Noise-Aware Quantum Amplitude Estimation

STEVEN HERBERT1,2, IFAN WILLIAMS1, ROLAND GUICHARD1, AND DARREN NG1

1Quantinuum, CB2 1NL Cambridge, U.K.
2Department of Computer Science and Technology, University of Cambridge, CB3 0FD Cambridge, U.K.
Corresponding author: Ifan Williams (e-mail: ifan.williams@quantinuum.com).

ABSTRACT In this article, based on some simple and reasonable assumptions, we derive a Gaussian noise model for quantum amplitude estimation. We provide results from quantum amplitude estimation run on various IBM superconducting quantum computers and on Quantuminum’s H1 trapped-ion quantum computer to show that the proposed model is a good fit for real-world experimental data. We also show that the proposed Gaussian noise model can be easily composed with other noise models in order to capture effects that are not well described by Gaussian noise. We give a generalized procedure for how to embed this noise model into any quantum-phase-estimation-free quantum amplitude estimation algorithm, such that the amplitude estimation is “noise aware.” We then provide experimental results from running an implementation of noise-aware quantum amplitude estimation using data from an IBM superconducting quantum computer, demonstrating that the addition of “noise awareness” serves as an effective means of quantum error mitigation.

INDEX TERMS Noise characterization, noisy intermediate-scale quantum, quantum algorithms, quantum amplitude estimation (QAE), quantum computing (QC), quantum error mitigation.

I. INTRODUCTION

The term noisy intermediate-scale quantum computer, coined by Preskill [1], has become the industry standard for early quantum computers, and the name immediately tells us two of their most important features: they are noisy, and they are too small (“intermediate-scale”) to run fault-tolerant algorithms. Thus, noise, as well as the consequent possibility of error, is something we have to live with.

Many ingenious suggestions for quantum error mitigation [that is, using circuit design and classical pre- and postprocessing to reduce the adverse effects of noise, without actually using any qubits for quantum error correction (QEC)] have been proposed, including zero-noise extrapolation [2], [3], randomized compilation [4], probabilistic error correction [5], and many more. It is also likely that some QEC (albeit falling short of the amount required for fault tolerance) will ultimately be deployed in early algorithms exhibiting genuine useful quantum advantage (see, e.g., [6] and [7]), and therefore, it is perhaps more appropriate to speak of resource-constrained quantum hardware, where the aim of algorithm design is to achieve such quantum advantage with minimal resource requirements.

One approach (which, in general, is complementary to the error mitigation techniques discussed earlier) is to characterize the noise as a “noise model” and handle the noise at the application level. This is particularly applicable for quantum sampling algorithms (see, e.g., [8], [9], [10], and [11]), where the effect of the noise is that the samples are from a different distribution than in the corresponding noiseless case, but in principle with an accurate noise model, the distribution actually being sampled from can still be expressed. Moreover, sampling provides the basis for many quantum optimization and estimation algorithms [8], [9], and with an accurate noise model, there may still be quantum advantage in speed of convergence even in the presence of noise (that is, if the noise is sufficiently mild, with some errors possibly having been mitigated/corrected by some of the techniques mentioned).

In this article, we address what is probably the simplest such instance of this concept, by proposing a suitable model for the noise accumulated when Grover iterations are repeated to perform quantum amplitude amplification. In particular, we address one such algorithm, quantum amplitude estimation (QAE), and focus on propositions for QAE, which do not call quantum phase estimation (QPE) as a
subroutine [12], [13], [14], [15]. Thus, in these algorithms the quantum circuits consist only of state preparation and then amplitude amplification—there is no other circuitry involved. Moreover, these proposed algorithms all share the property that only a single qubit is measured, which gives us the basis for a simple Gaussian noise model that is valid for some particular assumptions made about the noise. In addition, QAE not only provides a suitable subject for a noise model but is also an important algorithm, which underpins quantum Monte Carlo integration (QMC) [8], [16], [17], [18] and thus forms the basis for many important anticipated applications in, for example, quantum finance [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29].

This approach of translating the noise at the physical level to the resultant estimation uncertainty at the application level in QAE has begun to gain traction recently, with both Brown et al. [30] and Tanaka et al. [31] investigating how generic noise models, such as depolarizing, amplitude damping, and phase damping noise, impact the performance of amplitude estimation without phase estimation [12]. The results we present in this article are complementary to, and in some ways extend, these results. In particular, to our knowledge, this is the first time that a noise model specific to QAE has been derived, even though the aforementioned papers note the importance (and difficulty) of doing so (“It is indeed a difficult challenge to model a noise effect” [31, Sect. 5]; “… any lack of accuracy in one’s noise characterization will translate to a lack of accuracy in amplitude estimation. Understanding better this difficulty represents a fruitful line of future research” [30, Sect. IV-A]). Moreover, a noise model is a statistical characterization of some physical process and hence stands independently of any particular flavor of QAE and instead provides a quantification of the measurement uncertainty that can then be used to more accurately infer the amplitude from the measured data. To this end, our main example for how to use the proposed noise model to achieve noise-aware QAE concerns a quantification of the number of additional shots that are needed—given a maximal circuit depth at which it is possible to constrain a level of intrinsic uncertainty that arises in the model formulation—to achieve the same total mean square error (MSE) as in the noiseless case. This applies across the board to all QPE-free QAE algorithms that use repeated shots of the same circuit to infer the amplitude, including some suggestions for shallow-depth QAE [32], [33].

Looking more broadly at QPE-free QAE, Wang et al. [34] take a different approach and cast QAE as an instance of generic observable measurement. They take an adaptive Bayesian approach, showing that by tuning the circuit parameters adaptively as the parameters become known with greater confidence, in turn, the future expected information gain can be maximized. In future work, it will be interesting to see if, and to what extent, the bespoke noise model we propose here can further enhance the Bayesian approach to quantum sampling—however, this is beyond the scope of this article. In addition, there are instances of QAE algorithms that do not follow the structure of repeated shots of the same circuit, for example those that require single shots of many different circuits such as the recent algorithm by Akhalwaya et al. [35], which uses linear combination of unitary (LCU) operations to prepare initial states, leading to improved statistical robustness of the final amplitude estimate. It should be possible in future to adapt the methodology for these cases—however, this is also beyond the scope of this article. Finally, it is also worth noting that efforts are afoot to investigate the performance of QAE on noisy quantum computers, even without application-level handling of the noise [36].

### A. ARTICLE ORGANIZATION

The rest of this article is organized as follows. In Section II we describe in detail how the Gaussian noise model is derived from some reasonable initial assumptions, demonstrating that depolarizing noise is a special case of the Gaussian noise model and also that the Gaussian noise model can be easily composed with other noise channels to form extended models. In Section III, we present a series of experiments, run on actual quantum hardware, to probe the real-world applicability of our proposed noise model. Then, in Section IV, we include and discuss the results of these experiments. In Section V—with a view to proposing our noise-aware QAE algorithm later—we discuss a generalized procedure for collecting and analyzing additional QAE calibration data, which we will see is a key requirement for actually running noise-aware QAE. In Section VI, we then give the theory and general framework for this proposed noise-aware QAE algorithm. In Section VII, we give some further numerical results for the noise-aware QAE algorithm. Finally, Section VIII concludes this article.

### II. THEORETICAL DERIVATION OF THE GAUSSIAN NOISE MODEL

To derive the Gaussian noise model, it is first necessary to introduce QAE [37]: QAE uses a generalization of Grover’s search algorithm [38], amplitude amplification, to estimate the amplitude, \( a = \sin^2 \theta \), of a general \( n \)-qubit quantum state expressed in the form

\[ |\psi\rangle = \cos \theta |\Psi_0\rangle |0\rangle + \sin \theta |\Psi_1\rangle |1\rangle \] (1)

for some \( (n - 1) \)-qubit states \( |\Psi_0\rangle \) and \( |\Psi_1\rangle \). A circuit, \( A \), which prepares \( |\psi\rangle \), that is \( |\psi\rangle = A |0^n\rangle \), is taken as the input to the QAE algorithm, from which it is possible to build an operator \( Q = -A S_0 A^{-1} S_\chi \) (where \( S_0 = X \otimes (C_{n-1}) X \) and \( S_\chi = I_{2^{n-1}} \otimes Z \) do not depend on \( A \)), which performs Grover iteration

\[ Q^m |\psi\rangle = \cos((2m + 1) \theta) |\Psi_0\rangle |0\rangle + \sin((2m + 1) \theta) |\Psi_1\rangle |1\rangle \] (2)

Thus, for \( m \) applications of \( Q \), the probability of measuring the state \( |1\rangle \) on the last qubit is \( \sin^2((2m + 1) \theta) \), and the essential idea common to all QPE-free QAE algorithms is to run circuits for a variety of different values of \( m \) and then to use classical postprocessing to estimate \( \theta \) and hence \( a \). In the
presence of noise, each Grover iterate will not necessarily
enact a rotation of exactly $2\theta$, and thus, we can write the
actual rotation enacted by the $i$th Grover iterate as $2\theta + \epsilon_i$,
where $\epsilon_i$ is some error, which then gives

$$|\psi\rangle \xrightarrow{Q^n} \cos\left((2m + 1)\theta + \sum_{i=1}^{m} \epsilon_i\right) |\psi_0\rangle|0\rangle$$

$$+ \sin\left((2m + 1)\theta + \sum_{i=1}^{m} \epsilon_i\right) |\psi_1\rangle|1\rangle. \quad (3)$$

Note that (3) is a general expression representing the final state
of a QAE circuit (after $m$ Grover iterations) run on a
noisy machine that characterizes the effects of most sources
of noise: that is, by treating each Grover iteration to be a
rotation by the desired (noiseless) angle plus some additional
erroneous components. However, it is worth stating that there
are certain types of noise channel that one would not neces-
ecessarily expect to be modeled in this manner, and this
is discussed in detail in Section II-B. It is also pertinent that,
by treating the Grover iteration as a “closed box” in this way,
(3) is agnostic to the specific realization of the algorithm
as a quantum circuit—covering even cases where QEC is
deployed, for example. In order to turn this into a useful
noise model, it is necessary to introduce some assumptions,
namely that the various $\epsilon_i$ are independent and identically
distributed (i.i.d.), and importantly arise from noise effects
that are independent of the quantum state itself. In the case
of applications of $Q^n$ in quantum amplitude amplification,
the fact that the entire evolution of the state occurs in the
Bloch sphere spanning $|\psi_0\rangle|0\rangle$ and $|\psi_1\rangle|1\rangle$ means that this
independence is easy to visualize: the distribution of $\epsilon_i$ does
not depend on the point on the Bloch sphere that represents
the state. Then, if $m$ is sufficiently large, we can invoke the
central limit theorem (CLT) to approximate the sum of the
random variables $\epsilon_i$ as a single random variable drawn from a
Gaussian distribution. In this case, the state can be expressed as

$$Q^n|\psi\rangle = \cos((2m + 1)\theta + \theta_\epsilon)|\psi_0\rangle|0\rangle$$

$$+ \sin((2m + 1)\theta + \theta_\epsilon)|\psi_1\rangle|1\rangle \quad (4)$$

such that $\theta_\epsilon \sim \mathcal{N}(k_\mu m, k_\sigma m)$, where $\mathcal{N}$ (mean, variance) is
the normal distribution and $k_\mu$ and $k_\sigma$ are constants.

This noise model applies because we are only interested
in the error on a single qubit, and this allows us to use a
classically inspired approach and invoke the CLT. The result
is a noise model that is somewhat different in appearance to
that which may conventionally be derived for modeling noisy
quantum channels. In particular, we do not express a mixed
state, but rather notice that a mixed state simply captures
classical uncertainty about which pure state some quantum
system is in. Thus, our proposed Gaussian noise model can
be thought of as always treating the quantum state as pure and
then quantifying the relevant part of our uncertainty about
what that quantum state is separately.

It is worth discussing in a bit more detail the assumptions
made regarding the noise model.

First, the assumption that the various $\epsilon_i$ are i.i.d. is actu-
ally an unnecessarily strong requirement. In fact, all that is
required is that a sufficiently large number of independent
underlying “factors”\(^1\) contribute to the randomness that a
Gaussian random variable well models the value in ques-
tion. This assumption is identical to that ubiquitously used in
wireless communication literature to model noise as additive
white Gaussian noise [39].

Second, while the noise model assumes that noise effects
are independent of the quantum state itself, this assump-
tion does not generally hold for all generic noise channels.
We thus might not expect the model to be able to capture
such effects. However, “extended” noise models can be built
corresponding to composite random processes, such that ef-
fects that are not expected to be captured by Gaussian noise
can also be incorporated. Such a model is introduced in
Section II-B.

Finally, the Gaussian noise model relies on the assumption
that the Gaussian iterate circuit remains in place; that is, the
Gaussian noise model explicitly requires that the circuit
structure is a succession of Grover iterates. It is reasonable
to question whether this is a realistic assumption if the cir-
cuit is passed through a compiler that returns a functionally
equivalent shallower circuit. In this case, one may ostensibly
expect the required structure to be disturbed; however, a more
detailed look at the structure of QAE circuits reveals that
such an approach would still leave a repeated sequence of
identical (or near-identical) circuit blocks. This is because,
onece the number of qubits has exceeded that for which the
entire unitary could be resynthesized from scratch, all that
can be done is a peephole-type optimization. In this case,
the multicontrolled CZ gate that is central to the Grover
iterate circuit acts as a barrier to too much commutation
of other gates past these regularly recurring multicontrolled
CZ gates, and hence, even a rewritten circuit would have a
similar overall structure. Indeed, an alternative would simply
be to pass the subcircuit $Q$ to the compiler to obtain circuit
optimizations wholly contained therein while maintaining a
succession of strictly identical blocks.

A. DEPOLARIZING NOISE IS A SPECIAL CASE OF THE
GAUSSIAN NOISE MODEL

Of the existing generic noise models, our proposed Gaussian
noise model is most similar to the depolarizing noise model.
The depolarizing noise model supposes that each layer of
quantum gates has some probability of completely depolar-
izing the quantum state, whereas the Gaussian noise model
supposes that the state gradually depolarizes in a continuous
manner. It turns out that when $k_\mu = 0$, the depolarizing noise

\(^1\)When running a QAE circuit on a noisy machine, such “factors” can
include systematic effects such as hardware and calibration inaccuracies,
systematic biases in the quantum gates or operations themselves, or effects
such as crosstalk, temperature-induced errors, etc.
model and the Gaussian noise model are identical, as we now show.

To begin, consider the probability of the measurement outcome being zero in depolarizing noise, using the fact that the circuit depth is approximately proportional to $m$

$$p(0) = p_{\text{coh}}^m \cos^2((2m + 1)\theta) + \left(1 - p_{\text{coh}}^m\right) \frac{1}{2}$$ (5)

where $p_{\text{coh}}$ is some (fixed) finite probability of the state remaining coherent (not depolarizing) that scales with the number of Grover iterates. For comparison with the Gaussian noise model, it is easiest to express the probability of measuring 1 subtracted from the probability of measuring 0

$$p(0) - p(1) = \left(p_{\text{coh}}^m \cos^2((2m + 1)\theta) + \left(1 - p_{\text{coh}}^m\right) \frac{1}{2}\right) - \left(p_{\text{coh}}^m \sin^2((2m + 1)\theta) + \left(1 - p_{\text{coh}}^m\right) \frac{1}{2}\right)$$

$$= p_{\text{coh}}^m \cos^2((2m + 1)\theta) - \sin^2((2m + 1)\theta).$$ (6)

We now show that the same asymptotic behavior occurs in the proposed Gaussian noise model. To do so, first, we consider the probability of the measurement outcome being 0 when the noise is Gaussian, which can be expressed as

$$p(0) = \int_{-\infty}^{\infty} p(0|\theta_c) p(\theta_c) d\theta_c$$

$$= \int_{-\infty}^{\infty} \cos^2((2m + 1)\theta + \theta_c) e^{-\frac{(\theta - \theta_c)^2}{2\sigma^2}} d\theta_c.$$ (7)

We can similarly express the probability of outcome 1

$$p(1) = \int_{-\infty}^{\infty} \sin^2((2m + 1)\theta + \theta_c) e^{-\frac{(\theta - \theta_c)^2}{2\sigma^2}} d\theta_c.$$ (8)

Next, we subtract (8) from (7)

$$p(0) - p(1) = \int_{-\infty}^{\infty} \cos^2((2m + 1)\theta + \theta_c) e^{-\frac{(\theta - \theta_c)^2}{2\sigma^2}} d\theta_c - \sin^2((2m + 1)\theta + \theta_c) e^{-\frac{(\theta - \theta_c)^2}{2\sigma^2}} d\theta_c$$

$$= \frac{1}{\sqrt{2\pi \sigma^2}} \Re \left(e^{2i((2m + 1)\theta + 2i\gamma_m \sqrt{2\pi \sigma^2} \theta)} e^{-2k_\mu m}\right)$$

$$= e^{-2k_\mu m} \cos(2((2m + 1)\theta + k_\mu m))$$

$$= e^{-2k_\mu m} \left(\cos^2((2m + 1)\theta + k_\mu m) - \sin^2((2m + 1)\theta + k_\mu m)\right).$$ (9)

where $\Re$ denotes the real part, and the substitution $\bar{\theta}_e = \theta_e - k_\mu m$ is used to simplify the integral.

We can see that when $k_\mu = 0$ (which is effectively implicit in the depolarizing noise model), the depolarizing and Gaussian noise models have identical measurement statistics, that is, when the parameters are matched accordingly ($\bar{p}_{\text{coh}} \equiv e^{-2k_\mu}$), (6) and (9) are equivalent. Note that, as there are only two measurement outcomes, namely, 0 and 1, and probabilities sum to one, the measurement statistics are completely determined by the value of $p(0) - p(1)$.

This correspondence between the depolarizing and Gaussian noise models is important for a number of reasons. First, the fact that the Gaussian noise model we propose here corresponds to a widely used and accepted noise model (i.e., depolarizing noise) adds credence to its validity. Conversely, the fact that the Gaussian noise model has been derived from simple and justified physical assumptions gives a phenomenological (rather than ad hoc) basis to use the depolarizing noise model in appropriate circumstances.

B. COMPOSITE NOISE MODEL INCLUDING THE EFFECTS OF AMPLITUDE DAMPING

The proposed Gaussian noise model assumes that the effects of noise are independent of the quantum state itself. However, there exists (at least) one widely used and experimentally observed noise channel that violates this assumption of independence, namely, amplitude damping.

Amplitude damping assumes that the probability amplitudes associated with quantum states tend to decay over time due to interactions with an external environment. Since qubit states of a machine correspond to physical states of some system with given physical properties, amplitude damping will affect the probability amplitudes differentially depending on these properties. For instance, if qubit states are mapped to physical states of a quantum system that correspond to different energies, then the dynamics of noisy interactions with the environment from effects such as spontaneous emission will differ between the physical states. This will lead to an explicit dependence of the noise on the state of the quantum system itself.

Amplitude damping is characterized by a damping parameter $\gamma$, quantifying the rate at which the system loses coherence. Taking a very simplified example of a single-qubit system with qubit states $|0\rangle$ and $|1\rangle$ mapped to physical states such that the $|1\rangle$ state is mapped to a higher energy physical state—meaning that in this simple example, it is more likely to interact with the environment through effects such as spontaneous emission—we can model the amplitude
damping channel as affecting the amplitudes of the state such that each qubit state has a modified amplitude
\[ \tilde{p}(0) = p(0) + \gamma p(1) \]
\[ \tilde{p}(1) = (1 - \gamma)p(1). \]  
(10)

As this random process is not independent of the state, the proposed Gaussian noise model does not capture the error introduced by amplitude damping, which manifests for a single qubit as a bias to the measurement statistics in favor of one of the qubit states. However, it is possible to build a noise model corresponding to a composite random process that incorporates Gaussian noise and other effects not captured by Gaussian noise such as amplitude damping, leading to a more comprehensive model.

To build a composite noise model that captures both Gaussian noise and amplitude damping, we first note that for the Gaussian noise model, \( p(0) - p(1) \) is as given in (9) and also use the simple fact that \( p(0) + p(1) = 1 \) to obtain
\[ p(0) = \frac{1}{2} \left[ 1 + e^{-2k_\sigma m} \left( \cos^2((2m + 1)\theta + k_\mu m) - \sin^2((2m + 1)\theta + k_\mu m) \right) \right] \]
\[ p(1) = \frac{1}{2} \left[ 1 - e^{-2k_\sigma m} \left( \cos^2((2m + 1)\theta + k_\mu m) - \sin^2((2m + 1)\theta + k_\mu m) \right) \right]. \]  
(11)

Then, under the physical scenario in which the \( |1 \rangle \) state is more likely to interact with the environment than the \( |0 \rangle \) state, \(^2\) given (10) and defining \( \gamma = 1 - e^{-2k_\sigma m} \), we can compose the amplitude damping channel with Gaussian noise, such that the probability amplitudes of the state are then modified according to
\[ \tilde{p}(0) = \frac{1}{2} \left[ 1 + e^{-2k_\sigma m} \left( \cos^2((2m + 1)\theta + k_\mu m) - \sin^2((2m + 1)\theta + k_\mu m) \right) + (1 - e^{-2k_\sigma m}) \right] \times \left[ 1 - e^{-2k_\sigma m} \left( \cos^2((2m + 1)\theta + k_\mu m) - \sin^2((2m + 1)\theta + k_\mu m) \right) \right] \]
\[ \tilde{p}(1) = \frac{1}{2} e^{-2k_\sigma m} \left[ 1 - e^{-2k_\sigma m} \left( \cos^2((2m + 1)\theta + k_\mu m) - \sin^2((2m + 1)\theta + k_\mu m) \right) \right]. \]  
(12)

It is important to note here that by building such a composite model, we are implicitly making the assumption that the effects of the amplitude damping channel can be modeled as if it was a pure channel applied to the final state, i.e., to the state after the pure Gaussian noise channel has already been applied, that is, rather than amplitude damping being continuously interleaved with Gaussian noise throughout the evolution. Even though this does not perfectly capture the true physical dynamics, empirical results given later in Section IV demonstrate that such a model is appropriate.

In a similar manner, it is possible to rigorously compose the Gaussian noise model with any other independent noise channel (or several such channels), as long as one can also make the assumption that the effects of the noisy channels can be modeled as if occurring purely to the initial or final state i.e., before or after the Gaussian noise channel has been applied. One such example of additional effects not expected to be captured by Gaussian noise that can be modeled in this way is “state-preparation and measurement” (SPAM) errors. State-preparation errors arise from nonidealities in initializing quantum states, e.g., some qubits assumed to be initialized in the \( |0 \rangle \) state may not be, and measurement errors refer to imperfections in the readout process, e.g., where a final state, such as \( |1 \rangle \), is incorrectly measured as \( |0 \rangle \). However, the reason that the Gaussian model composed of the amplitude damping channel is studied in particular is based on preliminary studies, which demonstrated that amplitude damping effects were clearly observed in the data, whereas the contributions from other sources were observed to generally be negligible.\(^3\)

The ability to compose the Gaussian noise model with other channels means that arbitrary noise effects that are not well captured by Gaussian noise can be included in an extended model. This significantly increases the utility of the Gaussian noise model.

III. NOISE-MODEL EXPERIMENTAL SETUP

In order to gather experimental results to assess the real-world applicability of the proposed Gaussian noise model, it was necessary to select some state-preparation circuits, \( A \), and run QAE on real quantum hardware. A number of different state-preparation circuits were thus defined, and for each round of experiments, given subsets of these circuits were run (depending on the particular quantum hardware that was run and the dates that the experiments were carried out).

Generally, when designing the state-preparation circuits, we were guided by the following two principles: first, we minimized the number of qubits used in \( A \) in order that we could run deeper circuits (more applications of \( Q \)), which thus allowed us to better observe how well the Gaussian noise model fits the experimental data; second, we chose circuits that have the property that noiselessly \( \theta \) is such that on periodic numbers of Grover iterations, the measurement outcome is either \( |0 \rangle \) or \( |1 \rangle \) with certainty, which allowed us to display plots to illustrate how well the noise model fits the experimental data. Two circuits that achieved these aims are \( A_1 \) and \( A_2 \), shown in Fig. 1, which prepare the states
\[ A_1 |00 \rangle = \cos(\pi/6)|\Psi_0\rangle |0 \rangle + \sin(\pi/6)|\Psi_1\rangle |1 \rangle \]  
(13)
\[ A_2 |00 \rangle = \cos(\pi/3)|\Psi_0\rangle |0 \rangle + \sin(\pi/3)|\Psi_1\rangle |1 \rangle. \]  
(14)

\(^2\) Such a choice is, however, arbitrary, and in reality, the choice will depend on the particular qubit mapping of the hardware. If the physical setup is such that the \( |0 \rangle \) state is more likely to interact with the environment, then the qubit labels can be simply swapped in these expressions.

\(^3\) However, contribution from SPAM errors in particular may have been observed for some of the data (see Section IV).
Therefore, we can see that
\[
Q_m^1 A_1 |00⟩ = \cos((2m + 1)\pi/6)|Ψ_0⟩|0⟩ + \sin((2m + 1)\pi/6)|Ψ_1⟩|1⟩
\]
(15)
\[
Q_m^2 A_2 |00⟩ = \cos((2m + 1)\pi/3)|Ψ_0⟩|0⟩ + \sin((2m + 1)\pi/3)|Ψ_1⟩|1⟩
\]
(16)

where \(Q_1\) and \(Q_2\) are the Grover iterate circuits for \(A_1\) and \(A_2\), respectively. We can see that \(Q_m^1 A_1 |00⟩\) (respectively, \(Q_m^2 A_2 |00⟩\)) has the property that the measurement of the last qubit is 1 (respectively, 0) with certainty when \((m - 1) \mod 3 = 0\) (that is for \(m = 1, 4, 7, \ldots\)). Fig. 2 illustrates this for the case of \(A_1\). Even though, in principle, QAE could be

applied to a single-qubit circuit, \(A\), the lack of entanglement therein would make it an unsuitable experiment, and thus, by selecting two-qubit circuits, we have used the minimum number of qubits that we can reasonably expect to lead to meaningful results.

In addition, a three-qubit circuit, \(A_5\), was also defined, which prepares the state
\[
A_5 |000⟩ = \cos(\pi/6)|Ψ_0⟩|0⟩ + \sin(\pi/6)|Ψ_1⟩|1⟩
\]
(17)
\[
A_5\) has the same “periodic” property as \(A_1\), where noiselessly \(Q_m A_5 |000⟩\) yields measurement outcome 1 with certainty on the third qubit when \((m - 1) \mod 3 = 0\).

While these circuits achieved our aim of providing results that can readily be plotted, it was also beneficial to include a range of other circuits. To this end, we also ran QAE for \(A_1\) and \(A_2\) simultaneously (to introduce the possibility of crosstalk) and additionally defined the circuits \(A_3\) and \(A_4\), also shown in Fig. 1, which prepare the states
\[
A_3 |00⟩ = \cos(1/2)|Ψ_0⟩|0⟩ + \sin(1/2)|Ψ_1⟩|1⟩
\]
(18)
\[
A_4 |00⟩ = \cos(1)|Ψ_0⟩|0⟩ + \sin(1)|Ψ_1⟩|1⟩
\]
(19)

\(A_3\) and \(A_4\) are such that the final qubit does not align with the \(|0⟩\) or \(|1⟩\) axes after any number of applications of \(Q\).

Quantinuum’s TKET compiler [40] was used to run the circuits—the compiler was set to map the circuit to the physical qubit connectivity and native gateset but not to perform any “optimization,” that is circuit rewriting to achieve a functionally equivalent but shallow-depth circuit (compiled circuits for all rounds of experiments are shown in Appendix A). In this way, we guaranteed that the circuit executed consisted of the appropriate number of repeated Grover iterations, as was necessary to probe the validity of the proposed noise model.

We used minimum mean square error (MMSE) parameter fitting [41], pp. 344–350] based on the least-squares method to find the noise-model parameters that best fit the experimental data. That is, each noise model can be expressed as

---

**FIGURE 1.** State-preparation circuits.

**FIGURE 2.** Illustration of the action of \(Q_m\) on \(|ψ⟩ = A_1|00⟩\). Here, we show for \(m = 0, \ldots, 4\) but we can easily see that the pattern will repeat after \(m = 6\) (i.e., so that we can take \(m\) modulo 6 to find the rotation angle).
a parameterized function (i.e., measurement outcome probability as a function of number of Grover iterates), and hence, the specific instance of the set of parameters of a given function that minimizes the mean square discrepancy between the experimental values and the values predicted by the noise model was selected as the “best fit.” In the fits, we accounted for the statistical uncertainties in the data that arise from the finite number of shots used to estimate the amplitudes.

For each experiment, we fitted three different models: the Gaussian noise model proposed herein; the depolarizing noise model (which is equivalent to the Gaussian noise model with $k_{ij}$ set to zero, as shown in Section II-A); and the Gaussian noise model composed with the amplitude damping channel discussed in Section II-B (where the qubit state that is physically more likely to interact with the environment is specified depending on the particular experiment).

For the first round of experiments, we used four of IBM’s five-qubit machines (Athens, Bogota, Rome, and Santiago—all of the five-qubit machines online at the time of the experiments), each of which uses transmon superconducting qubits connected in a line [42]. For IBM transmon qubits, in the amplitude damping model, the $|0\rangle$ state is mapped to the physical ground state of the system, and the $|1\rangle$ state is mapped to the physical excited state. We ran all of the two-qubit QAE circuits for $m$ applications of $Q$ where $m = 0, \ldots, 67$. This range was set on the basis of preliminary studies to discover the reasonable maximum circuit depths (i.e., before noise becomes overwhelming). We ran each circuit for the maximum number of shots allowed at the time, which was 8192.

For the second round of experiments, we ran shots of the circuit $A_1$ on three of IBM’s five-qubit machines online at the time of the experiments (Lagos, Nairobi, and Perth), each of which also uses superconducting qubits, but these are instead laid out in the shape of a sideways “H” [42]. We ran 20,000 shots of each circuit—the maximum number of shots allowed at the time—and the circuit depths probed were greater than in the first round of IBM experiments, with depths $m = 0, \ldots, 80$.

Finally, for the third and fourth rounds of experiments, we used Quantumium’s H1 trapped-ion quantum computer [43]. For Quantumium’s ionic qubits, in the amplitude damping model, the $|0\rangle$ state is mapped to a physical state of the system that is more likely to interact with the environment, and the $|1\rangle$ state is mapped to a physical state of the system that is less likely to interact with the environment. Because of the slower gate times and more restricted device availability, we were only able to run a smaller number of shots than when using the IBM devices. However, because of the world-leading (at the time of writing) quantum volume of the Quantumium machine [44], we were able to confidently run a three-qubit circuit, $A_5$, also shown in Fig. 1, which prepares the state given in (17).

For the third round of experiments, circuits corresponding to $m = 0, 1, \ldots, 12$ were run, and for each circuit, 1024 shots were performed. This is denoted “run 1.” On a different day, further experiments were run for $m = 0, 1, \ldots, 13$, and for each circuit, 1500 shots were performed. This is denoted “run 2.”

For the fourth round of experiments, the circuit depths probed were much greater, corresponding to $m = 0, \ldots, 80$, and 1000 shots of each circuit were performed. This is denoted “run 3.”

The first and second rounds of experiments using IBM hardware were conducted in December 2020, January 2021, and October 2023, respectively, and the third and fourth rounds of experiments using Quantumium’s H1 hardware were conducted in June and July 2021, and August and September 2023, respectively.

### IV. NOISE-MODEL RESULTS AND DISCUSSION

From the MMSE-fitted parameters, we calculate $R^2$ values to evaluate the goodness of fit of the three noise models under consideration. These $R^2$ values are given in Table 1—note that $R^2$ is at most equal to 1 and $R^2 = 1$ means that all of

| Circuit | Machine | Gauss | Dep | Gauss AD |
|---------|---------|-------|-----|----------|
| $A_1$   | Athens  | 0.9563 | 0.8137 | 0.9563 |
|         | Bogota  | 0.9419 | 0.5958 | 0.9419 |
|         | Rome    | 0.9690 | 0.8772 | 0.9690 |
|         | Santiago| 0.8994 | 0.8783 | 0.8994 |
|         | Lagos   | 0.9971 | 0.4663 | 0.9971 |
|         | Nairobi | 0.9887 | 0.4900 | 0.9943 |
|         | Perth   | 0.9682 | 0.5784 | 0.9817 |
| $A_2$   | Athens  | 0.8856 | 0.7801 | 0.8891 |
|         | Bogota  | 0.8692 | 0.7958 | 0.8692 |
|         | Rome    | 0.6732 | 0.6404 | 0.6732 |
|         | Santiago| 0.8975 | 0.8972 | 0.8975 |
| $A_1'$  | Athens  | 0.9132 | 0.8450 | 0.9132 |
|         | Bogota  | 0.7448 | 0.6870 | 0.7448 |
|         | Rome    | 0.8241 | 0.8224 | 0.8241 |
|         | Santiago| 0.8026 | 0.7839 | 0.8026 |
| $A_2'$  | Athens  | 0.9993 | 0.9838 | 0.9993 |
|         | Bogota  | 0.7817 | 0.6854 | 0.7998 |
|         | Rome    | 0.8973 | 0.6421 | 0.8973 |
|         | Santiago| 0.9228 | 0.8773 | 0.9228 |
| $A_3$   | Athens  | 0.8890 | 0.8550 | 0.8890 |
|         | Bogota  | 0.8676 | 0.7980 | 0.8676 |
|         | Rome    | 0.8040 | 0.7301 | 0.8040 |
|         | Santiago| 0.8648 | 0.8217 | 0.8648 |
| $A_5$   | H1 (Run 1) | 0.9859 | 0.9854 | 0.9928 |
|         | H1 (Run 2) | 0.9716 | 0.9297 | 0.9889 |
|         | H1 (Run 3) | 0.9060 | 0.7920 | 0.9613 |

$^1$Gauss stands for ‘Gaussian’, Dep for ‘depolarizing’ and ‘Gauss AD’ for ‘Gaussian with amplitude damping’. $A_1$ and $A_2$ denote the results from $A_1$ and $A_2$, respectively, when $A_1$ and $A_2$ were run simultaneously.

### TABLE 1. $R^2$ Values for the Various State-Preparation Circuits Running on IBM Superconducting Quantum Computers and Quantumium’s H1 Trapped-Ion Quantum Computer

- Herbert et al.: NOISE-AWARE QAE
- IEEE Transactions on Quantum Engineering
- VOLUME 5, 2024
- 3103623
- Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
the experimentally observed variance is explained by the proposed model. In addition, while the key quantities for assessing how well the theoretical noise models fit the experimental data are indeed the $R^2$ values, the values of the parameters themselves may also be of general interest, and these are included in Table 2 in the appendixes.

Fig. 3 shows plots for $A_1$ run on the various IBM machines, with circuit depths $m = 1, 4, 7, \ldots$ (i.e., number of Grover iterates such that the noiseless measurement outcome is always equal to 1), and Fig. 4 shows plots for $A_1$ run on the various machines, with circuit depths $m = 0, 2, 3, 5, 6, 8, 9, \ldots$ (i.e., number of Grover iterates such that the noiseless measurement outcome is equal to zero with probability 0.75).

From these results, it is immediately apparent that the proposed Gaussian noise model (with variable mean, and both with and without amplitude damping) explains the experimental data better than the depolarizing noise model (Gaussian noise model with mean fixed at zero)—in almost all cases having (sometimes considerably) higher $R^2$ values. The fact that Gaussian noise model better fits the data can be explained by the fact that the Gaussian noise model with parameterized mean can capture a constant “bias” or “drift,” where each Grover iterate performs a rotation that is offset from the theoretical rotation angle by some constant amount (plus some zero-mean random noise). The results for rotation angles $\pi/6$ or $\pi/3$ (as in $A_1$, $A_2$, and $A_3$) clearly illustrate this, as the constant offset is manifested as the probability of measuring zero (or one) being alternately higher and lower (as the number of Grover iterations is incremented) than the simple depolarizing case, as shown in Fig. 4 (specifically, Fig. 4 illustrates that noiselessly for all $m$ such that $(m-1) \mod 3 \neq 0$, the probability of measurement outcome being $|0\rangle$ is 0.75). To see an example of this effect, consider the case of a small positive offset to the rotations in Fig. 2. In the case of $Q^2$, this offset would move the superposition closer to the direction $|0\rangle$, and so there would a corresponding increase in the probability of measuring 0; conversely, in the case of $Q^3$, the offset would move the superposition closer...
to the direction $-|1\rangle$, and so there would be a corresponding increase in the probability of measuring 1. Thus, we can see the grounds for the oscillation observed in Fig. 4.

It is worth remarking upon the fact that the presence of this oscillatory behavior demonstrates a deficiency of the depolarizing noise model in certain circumstances. Whereas the depolarizing model aims to capture the overall degradation of the quantum state from a variety of physical effects (as is also our motivation in proposing the Gaussian noise model), it is clearly the case that “biased” behavior in the state degradation cannot be captured. That this behavior is evident in both the IBM machines and when the larger three-qubit circuit, $A_5$, is run on Quantumuin’s machine—machines with fundamentally different physical manifestations of quantum computation—highlights the necessity of the Gaussian noise model. Most strikingly, for the circuit $A_1$ run on the newer IBM machines (Lagos, Nairobi, and Perth), as well as on Bogota, the Gaussian noise models (with variable mean) have very high $R^2$ values—thus explaining almost all of the experimental data—whereas the equivalent fits for the depolarizing noise model have much lower $R^2$ values. This contrast demonstrates clearly the effect of this constant bias. A similar contrast between the fit results (though the difference is not as large) is seen for the fit to the Run 3 data from the circuit $A_5$ run on H1, demonstrated in Fig. 5; this is not surprising given the fact that $A_1$ and $A_5$ have identical measurement statistics. It should also be noted that, while less amenable to visualization, the same principle explains the better fit of the Gaussian noise model with parameterized mean for $A_3$ and $A_4$.

Comparing the results for the Gaussian noise model and the Gaussian noise model composed of the amplitude damping channel, we see that there is a clear contrast between the results for the older IBM machines and the newer IBM machines, alongside Quantumuin’s machine. In the case of the older IBM machines, the fits suggest that the effects of amplitude damping can be generally considered negligible, as in the majority of cases, the parameter $k_{AD}$ is found to be negligible, indicating no significant amplitude damping effect observed in the data. By contrast, for the newer IBM machines and the Quantumuin machine, the Gaussian noise model with amplitude damping is a better fit in all cases except Lagos, as the $R^2$ results are always greater (significantly so for Quantumuin Run 3).
The fact that the effects of the constant bias and effects, such as amplitude damping, seem to be more significant for the newer IBM machines and for Quantinuum H1 could plausibly be explained by considering the improved quantum volume of the newer IBM machines as compared to the older generation and of the world-leading (at the time of writing) quantum volume of the Quantinuum machine. While quantum volume is not the only metric that quantum computing manufacturers strive to improve, there has generally been a strong focus on increasing quantum volume, which has been shown to mostly scale as function of the total gate-error magnitude [45]. Because the combined effect of the dominant sources of noise such as the effect of applying entangling gates will thus have the strongest impact on quantum volume and will thus be more suppressed on these machines, then subdominant effects will be clearer to distinguish in the data, i.e., they can give a nonnegligible contribution to the overall bias. For the machine with the best quantum volume (H1), this is clear to see as both the Gaussian noise model and the amplitude damping component considerably improve the fit. By contrast, in the case of the older IBM machines, because the effects of the dominant sources of noise are more significant, then it is likely that the subdominant effects cannot be distinguished from the total noise, i.e., they will be comparatively negligible. Specifically, modeling effects such as amplitude damping in this case would appear to be unnecessary. However, the Gaussian noise model still provides a better fit for these machines, suggesting that the “grouped effects” that are reasonably well captured by Gaussian noise are significant in this case.

To further analyze the results, there is merit in discussing the goodness of fit of the proposed Gaussian noise model from both a phenomenological and operational point of view. Regarding the former, the predictions of the noise model clearly do not explain all of the variation observed in the experimental data (that would correspond to $R^2 = 1$ for all experiments), and thus, we can conclude that there are likely underlying physical causes of the variation that are not captured by the model.

It is worth remarking that the fact that the proposed Gaussian noise model (with and without amplitude damping) does not capture everything is to be expected, as the experiments were explicitly designed to capture raw data from the quantum hardware and so do not even accommodate the correction of other effects such as SPAM errors, for instance. Indeed, Fig. 6 shows the two occasions when the $R^2$ value is particularly low, and the SPAM error is a plausible explanation for each. In the case of $A_2$ on Rome, we can see that the experimental data have a bias toward measuring the $|1\rangle$ state as the number of Grover iterations grows large (which is the opposite effect to what one would expect from amplitude damping). One possible explanation for this (there could be others) is that there is a readout error, where (in this case) a bit-flip from 0 to 1 is more likely than the converse—which can be thought of as a type of SPAM error. In the case of $A_1\prime$ on Rome, even for a shallow circuit that does little more than simply prepare and measure a state, we have that $p(1) = 0.7$ rather than $\approx 1$, so clearly—indeed, almost by definition—there is a SPAM error. As discussed previously, however, it is in principle possible to compose any number of physically motivated noise channels (i.e., not just amplitude damping) with the Gaussian noise model in order to be able to capture additional effects such as SPAM, thus making the Gaussian noise model a very versatile and useful tool.

We now turn to the operational benefits of the Gaussian noise model, which are twofold. First, we note that the Gaussian noise model is parameterized in a way that can readily be used in the application—as expounded on in detail in the following sections. This corresponds to our original motivation to model the noise such that it can be handled at the application level. Second, the fact that the proposed noise model does capture the bias is important for calibration, which turns out to be a necessary requirement for running QAE.
V. NOISE-AWARE QAE: CALIBRATION

For the rest of this article, we only discuss the Gaussian noise model without amplitude damping included, for simplicity. Turning our attention to how the Gaussian noise model can be used to improve amplitude estimates, we must first deal with the fact that when running QAE, an unknown constant bias (i.e., an unknown \( k_\mu \)) in the result cannot be easily observed. To see this, consider the simplest case in which circuits \( A(0^\mu) \) are used to prepare \( |\psi\rangle = \cos \theta |\Psi_0\rangle |0\rangle + \sin \theta |\Psi_1\rangle |1\rangle \), in the ideal case, but in practice actually prepare \( \cos(\theta + \theta_c)|\Psi_0\rangle |0\rangle + \sin(\theta + \theta_c)|\Psi_1\rangle |1\rangle \) for some nonzero constant \( \theta_c \). In this case, averaging a large number of samples will return an estimate converging on \( \theta + \theta_c \), i.e., there is no way to distinguish the actual value, \( \theta \), from the bias, \( \theta_c \). Of course, if \( \theta_c \) is nonzero, but known, then it is trivial to subtract this from the averaged value and obtain an unbiased estimate.

A similar principle concerns the effect of \( k_\rho \) in the QAE circuits. In particular, if one considers the simplest scenario in which there is only an unknown bias, i.e., \( k_\rho = 0 \), then according to the Gaussian noise model, the prepared states are of the form: \( \cos((2m + 1)\theta + k_\mu m)|\Psi_0\rangle |0\rangle + \sin((2m + 1)\theta + k_\mu m)|\Psi_1\rangle |1\rangle = \cos((2\theta + k_\mu m + 1)|\Psi_0\rangle |0\rangle + \sin((2\theta + k_\mu m + 1)|\Psi_1\rangle |1\rangle \). For large \( m \), as are needed for QAE to deliver a quadratic advantage, the total rotation angle is, therefore, dominated by \( (2\theta + k_\mu m) \) and so there is again no way to distinguish \( \theta \) and \( k_\mu \) from samples thereof. This is clearly an issue as such biases (i.e., nonzero values of \( k_\mu \)) are observed to occur in data, as demonstrated in the previous section, and the magnitude of these biases cannot be known a priori for a given circuit; this can, therefore, significantly affect the accuracy of the final estimates output by QAE. However, all is not lost—the Gaussian noise model with parameterized mean is able to capture the bias via the parameter \( k_\mu \). This feature allows us to propose a general methodology for running QAE while also accounting for this bias, but this requires making use of additional calibration information.

We now describe a procedure for collecting and analyzing this calibration data. In Section VI, we discuss how this can then be exploited when running QAE. The basic idea is that, given a QAE circuit \( A \)—whose amplitude is to be estimated by running QAE on a given machine—one can construct a set of calibration circuits \( \{A'\}_\rho \), where the calibration circuits have similar structures to \( A \) but prepare states with known amplitudes. In particular, the calibration circuits should have the same number of two-qubit gates as \( A \) (and preferably similar placements of these gates within the circuits), as the assumption is that the error of a final state will be dominated by the effect of entangling gates, and therefore, similar-structured circuits containing the same number of two-qubit gates should have similar error profiles to the desired circuit.

The numerical results in Section VII suggest that the calibration technique discussed is effective for the two-qubit circuit used; however, as hardware matures to enable larger scale noise-aware QAE experiments to be conducted, an important question is how to conduct such calibration in general. The principle that we have used here is that the circuit structure—i.e., number and relative positions of the gates—is the most significant factor determining the values (or range of possible values) of \( k_\mu \) and \( k_\rho \). If this holds for larger circuits, then a suitable strategy will be to take the circuit \( A \), assumed to be compiled into a standard gateset of single-qubit gates and CNOTs, and to replace each single-qubit gate with a Hadamard or \( S \) gate (say chosen at random). In this way, one will obtain a stabilizer circuit with the same structure, and hence, it will be possible to obtain the ground truth of \( \theta \) using standard stabilizer simulation techniques and hence to achieve calibration by comparing this to numerical values obtained by running the stabilizer circuits (thus essentially following the same template as set out herein).

A sample of calibration data can be collected by running repeated shots of these calibration circuits on the given hardware—with the circuit submission preferably spaced out across a period of time that is sufficiently long as to maximize the variation in the noise captured by the calibration sample.\(^6\)

\(^6\)In particular, running the same circuit at different time periods is useful as it allows one to characterize the variation in the noise of the machine purely as a function of time.
By fitting the Gaussian noise model to the calibration data, for example by using MMSE parameter fitting as in the previous section, one can extract distributions of $k_\mu$ and $k_\sigma$ values that describe the variation in the noise for a given machine.

In particular, the distribution of $k_\mu$ values for the particular circuit run on the particular hardware can then be directly exploited for running QAE at the application level; for example, empirical bounds on the expected maximum size of $k_\mu$ can be constructed based on the distribution, and this bound can then be propagated into the Gaussian noise model, or instead, the distribution itself can be used directly at the application level. In this article, we discuss both examples.

Additional experimental studies were carried out during November and December 2023 to explicitly demonstrate the methodology and also explore the validity of the calibration protocol. These were based on running two-qubit circuits of the same structure as $A_1$-$A_4$, defined previously, using an IBM five-qubit machine online at the time (Perth). First, we note that all of these circuits prepare states of the form

$$A'(00) = \cos(\phi/2)|\psi_0\rangle |0\rangle + \sin(\phi/2)|\psi_1\rangle |1\rangle.$$  \hspace{1cm} (20)

Based on this, a “template” calibration circuit $A'(\phi)$ is defined, as shown in Fig. 7. This template circuit defines the set of calibration circuits $\{A’\}$, which was obtained by randomly sampling 50 different values for the angle $\phi$ in the interval $[0, 4\pi]$ and defining a corresponding calibration circuit based on each sampled value. A total of 20,000 shots of each circuit were run for $m$ applications of $Q$, where $m = 0, \ldots, 40$. We ran all 50 circuits 100 different times, i.e., with the intention of running the same circuits at different time periods in order to probe the variation of the noise parameters for a given circuit. However, due to the fair-use queuing system that exists for submitting circuits to IBM hardware, in practice, it was impossible to schedule exactly when a given circuit would run, and occasionally, a significant fraction of scheduled jobs for particular circuits that were intended to be run over different periods were found to run all at once, i.e., sequentially. Because the data from these circuits were not particularly useful for characterizing the variation in $k_\mu$, we chose to omit all samples from circuits where greater than 50% of jobs were run sequentially from the subsequent analysis. In addition, for some circuits, some scheduled jobs did not complete, and therefore, for certain samples, the total number of repetitions ended up being less than the 100 requested.\footnote{Unfortunately, it was not possible to submit replacement jobs or re-submit failed jobs because IBM retired these machines shortly after these studies were carried out.} \footnote{This left 42 different circuits to analyze with a variable number of repetitions: in total, 3724 independent data points. For the remaining samples, MMSE fitting was again used to find the Gaussian noise-model parameters (i.e., $k_\mu$ and $k_\sigma$ values as $\theta$ was fixed in the fit to the $\phi/2$ value of the particular calibration circuit) that best fit each data sample.}

The distribution of $k_\mu$ values across all circuits and all repeated runs is given in Fig. 8. The distribution of $k_\mu$ values grouped by the calibration circuit and grouped by the particular 24-h period a circuit was run is given in Figs. 9 and 10, respectively.

We first focus on the distribution of values across all circuits (i.e., all of the data combined), which can be used to constrain $k_\mu$ for the given circuit structure and hardware. From this distribution, the average value of $k_\mu$ can be seen to be small (sample mean $\mu_{bias} = 0.013$), and the variance is also fairly small (sample standard deviation $\sigma_{bias} = 0.169$)—however, clearly, there is a reasonable amount of variation in the parameter.

In principle, one can use this distribution to determine a bound on the expected maximum absolute size of $k_\mu$ for the particular circuit run on the particular hardware, and then, this can be propagated directly into noise-aware QAE. As is usual when handling statistical quantities, there are various ways to use the calibration data for $k_\mu$, and the most appropriate one depends on the overall goals of application. For example, if one requires a high level of confidence that any possible bias arising from nonzero $k_\mu$ is always accounted for, then a strict bound should be set; however, as described...
in Section VI, this typically leads to a large number of additional shots. Assuming that for some application we require a high confidence, we could directly use the largest absolute magnitude, corresponding to

\[ |k_{\mu}^{(\text{strict})}| \leq 0.549 \]  \hspace{1cm} (21)

to set a strict upper bound.

Conversely, if a lower confidence is acceptable for the particular application, then a looser bound can be set, with correspondingly fewer additional shots. In particular, we could set a loose upper bound based on the spread around the mean; given that the sample mean and sample standard deviation
for 3724 samples will well estimate the population mean and standard deviation, we could set
\[ |k^{\text{(loose)}}_\mu| \leq \max(|\mu \pm \sigma|) = 0.182. \] (22)

Thus, if the priority is to aggressively minimize the total number of samples, then setting a loose bound may be sensible, whereas if the priority is to be highly confident in the precision of a result, then a stricter bound may be sensible. It is worth noting that in Section VI, we also discuss a different method for constraining \( k_\mu \) when running QAE at the application level that does not rely on setting a bound and instead uses the distribution directly in a completely data-driven approach.

Moving on to discuss the distributions grouped by the calibration circuit, we see generally that there is a fairly significant variation in the noise profile for a given circuit; this demonstrates that the particular running conditions of the machine have a significant effect on the noise introduced to a given computation. However, it is worth noting that across all circuits, the overall variance in these distributions is reasonably small (the largest difference between the largest and smallest value in a sample (range) is 0.665 and largest standard deviation is 0.181), and also that the average \( k_\mu \) values themselves are generally small (the largest absolute average is 0.354).

Finally, considering the distributions grouped by 24-h running period, we see again that there is a fairly significant variation in the noise profile across each period when running different circuits. This, in contrast to the previous figure, demonstrates that the particular circuits that run on a given machine also have a significant effect on the noise introduced. Although here again we see that the overall variances are generally small (the largest range is 1.052 and largest standard deviation is 0.701), they are larger than those in the previous case. Here again, the average \( k_\mu \) values themselves are generally small (the largest absolute average is 0.301).

Overall, the grouped distributions demonstrate that the effects of both running conditions and particular circuit structure are important factors for determining the noise, and thus, by varying both across the total calibration sample, we can be confident that our sample is representative of the effects of noise and can, therefore, be used to characterize \( k_\mu \).

In addition, to compare the relative sizes of the parameters \( k^2_\mu \) and \( k_\sigma \)—which is important for considering the total MSE of the estimate when running QAE, as discussed in Section VI—the distribution of the ratio \( k^2_\mu/k_\sigma \) for all circuits is given in Fig. 11. Across all samples, this ratio has an average value of 0.069, which demonstrates that the value of \( k^2_\mu \) is typically smaller than the value of \( k_\sigma \) by an order of magnitude. The full range of ratios varies from approximately 10^{-10} to 10^2. However, importantly, only \( 1\% \) of samples have a ratio that is an order of magnitude or more greater than the average, and only \( 0.5\% \) of samples have the ratio being greater than or equal to unity.

Based on these findings, it is reasonable to assume, in what follows, that for circuits of the form \( A_1–A_4 \) run on IBM Perth, that in general \( k^2_\mu \) and \( k_\sigma \) differ by around an order magnitude, and that \( k^2_\mu \) is smaller than \( k_\sigma \). This fact will be significant in Section VI.

**VI. NOISE-AWARE QAE: THEORY**

We now show how the Gaussian noise model can be used to derive a rule for the necessary factor increase in the number of shots required to counteract the noise at a certain circuit depth, and how it can be used to perform Bayesian inference to estimate \( \theta \). To do this, we assume that, even at the maximum circuit depth, the error \( \theta_e \) is a “small angle,” that we have collected calibration data for \( k_\mu \) for the particular circuit and particular hardware we aim to run based on the methodology outlined in Section V, that this information can be used to construct a strong initial prior probability distribution on \( k_\mu \), and that \( k_\sigma \) is approximately known from device characterization—here, we merely require there to be a weak prior on \( k_\sigma \).

Consider that the measurement outcome for a single shot of a circuit with some \( m \) Grover iterations is Bernoulli distributed with parameter \( \sin^2((2m+1)\theta + \theta_e) \); using the aforementioned small angle assumption, we get

\[
\sin^2((2m+1)\theta + \theta_e) = \frac{1 - \cos(2((2m+1)\theta + \theta_e))}{2} = \frac{1}{2} \left(1 - \cos(2(2m+1)\theta) \cos(2\theta_e) + \sin(2(2m+1)\theta) \sin(2\theta_e)\right) \\
\approx \frac{1}{2} \left(1 - \cos(2(2m+1)\theta) + 2\theta_e \sin(2(2m+1)\theta)\right) = \sin^2((2m+1)\theta) + \theta_e \sin(2(2m+1)\theta). \] (23)
Next, we note that, by the Gaussian noise model, \( \theta \) is normally distributed, with mean \( k_{\mu} m \) (where \( k_{\mu} \) has been characterized based on the previous assumption of calibration) and variance \( k_{\sigma} m \); because \( \sin(2(2m + 1)\theta) \) is a constant whose magnitude is at most equal to 1, it therefore follows that \( \theta \sin(2(2m + 1)\theta) \) is also normally distributed

\[
\theta \sin(2(2m + 1)\theta) \sim \mathcal{N}(\mu, \sigma^2)
\]

(24)

where \( \mu \leq k_{\mu} m \) and \( \sigma^2 \leq k_{\sigma} m \).

Next, we let \( \alpha_m = \sin^2((2m + 1)\theta) \) be the amplitude of the qubit after \( m \) Grover iterates, and let \( N_m \) be the number of shots of this circuit. There are various ways in which the measurement outcomes for all of the different values of \( m \) can be combined to infer \( \theta \) (and hence \( \alpha \)), and therefore, to keep things general, here, we simply assume that the objective is to use the \( N_m \) shots to provide a point estimate of \( \alpha_m \), which we denote \( \hat{\alpha}_m \). Taking this approach allows us to determine the increase in the number of shots [i.e., (30)] to counteract the noise, and the numerical results in Section VII show that it does indeed yield performance improvements. We use the maximum likelihood estimate of \( \alpha_m \), which is simply the mean of the \( N_m \) samples; using a Gaussian approximation of the binomial distribution, we get that our estimate of the amplitude is normally distributed according to

\[
\hat{\alpha}_m \sim \mathcal{N}\left(\alpha_m + \frac{1}{N_m} \sum_{i=1}^{N_m} \theta^{(i)} \sin(2(2m + 1)\theta), \sigma^2\right)
\]

(25)

where \( \sigma^2 \leq 1/(4N_m) \) using the fact that a Bernoulli random variable has variance at most one quarter. However, when the mean of a Gaussian is itself normally distributed, it is easy to express the resultant distribution

\[
\hat{\alpha}_m \sim \mathcal{N}\left(\alpha_m + \mu, \frac{1}{N_m} \sigma^2 + \mu^2\right).
\]

(26)

It is worth further remarking that the sum in (25) can be thought of as further loop of summation of the errors, \( \epsilon \), associated with each Grover iterate [as in (3)] and hence “compounds” the use of the CLT, making the assumption of Gaussianity more plausible when the underlying noise model is used in this way.

The MSE of the estimate for \( \theta_m = (2m + 1)\theta \) is

\[
\text{MSE}(\hat{\theta}_m) = \text{Var}(\hat{\theta}_m) + \text{Bias}(\hat{\theta}_m, \theta_m)^2 = k_{\sigma} m + k_{\mu}^2 m^2.
\]

(27)

If \( k_{\mu} \) is exactly known a priori, then it is trivial to adjust the estimator to account for the bias, and hence, the term corresponding to the bias can be set to zero (i.e., in this case \( \text{MSE}(\hat{\theta}_m) = k_{\sigma} m \)). Letting \( N_{\text{shot}} \) be the number of shots that would have been selected in the noiseless case (where \( k_{\sigma} = 0 \) and \( k_{\mu} = 0 \)), we can express the factor increase in the number of shots required to obtain a total MSE equal to the noiseless worst case variance for the estimate of \( \alpha_m \). In particular, \( N_m \) is the number of shots now required when set such that the

MSEs for the noisy and noiseless cases are equal

\[
\frac{1}{4N_{\text{shot}}} = \frac{4k_{\sigma} m + 1}{4N_m} + k_{\mu}^2 m^2
\]

(28)

when \( k_{\mu} \) is unknown, and

\[
\frac{1}{4N_{\text{shot}}} = \frac{4k_{\sigma} m + 1}{4N_m}
\]

(29)

when \( k_{\mu} \) is known, giving

\[
\begin{align*}
N_m &= \frac{4k_{\sigma} m + 1}{4N_{\text{shot}}} , & \text{if } k_{\mu} \text{ is unknown} \\
N_m &= (4k_{\sigma} m + 1)N_{\text{shot}}, & \text{if } k_{\mu} \text{ is known.}
\end{align*}
\]

(30)

From this, the need for the calibration procedure to give some indication of the value of \( k_{\mu} \) is immediately obvious, as the case of “unknown” \( k_{\mu} \) has \( N_m \) depending on \( k_{\mu} \). Note also that when \( k_{\mu} \) is unknown, the first equation for \( N_m \) only has positive solutions up to a particular value of \( m \) (where this value depends on the exact values of \( N_{\text{shot}}, k_{\mu}^2 \), and \( k_{\sigma} \)). This means that after some point, it is not possible to obtain a total MSE equal to the worst case noiseless variance, regardless of how many additional shots are performed. This follows from the fact that at some value of \( m \), the component of the total MSE introduced by the bias becomes dominant over the component introduced by the variance due to the former growing quadratically faster as a function of \( m \). As the bias component cannot be suppressed by increasing the number of shots, noise-aware QAE is only effective up to some particular value of \( m \).

Turning to the specific case considered in the calibration procedure of Section V, namely circuits of the form \( A_1 \rightarrow A_4 \) run on IBM Perth, it was experimentally found that the average ratio of \( k_{\mu}^2/k_{\sigma} \) is 0.069. From this, (27) can be used to determine the value of \( m \) at which the MSE resulting from the bias begins to dominate by equating the two terms on the right-hand side and solving for \( m \), i.e.,

\[
k_{\sigma} m = k_{\mu}^2 m^2 \Rightarrow m = \frac{k_{\sigma}}{k_{\mu}^2} = \left( \frac{k_{\sigma}^2}{k_{\mu}} \right)^{-1}
\]

(31)

which, for the average value of \( k_{\mu}^2/k_{\sigma} = 0.069 \), gives \( m = 14 \). This is confirmed by plotting the average behavior of the MSE alongside the individual components arising from the bias and variance in Fig. 12, from which it can indeed be seen that the contribution to the MSE arising from the bias starts to become significant at approximately \( m = 14 \). This then dictates the approximate regime of validity of our protocol, i.e., by increasing the number of shots, we can on average expect the total MSE of the estimate to be able to be constrained to be equal to the worst case noiseless case (for circuits of this structure when run on IBM Perth).

In general, as preempted in Section V, it is unreasonable to suppose that \( k_{\mu} \) and \( k_{\sigma} \) will generally be exactly known a priori, and hence, these must be coestimated alongside \( \theta \). To do so, a prior (in the Bayesian sense) on each is required,
and for the reasons outlined in the opening of Section V, the prior on \( k_\mu \) must be relatively strong, to account for the fact that it is hard to distinguish the corresponding offset from the noiseless angle, \((2m + 1)\theta\), in circuits of large \( m \). To elaborate on this discussion a little, we do expect a strong but still imperfect prior on \( k_\mu \) to work in principle (i.e., we do not expect \( k_\mu \) being exactly known is strictly necessary), as for small values of \( m \), the noiseless angle and \( k_\mu \) can be distinguished, and so a strong (low variance) prior on \( k_\mu \) with variance further reduced by Bayesian updates from the measurement outcomes of circuits with small \( m \) ought to still enable the estimation of \( \theta \).

Conversely, a strong prior is not required for \( k_\sigma \), as its contribution is completely separable from \( \theta \) in (11), and therefore, the individual uncertainties can be suppressed simultaneously, i.e., if \( k_\mu \) was equal to zero, then increasing both \( m \) and the number of shots will naturally suppress the individual uncertainties in \( \theta \) and \( k_\mu \), respectively.

We now discuss some possible methods to construct the prior for \( k_\mu \) using the calibration data for \( k_\mu \) and specifically consider circuits of the form \( A_1-A_4 \) run on IBM Perth (i.e., circuits for which we have calibration data). One possible method is to set a uniform prior on \( k_\mu \) based on the bound set from the calibration data, which, as discussed previously, can either be a strict bound as in (21) or a looser bound as in (22), dependent on the particular application. Previously, the use of a uniform prior was chosen for convenience (and in the strict case such that the prior spans the empirically determined full range of the parameter) rather than because of any phenomenological reasons; hence, we are motivated to consider alternatives too. One such possibility is to take a nonparametric approach and set the prior using a completely data-driven approach, by directly extracting a probability density function (PDF) that approximates the distribution from the data. Because Fig. 8 suggests the absence of an evident parametric distribution to describe the data, employing a nonparametric approach to derive the PDF in this case is reasonable.\(^8\) Utilizing Gaussian kernel density estimation (KDE) [46] applied to the densitized distribution, we directly extract a PDF that approximates the PDF in order to use as the prior distribution, as demonstrated in Fig. 13.

We can bring all this together to give a general framework for embedding noise awareness into any QPE-free QAE algorithm that uses repeated shots of the same circuit to infer the amplitude, as shown in Algorithm 1. If lines 2–7 are omitted and \( p(\theta, k_\sigma, k_\mu) \) is reinterpreted as simply the marginal probability \( p(\theta) \), then we can see that this is a generic framework covering all QPE-free QAE algorithms. As discussed previously, updating the number of shots according to (30), as given in line 4, is only possible up to a particular \( m \) value, and therefore, if the algorithm gives no valid solutions for the adjusted number of shots for a particular \( m \) value, then

\(^8\)We should note that one would not expect this to necessarily always be the case, and in other cases, there may be an appropriate parameterized function that well describes the data.

![FIGURE 12. Average behavior of the MSE as a function of \( m \) value for circuits of the form \( A_1-A_4 \) run on IBM Perth.](image1)

![FIGURE 13. Densitized distribution of \( k_\sigma \) values for circuits of the form \( A_1-A_4 \) run on IBM Perth overlaid with a Gaussian kernel density estimation PDF that approximates the empirical distribution.](image2)
$N_{\text{max}}^{\text{shots}}$ shots are run instead. Clearly, in this case, there are no guarantees to ensure that the total error on the final estimate will be well constrained, and therefore, an actual implementation should warn the user that this is happening and that, subsequently, the performance may suffer. Likewise, in order that the resources consumed by the user remain bounded, i.e., in order to ensure that the algorithm does not decide to exhaust an unexpectedly large number of shots, if the adjusted number of shots exceeds $N_{\text{max}}^{\text{shots}}$, then again $N_{\text{max}}^{\text{shots}}$ shots are run instead (and a similar warning should be given). Both these conditions are given in line 3. To avoid the algorithm running infinitely, some (user-specified) stopping criteria should be specified, as checked in line 10. This could be one (or a combination) of the following: a maximum number of uses of the circuit $A$ has been reached; a maximum value of $m$ has been reached; the desired accuracy has been reached; or some limit on the total runtime has been reached. Note also that normally, but not exclusively, the initial value of $m$ is zero. Also note that all QPE-free QAE algorithms implicitly set a uniform prior for $\theta$ between 0 and $\pi/2$. It is also the case that this algorithm covers the scenario where either $\mu_{\text{loose}}$, $\mu_{\text{true}}$, or both are known exactly a priori, as this certainty can easily be encoded in the prior, and the rest of the algorithm will play out using the values as constant inputs. It is also worth highlighting that the framework set out in Algorithm 1 captures both adaptive and nonadaptive QAE: in line 12, the update to $m$ and $N_{\text{shots}}$ can either be done by following a preset list, or according to some design based on the current uncertainty of $\theta$.

It is also important to briefly address the challenge of state preparation for QAE within the context of QMCI and its connection to noise-aware QAE. The circuit $A$ is constructed from a state-preparation circuit $P$, which encodes a probability distribution for sampling (see [8] and [16]). $P$ can represent any arbitrary distribution, and in practical near-term applications, it is crucial to design $P$ to be as shallow as possible. This ensures that the total number of Grover iterations can be maximized, thus improving overall performance. Using a shallow circuit naturally limits the probability distributions that can be prepared in practice. However, noise-aware QAE mitigates this restriction by enabling deeper circuits to be run than would be feasible without. In principle, this allows for sampling from a broader range of distributions. Nonetheless, $P$ should still be designed to be as shallow as possible, as in practice there remains a limit to the effectiveness of noise-aware QAE in terms of the total number of Grover iterations that can be performed, as discussed. Several methods have been proposed for the state preparation of probability distributions (see, e.g., [47], [48], [49], [50], [51], [52], and [53]). One frequently cited approach is the Grover–Rudolph method [54]; however, it has been demonstrated that this method offers no quantum advantage when its computational complexity is fully analyzed [55]. A more recent and promising technique from Rosenkranz et al. [56] leverages LCU to prepare multivariate probability distributions. Overall, state preparation for QAE remains an active area of research, and any advancements in this field will greatly enhance the performance of QMCI and other algorithms that make use of QAE as a key subroutine.

VII. NOISE-AWARE QAE: RESULTS

To assess the performance of noise-aware QAE, we reused the results of the existing experiment running the circuit $A_1$ on IBM’s Perth superconducting quantum computer (which is the machine that we have calibration data for) and used the measurement outcome results for each of $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$ uses of $A$. This is a “linearly increasing sequence” in the terminology of [12]. It is important to note that we used this not because it is the best performing way to run QAE, but because we only considered values up to $m = 14$ as this is the average approximate regime of validity for the protocol, as discussed previously, and this way, we were using the maximum number of data points in this range (i.e., an $m$ for every integer). However, in reality, we know from Table 2 that for this particular dataset, the $\mu_{\text{loose}}$ and $\mu_{\text{true}}$ values extracted from the entire dataset are the same order of magnitude, i.e., $\mu_{\text{true}}$ is larger than average, and therefore, the true regime of validity is below $m = 14$; therefore, we may expect the performance to be suboptimal. We also set $N_{\text{max}}^{\text{shots}} = 5N_{\text{shots}}$ as a compromise between giving enough freedom for the algorithm to be able to run as it should, but also making sure that the algorithm does not run an excessive number of shots for a given $m$. Because the total number of shots run when running the algorithm is nondeterministic, the total number of independent runs of noise-aware QAE varied for different experimental configurations tested. As Suzuki et al. [12] only specify that a constant number of shots should be used for each element of the sequence, not a specific value, we chose 10 as a compromise between final precision and the ability of the algorithm to probe higher $m$ values while still being able to constrain the total error by adjusting the number of shots—we shall see that this is sufficient for the noiseless case.

We used the experimental data to compare six different configurations, alongside simulated data for another configuration, where $N_m = 10$ in all cases:

1) QAE and no use of the noise model in the parameter estimation, labeled “Vanilla”;
2) full noise-aware QAE with $p(\theta, k_0, k_{\text{true}})$ set uniformly over $0, \ldots, \pi/2$ for $\theta$, $0.5\mu_{\text{true}}$, $1.5\mu_{\text{true}}$ for $k_0$ (with $k_{\text{true}}$ being the “true” value from Table 2) and $-|k_{\text{true}}^{\text{true}}|, \ldots, |k_{\text{true}}^{\text{true}}|$ for $k_1$ (with $k_{\text{true}}$ as given in (22)), labeled “Full noise aware: $k_{\text{true}}^{\text{true}}$”;
3) full noise-aware QAE with $p(\theta, k_0, k_{\text{true}})$ set uniformly over $0, \ldots, \pi/2$ for $\theta$, $0.5\mu_{\text{true}}$, $1.5\mu_{\text{true}}$ for $k_0$.

9We note that some of these configurations may not be experimentally possible to realize in reality, i.e., it may not be possible to know the parameter values beforehand. However, we tested them regardless in order to better understand the performance of the algorithm.
FIGURE 14. QAE for various configurations for the experiment using IBM’s Perth five-qubit machine. Bootstrapping—implemented via the bias-corrected and accelerated bootstrap interval [57]—is used to determine 68% confidence intervals from which the uncertainties on the data points are determined.

and \(|k_{\mu}^{(\text{strict})}|-|k_{\mu}^{(\text{strict})}|\) for \(k_{\mu}\) [with \(k_{\mu}^{(\text{strict})}\) as given in (21)], labeled “Full noise aware: \(k_{\mu}^{(\text{strict})}\),

4) full noise-aware QAE with \(p(\theta, k_{\sigma}, k_{\mu})\) set uniformly for \(\theta\) and \(k_{\sigma}\) as \(0, \ldots, \pi/2\) and \(0.5k_{\sigma}^{(\text{true})}, \ldots, 1.5k_{\sigma}^{(\text{true})}\), respectively, and \(k_{\mu}\) set according to the KDE PDF given in Section VI, labeled “Full noise aware: \(k_{\mu}^{(\text{kde})}\),

5) noise-aware QAE, but with \(k_{\mu}\) and \(k_{\sigma}\) set to the values extracted from the entire dataset (and so these were not “learned” throughout the algorithms run, but set to their “true” values \(k_{\mu}^{(\text{true})}\) and \(k_{\sigma}^{(\text{true})}\)), labeled “Known \(k_{\mu}, k_{\sigma}\),

6) noise-aware QAE with \(k_{\mu}\) set to \(k_{\mu}^{(\text{true})}\) and the marginalized prior \(p(\theta, k_{\sigma})\) set uniformly over \(0, \ldots, \pi/2\) for \(\theta\) and \(0.5k_{\sigma}^{(\text{true})}, \ldots, 1.5k_{\sigma}^{(\text{true})}\), labeled “Known \(k_{\mu}\),

7) simulation of a noiseless quantum computer running QAE, labeled “Noiseless.”

The results are given in Fig. 14. First, we note that for noise-aware QAE where both parameters were fixed to their “true” values, then the algorithm was able to run a deterministic sequence of shots; this number of shots varied as \(N_m = 10, 11, 12, 13, 14, 15, 16, 18, 19, 20, 21, 22, 23, 24, 25\), and we were able to run 800 independent runs of noise-aware QAE. For all other runs of noise-aware QAE—where one or both of the parameters were learned on the fly—the numbers of shots run were nondeterministic, and thus, the total number of independent runs available varied for the different configuration tested.

Focusing on the full noise-aware QAE configurations, we can immediately see that there are clear benefits to using the noise model; without using the noise model (Vanilla), QAE fails to converge at all on the amplitude, whereas the root mean square error (RMSE) can generally (the exception is possibly for “Full noise aware: \(k_{\mu}^{(\text{strict})}\)” be seen to be converging better when the noise model is used to adjust \(N_m\) such that more shots are executed when the inclement noise becomes more severe. We see that, as expected, the choice of prior for \(k_{\mu}\) has a significant effect on the performance of the algorithm, with the configurations corresponding to setting stronger initial priors (“Full noise aware: \(k_{\mu}^{(\text{loose})}\)” and “Full noise aware: \(k_{\mu}^{(\text{kde})}\)” performing much better than the configuration corresponding to the weaker prior (“Full noise aware: \(k_{\mu}^{(\text{strict})}\)”), which does not appear to really converge at all. Of the three, we see that “Full noise aware: \(k_{\mu}^{(\text{loose})}\)” is the best performing. However, we also see that the convergence of the two performing configurations also begins to flat line at around \(m = 5\), and in principle, this is to be expected given the approximate regime of validity of the protocol for this particular dataset, i.e., after \(m = 5\), there are no valid
solutions to (30), and therefore, no increase in the number of shots can counteract the bias. This regime of validity, however, does not apply for the configurations with fixed $k_\sigma$ as the bias is accounted for, and therefore, the number of shots is updated based on the second equation in (30); in contrast, we see that in these cases, the convergence continues past $m = 5$.

Encouragingly, we see that the performance of the algorithm is fairly agnostic to the assumptions made regarding $k_\sigma$, because even with a relatively high degree of ignorance about $k_\sigma$ assumed—as is the case for the prior that is set—the performance is comparable between the configurations “Known $k_\mu$” and “Known $k_\sigma, k_\mu$,” suggesting that the algorithm is robust to the prior set for $k_\sigma$; we speculate that using published device information (fidelity, quantum volume, etc.) along with coarse information about the circuit, $A$, and making some assumptions regarding the noise, it should be possible to set a prior on $k_\sigma$ which allows noise-aware QAE to work successfully. Based on the configurations with fixed parameter values, it also appears to be the case that noise-aware QAE continues to give decent results in practice, even when the small-angle assumption no longer holds: for instance, using the parameter values from Table 2, we get that the RMSE in the angle is about 1.8 rad for $m = 10$, which cannot reasonably be called a “small angle,” even though there appears to be reasonably decent convergence at this number of Grover iterations.

The final line, “Noiseless,” however, shows that even when the parameters are known precisely, there is still an inherent ceiling to the improvement that can be made by employing noise-aware QAE, and also given the variation in the noise parameters, generally one cannot predict a priori how much improvement one might expect for a given problem, particularly given the imperfect method of constraining $k_\sigma$ and $k_\mu$ from calibration data. However, we still see that there are clear benefits to running noise-aware QAE, and indeed, error mitigation protocols are not expected to improve results to the extent that they match the noiseless case. An important future direction of research is, therefore, to discover effective general ways of setting strong (low-variance) priors on $k_\mu$ and indeed to better use the data from circuits with low values of $m$ (i.e., when $\theta$ and $k_\mu$ can be better distinguished) to narrow the distribution on $k_\mu$ further.

The challenge now is to use the proposed Gaussian noise model, together with ever-improving quantum hardware, to obtain compelling QAE results for problems of real-world interest. Furthermore, the linearly increasing sequence of Suzuki et al. [12] is known to not be optimal and is only rarely used here for illustrative purposes. An important future line of research and development, therefore, will be to embed the noise model into other proposals for QPE-free QAE to obtain noise-aware versions of the state-of-the-art QPE-free QAE algorithms.

VIII. CONCLUSION

QAE is a subroutine that powers many of the most promising potential applications of quantum computing, notably QMCI. In order to run QAE-based algorithms on near-term hardware, it is desirable to find ways to enhance the computational performance in the face of the unavoidable inclement noise. While generic error mitigation techniques may offer some such enhancements, we have identified that the regular structure of QAE circuits offers the opportunity to treat the machine noise as estimation uncertainty and thus handle the noise at the application level. This approach is attractive as it is tailored to the particular algorithm and offers rigorous performance guarantees even in the face of noisy hardware.

In this article, we have proposed a simple Gaussian noise model applicable to all near-term QAE algorithms that do not use quantum phase estimation. In addition, we discussed how to extend the model to account for noise effects not well captured by Gaussian noise, such as amplitude damping, by incorporating them into an extended model. We conducted experiments on various IBM superconducting quantum computers and Quantumum’s H1 trapped-ion quantum computer to validate the model on real hardware. The proposed Gaussian noise model fits the experimental data well, with $R^2$ values for the fits to data having a percentage reduction in error (improvement in closeness to unity) for IBM quantum hardware of up to $99\%$ (circuit $A_1$, Lagos) and for Quantumum quantum hardware of up to $55\%$ (circuit $A_3$, Run 3), compared to fitting a depolarizing noise model. Notably, this is because the Gaussian noise model effectively captures a systematic bias introduced by noise during the application of the Grover iterate in QAE—something generic noise models like depolarizing noise fail to do. Furthermore, we have demonstrated how our noise model can be used to inform the design of, and improve parameter estimation in, QAE, resulting in a noise-aware QAE algorithm. Experimentally, on IBM quantum hardware (Perth), we showed that for a two-qubit circuit with a particular structure, and for a suitable prior distribution on the Gaussian noise parameters, the convergence of amplitude estimation in noise-aware QAE continues to improve up to circuit depths corresponding to approximately five Grover iterates (the expected upper limit of applicability of the methodology given the particular dataset). In contrast, without noise awareness, convergence only improved up to about a single Grover iterate.

An important direction for future research is to combine noise-aware QAE with traditional techniques for error mitigation and to assess how much further improvement the two complementary approaches give in combination. In addition, another intriguing—although speculative—research avenue would be to explore how noise-aware QAE and QEC might complement each other, both in terms of enhancing QAE performance and reducing the overall resources required for QEC.

---

10It is likely that the prior itself will not be particularly accurate based on this information alone, but this is likely not an issue when actually running the algorithm, as discussed.
APPENDIX A
COMPILED CIRCUITS
In this appendix, we give compiled circuits for $A_1$–$A_5$ (see Fig. 15).

(a) $A_1$ compiled for IBM (Athens, Bogota, Rome, and Santiago).
(b) $A_1$ compiled for IBM (Lagos, Nairobi, and Perth).
(c) $A_2$ compiled for IBM.
(d) $A_3$ compiled for IBM.
(e) $A_4$ compiled for IBM.
(f) $A_5$ compiled for Quantinuum H1 (Run 1, Run 2).
(g) $A_5$ compiled for Quantinuum H1 (Run 3).

FIGURE 15. (a) $A_1$ compiled for IBM (Athens, Bogota, Rome, and Santiago). (b) $A_1$ compiled for IBM (Lagos, Nairobi, and Perth). (c) $A_2$ compiled for IBM. (d) $A_3$ compiled for IBM. (e) $A_4$ compiled for IBM. (f) $A_5$ compiled for Quantinuum H1 (Run 1, Run 2). (g) $A_5$ compiled for Quantinuum H1 (Run 3).
TABLE 2. Parameter Values for the Various State-Preparation Circuits Run on IBM Superconducting Quantum Computers and Quantinuum’s H1 Trapped-Ion Quantum Computer

| Circuit | Machine  | $k_e$ | $k_{GA}$ | $k_{GA}$ with depolarizing $p_{\text{in}}$ | $k_f$ | $k_{GA}$ | $k_{FAD}$ |
|---------|----------|-------|--------|---------------------------------|-------|--------|----------|
| $A_1$   | Athens   | -0.0153 ± 0.0002 | 0.2575 ± 0.0002 | 0.9159 ± 0.0007 | 0.0315 ± 0.0002 | -0.0153 ± 0.0002 | 0.2599 ± 0.0002 | 0.0300 ± 0.0000 |
| $A_2$   | Bogota   | -0.0035 ± 0.0002 | 0.2575 ± 0.0002 | 0.9389 ± 0.0004 | 0.0435 ± 0.0004 | 0.0265 ± 0.0002 | 0.0335 ± 0.0002 | 0.0000 ± 0.0000 |
| $A_3$   | Rome     | -0.0147 ± 0.0003 | 0.3030 ± 0.0003 | 0.9277 ± 0.0006 | 0.0375 ± 0.0003 | -0.0147 ± 0.0003 | 0.0303 ± 0.0003 | 0.0000 ± 0.0000 |
| $A_4$   | Santiago | -0.0214 ± 0.0002 | 0.0108 ± 0.0002 | 0.9465 ± 0.0002 | 0.0275 ± 0.0001 | -0.0214 ± 0.0000 | 0.0108 ± 0.0002 | 0.0000 ± 0.0000 |
| $A_5$   | Nairobi  | 0.0165 ± 0.0000 | 0.113 ± 0.0000 | 0.841 ± 0.0000 | 0.0131 ± 0.0000 | 0.0165 ± 0.0000 | 0.113 ± 0.0000 | 0.0000 ± 0.0000 |
| $A_6$   | Perth    | 0.0076 ± 0.0001 | 0.0270 ± 0.0001 | 0.8932 ± 0.0005 | 0.0571 ± 0.0003 | 0.0370 ± 0.0001 | 0.0267 ± 0.0001 | 0.0000 ± 0.0000 |

ACKNOWLEDGMENT

The authors thank the anonymous reviewers and the editorial team for their careful consideration of the manuscript, Cristina Cirstoiu for helpful conversations leading to the development and refinement of the theoretical model, and Ross Duncan for reviewing.

REFERENCES

[1] J. Preskill, “Quantum computing in the NISQ era and beyond,” Quantum, vol. 2, Aug. 2018, Art. no. 79, doi: 10.22331/q-2018-08-06-79.

[2] K. Temme, S. Bravyi, and J. M. Gambetta, “Error mitigation for short-depth quantum circuits,” Phys. Rev. Lett., vol. 119, no. 18, Nov. 2017, Art. no. 180509, doi: 10.1103/PhysRevLett.119.180509.

[3] Y. Li and S. C. Benjamin, “Efficient variational quantum simulator incorporating active error mitigation,” Phys. Rev. X, vol. 7, no. 2, Jun. 2017, Art. no. 021050, doi: 10.1103/PhysRevX.7.021050.

[4] J. J. Wallman and J. Emerson, “Noise tailoring for scalable quantum computation via randomized compiling,” Phys. Rev. A, vol. 94, no. 5, Nov. 2016, Art. no. 052325, doi: 10.1103/PhysRevA.94.052325.

[5] A. N. Chowdhury and R. D. Somma, “Quantum algorithms for Gibbs sampling and hitting-time estimation,” in Proc. IEEE Internat. Symp. Comput. Des., 2020, pp. 9–12, doi: 10.1109/ICCD50377.2020.00015.

[6] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, “Amplitude estimation without phase estimation,” Quantum Inf. Process., vol. 19, no. 2, Jan. 2020, doi: 10.1007/s11128-019-2565-2.

[7] D. G. S. L. Brandão, A. Kalev, T. Li, C.-Y.-Y. Lin, K. M. Svore, and X. Wu, “Quantum SDP solvers: Large speed-ups, optimality, and applications to quantum learning,” 2019, arXiv:1710.02581, doi: 10.48550/arXiv.1710.02581.

[8] S. Aaronson and A. Arkhipov, “The computational complexity of linear optics,” in Proc. 43rd Annu. ACM Symp. Theory Comput., 2011, pp. 333–342, doi: 10.1145/1993636.1993682.

[9] A. N. Chowdhury and R. D. Somma, “Quantum algorithms for Gibbs sampling and hitting-time estimation,” 2016, arXiv:1603.02940, doi: 10.48550/arXiv.1603.02940.

[10] Y. Suzuki, S. Uno, R. Raymond, T. Tanaka, T. Onodera, and N. Yamamoto, “Amplitude estimation without phase estimation,” Quantum Inf. Process., vol. 19, no. 2, Jan. 2020, doi: 10.1007/s11128-019-2565-2.

[11] D. Grinko, J. Gacon, C. Zoufal, and S. Woerner, “Iterative quantum amplitude estimation,” 2020, arXiv:1912.05559, doi: 10.48550/arXiv.1912.05559.

[12] K. Nakaji, “Faster amplitude estimation,” Quantum Inf. Comput., vol. 20, pp. 1109–1123, 2020, doi: 10.26421/QIC20.13-14-2.

[13] S. Aaronson and P. Rall, “Quantum approximate counting, simplified,” in Proc. Simplicity Algorithms, Jan. 2020, pp. 24–32, doi: 10.1137/19781197604.5.

[14] S. Herbert, “Quantum Monte Carlo integration: The full advantage in minimal circuit depth,” 2021, arXiv:2105.09100, doi: 10.48550/arXiv.2105.09100.

[15] S. Herbert, “Quantum computing system and method,” Patent application GB2102902.0 and SE2130060-3, 2021.

[16] D. An, N. Linden, J.-P. Liu, A. Montanaro, C. Shao, and J. Wang, “Quantum-accelerated multilevel Monte Carlo methods for stochastic differential equations in mathematical finance,” 2020, arXiv:2012.06283, doi: 10.48550/arXiv.2012.06283.

[17] P. Rebentrost, B. Gupt, and T. R. Bromley, “Quantum computational finance: Monte Carlo pricing of financial derivatives,” Phys. Rev. A, vol. 98, no. 2, Aug. 2018, Art. no. 022321, doi: 10.1103/PhysRevA.98.022321.

[18] N. Stamatopoulos et al., “Option pricing using quantum computers,” 2019, arXiv:1905.02666, doi: 10.48550/arXiv.1905.02666.
[21] S. Woerner and D. J. Egger, “Quantum risk analysis,” NPJ Quantum Inf., vol. 5, no. 1, Feb. 2019, Art. no. 10.5058/S41534-019-0120-3.

[22] A. Bouland, W. van Dam, H. Joorati, I. Kerenidis, and A. Prakash, “Prospects and challenges of quantum finance,” 2020, arXiv:2011.06492, doi: 10.48550/arXiv.2011.06492.

[23] F. Orúis, S. Murgul, and E. Lizaso, “Quantum computing for finance: Overview and prospects,” Rev. Phys., vol. 4, 2019, Art. no. 100028, doi: 10.1103/RevModPhys.2019.100028.

[24] D. J. Egger, R. G. Gutiérrez, J. C. Mestre, and S. Woerner, “Credit risk analysis using quantum computers,” 2019, arXiv:1907.03044, doi: 10.48550/arXiv.arXiv.1907.03044.

[25] K. Kaneko, K. Miyamoto, N. Takeda, and K. Yoshino, “Quantum pricing with a smile: Implementation of local volatility model on quantum computer,” 2020, arXiv:2007.01467, doi: 10.48550/arXiv.2007.01467.

[26] S. Chakrabarti, R. Krishnakumar, G. Mazzola, N. Stamatopoulos, S. Woerner, and W. J. Zeng, “A threshold for quantum advantage in derivative pricing,” 2020, arXiv:2012.03819, doi: 10.48550/arXiv.2012.03819.

[27] P. Rebentrost and S. Lloyd, “Quantum computational finance: Quantum algorithm for portfolio optimization,” 2018, arXiv:1811.03975, doi: 10.48550/arXiv.arXiv.1811.03975.

[28] D. J. Egger et al., “Quantum computing for finance: State-of-the-art and future prospects,” IEEE Trans. Quantum Eng., vol. 1, 2020, Art. no. 310124, doi: 10.1109/TQE.2020.3030314.

[29] J. Alcazar et al., “Quantum algorithm for credit valuation adjustments,” New J. Phys., vol. 24, 2022, Art. no. 020306, doi: 10.1088/1367-2630/ac5003.

[30] E. G. Brown, O. Goktas, and W. K. Thom, “Quantum amplitude estimation in the presence of noise,” 2020, arXiv:2006.14145, doi: 10.48550/arXiv.arXiv.2006.14145.

[31] T. Tanaka, Y. Suzuki, S. Uno, R. Raymond, T. Onodera, and N. Yamamoto, “Amplitude estimation via maximum likelihood on noisy quantum computer,” 2020, arXiv:2006.16223, doi: 10.48550/arXiv.2006.16223.

[32] T. Giurgica-Tiron, I. Kerenidis, F. Labib, A. Prakash, and W. Zeng, “Low depth algorithms for quantum amplitude estimation,” 2020, arXiv:2012.03348, doi: 10.48550/arXiv.2012.03348.

[33] I. Kerenidis and A. Prakash, “A method for amplitude estimation with a smile: Implementation of local volatility model on quantum computer,” 2020, arXiv:2005.02081, doi: 10.48550/arXiv.2005.02081.

[34] T. J. DiCiccio and B. Efron, “ Bootstrap confidence intervals,” Stat. Sci., vol. 11, no. 3, pp. 189–228, 1996, doi: 10.1214/ss/1032280214.

[35] S. Herbert, “ No quantum speedup with Grover-Rudolph state preparation for quantum Monte Carlo integration,” Phys. Rev. E, vol. 103, Jun. 2021, Art. no. 063302, doi: 10.1103/PhysRevE.103.063302.

[36] M. Rosenberg et al., “ Quantum state preparation for multivariate functions,” 2024, arXiv:2405.21058, doi: 10.48550/arXiv.2405.21058.

[37] T. J. DiCiccio and B. Efron, “Bootstrap confidence intervals,” Stat. Sci., vol. 11, no. 3, pp. 189–228, 1996, doi: 10.1214/ss/1032280214.

Steven Herbert received the B.A., M.A., and M.Eng. (Cantab) degrees in engineering and the Ph.D. degree in computer science from University of Cambridge, Cambridge, U.K., in 2012, 2013, and 2014, respectively. He was a Postdoctoral Researcher with the University of Cambridge, the University of Oxford, Oxford, U.K., and the University of Edinburgh, Edinburgh, U.K. He is currently the Head of Quantum Algorithms with Quantumin, Cambridge, U.K., and an Affiliated Lecturer in Computer Science with the University of Cambridge. His research interests include all aspects of quantum information and computation.

Ilan Williams received the integrated M.Sc. degree in physics with theoretical physics from Imperial College London, London, U.K., in 2015, and the Ph.D. degree in high-energy physics from the University of Cambridge, Cambridge, U.K., in 2022. In 2013, he was a Undergraduate Research Intern with the Institut Laue-Langevin, Grenoble, France, investigating novel small-angle neutron scattering instruments. In 2014, he was a Technology Intern with fintech startup Currencycloud, London. In 2017, he was a Graduate Research Intern with the Institute for Particle Physics Phenomenology, University of Durham, Durham, U.K., studying the properties of particle decays in heavy-flavor physics. He is currently a Quantum Algorithms Research Scientist with Quantumin, Cambridge, U.K. His research interests include aspects of quantum Monte Carlo integration algorithms and the application of classical machine learning techniques to problems in quantum computing.
Roland Guichard was born in St-Julien-en-Genevois, France, in 1980. He received the B.S. degree in physics, the M.S. degree in physics, and the Ph.D. degree in theoretical quantum mechanics from the University of Bordeaux, Bordeaux, France, in 2003, 2004, and 2007, respectively. From 2008 to 2010, he was a Postdoctoral Fellow with the Max-Planck Institute for the Physics of Complex Systems, Dresden, Germany. From 2010 to 2013, he was a Postdoctoral Fellow with Sorbonne University, Paris, France. From 2014 to 2015, he was a Postdoctoral Fellow with University College London, London, U.K. From 2014 to 2017, he was a Software Engineer with Connected Cities Catapult, Milton Keynes, U.K. From 2017 to 2020, he was a Research Software Engineer with University College London. From 2020 to 2023, he was a Senior Software Engineer with Quantinuum, Cambridge, U.K. He is currently the Head of Quantum Libraries, Pasqal, London, U.K.

Darren Ng received the Integrated M.Sc. degree in theoretical physics from the University of Bristol, Bristol, U.K., in 2020. In 2018, he was a Student Researcher with the School of Physics, University of Bristol, working on the visualization of the correspondence principle of the hydrogen atom. In 2019, he then interned with the Quantum Engineering Technology Lab, Bristol, under the EPSRC Summer Research Bursary Fund, working on benchmarking quantum gates and quantum noise models in quantum computation. Shortly after graduation in 2020, he joined Quantinuum, Cambridge, U.K., as a Quantum Algorithms Research Intern, working on quantum amplitude estimation algorithms. He is currently a Software Engineer in finance.