quantum scaling advantage. This demonstration on IonQ hardware (Fig. 2) is the main result of this paper.

Note that this is not a demonstration of the scaling advantage of AE. To show practical quantum advantage, the quantum AE approach would need to best classical sampling with classical processors that today have a much higher clock speed and negligible error rates compared to QPUs. Significant improvements in QPU performance will be needed before such a demonstration.

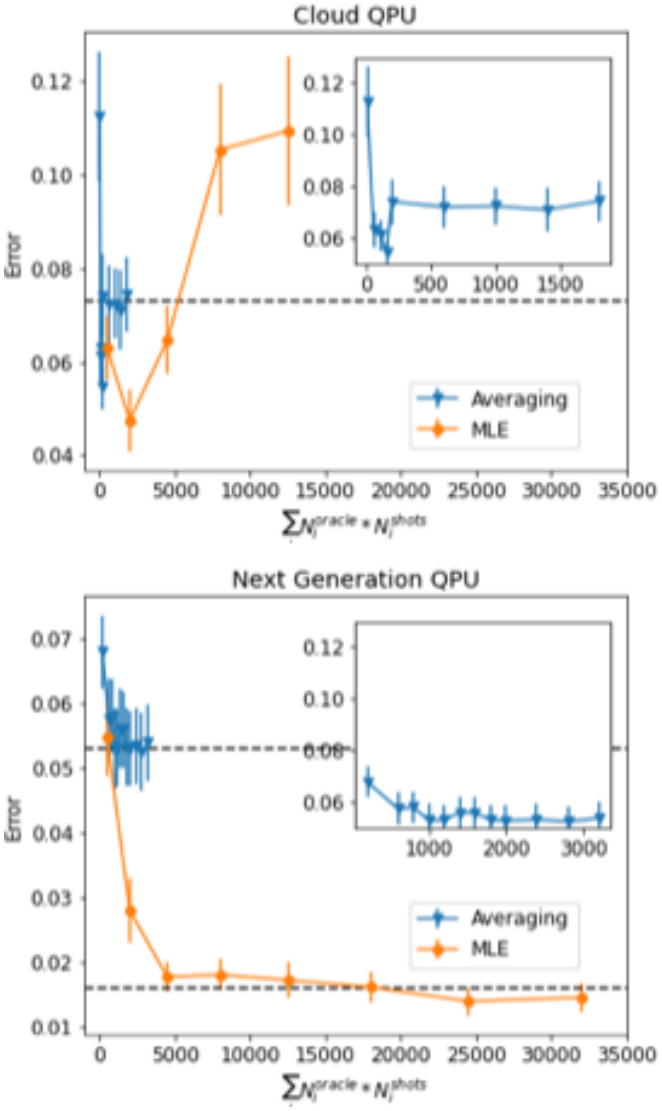


FIG. 2. Comparison of MLE-based AE algorithm versus simple averaging over samples from the evaluation oracle, namely, the inner product estimation circuit. The mean error and error bars for MLE-based AE are shown with orange circles and for simple averaging over many shots with blue triangles. The top figure corresponds to the cloud-based QPU and the bottom figure to the next-generation QPU with $N_{\text{shot}} = 500$. The black lines are a guide to the eye. The x axis denotes the total oracle calls. For simple averaging, $N^{\text{oracle}} = 1$ for all points. The insets zoom in on the error from simple averaging showing its convergence to a constant value as the number of shots increases. For convenience of averaging over several circuits at each data point, the number of shots used is the total number of shots taken for each experiment even though a symmetry-based postprocessing technique was utilized which discards some of the shots.

For our experiments, AE is used to estimate the inner product between randomly chosen four-dimensional vectors. The inner product estimation circuits were developed using quantum data loaders, which are logarithmic depth quantum circuits for preparing vector states [28]. The evaluation oracle for inner product estimation compiles down to a circuit on four qubits with a total depth 6, it uses eight two-qubit gates that are directly applied on hardware.

The high gate fidelity for the trapped ion quantum computer allow us to report experimental results for considerably deep AE circuits, where the oracle circuit is repeated sequentially up to 15 times for a total depth of more than 60 and a total number of two-qubit gates of more than 90. The level of noise in the circuits has some interesting consequences: First, samples beyond a certain depth are ineffective at further improving the accuracy since they become too noisy and, second, even for sampling from the evaluation oracle directly, a certain approximation error remains as we increase the number of samples.

Last, we analyze two more algorithms that take advantage of knowledge of the hardware noise. This information may come from the calibration of the hardware or by running the algorithms on some training data beforehand. The first algorithm is a hybrid algorithm that chooses between MLE estimate for small depths and a CRT-based estimate for two large moduli, and whose test condition is tuned experimentally. The second algorithm is an MLE algorithm with a power-law schedule [12], where the exponent is tuned experimentally.

II. EXPERIMENTAL SETTING FOR AMPLITUDE ESTIMATION

The general setting for AE is one where the quantum algorithm is given access to a quantum circuit $\mathcal{A}$ known as the evaluation oracle such that $\mathcal{A}\,|0^k\rangle = \sin(\theta)\,|1, x\rangle + \cos(\theta)\,|0, x'\rangle$, where $|x\rangle$, $|x'\rangle$ are arbitrary states on $(k-1)$ qubits. The algorithm's goal is to estimate the angle θ within an additive factor ϵ.

The iteration circuit consists of the following four unitaries: $\mathcal{A}S_0\mathcal{A}^\dagger S_\chi$, where the reflection S_χ puts a minus phase in front of the states $|1\rangle\,|x\rangle$ and $S_0 = 2\,|0^n\rangle\,\langle 0^n| - I$ is a reflection around the all-zero state.

The low-depth quantum AE algorithm executes a number of different circuits U^t for different depths t, where we define these circuits as

$$U^t = (\mathcal{A}S_0\mathcal{A}^\dagger S_\chi)^t\mathcal{A}. \tag{1}$$

In other words, the depth 0 circuit is a simple application of the evaluation oracle $\mathcal{A}$, while for a depth t we add t applications of the iteration circuit to an initial application of the oracle. Thus, for depth t the number of oracle calls is $(2t+1)$ and the state at the end of the circuit U^t is

$$U^t\,|0\rangle = \sin((2t+1)\theta)\,|1, x\rangle + \cos((2t+1)\theta)\,|0, x'\rangle\,.$$

The evaluation oracle used for our experiments is a simple circuit for estimating the square inner product between two random unit vectors that are encoded with a unary amplitude encoding method. Such circuits have been used before for estimating the Euclidean distance between data points in Ref. [28] and are specified in more detail in Appendix A. In particular, the hardware implementation consisted of circuits up to $t = 7$, with a total of 92 two-qubit gates and a circuit depth of 62. These figures can be read from the circuit diagram and descriptions provided in Appendix A.

We remark here that one could use many different evaluation oracle circuits to test the AE algorithms. The inner product estimation circuits used in the experiments as the evaluation oracle are known to be optimal both with respect to the number of gates and circuit depth and they have a number of different applications in estimating the Euclidean distance between data points in similarity-based classification algorithms [28] or for assisting in the training of neural networks [29–31]. They also provide a great benchmark for the AE algorithms, since one can easily increase the complexity of the oracle by increasing the dimension of the unit vectors.

The experimental demonstration was performed on the newest generation IonQ quantum processing unit (QPU) illustrated in Fig. 1. This system, as in previous IonQ QPUs [32], utilizes trapped Ytterbium ions where two states in the ground hyperfine manifold are used as qubit states. These states are manipulated by illuminating individual ions with pulses of 355 nm light that drive Raman transitions between the ground states defining the qubit. By configuring these pulses, arbitrary single qubit gates and Molmer-Sorenson type two-qubit gates can both be realized. This QPU features not only an order of magnitude better performance in terms of fidelity but also is considerably more robust compared to its current QPU on the cloud. This allows for deep circuits with many shots to be run over a very reasonable period of time. This increased data collection rate has made it possible to run experiments of this nature.

III. EXPERIMENTAL RESULTS FOR MLE-BASED AMPLITUDE ESTIMATION

The MLE-based AE algorithms combine information from samples taken from quantum circuits of the form U^t at different depths t in a statistically efficient manner. The MLE-based algorithms used for the experiments use linear schedules [12,16] and are described in more detail in Appendix B.

With the high fidelities of the state-of-the-art IonQ quantum computer, we managed to run the AE algorithm with depths ranging from 0–7, where the circuit U^7 has 92 two-qubit gates and a depth of 62, and keeping the additive error for the estimate very small, in fact, below 0.02. The schedules used 500 shots per depth.

In addition, efficient error mitigation was performed in the measurement results as in Ref. [28] by taking advantage of the unary encoding.

The goal for the experiments was to establish an advantage for MLE-based AE implementation in a noisy setting, that is, to show that this family of algorithms can achieve more accurate estimates compared to the baseline of simply sampling from the evaluation oracle (the circuits corresponding to U^0), which is akin to classical sampling in the noiseless setting. In the noisy setting, the mean accuracy for sampling from the evaluation oracle saturates at some threshold determined by the noise rate, and thus an AE algorithm in the noisy setting achieves an advantage if it can obtain more accurate estimates by sampling from higher depth circuits.

The results for the MLE-based AE algorithm are presented in Fig. 2. The plot compares the results of estimating the angle of the evaluation oracle when we only sample from the evaluation oracle for different numbers of shots (blue line) versus when we perform an MLE-based AE algorithm using a fixed number of shots for different circuit depths. The results are averaged over a set of 50 randomly chosen input vectors,

500 shots per circuit, and the mean and standard deviations are shown—for the top plot, the IonQ cloud-based QPU, while for the bottom plot we used the state-of-the-art next-generation IonQ QPU.

The blue lines correspond to the classical case where in a noiseless setting $O(1/\epsilon^2)$ samples are needed to achieve accuracy ϵ. Since each point is averaged over many circuits with the same structure and a different angle, depolarizing noise can be used as an effective error model. Then, given the fact that there effectively exists a level of depolarizing noise even at the level of single oracle circuits, the error does not converge to zero but saturates at a higher threshold, which in the experiments (and after the postprocessing technique described in step 6 of Algorithm B.1) is about 0.073 for the cloud QPU and 0.053 for the next-generation QPU.

The orange lines correspond to the MLE-based AE algorithm which uses samples from different depths and combines the estimates via an MLE calculation as in Algorithm B.1. With the cloud-based QPU, we see that the MLE-based algorithm beats the simple algorithm of sampling only from the evaluation oracle for depths up to 2. In fact, the minimum mean error is about 0.048 and this value is achieved for the MLE algorithm with maximum depth 1, namely, three sequential oracle applications. As we increase the depth further, samples are detrimental to the accuracy and make the approximation error shoot up. This is due to the effective depolarizing error that results from noise in the QPU operations.

On the other hand, the results on the new next-generation IonQ QPU show remarkable improvements, allowing us to provide estimates with much smaller additive errors for much larger depth circuits. The minimum mean approximation error is 0.0138 and is achieved for the MLE algorithm with depth-6 circuits. The difference between the mean errors between depths 2 and 6 is rather small and thus one can argue that the saturation of the error at depth 6 is not robust. However, if we examine the error bars carefully, the error bars at depths 2 and 6 are in fact disjoint, so we can conclude that the event that the mean error at depth 2 is smaller than that at depth 6 has low probability over experimental runs and that in most experimental runs a saturation of the mean error at some depth greater than 2 will be observed. The improved gate fidelity for the next-generation system accounts for the stark differences in the two plots and the improvement for the AE results.

Let us give some more details now about the depolarizing error model and relate it to the results of the hardware demonstration. Let $p = \sin^2(\theta)$ be the quantity being estimated by the AE algorithm. There are two different sources of effective depolarizing noise for this experiment. The combined readout and initial state preparation error β is independent of the circuit depth. The gate error α_t is the probability of depolarizing noise due to the loss of coherence for gates in the depth t evaluation. The combined effect of these errors is given as

$$\eta_t = \beta + \alpha_t. \tag{2}$$

The above equation approximates $(1-\alpha_t)(1-\beta)$ by $1-\alpha_t-\beta$; this is a good approximation in the regime when α_t, β are small. For the depth t experiment, the outcome $|0\rangle$ is observed with probability $\overline{p_t} = p_t(1-\eta_t) + \eta_t/2$.

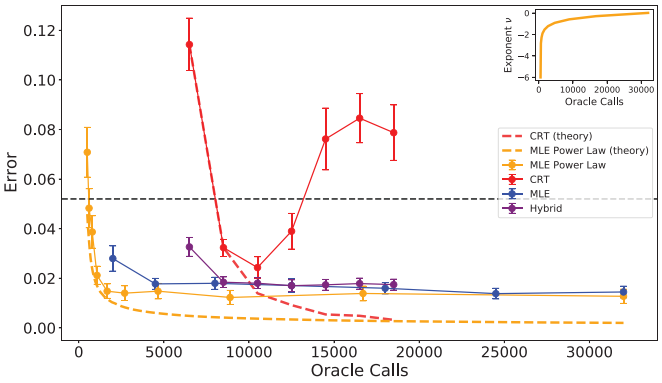


FIG. 3. Comparison of the all the amplitude estimation algorithms discussed in this paper. The maximum-likelihood methods perform the best, particularly when the schedules are optimized with the power-law method (orange). The CRT-based algorithms have a higher mean error, despite its theoretical guarantees. The black dashed line represents the baseline mean error for sampling from the evaluation oracle. The inset shows the optimal exponent ν chosen at various error levels by the power-law algorithm.

The difference between actual and estimated probability is given by

$$|\overline{p_t} - p_t| = \eta_t |1/2 - p_t|. \tag{3}$$

The above equation shows that the mean error for the experiment is nonzero; more precisely the mean error can be computed as the integral $\eta_t \int |1/2 - p_t| dp$ over the distribution from which the probabilities are sampled. This allows us to extrapolate the blue lines in Fig. 2 and conclude that the mean error for sampling from the evaluation oracle saturates and does not improve by taking even more samples from the evaluation oracle.

The orange line in Fig. 2 corresponds to the MLE-based AE algorithm that cases achieve substantially better estimation accuracy compared to taking samples only from the evaluation oracle for the next-generation QPU experiments. As noisier samples at higher depths are taken into account, the accuracy for the MLE-based algorithm worsens for the cloud-based QPU. The lower noise rates for the next-generation QPU enable the algorithm to avoid this degradation in performance for circuits with up to depth 7 that consist of more than 90 two-qubit gates and two-qubit depth of more than 60. Some additional results for the performance of the MLE-based algorithm are given in Appendix B.

IV. COMPARISON OF AMPLITUDE ESTIMATION ALGORITHMS

In addition to the MLE-based AE, we also implemented a simplified version of the CRT-based AE described in Appendix C and two noise-aware AE algorithms that combine these two techniques described in Appendix D.

In Fig. 3, we provide a comparison between all different AE methods we described above, namely, sampling from the evaluation oracle, MLE based with linear schedules, MLE based with power-law schedules, CRT based, and hybrid. We see that the MLE-based algorithms perform better on hardware, and the one using power-law schedules with expo-

nents chosen experimentally achieves the best performance. The CRT-based algorithms have a higher mean error than the MLE-based algorithm, though they have better theoretical guarantees and provide good estimates for many inputs. The black dashed line represents the baseline error for sampling from the evaluation oracle.

We also applied the iterative quantum AE algorithm [11] on the same data, and saw that it systematically underperformed compared to the maximum-likelihood and coprime methods due to the shallow depths accessible in the experiment. Indeed, the algorithm in Ref. [11] generates a sampling schedule which is exponentially spaced out in depth; when applied to our experimental data, only samples from two or three low depths end up being used, thus it does not achieve competitive results. To make this iterative quantum AE methods competitive, we would need to access larger depths and also a larger, adaptive number of shots according to Ref. [11]. When such larger depths become available, also benchmarking against methods with exponentially spaced out schedules will be important.

V. CONCLUSIONS

We reported the results for experiments on low-depth AE algorithms using MLE- and CRT-based approaches on a state-of-the-art trapped ion quantum computer. The high fidelity of the quantum hardware allowed us to run oracle circuits with depths ranging up to 7, which translates to four qubit circuits with more than 90 two-qubit gates and depth 60. Similar experiments on quantum hardware available on the cloud provided considerably worse results.

The MLE-based algorithms showed significant improvements in accuracy when higher depth samples were taken into account, reaching mean errors of less than 0.014 at depth 6, while the mean error when samples from the evaluation oracle are taken saturates to 0.053. Further, the error bars indicate that with high probability the mean error will saturate at depths greater than 2 for the next-generation QPU. We also developed a more sophisticated version of maximum-likelihood AE based on a power-law schedule. This introduced two improvements: First, the asymptotic precision improved since the power-law algorithm incorporates a noise model, albeit an imperfect one. Second, this noise floor is reached much faster in terms of oracle calls since the optimal power-law schedules spend fewer shots at costly higher depths. Note that all maximum-likelihood methods can naturally accommodate any probabilistic noise model in the definition of the likelihoods.

The CRT-based algorithm is more sensitive to noise and was affected by the hardware noise as well as the correlated errors across experiments. It achieved a minimum mean error of 0.024 at depth 3, following its design precision curve, before departing from it at larger depths. A hybrid algorithm that combines small-depth MLE estimates with CRT estimates achieved minimum mean error of 0.017, an improvement over the depth-2 MLE estimates with an average error of 0.018. With improvements in hardware fidelity and calibration, the CRT-based and hybrid algorithms are expected to become competitive with the MLE-based approach.

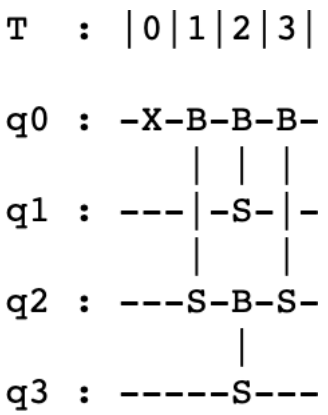


FIG. 4. The inner product estimation oracle for four-dimensional inputs. It consists of an initial X gate and four *RBS* gates whose parameters depend on the input vectors.

We restricted the experiments to four qubits because our main goal was to probe the regime where the evaluation oracle is invoked a large number of times in a noisy setting, achieving up to 15 sequential oracle applications with still remarkable results. A next step would be to establish experimentally trade-offs between circuit depth and oracle calls, as proved in Ref. [12], and this may soon be feasible with further hardware improvements. A back of the envelope calculation for the allowed error rate per gate for which these trade-offs become effective is adapted from Ref. [2] and appears in Appendix E—these calculations indicate that the higher gate fidelity is required for the depth versus oracle call trade-off for AE to be observable.

Overall, the experimental results provide a proof-of-concept that AE algorithms can be used on near-term quantum computers to reduce the error in the estimate of the expectation value of an observable, even if they do not yet provide speedup over classical algorithms.

ACKNOWLEDGMENTS

A.P., T.G.-T, W.Z., and I.K. designed the algorithms and performed the analysis of the experimental data. J.N., N.P., K.S., K.W., and S.J. performed the experiments on the IonQ hardware.

APPENDIX A: THE AMPLITUDE ESTIMATION ALGORITHM AND CIRCUITS

We specify the evaluation oracle that will be used as our AE benchmark. We use a simple circuit for estimating the square inner product between two random unit vectors that are encoded with a unary amplitude encoding method. Such circuits have been used before for estimating the Euclidean distance between data points in Ref. [28]. In Fig. 4, we see the inner product estimation circuit for four-dimensional vectors that use four qubits and four RBS gates (denoted as B-S in the circuit and defined in equation (A1)).

An $RBS(\phi)$ gate is a two-qubit gate defined as

$$RBS(\phi) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\phi & \sin\phi & 0 \\ 0 & -\sin\phi & \cos\phi & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}. \tag{A1}$$

Note also that $RBS^{\dagger}(\theta) = RBS(-\theta)$.

Given two unit vectors x and y, one can find angles ϕ for the four *RBS* gates in the circuit so the final state at the end of the quantum circuit is of the form

$$x \cdot y\,|e_1\rangle + |G\rangle\,, \tag{A2}$$

where $|G\rangle$ is an unnormalized unary state orthogonal to e_1; in other words, whose first qubit is in state $|0\rangle$. This implies that when we measure the first qubit of this quantum state, the probability we get outcome 1 is exactly $(x \cdot y)^2$, while the outcome is 0 with the remaining probability $1 - (x \cdot y)^2$. To match the notation for the AE algorithm, one can rewrite the above state as

$$\sin\theta\,|1\rangle\,|000\rangle + \cos\theta\,|0\rangle\,|000^{\perp}\rangle \tag{A3}$$

for θ such that $\sin\theta = x \cdot y$. Hence, we can use the inner product estimation circuits as the evaluation oracle for an AE implementation. We see that in total the evaluation oracle consists of four *RBS* gates with a circuit depth of 3. Here we focus on the number and depth of the two-qubit gates, since single-qubit gates are much faster and easier to implement.

We remark here that one could use many different evaluation oracle circuits to test the AE algorithms. The inner product estimation circuits used in the experiments as the evaluation oracle are known to be optimal both with respect to the number of gates and circuit depth and they have a number of different applications in estimating the Euclidean distance between data points in similarity-based classification algorithms [28] or for assisting in the training of neural networks [29–31]. They also provide a great benchmark for the AE algorithms, since one can easily increase the complexity of the oracle by increasing the dimension of the unit vectors. For example, using eight-dimensional vectors increases the complexity of the evaluation oracle to a circuit with eight qubits, ten *RBS* gates, and depth 5.

We will now see how to optimize the circuits of the AE before moving to the real hardware implementation. First, we define the operators in the iteration circuit. For implementing $\mathcal{A}^{\dagger}$, one only needs to invert the circuit and negate the angles of the *RBS* gates. The fact of using a unary encoding also greatly simplifies the implementation of the two reflections; in fact, it is easy to see that the reflections both simplify to adding a single Z gate on the first qubit between applications of the oracle and its adjoint.

The last optimization of the circuit is to notice that two *RBS* gates with a Z gate between them can be simplified to only one *RBS* gate and one Z gate by changing the angle parameter of the *RBS* gate. This simplification happens in all applications of the oracle apart from the last one. Thus, the final circuits we implement appear in Fig. 5, where we show the circuits for depths 1–3. Note that we implemented circuits up to depth 8.

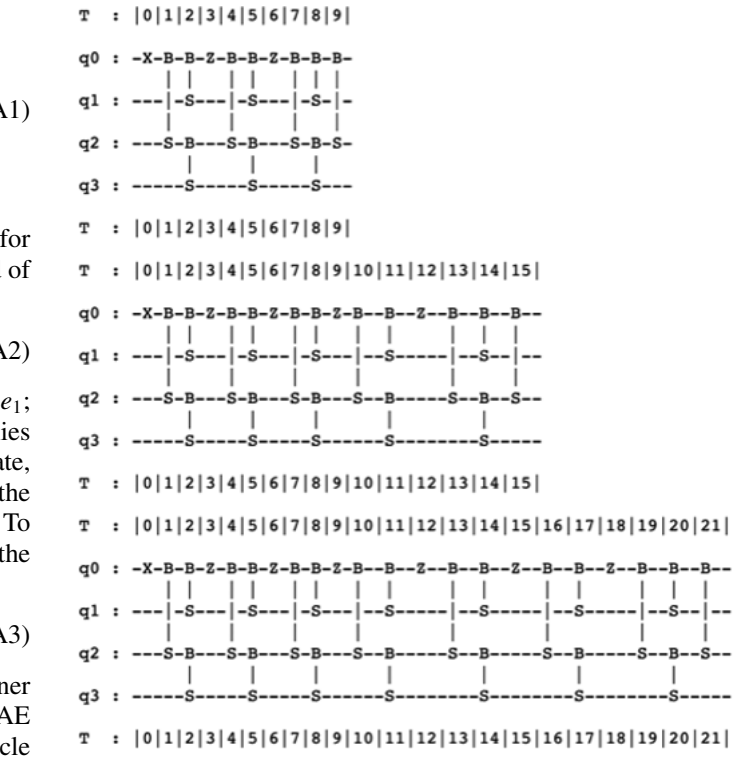


FIG. 5. The circuits U^t for depth $t = 1, 2, 3$. The hardware implementation consisted of circuits up to $t = 7$, with a total of 92 two-qubit gates and a circuit depth of 62

Lastly, we use the decomposition of the *RBS* gate into single-qubit gates and two CZ (two qubit controlled Z gate with first quit acting as control qubit) gates as in Fig. 6 [29] to implement it efficiently with the two-qubit gates available on the IonQ platform.

Hence, for the circuit U^t, the number of two-qubit gates implemented on the hardware are $(12t + 8)$ for a circuit depth of $(8t + 6)$.

APPENDIX B: MAXIMUM LIKELIHOOD ESTIMATION BASED AMPLITUDE ESTIMATION

The MLE-based AE algorithms combine information from samples taken from quantum circuits of the form $U^t = (\mathcal{A}S_0\mathcal{A}^{\dagger}S_{\chi})^t\mathcal{A}$ at different depths t in a statistically efficient manner to provide an accurate estimate of the parameter θ of the evaluation oracle $\mathcal{A}$, where $\mathcal{A}\,|0^k\rangle = \sin(\theta)\,|1, x\rangle + \cos(\theta)\,|0, x'\rangle$. An MLE-based algorithm is specified by a schedule of measurements (t, N_t) where N_t is the number of measurements made at depth t. A uniform (or more generally a beta distributed) prior on the success probabilities is updated according to the maximum likelihood rule applied to

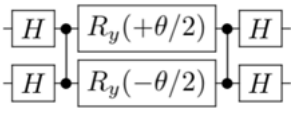


FIG. 6. A decomposition of the $RBS(\theta)$ using Hadadard gates H, two single-qubit rotations R_y, and two two-qubit CZ gates.

the results of the measurements for the different depth circuits. Different schedules for MLE-based AE algorithms yield different trade-offs between the number of oracle calls and the accuracy of the algorithm, the trade-off can be analyzed using the notion of Fisher information of the schedule. Suzuki *et al.* [9] proposed exponential schedules $(2^t, N_{\text{shot}})$, where samples are taken at depths that are multiples of powers of 2 ranging from 1 up to $2^{\log(1/\epsilon)}$ and showed that such exponential schedules can be used for AE without the need of a QFT. Linear and super polynomial schedules $(t^\alpha, N_{\text{shot}})$ for $\alpha > 1$ have been proposed [9,16] along with analysis supporting the robustness of these schedules against depolarizing noise. More generally, power law schedules $(t^\alpha, N_{\text{shot}})$ for $\alpha > 0$ were proposed in Ref. [12] with optimal trade-offs between circuit depth and total number of oracle calls.

Let us now describe the schedules we used in the experiments. First, we used linear schedules, since for small depths they provide a good and simple alternative. Furthermore, for MLE-based AE algorithms in a setting with depolarizing noise, it is known that the accuracy of the estimation improves only up to a certain depth, depending on the noise rate [12,16]. Taking further samples at greater depths actually degrades the performance of the algorithm since it basically adds more noise to the estimate. Thus, taking advantage of the high fidelities of the state-of-the-art IonQ quantum computer, we managed to run the AE algorithm with depths ranging from zero to seven, where circuit U^7 has 92 two-qubit gates and a depth of 62, and keeping the additive error for the estimate very small, in fact, below 0.02. The schedules used 500 shots per depth.

Algorithm B.1 Maximum likelihood estimation-based AE algorithm

Require: Parameter ϵ, maximum depth D and number of shots N_{shot}. Access to a unitary $\mathcal{A}$ such that $\mathcal{A}|0\rangle = \sin(\theta)|1\rangle|000\rangle + \sin(\theta)|0\rangle|000^\perp\rangle$ and to $U^d = (\mathcal{A}S_0\mathcal{A}^\dagger S_\chi)^d\mathcal{A}$ for $d \in [D]$.
1: Initialize the prior to the uniform distribution on angles $\theta = \frac{\pi k\epsilon}{2}$ for integer valued $k \in [0, \frac{1}{\epsilon})$, i.e. $p(\theta) = \frac{1}{\epsilon}, \forall\theta$.
2: **for** $t = 1$ **to** D **do**
3: Initialize $N_{d_0} = N_{d_1} = 0$, these variables record the 0 and 1 counts for the depth-d measurements.
4: **for** $i = 1$ to N_{shot}
5: Apply the unitary U^d and measure the resulting quantum state in the standard basis.
6: If the outcome is the state $|1000\rangle$ then $N_{d_0} = N_{d_0} + 1$, else if the outcome is a different unary string then $N_{d_1} = N_{d_1} + 1$, else do nothing.
7: **end for**
8: Perform Bayesian updates $p(\theta) \to p(\theta)\cos((2d+1)\theta)^{N_{d_0}}\sin((2d+1)\theta)^{N_{d_1}}$ and renormalize to obtain the posterior probability distribution.
9: **end for**
10: **Output** θ with the highest probability according to the posterior probability distribution.

In addition, efficient error mitigation was performed in the measurement results as in Ref. [28] by taking advantage of the unary encoding.

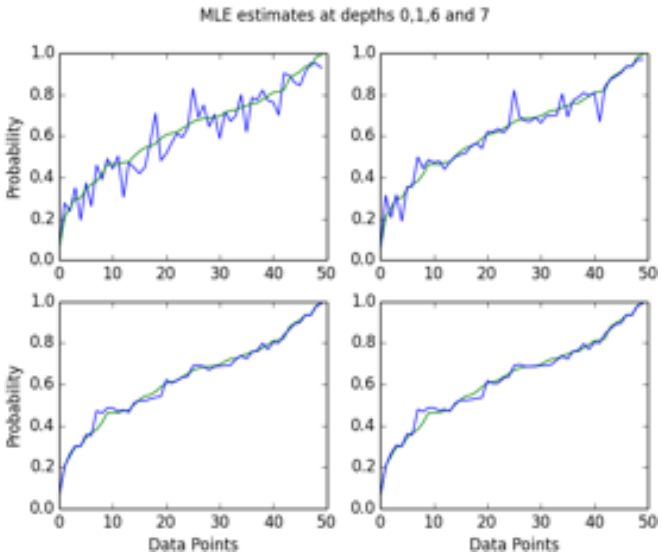


FIG. 7. The green curves represent the true probabilities for the data points and the blue curves correspond to the MLE-based estimates at depths $D = 0$ (top left), $D = 1$ (top right), $D = 6$ (bottom left), and $D = 7$ (bottom right), where the $D = 6$ was empirically observed to have the minimum mean error.

The overall algorithm used is presented in Algorithm B.1. It is an MLE-based AE algorithm with a linear schedule $(t, 500)$ with maximum depth equal to 7 and parameter $\epsilon = 0.001$. The algorithm uses $1/\epsilon = 1000$ buckets for the MLE estimation; ϵ is a lower bound on the accuracy for the algorithm. The evaluation oracle used was an inner product estimation circuit for four-dimensional unit vectors.

In Fig. 7, we plot the curves for the actual probabilities (green) and the MLE estimates (blue) at depths 0, 1 6, and 7, giving a global view of the results produced by the MLE-based AE algorithm. The minimum mean error over the 50 amplitudes was computed for MLE on circuits of depths up to 7. The depth-6 circuits had 80 two-qubit gates and depth 54 and were empirically observed to have the minimum mean error. The results can be viewed as a demonstration that MLE-based AE algorithms can achieve improvements in accuracy compared to sampling from the evaluation oracle for near-term quantum devices and in a noisy setting.

APPENDIX C: CHINESE REMAINDER THEOREM BASED AMPLITUDE ESTIMATION ALGORITHMS

In this Appendix, we look at a different type of low-depth AE algorithm that uses the CRT instead of MLE techniques to combine the samples at different circuit depths. This algorithm, called QoPrime, has a rigorous proof of convergence and in the noiseless case or in the case of depolarizing noise applied independently over the samples, needs fewer samples than the MLE based algorithm with linear schedules and matches the one with power-law schedules [12].

For the experiment, we use a variant of the QoPrime algorithm that samples at two consecutive depths to find low precision estimates for the amplitude that are then combined using the CRT to provide better estimates.

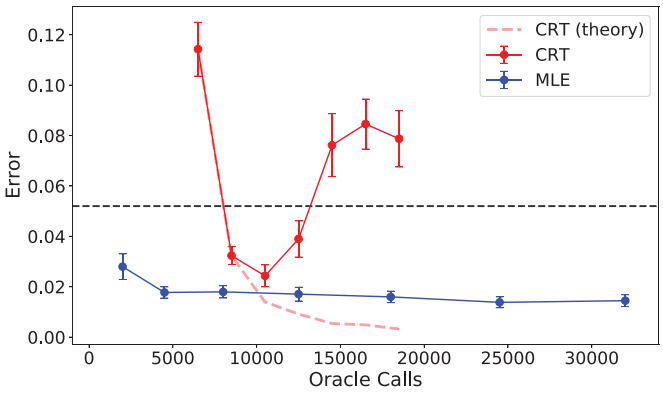


FIG. 8. Comparison of mean error against total number of oracle calls for MLE, exact CRT, and CRT algorithms for depths D ranging from 1 to 7. The black dashed line represents the baseline error for sampling from the evaluation oracle, same as the black line in Fig. 2.

Algorithm C.1 Chinese remainder theorem-based AE algorithm

Require: Accuracy parameter ϵ, maximum depth D, and number of shots N_{shot}. Access to a unitary $\mathcal{A}$ such that $\mathcal{A}|0\rangle = \sin(\theta)|1\rangle|000\rangle + \sin(\theta)|0\rangle|000^{\perp}\rangle$ and to $U^t = (\mathcal{A}S_0\mathcal{A}^{\dagger}S_{\chi})^t\mathcal{A}$ for $t \in [D]$.

1: Compute an MLE estimate p_0 for the quantity $\sin^2(\theta)$ using Algorithm IV.1 with maximum depth equal to 2 and define angle $\theta' = \arcsin(\sqrt{p_0})$.

2: Compute estimates p_D and p_{D-1} of the quantities $\sin^2((2D+1)\theta)$ and $\sin^2((2D-1)\theta)$ by using samples from circuits U^D and U^{D-1}, respectively.

3: Compute $l = \frac{2(2D-1)}{\pi}\arcsin(\sqrt{p_D})$ and $h = \frac{2(2D+1)}{\pi}\arcsin(\sqrt{p_{D-1}})$.

4: Compute $s_1 = \text{sgn}(\sin(2(2D-1))\theta')$ and $s_2 = \text{sgn}(\sin(2(2D+1)\theta')$.

5: Compute integers $v_t \mod (4D^2-1)$ for $1 \leqslant t \leqslant 4$ such that $v_t \mod (2D-1) = \lfloor s1.l \rfloor + (t \mod 2)$ and $v_t \mod (2D+1) = \lfloor s2.h \rfloor + t/2$ using the CRT.

6: Out of the four values computed in step 4, select v_t minimizing $|\sin^2(\frac{v_t\pi}{4D^2-1}) - p_0|$.

7: Output the value $\frac{v_t\pi}{4D^2-1}$ as an estimate for the angle θ.

The main idea for the CRT-based AE Algorithm C.1 is that if estimates obtained from running the evaluation oracle at depths D and $D-1$ are sufficiently accurate, then they can be combined to get a higher accuracy estimate using the CRT.

The depth D and $D-1$ estimates are determined up to an ambiguity in sign, which are resolved using an approximation to the true success probability obtained with a depth-2 MLE estimate. For The QoPrime algorithm [12], the sign resolution is carried out recursively using depth-0 samples and the analysis uses the Chernoff bounds. A depth-2 MLE estimate was used for sign resolution in this work, as for the experiments the depth-0 error saturates to a large additive error of 0.06 due to depolarizing noise, so the depth-0 samples are not accurate enough to resolve the sign ambiguities.

Figure 8 shows the results of Algorithm C.1 with the inner product estimation circuit as the evaluation oracle on the IonQ next-generation QPU for depths ranging from 1–7. As we said, here the CRT-based algorithm of depth D takes samples only from circuits of depths D and $D-1$, while it also uses an MLE-based algorithm for depth 2 to differentiate the signs of the estimates. Note as well that the cloud QPU was not able to show that this algorithm improves the approximation error with respect to sampling directly from the evaluation oracle, and the next-generation QPU was necessary to show such an improvement.

In Fig. 8, the dashed pink curve represents the error if the exact success probabilities $\sin^2((2D-1)\theta)$ and $\sin^2((2D+1)\theta)$ are used instead of the estimated probabilities in step 2 of the CRT algorithm. Using the exact success probabilities instead of the estimates obtained from depth D and $D-1$ runs of the oracle corresponds to the accuracy of the Algorithm C.1 for a noiseless setting where the number of shots is large enough to ensure convergence to the true success probabilities. The red curve represents the accuracy of the CRT algorithm C.1. This achieves a minimum mean error of 0.024 at depth three. The figure shows that the errors for the CRT algorithm C.1 match the errors for the noiseless algorithm for depths up to two corresponding to the coprime moduli (3,5). The CRT algorithm is less robust against noise compared to the MLE as it requires both probabilities p_D and p_{D-1} to be good approximations to the true probabilities. The result for the CRT will be incorrect if either of these probabilities is not estimated correctly.

We also remark that while the MLE-based algorithm performs better on average than the CRT-based algorithm, the noiseless success probabilities for the CRT algorithm represented by the green line are lower than the empirically observed probabilities for the MLE-based algorithm in Fig. 2. However, the CRT-based AE Algorithm C.1 requires fewer oracle calls than the MLE algorithm as it considers samples only at depths 0,1,2 and at D, $D-1$.

Let us try to understand in more detail the performance of the CRT-based algorithm. Incorrect estimates at higher depths are more common due to greater noise for deeper circuits, resulting in a high average error for the CRT-algorithm for samples of depth greater than three. Figure 9 plots the true success probabilities at depths 0, 3, 5, and 7 (blue) and the estimates from the runs of the corresponding oracles with 500 shots (red). It can be seen from the figures that the AE estimates are not always a close approximation for the true success probabilities. The inputs to the CRT procedure computed in step 3 of the algorithm differ from the correct estimates for the outlying data points resulting in an error in the final answer. This results in the CRT algorithm having a higher mean error than the MLE algorithm.

We can also relate the error models and the experimental results of the CRT-based algorithm. The theoretical analysis of the QoPrime algorithm assumed independence of errors over samples so Chernoff bounds could be applied. However, in the hardware implementation, the errors over the 500 shots are not independent but correlated in time with larger errors sometimes occurring on consecutive runs of the circuit, and this results in a nonsmooth approximation to the success probabilities. It should be noted that the next-generation QPU is an experimental system and further improvements in gate fidelity and calibration should improve the quality of approximation

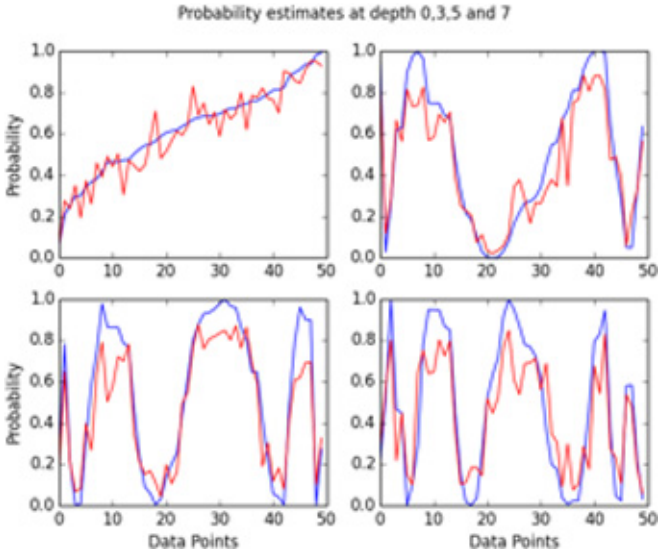


FIG. 9. Exact probabilities $\sin^2((2D+1)\theta)$ and the probability estimates with 500 shots for the depths $D = 0$ (top left), $D = 3$ (top right), $D = 5$ (bottom left), and $D = 7$ (bottom right).

for the data points and the performance of the CRT-based AE algorithm.

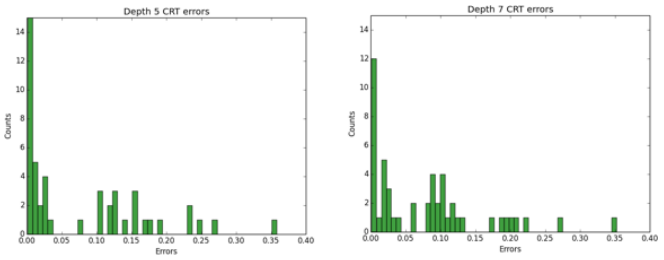
APPENDIX D: AMPLITUDE ESTIMATION WITH EXPERIMENTAL HYPERPARAMETER TUNING

In this Appendix, we show that we can further improve the performance of the AE algorithms by using experimental results on the quantum hardware to tune some hyperparameters of the algorithms.

1. Hybrid amplitude estimation algorithm

We start with Algorithm D.2, a simple hybrid algorithm that compares between a low-depth MLE-based estimate and the CRT-based estimate and tries to output the one with the smaller error. Such an algorithm tries to use the higher CRT estimates when they are correct, but falls back to an MLE-based estimate when the CRT estimate seems to be wrong. To choose between the CRT- and MLE-based estimates, we cannot use the estimate with the real smaller error, as the true success probabilities are unknown. Instead, we use a heuristic test condition that outputs the CRT estimate if the MLE and CRT estimates do not differ more than a β factor times some average difference $|\mathrm{MLE}_{\mathrm{av}}(2) - \mathrm{CRT}_{\mathrm{exact}}(D)|$ between the two estimates, which are quantities that can be computed through experiments on some training data.

More precisely, to define these quantities, we fix a prior distribution on probabilities—this can be the uniform or a beta distributed prior, for example. $\mathrm{MLE}_{\mathrm{av}}(2)$ is the average mean error for the MLE algorithm at depth 2 (or some other low depth) with the probabilities drawn from the prior distribution. $\mathrm{CRT}_{\mathrm{exact}}(D)$ are the mean errors for the classically simulated CRT algorithm with probabilities drawn from the prior distribution. In this case, they can be read off from the blue and green lines as in Fig 8. The factor β can also be tuned experimentally.



(a) Depth 5 (b) Depth 7

FIG. 10. Histograms for errors for the CRT-based AE algorithm at depths $D = 5$ and $D = 7$. This shows a distinct *failure mode* for the CRT algorithm, in that many of the points are estimated accurately, while others are entirely miscalculated by the algebraic reconstruction of the CRT. The hybrid algorithm D.2 is designed to diagnose this issue.

Algorithm D.1 Hybrid AE algorithm

Require: Accuracy parameter ϵ, parameter β, maximum depth D, and number of shots N_{shot}. Access to a unitary $\mathcal{A}$ such that $\mathcal{A}|0\rangle = \sin(\theta)|1\rangle|000\rangle + \sin(\theta)|0\rangle|000^{\perp}\rangle$ and to $U^t = (\mathcal{A}S_0\mathcal{A}^{\dagger}S_{\chi})^t\mathcal{A}$ for $t \in [D]$.
1: Compute an MLE estimate p_0 for the quantity $\sin^2(\theta)$ using Algorithm IV.1 with maximum depth equal to 2.
2: Compute CRT estimate q_0 with maximum depth D using the CRT algorithm C.1.
3: If $|p_0 - q_0| > \beta|\mathrm{MLE}_{\mathrm{av}}(2) - \mathrm{CRT}_{\mathrm{exact}}(D)|$ return p_0 else return q_0.

Another way to see the potential advantage of the CRT-based approach is to look at the histograms for the errors achieved by the algorithm at high depths. Figure 10 shows the error histograms for the CRT-based algorithm with maximum depths 5 and 7. It can be seen from the figure that even at high depths, the CRT-based AE algorithm is able estimate a large fraction of the points with very low errors. The larger mean errors compared to MLE can be explained by a few outlying points with large errors. The algorithm is close to the theoretical accuracy for depths up to 3.

The performance of the hybrid AE algorithm compared to MLE and the CRT is illustrated in Fig. 3. The estimates output by the hybrid algorithm are considerably better than the CRT-based ones and match the MLE estimates for up to depth equal to 5. The hybrid algorithm achieves a minimum mean error of 0.017 at depth 4, which improves upon the mean error 0.018 for the depth-2 MLE algorithm. Note also that the hybrid algorithm uses fewer oracles calls than the MLE-based algorithm for the same maximum depth.

2. Power law amplitude estimation

The MLE-based amplitude estimation Algorithm B.1 used a linear schedule and a fixed number of shots at each depth. Namely, we used a schedule of the form (d, N_{shot}) for d all integers up to D, the maximum depth. Here we introduce an additional degree of freedom in the choice of the exponents in the *power-law AE algorithm*, studied in Ref, [12]. We propose a noise-aware power-law AE method that solves an

optimization problem to determine the optimal exponent for a given target error. The sampling schedules will be of the form

$$N_d = \lfloor N_{\text{shots}} \times (2d+1)^{\nu} \rfloor, \tag{D1}$$

where N_d is the number of samples taken at depth d, for $d = 0, 1, 2, 3, \ldots$. The number of shots N_{shots} was taken to be 500 and the exponent ν is the hyper-parameter of the algorithm that we tune experimentally.

Assuming an unbiased estimator, the Cramér-Rao bound can be used to connect the schedule to the estimation error. When there is a specific level of (depolarizing) noise γ_d associated with depth d, the classical Fisher information is exponentially damped due to the noise. Given the sampling schedule in Eq. (D1), the problem becomes finding the exponent ν that minimizes the oracle calls for given a desired target error, namely,

$$\text{minimize}_{\nu} \quad \mathcal{O} \sim \sum_{d=0}^{D} (2d+1)^{\nu+1}, \tag{D2}$$

$$\text{subject to} \quad \mathcal{F}_{\text{noisy}} \equiv N_{\text{shots}} \sum_{d=0}^{D} (2d+1)^{\nu+2} e^{-2\gamma_d} \geqslant \epsilon^{-2}, \tag{D3}$$

where we introduce a parameter-agnostic approximation $\mathcal{F}_{\text{noisy}}$ to the true Fisher information. This optimization problem is solved classically, given experimental estimates of the noise levels γ_d, and allowing the exponent ν to take values smaller than -1. For the data under consideration, a depolarizing model was fit with noise parameter γ_d ranging from $\gamma_0 = 0.035$ at depth 0 (evaluation oracle), up to $\gamma_7 = 0.35$ at depth seven.

The last step of the algorithm is similar to the first MLE method in B.1 in that it performs a noise-aware Bayesian update on the parameter θ given the data at each depth d:

$$p(\theta \,|\, N_{d_0}, N_{d_1}) \sim p_d(0)^{N_{d_0}} p_d(1)^{N_{d_1}} p(\theta), \tag{D4}$$

where N_{d_0} and N_{d_1} are the number of 0s and 1s measured at depth d, and in the depolarizing noise assumption the measurement probabilities are

$$p_d(1) = \frac{1 - e^{-\gamma_d} \cos(2(2d+1)\theta)}{2}, \quad p_d(0) = 1 - p_d(1). \tag{D5}$$

The maximum-likelihood estimates for the power-law method were performed on data which is sampled without replacement from the measurements at each depth, namely, for every target error we construct a data set with $N_d \leqslant N_0 = 500$ Bernoulli trials for depths $d \leqslant 7$, drawn from the data set of all measurements.

Finally, one could question the use of the power-law ansatz (D1) in the first place, when the optimization problem in (D2) could be formulated more fundamentally as a linear program over the sampling schedule N_d. The answer lies in the statistical subtleties which appear when translating the AE problem into the language of classical inference. Specifically, the simple schedules arising out of solving a naïve linear program would most likely *not lead to unbiased estimators* due to the periodic nature of the likelihood function, as pointed out in [12]. In short, the choice of a power law schedule is an empirically-validated compromise which ensures an unbiased estimator, and thus legitimizes the use of the simple Cramér-Rao bound above. More work is needed in order to obtain provable guarantees with such sampling schedules.

	Low depth amplitude estimation framework									
	classical $\beta = 1$	$\beta = 2/3$			$\beta = 1/3$			standard quantum $\beta = 0$		
ϵ	classical samples	Oracle depth	Allowed error rate	speedup	Oracle depth	Allowed error rate	speedup	Oracle depth	Allowed error rate	speedup
10^{-2}	10^4	11	9×10^{-4}	5	45	2×10^{-4}	22	2×10^3	5×10^{-5}	10^2
10^{-3}	10^6	21	5×10^{-4}	10	200	5×10^{-5}	100	2×10^3	5×10^{-6}	10^3
10^{-4}	10^8	45	2×10^{-4}	22	900	1×10^{-5}	450	2×10^4	5×10^{-7}	10^4
10^{-5}	10^{10}	93	1×10^{-5}	46	4000	2×10^{-6}	2000	2×10^5	5×10^{-8}	10^5

FIG. 11. Back-of-the-envelope resource estimates for quantum Monte Carlo algorithms.

Algorithm D.2 Noise-aware power-law MLE algorithm

Require: Target accuracy ϵ, maximum depth D, number of shots N_{shots}, noise levels γ_d for $d \in [D]$. Access to a unitary $\mathcal{A}$ such that $\mathcal{A}|0\rangle = \sin(\theta)|1\rangle|000\rangle + \sin(\theta)|0\rangle|000^{\perp}\rangle$ and to $U^d = (\mathcal{A}S_0\mathcal{A}^{\dagger}S_{\chi})^d\mathcal{A}$ for $d \in [D]$.
1: Find optimal power-law schedule $S = \{(d, N_d)\}$, with $d \in [D]$, using Eqs. (D1) and (D2).
2: Initialize the prior to the uniform distribution on angles $\theta = \frac{\pi k \epsilon}{2}$ for integer valued $k \in [0, \frac{1}{\epsilon})$, i.e., $p(\theta) = \frac{1}{\epsilon}, \forall \theta$.
3: **for** $(d, N_d) \in S$
4: Initialize $N_{d_0} = N_{d_1} = 0$, these variables record the 0 and 1 counts for the depth-d measurements.
5: **for** $i = 1$ to N_d **do**
6: Apply the unitary U^d and measure the resulting quantum state in the standard basis.
7: *Postprocessing:* If the outcome is the state $|1000\rangle$ then $N_{d_0} = N_{d_0} + 1$, else if the outcome is a different unary string then $N_{d_1} = N_{d_1} + 1$, else do nothing.
8: **end for**
9: Perform Bayesian updates according to Eq. (D4) and renormalize to obtain the posterior probability distribution.
10: **end for**
11: Output θ with the highest probability according to the final posterior probability distribution.

The performance of the power-law AE algorithm compared to previous AE algorithms is illustrated in Fig. 3. The power law algorithm was performed for ten different target errors ϵ, where for each case a new optimization problem was solved to find the optimal schedules for the algorithm. We also plot the theoretical curve that corresponds to Eq. (D3).

APPENDIX E: BACK OF ENVELOPE CALCULATIONS

Back-of-the-envelope resource estimates for quantum Monte Carlo algorithms interpolating between classical sampling and the standard standard algorithm [3] using the low-depth variants are presented in Fig. 11. The figure is adapted from Ref. [2] and computes the raw quantum speedup obtained—i.e., the ratio of the number of classical samples needed to the number of quantum samples needed for each setting, where a setting is parametrized by a number $\beta \in [0, 1]$ that selects the depth versus oracle calls trade-off for the low-depth AE algorithm. The error rates per evaluation oracle

call that are permissible in the quantum sampling algorithm to achieve a 99% confidence interval at the end of the algorithm are given in the figure.

One can imagine extending the figure for more values of β as required; the resource estimates can be applied to a specific setting using the extended figure. The gate fidelity for the quantum device and the evaluation oracle circuit together determine for each column in the figure the actual error rate for the oracle; the constraint that the actual error rate is less than the allowed error rate in turn determines the precision ϵ up to which the AE algorithm yields meaningful results and also the raw speedup obtained. The second factor to be considered is the slowdown in clock speed for the quantum device compared to the classical device—the algorithm obtains an effective speedup over classical if this factor is less than the raw speedup. The speedup calculation using the figure takes an asymptotic viewpoint; the actual speedups for the quantum algorithm may be better or worse by a constant factor. However, such a calculation is useful as it provides a ballpark estimate of the extent of speedup achievable by low-depth AE for a specific quantum device and evaluation oracle.

[1] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Quantum amplitude amplification and estimation, Contemp. Math. **305**, 53 (2002).

[2] A. Bouland, W. van Dam, H. Joorati, I. Kerenidis, and A. Prakash, Prospects and challenges of quantum finance, arXiv:2011.06492.

[3] A. Montanaro, Quantum speedup of Monte Carlo methods, Proc. R. Soc. A **471**, 20150301 (2015).

[4] D. S. Abrams and C. P. Williams, Fast quantum algorithms for numerical integrals and stochastic processes, arXiv:quant-ph/9908083.

[5] L. K. Grover, A framework for fast quantum mechanical algorithms, in *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing* (ACM, New York, 1998), pp. 53–62.

[6] G. Brassard, F. Dupuis, S. Gambs, and A. Tapp, An optimal quantum algorithm to approximate the mean and its application for approximating the median of a set of points over an arbitrary distance, arXiv:1106.4267.

[7] T. Li and X. Wu, Quantum query complexity of entropy estimation, IEEE Trans. Inf. Theory **65**, 2899 (2018).

[8] Y. Hamoudi and F. Magniez, Quantum Chebyshev's inequality and applications, in *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)* (Schloss Dagstuhl-Leibniz-Zentrum for Informatics, Germany, New York, 2019).

[9] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, Amplitude estimation without phase estimation, Quant. Info. Proc. **19**, 75 (2020).

[10] S. Aaronson and P. Rall, Quantum approximate counting, simplified, in *Symposium on Simplicity in Algorithms* (SIAM, Germany, New York, 2020), pp. 24–32.

[11] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, Iterative quantum amplitude estimation, npj Quantum Inf. **7**, 52 (2021).

[12] T. Giurgica-Tiron, I. Kerenidis, F. Labib, A. Prakash, and W. Zeng, Low depth algorithms for quantum amplitude estimation, Quantum **6**, 745 (2022).

[13] P. Burchard, Lower bounds for parallel quantum counting, arXiv:1910.04555.

[14] S. Jeffery, F. Magniez, and R. De Wolf, Optimal parallel quantum query algorithms, Algorithmica **79**, 509 (2017).

[15] C. Zalka, Grover's quantum searching algorithm is optimal, Phys. Rev. A **60**, 2746 (1999).

[16] T. Tanaka, Y. Suzuki, S. Uno, R. Raymond, T. Onodera, and N. Yamamoto, Amplitude estimation via maximum likelihood on noisy quantum computer, Quant. Info. Proc. **20**, 293 (2021).

[17] P. Rebentrost, B. Gupt, and T. R. Bromley, Quantum computational finance: Monte Carlo pricing of financial derivatives, Phys. Rev. A **98**, 022321 (2018).

[18] D. J. Egger, R. García Gutiérrez, J. C. Mestre, and S. Woerner, Credit risk analysis using quantum computers, IEEE Transactions on Computers **70**, 2136 (2020).

[19] S. Woerner and D. J. Egger, Quantum risk analysis, npj Quantum Inf. **5**, 15 (2019).

[20] K. Miyamoto, Bermudan option pricing by quantum amplitude estimation and chebyshev interpolation, EPJ Quantum Technology **9**, 3 (2022).

[21] N. Stamatopoulos, D. J. Egger, Y. Sun, C. Zoufal, R. Iten, N. Shen, and S. Woerner, Option pricing using quantum computers, Quantum **4**, 291 (2020).

[22] S. Chakrabarti, R. Krishnakumar, G. Mazzola, N. Stamatopoulos, S. Woerner, and W. J. Zeng, A threshold for quantum advantage in derivative pricing, Quantum **5**, 463 (2021).

[23] K. Miyamoto and K. Shiohara, Reduction of qubits in a quantum algorithm for Monte Carlo simulation by a pseudo-random-number generator, Phys. Rev. A **102**, 022424 (2020).

[24] S. Ramos-Calderer, A. Pérez-Salinas, D. García-Martín, C. Bravo-Prieto, J. Cortada, J. Planagumà, and J. I. Latorre, Quantum unary approach to option pricing, Phys. Rev. A **103**, 032414 (2021).

[25] A. C. Vazquez and S. Woerner, Efficient State Preparation for Quantum Amplitude Estimation, Phys. Rev. Applied **15**, 034027 (2021).

[26] K. Kaneko, K. Miyamoto, N. Takeda, and K. Yoshino, Quantum pricing with a smile: Implementation of local volatility model on quantum computer, EPJ Quantum Technology **9**, 7 (2022).

[27] P. Rao, Kwangmin Yu, Hyunkyung Lim, D. Jin, and D. Choi, *Quantum amplitude estimation algorithms on IBM quantum devices*, in *Quantum Communications and Quantum Imaging XVIII* (International Society for Optics and Photonics, 2020), Vol. 11507, p. 115070O.

[28] S. Johri, S. Debnath, A. Mocherla, A. Singk, A. Prakash, J. Kim, and I. Kerenidis, Nearest centroid classification on a trapped ion quantum computer, npj Quantum Inf. **7**, 122 (2021).

[29] N. Mathur, J. Landman, Yun Yvonna Li, M. Strahm, S. Kazdaghli, A. Prakash, and I. Kerenidis, Medical image classification via quantum neural networks, arXiv:2109.01831.

[30] J. Allcock, C. Y. Hsieh, I. Kerenidis, and S. Zhang, Quantum algorithms for feedforward neural networks, ACM Trans. Quantum Comput. **1**, 1 (2020).

[31] I. Kerenidis, J. Landman, and A. Prakash, Quantum algorithms for deep convolutional neural networks, *Eighth International Conference on Learning Representations ICLR*, 2019.

[32] K. Wright, K. M. Beck, S. Debnath, J. M. Amini, Y. Nam, N. Grzesiak, J.-S. Chen, N. C. Pisenti, M. Chmielewski, C. Collins *et al.*, Benchmarking an 11-qubit quantum computer, Nat. Commun. **10**, 5464 (2019).