Efficient Bayesian Phase Estimation

Nathan Wiebe[1] and Chris Granade[2,3]

[1] *Quantum Architectures and Computation Group, Microsoft Research, Redmond, WA (USA)*
[2] *Centre for Engineered Quantum Systems, Sydney, NSW (Australia)*
[3] *School of Physics, University of Sydney, Sydney, NSW (Australia)*

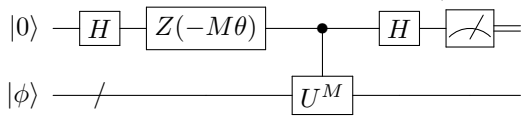
We provide a new efficient adaptive algorithm for performing phase estimation that does not require that the user to infer the bits of the eigenphase from highest- to lowest-order; rather it directly infers the phase and estimates the uncertainty in the phase from experimental data. Our method is highly flexible, can be run in the presence of substantial decoherence and other experimental imperfections and is as fast or faster than existing approaches to phase estimation. The simplicity and low memory footprint of our method also make it amenable to embedded or FPGA applications.

I. INTRODUCTION

Phase estimation (PE) is a technique of fundamental importance to quantum information. The purpose of the algorithm is to estimate the eigenphase corresponding to an input eigenstate of a unitary U within a user specified precision. The protocol forms the core of quantum simulation algorithms, linear systems algorithms and the amplitude estimation algorithm to name a few applications [1–5]. As a result, methods that even modestly improve the time and space complexities of phase estimation or increase the algorithm's robustness to experimental errors are of great importance to both quantum computing and metrology [6–9]. Our work not only provide a fast PE algorithm but also a more general one that can tolerate substantial experimental imperfections.

Iterative phase estimation (IPE) seeks to learn the eigenvalue of a given eigenvector of U using a classical inference algorithm run in conjunction with a simple quantum circuit to infer the bits of a binary expansion of the phase. It is typically used in preference to traditional phase estimation for its simplicity and minimal qubit requirements

The quantum circuit for IPE is [9]



where $U|\phi\rangle = e^{i\phi}|\phi\rangle$ for an unknown eigenphase $\phi \in \mathbb{R}$, and where $Z(M\theta) = e^{iM\theta Z}$ is a rotation that can be used to compare the eigenphase of U to a known reference value. Most PE algorithms use this circuit to infer the bits in a binary expansion of ϕ in order from least significant to most significant. The process is near optimal (up to $\log^*$ factors [9]) and affords a classically efficient and deterministic process for inferring these bits.

A drawback to iterative phase estimation is that it is not necessarily robust to experimental errors. This is because the inference process used to find the bits of the unknown phase is specialized for the probability distribution induced by an ideal phase estimation experiment, which may not accurately model that yielded by an imperfect implementation. This issue becomes particularly significant as more bits of precision are required since typical phase estimation algorithms require that M grow exponentially. This means that initially weak decoherence eventually blocks the ability to learn an unknown eigenphase using traditional phase estimation.

Bayesian methods for PE provide a potential resolution to this problem. This approach, originally proposed by Svore *et al* [9], allows information to be gleaned from arbitrary experiments. This feature allows short experiments to be used to learn the eigenphase, which minimizes the impact of decoherence on the inference. Despite this, existing proposals for performing Bayesian phase estimation require an exponential amount of processing time to process the data from the experiments and also the use of randomly chosen experiments can cause the algorithm to become confused if a continuum of eigenphases are possible.

Our work addresses these problems by not only providing a method for performing an efficient form of Bayesian phase estimation but also giving an expedient heuristic for adaptively choosing approximately optimal experiments. We illustrate this by showing that our algorithm not only exhibits the same Heisenberg limited scaling of phase estimation and continues to learn even when decoherence or other experimental imperfections prevent traditional phase estimation methods from working.

II. BAYESIAN PHASE ESTIMATION

Bayesian phase estimation, as introduced by Svore *et al* [9], involves performing a set of experiments and then updating the prior distribution using Bayes' rule. For example, if an experiment is performed with using M repetitions of U, $Z(M\theta)$ and a measurement outcome $E \in \{0, 1\}$ is observed then Bayes' rule states that the *posterior probability* distribution for ϕ after observing the datum is

$$P(\phi|E;\theta,M) = \frac{P(E|\phi;\theta,M)P(\phi)}{\int P(E|\phi;\theta,M)P(\phi)\mathrm{d}\phi}. \quad (1)$$

The final ingredient that is needed to perform an update is the likelihood function, $P(0|\phi;\theta,M)$, is easy to com-

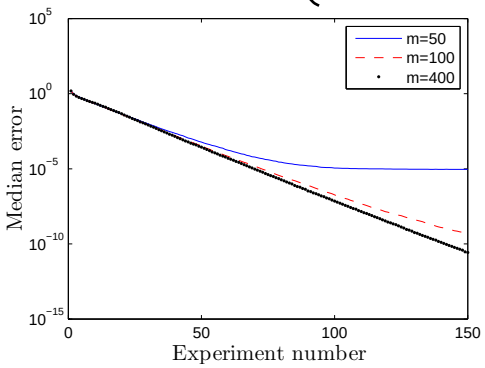


FIG. 1. Median errors in phase estimation for 10 000 random initial choices of the true eigenphase.

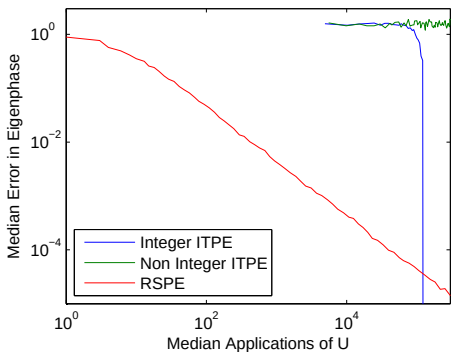


FIG. 2. Comparison of RSPE to ITPE from [9] for $t = 10\,000$ for $\phi_{\text{true}} = 2\pi k/t$ for integer and non-integer k.

pute for phase estimation:

$$P(0|\phi;\theta,M) = \frac{1+\cos(M[\phi+\theta])}{2}, \qquad P(1|\phi;\theta,M) = \frac{1-\cos(M[\phi+\theta])}{2}. \tag{2}$$

After using (1) to update the posterior distribution we then set the prior distribution to equal the posterior distribution. This process is then repeated for each of the random experiments in the data set.

Unlike conventional methods, Bayesian inference returns a posterior distribution over the phase. The mean and standard deviation of this distribution then provide an estimate of the true eigenvalue and the algorithm's uncertainty in that value. More sophisticated estimates of uncertainty, such as credible regions, can also be easily extracted from the posterior distribution [10]. This estimate of the uncertainty is vital for these algorithms because it gives the confidence in the result and also provides a stopping criteria. In particular, the inference can be prematurely ended if an accuracy target is met.

III. APPROXIMATE BAYESIAN PHASE ESTIMATION

Exact Bayesian inference is impossible in cases where the eigenphase is continuous. Approximate Bayesian inference nonetheless is not only possible but is also a mainstay in computer vision and machine learning [11, 13]. Such methods typically use sampling methods to discretize the prior distribution and then perform Bayesian inference on the discrete set of samples (often called "particles").

Rather than using a set of hypotheses that implicitly define a model for the system, we posit a prior model and directly update it to find a model for our posterior distribution. We achieve this by using a Gaussian to model our initial prior, perform a Bayesian update on samples drawn from the distribution and then refit the updated samples to a Gaussian. This strategy is used in a number of particle filter approximations to the posterior such as the extended Kalman filter and assumed density filtering [11, 12]. These methods, although efficient, can require tracking several thousand discrete hypotheses for ϕ and can be challenging to implement.

We propose using a much simpler approach that we call Rejection Sampling Phase Estimation (RSPE). The algorithm is described below and pseudocode is given in the appendix.

1. Perform experiment for given θ, M and observe outcome $E \in \{0, 1\}$.
2. Draw m samples from $\mathcal{N}(\mu, \sigma^2)$.
3. For each sample, ϕ_j, assign ϕ_j to Φ_{accept} with probability $P(E|\phi_j;\theta,M)/\kappa_E$, where $\kappa_E \in (0, 1]$ is a constant s.t. $P(E|\phi_j;\theta,M)/\kappa_E \leq 1$ for all ϕ_j, E.
4. Return $\mu = \mathbb{E}(\Phi_{\text{accept}})$ and $\sigma = \sqrt{\mathbb{V}(\Phi_{\text{accept}})}$.

The samples drawn from this distribution are sampled according to the posterior distribution $P(\phi|E;M,\theta)$. To see this, note that the probability density of a sample being accepted at $\phi = \phi_j$ is the product of the probability of acceptance and the probability density of drawing the sample from $\mathcal{N}(\mu, \sigma^2) = P(\phi)$. Eqn (1) then implies that

$$P(E|\phi;\theta,M)\mathcal{N}(\mu,\sigma^2) \propto P(\phi|E;\theta,M). \tag{3}$$

Thus the distribution of the accepted samples is equivalent to the posterior distribution.

A further advantage of RSPE is that only a few hundred bits of memory are required, even if $m \gg 1$, because μ and σ can be computed incrementally. This not only allows PE to be run as a streaming algorithm but also enables execution in memory restricted environments such as embedded controllers or FPGAs.

The main issue that remains is how to optimally choose the parameters θ and M. This issue can be solved by local optimization of the Bayes risk [10], but the resulting calculation can be too expensive to carry out in online experiments that provide experimental results at a rate

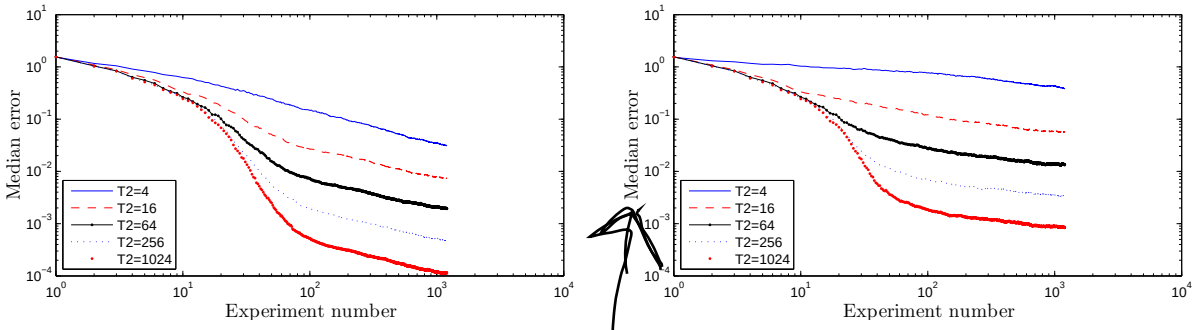


FIG. 3. Median errors for phase estimation in decohering systems for experiments constrained to use $M \leq T_2$ (left) and $M \leq T_2/10$ (right). We take $m = 12\,000$ and use 1 000 random samples to generate the data above. The initial state is taken to be a randomly chosen eigenstate in all cases.

of tens of Megahertz or faster. Fortunately, there is an expedient guess heuristic, known as the particle guess heuristic (PGH) that can be used to choose a near–optimal experiment for this class of likelihood functions [14]. Specifically we take

$$M = \left\lceil \frac{1.25}{\sigma} \right\rceil, \ \theta \sim P(\phi). \tag{4}$$

The factor of 1.25 comes from optimizing the cost of RSPE. Non–integer M are appropriate if $U = e^{-iHM}$.

Figure 1 shows the error incurred using RSPE for non–integer M. The most obvious feature is that the error shrinks exponentially with the number of experiments (which is proportional to the evolution time under the PGH) for $m > 100$. Roughly 150 experiments are needed for the method to provide 32 bits of accuracy in the median case. We discuss the scaling in the mean in the appendix.

The number of experiments needed to reach error ϵ scales as $O(\log(1/\epsilon))$ rather than $O(\log(1/\epsilon)\log\log(1/\epsilon))$ in Kitaev's method for iterative phase estimation [6, 7]. Although the number of experiments needed is small, the experimental time required to achieve this is not necessarily small. If the error shrinks as $\epsilon \in \Theta(e^{-\lambda N})$ where N is the experiment number then $T_{\text{exp}} \in O(\sum_{N=1}^{N_{\max}} e^{\lambda N}) \in O(e^{\lambda N_{\max}}) \in O(1/\epsilon)$. Thus the total experimental time required saturates the Heisenberg limit, up to a multiplicative constant. In practice, $\lambda \approx 0.17$ for RSPE.

Figure 2 compares RSPE to the Bayesian PE method of [9] (ITPE). Although ITPE is inefficient and is non–adaptive, it is exact and as a result it is a natural benchmark to compare RSPE against. ITPE requires nearly three times the applications of U to estimate ϕ if the eigenphase is of the form $\phi = 2\pi k/t$ for integer $k < t$ and $t = 10\,000$. This shows that although ITPE makes excellent use of sub–optimal experiments, optimized experiments can reduce the requisite experimental time.

Conversely, if we focus on the number of experiments needed then we find that ITPE requires only 25 measurements to identify the phase with 50% probability whereas RSPE requires 55 experiments if k is an integer. However, RSPE is handicapped here because it does not assume that k is an integer. On the other hand, if the true value of k is real valued and ITPE is left unmodified then it no longer performs exact inference. Figure 2 also shows no evidence of learning for non–integer k in contrast to RSPE. This underscores the fact that intelligent experimental designs and approximations to the prior are crucial for the success of approximate Bayesian PE.

A. Phase estimation with depolarizing noise

A criticism that has been levied lately at the phase estimation algorithm is that it can be impractical to execute on non–fault tolerant quantum hardware [15, 16]. This is because phase estimation attains its quadratic advantage over statistical sampling by using exponentially long evolutions. Decoherence causes the resultant phases to become randomized as time progresses, ultimately resulting in $\lim_{M\to\infty} P(0|\phi;\theta,M) = \frac{1}{2}$. In this limit, the measurements convey no information and inference becomes impossible. It similarly may be tempting to think that decoherence fundamentally places a lower limit on the accuracy of PE. It is possible, however, to use Bayesian inference to estimate ϕ using experiments with $M \approx T_2$. This observation is analogous to that in [10], where particle filter methods are used to infer a frequency in the presence of uncharacterized decoherence.

We model the effect of decoherence on the system by assuming the existence of a *decoherence time* T_2 such that

$$P(0|\phi) = \mathrm{e}^{-M/T_2}\left(\frac{1+\cos(M[\phi-\theta])}{2}\right) + \frac{1-\mathrm{e}^{-M/T_2}}{2},$$
$$P(1|\phi) = \mathrm{e}^{-M/T_2}\left(\frac{1-\cos(M[\phi-\theta])}{2}\right) + \frac{1-\mathrm{e}^{-M/T_2}}{2}. \tag{5}$$

In the limit as $T_2 \to \infty$ the likelihood function (5) approaches (2). Similarly as $T_2 \to 0$ it approaches the uniform distribution. This model is appropriate when the time required to implement the controlled operation $\Lambda(U)$ is long relative the that required to perform H and an arbitrary Z–rotation, as is appropriate in quantum

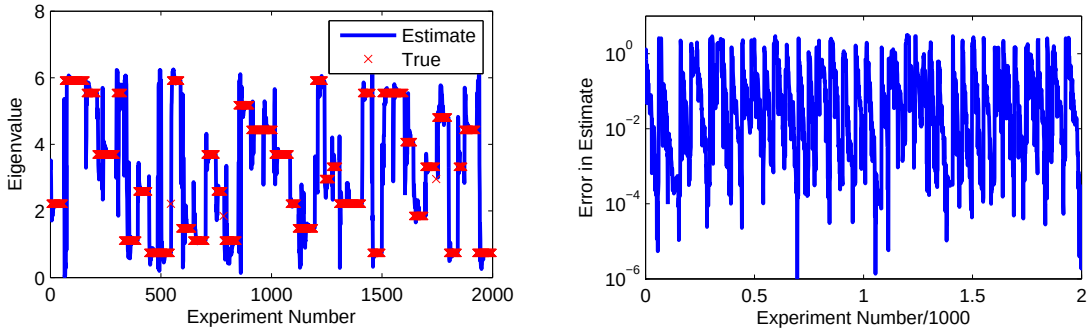


FIG. 4. Tracking the current estimate of an eigenphase during phase estimation for a system with 16 eigenvalues and $T_2 = 10^4$.

simulation. For simplicity, in the following we will assume that T_2 is well characterized.

Since T_2 places a limitation on our ability to learn, we propose a variation to (4):

$$M = \min\left\{\left\lceil\frac{1.25}{\sigma}\right\rceil, T_2\right\}. \tag{6}$$

This heuristic qualitatively matches optimized solutions for frequency estimation, which are chosen to saturate the Cramér-Rao bound [17]. This can be seen through the following argument. The Cramér-Rao bound scales as $O(M^{-2})$ for phase estimation [18] in the absence of decoherence. Eqn. (5) suggests that depolarizing noise causes the variance to increase as $O(\exp(2M/T_2))$. Calculus reveals that $M = T_2$ optimally trades off the two tendencies, as also observed in Figure 3.

The same methodology applies when T_2 is unknown [10]. The only modification needed is that the T_2 in (6) should be drawn from the prior $P(\phi, T_2)$.

Figure 3 shows that RSPE smoothly transitions between the exponential scaling expected at short times and polynomial scaling when decoherence becomes significant. The error scales roughly as $1/N^{0.6}$ in this polynomial regime, which is comparable to the $1/\sqrt{N}$ scaling expected from statistical sampling. Decoherence therefore does not necessarily impose a fundamental limitation on the accuracy that PE can achieve. Figure 3 also shows that confining the experiments to stay in a more coherent regime also can actually hurt the algorithm's ability to infer the phase, as expected.

IV. TRACKING EIGENPHASES

Figure 3 shows the performance of RSPE when the initial quantum state is an eigenstate and is discarded after each experiment. Performing phase estimation in this way minimizes the number of experiments, but can be prohibitively expensive if preparation of the initial eigenstate is prohibitively costly. In such cases, it makes sense to follow the standard prescription for phase estimation by keeping the quantum state until it is clear that the initial eigenstate has been depolarized. These depolarizations can cause RSPE to become confused because the new data that comes in is only consistent with hypotheses that have been ruled out. We address this by performing inexpensive experiments to assess whether the state has depolarized and then restart the learning process. Restarting can even be valuable when $T_2 = 0$ if the input state is a superposition of eigenvectors or to recover if the RSPE becomes stuck (see appendix).

The following procedure addresses this issue in cases where the spectral gaps are promised to be at least Δ.

1. After each update with probability e^{-M/T_2} perform an experiment with $\theta = \mu$ and $M = \tau/\sigma$ for $\tau < 1$.
2. If result $= 1$ then prepare initial state and reset σ.
3. After restart, continue as normal until $\sigma < \Delta$ then set σ and μ to the previous best known values.

Steps 1 and 2 can be thought of as a one-sided test of whether the prior distribution is consistent with the current state. If the prior probability distribution is correct then the probability of measuring 0 is

$$\frac{1}{\sigma\sqrt{2\pi}}\int_{-\infty}^{\infty}\cos^2\left(\frac{(\mu-x)\tau}{2\sigma}\right)e^{-\frac{(\mu-x)^2}{2\sigma^2}}\mathrm{d}x = \frac{1+e^{-\tau^2/2}}{2}. \tag{7}$$

If $\tau = 0.1$ the probability of measuring 0 is approximately 0.998 and hence measuring 1 implies that the Gaussian distribution is wrong with probability 0.998.

If the test fails at a fixed level of significance then Step 3 restarts the learning process. It does not, however, make sense to throw away all the spectral information that has been learned so far after a restart. Step 3 reflects this by checking to see if the current estimate of the eigenphase corresponds to a known eigenstate and then sets μ to be the estimate of the eigenvalue and σ to be its uncertainty. This allows the algorithm to resume learning once the depolarized state is projected onto a known eigenstate.

This process not only allows the initial state to be reused, but also permits the eigenvalue of an eigenstate in a decohering system to be estimated in real time as seen

in Figure 4. There we take $\Delta = 0$, $\tau = 0.1$ and $T_2 = 10^4$ and notice that as the PE algorithm can rapidly detect a transition away from the instantaneous eigenstate and then begin inferring the eigenvalue of the system's new instantaneous eigenstate.

V. CONCLUSION

We have provided a new approach to phase estimation that is not only efficient but also is resilient to experimental noise. The simplicity of the algorithm further means that it can be implemented in a few hours but also that it can be executed on an embedded controller or an FPGA that is dedicated to controlling a quantum system. Since RSPE runs well in the presence of noise and imperfections in the system, it may allow more sophisticated quantum simulations to be run without raising the specter of decoherence ruining the experiment.

Looking forward, our work illustrates that phase estimation has not been solved in its entirety. Rather, the problem of how to best infer an unknown eigenvalue in the presence of experimental constraints is, and will likely remain, an important problem for the foreseeable future. The flexibility, efficiency and robustness of approximate Bayesian inference will no doubt make it an indispensable tool for facing such challenges.

[1] P. W. Shor, SIAM journal on computing **26**, 1484 (1997).
[2] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Contemporary Mathematics **305**, 53 (2002).
[3] A. Aspuru-Guzik, A. D. Dutoi, P. J. Love, and M. Head-Gordon, Science **309**, 1704 (2005).
[4] A. W. Harrow, A. Hassidim, and S. Lloyd, Physical review letters **103**, 150502 (2009).
[5] B. P. Lanyon, J. D. Whitfield, G. Gillett, M. E. Goggin, M. P. Almeida, I. Kassal, J. D. Biamonte, M. Mohseni, B. J. Powell, M. Barbieri, *et al.*, Nature Chemistry **2**, 106 (2010).
[6] A. Y. Kitaev, Electronic Colloquium on Computational Complexity **3** (1996).
[7] A. Y. Kitaev, A. Shen, and M. N. Vyalyi, *Classical and quantum computation*, Vol. 47 (American Mathematical Society Providence, 2002).
[8] B. L. Higgins, D. W. Berry, S. D. Bartlett, H. M. Wiseman, and G. J. Pryde, Nature **450**, 393 (2007).
[9] K. M. Svore, M. B. Hastings, and M. Freedman, Quantum Information & Computation **14**, 306 (2014).
[10] C. E. Granade, C. Ferrie, N. Wiebe, and D. G. Cory, New Journal of Physics **14**, 103013 (2012).
[11] S. Haykin, *Kalman filtering and neural networks*, Vol. 47 (John Wiley & Sons, 2004).
[12] M. Opper and O. Winther, On-line Learning in Neural Networks, ed. D. Saad , 363 (1998).
[13] A. Smith, A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice* (Springer Science & Business Media, 2013).
[14] N. Wiebe, C. Granade, C. Ferrie, and D. Cory, Physical Review Letters **112**, 190501 (2014).
[15] A. Peruzzo, J. McClean, P. Shadbolt, M.-H. Yung, X.-Q. Zhou, P. J. Love, A. Aspuru-Guzik, and J. L. OBrien, Nature communications **5** (2014).
[16] J. R. McClean, R. Babbush, P. J. Love, and A. Aspuru-Guzik, The Journal of Physical Chemistry Letters **5**, 4368 (2014).
[17] C. Ferrie, C. E. Granade, and D. G. Cory, Quantum Information Processing **12**, 611 (2013).
[18] N. Wiebe, C. Granade, and D. G. Cory, New Journal of Physics **17**, 022005 (2015).

To cite: Shalman et al, Reiley's FPGA paper

Appendix A: Variance reduction strategies

An important drawback of our approach is that in typical applications the tails can be quite fat, meaning that there is significant probability that the error in the inferred eigenphase is orders of magnitude greater than the median. Such a distribution can be seen in Figure 5 where we see that although the median error is roughly 10^{-10} radians after 100 updates for $m > 50$, a non–negligible fraction of the experiments have error on the order of 1.

Fortunately, the need to always repeat the algorithm and use a majority voting scheme to reduce the variance of the estimate is mitigated by the fact that the algorithm outputs σ which estimates the uncertainty in the resultant eigenphase. Nonetheless, alternative strategies exist for reducing the variance of the estimates of the phase after a fixed number of experiments. Standard approaches involve random restarting and the use of multi–modal distributions. Here we focus on heuristics for detecting when the learning algorithm fails to converge.

The central idea behind our restarting strategy is to examine the decay of σ with the number of experiments. In ideal cases, the error decays exponentially which means that it is easy to see when the algorithm fails by plotting σ on a semilog plot. Such intuition can be easily automated. The idea behind our restarting algorithm is to first estimate the derivative of $\log(\sigma)$ with respect to the experiment number and determine whether it is less than a threshold. If it is less than the threshold, perform an experiment to test to see if the value of μ that has been learned so far is accurate (as per our incoherent phase estimation algorithm). If it is found to be innacurate then restart the algorithm and abandon the information learned so far.

When a restarting strategy like this is employed, we need to modify the model selection criteria. Previously, we

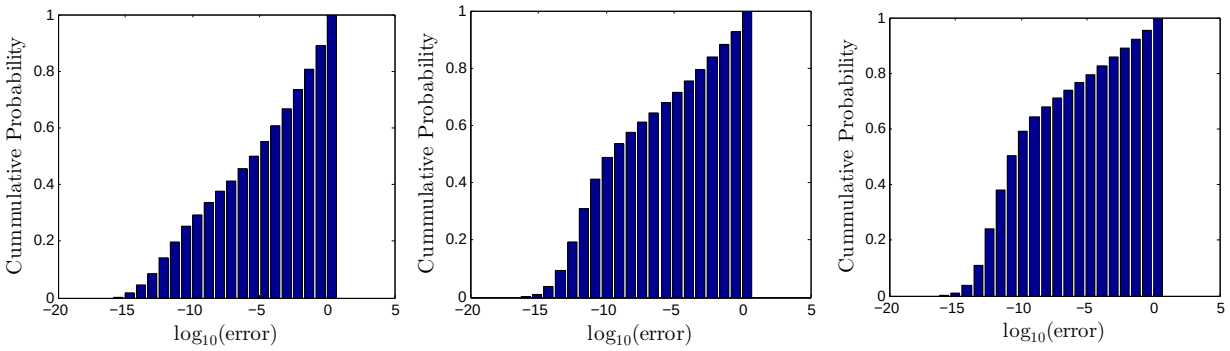


FIG. 5. Cumulative distribution function of probability that PE error is less than x after 150 updates for $m = 50$ (left) $m = 100$ (middle) $m = 200$ (right).

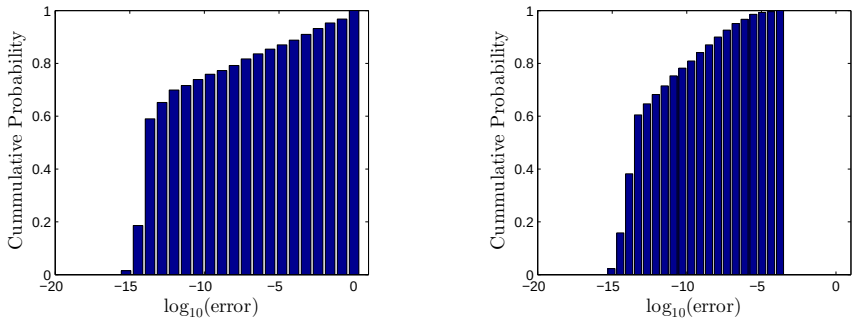


FIG. 6. Cumulative distribution function of probability that PE error is less than x after 200 updates for $m = 2000$ for $\Gamma = \infty$ (left) $\Gamma = 0.1$ (right) and $\tau = 0.1$. Estimate of CDF consists of 1000 randomly sampled eigenphase inference problems with $T_2 = \infty$.

used the value of μ yielded by the most recent experiment as our estimate. Given that a restart late in the algorithm can reset all information learned about a model, it makes sense in this context to use the value of μ corresponding to the smallest σ observed. This corresponds to the model that the inference algorithm has the greatest certainty.

We see in Figure 6 that this strategy of restarting substantially reduces the weights of the tails. In fact, the mean error in the inference falls from 0.0513 radians to 1.08×10^{-6} radians. This shows that this resetting strategy substantially reduce the probability of a large error occuring in the estimate of the eigenphase.

Appendix B: Stability against errors in the likelihood function

Our algorithm is similarly robust against *uncharacterized noise sources* as well. We demonstrate this in Figure 7 wherein we introduce depolarizing noise of strength γ to our experimental system, but do not include such noise in the likelihood function. For example, with $\gamma = 0.4$, the measurement outcome of phase estimation is replaced with a random bit with probability 40%. We find that while the inclusion of such noise causes the error decay exponent, λ, to shrink as roughly $0.17e^{-3.1\gamma}$ it does not prevent our algorithm from learning at an exponential rate (until depolarizing noise becomes significant). This illustrates that our method for phase estimation can continue to work even in the presence of noise sources that are both strong and uncharacterized.

These results may not be surprising in the context of previous work [14]. However, these results were only shown for sequential Monte–Carlo approximations rather than our comparably simpler rejection sampling approach.

Appendix C: Stability of rejection sampling PE

One way in which the rejection sampling algorithm can break down is if the likelihood function becomes too flat relative to the number of discrete samples drawn from the prior distribution, m. This breakdown occurs because the sample variance in the posterior mean is much greater than the difference between the prior mean and posterior

can this be combatted by batching & K-rescaling when likelihood is flat, heuristic: small shift in μ?

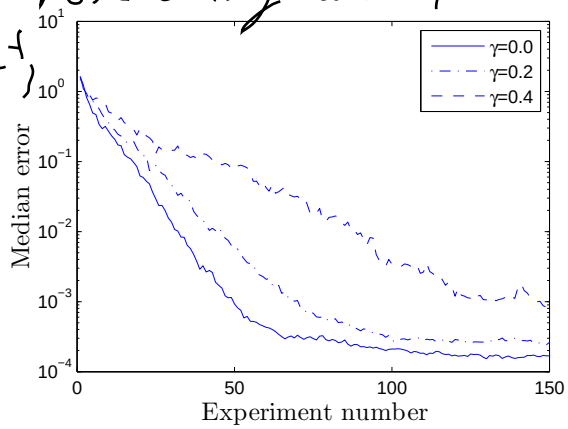


FIG. 7. Errors in inference for phase estimation with different levels of un-modeled noise, γ, for $T_2 = 1000$. In each instance the initial state was taken to be a fixed, but unknown, eigenstate of the Hamiltonian. The median was found for 100 randomly chosen values of the eigenphase of this fixed initial eigenstate.

means, which are not expected to be approximately the same if the likelihood function has little variation. This in effect implies that the dynamics of the mean will essentially be a random walk and as a result we do not expect that our rejection sampling method to perform well if the likelihood function is too flat.

The question remaining is: how flat is too flat? We show in the theorem below that if the number of samples taken from the true posterior distribution does not scale at least inverse quadratically with the scale of relative fluctuations in the likelihood function then we expect the posterior variance to be much greater than the shift in the means.

Theorem 1. *Assume that for all j $P(E|x_j) = \alpha + \delta_j$ where $|\delta_j| \leq \delta$ and $\alpha \geq 10\delta$ and assume that $x_j \sim P(x_j)$. If we then define $\mu_0 := \sum_j P(x_j)x_j$, $\mu_1 := \sum_j P(x_j|E)x_j$ and σ_1^2 to be the posterior variance then $|\mu_1 - \mu_0| \in \Omega(\sigma_1/\sqrt{m})$ only if $\frac{\alpha^2}{\delta^2} \in O(m)$.*

Proof. Bayes' rule gives us

$$\left|\sum_j P(x_j|E)x_j - \mu_0\right| = \left|\frac{\sum_j P(E|x_j)P(x_j)(x_j - \mu_0)}{\sum_j P(E|x_j)P(x_j)}\right| = \left|\frac{\sum_j \delta_j P(x_j)(x_j - \mu_0)}{\sum_j P(E|x_j)P(x_j)}\right|. \tag{C1}$$

Then using the Cauchy–Schwarz innequality, the triangle inequality and $\alpha \geq 2\delta$.

$$\left|\frac{\sum_j \delta_j P(x_j)(x_j - \mu_0)}{\sum_j P(E|x_j)P(x_j)}\right| \leq \left|\frac{\delta\sqrt{\sum_j P(x_j)|x_j - \mu_0|^2}}{\alpha - \delta}\right| \leq \frac{\delta\sigma}{\alpha - \delta} \leq \frac{2\delta\sigma}{\alpha}. \tag{C2}$$

Thus the maximum shift in the posterior mean shrinks as the the likelihood function becomes increasingly flat, as expected.

Next we need to lower bound the posterior variance in order ensure that the value of m chosen suffices to make the error small in the best possible case. To do so we use the reverse triangle inequality:

$$\sigma_1^2 = |\sigma_1^2 - \sigma^2 + \sigma^2| \geq \sigma^2 - |\sigma_1^2 - \sigma^2|. \tag{C3}$$

Thus it suffices to upper bound $|\sigma_1^2 - \sigma^2|$ to lower bound σ_1^2. To do so, note that $\alpha \geq 2\delta$ and hence

$$|P(x_j|E) - P(x_j)| = \left|\frac{P(x_j)(\delta_j - \sum_j P(x_j)\delta_j)}{\alpha + \sum_j P(x_j)\delta_j}\right| \leq \frac{P(x_j)2\delta}{\alpha - \delta} \leq \frac{P(x_j)4\delta}{\alpha}. \tag{C4}$$

Now the difference between the two variances can be written as

$$\begin{aligned}
|\sigma_1^2 - \sigma^2| &= \left| \sum_j P(x_j|E)(x_j-\mu_1)^2 - P(x_j)(x_j-\mu_0)^2 \right| \\
&\leq \left| \sum_j (P(x_j|E) - P(x_j))(x_j-\mu_1)^2 \right| + \left| \sum_j P(x_j)\left((x_j-\mu_0)^2 - (x_j-\mu_1)^2\right) \right| \\
&\leq \frac{4\delta}{\alpha} \left| \sum_j P(x_j)(x_j-\mu_1)^2 \right| + (\mu_1-\mu_0)^2 \\
&\leq \frac{4\delta\sigma^2}{\alpha} + \frac{4\delta}{\alpha} \left| \sum_j P(x_j)[(x_j-\mu_1)^2 - (x_j-\mu_0)^2] \right| + (\mu_1-\mu_0)^2 \\
&\leq \frac{4\delta\sigma^2}{\alpha} + (1+\frac{4\delta}{\alpha})(\mu_1-\mu_0)^2 \leq \frac{4\delta\sigma^2}{\alpha} + \frac{12\delta^2\sigma^2}{\alpha^2} \leq \frac{10\delta\sigma^2}{\alpha}.
\end{aligned} \tag{C5}$$

Thus we have that

$$\sigma_1^2 \geq \sigma^2(1 - 10\delta/\alpha). \tag{C6}$$

Now assuming $\delta \leq \alpha/10$ we have

$$\sigma_1^2 \in \Omega(\sigma^2). \tag{C7}$$

Finally we note that

$$|\mu_1 - \mu_0| \in \Omega(\sigma_1/\sqrt{m}) \Rightarrow \frac{\delta\sigma}{\alpha} \in \Omega(\sigma/\sqrt{m}), \tag{C8}$$

which is only true if $m \in O(\alpha^2/\delta^2)$. □

We therefore see that the number of samples needed to track the small changes in a posterior distribution that happens when the likelihood function becomes extremely flat. This condition is not sufficient because the actual components of the posterior mean may be shifted by a much smaller amount than the upper bounds used in the proof of Theorem 1.

In contrast, exact Bayesian inference requires a number of bits that scales as $O(\log(1/\delta))$ (assuming a fixed and discrete number of hypotheses). Thus exact Bayesian inference (or to a lesser extent particle filter methods) can be preferable in cases where the likelihood function is extremely flat. Such concerns can be somewhat be either be avoided entirely or batches of such experiments should be combined to produce a likelihood function that is much less flat and choosing an appropriate instrumental distribution to ensure that the success probability remains high.

Appendix D: Pseudocode for algorithms

In the main body we sketched the details of our phase estimation algorithm. Here we elaborate on this algorithm and discuss some of the subtle details needed to make the algorithm work. The first such subtlety stems from the fact that eigenphases are equivalent modulo 2π. To see this problem, consider a Gaussian distribution centered at 0. If we take the outputs of the distribution in the branch $[0, 2\pi]$ then we find that the mean of the distribution is π rather than 0. Since the support of the initial Gaussian may be small at $\phi = \pi$, such errors can be catastrophic during the inference procedure. This can be dealt by using the circular mean and by working with a wrapped normal distribution. For expedience, we eschew this approach and instead use a heuristic approach that does not require a large number of trigonometric calls. This is especially important in cases where the algorithm is executed on a microcontroller or an FPGA where native trigonometric functions are not likely to be available.

The heuristic approach, described in Algorithm 1, uses rejection sampling and incremental methods to estimate the mean and standard deviation of the posterior distribution. If $\sigma \ll 2\pi$ then the probability distribution is narrow and does not suffer from significant wrap around unless $\mu \mod 2\pi \approx 0$. We address this by keeping track of each of the accepted ϕ_j as well as $\phi_j + \pi$. If $\sigma \ll 1$ than it is impossible that both distributions suffer from substantial

wrap around. The arithmetic, rather than circular, mean and standard deviation are then computed for both using an incremental formula and the branch with the smallest variance is kept. If the branch that was shifted by π is chosen, then π is subtracted from the reported mean. The standard deviation does not need to be modified because it is invariant with respect to displacements of the mean.

While this approach is correct if $\sigma \ll 1$, it is only approximate if σ is on the order of 1. In such cases, computation of the circular mean is much better justified, however we find that using our heuristic approach continues to provide acceptable results while avoiding trigonometric calls that can be expensive in some contexts. An alternative approach to solving this problem is to begin each phase estimation run with a batch of random experiments, as per [9], before continuing to ensure that the posterior variance is small enough to neglect the wrap around effect.

The choice of the evolution time and the inversion angle strongly impacts the efficiency of the learning algorithm. We provide below code for a modified version of the particle guess heuristic of [14]. As discussed in the main body, we expect that choosing $M > T_2$ will typically lead to worse estimates of the eigenphase because the effects of decoherence overwhelm the information that can be gleaned from these long experiments. As a result, we modify the particle guess heuristic to never choose $M > T_2$. We formally state this procedure in Algorithm 3.

Algorithm 1 Bayes Update for RejS

Input: Prior mean and variance μ, σ, measurement E, settings M, θ, number of attempts m, scale κ_E

▷ Initialize accumulators to 0.

$(\mu_{\text{inc}}, \mu'_{\text{inc}}, V_{\text{inc}}, V'_{\text{inc}}, N_a) \leftarrow 0$

▷ Attempt each sample.

for $i \in 1 \to m$ **do**

▷ Draw a sample using each "cut" of the prior.

$x \sim \frac{e^{-(\phi-\mu)^2/2\sigma^2}}{\sigma\sqrt{2\pi}}$,

$x \leftarrow x \mod 2\pi$.

$x' \leftarrow x + \pi \mod 2\pi$.

▷ Accept or reject the new sample.

$u \sim \text{Uniform}(0, 1)$

if $P(E|x) \geq \kappa_E u$ **then**

▷ Accumulate using the accepted sample w/ each "cut."

$\mu_{\text{inc}} \leftarrow \mu_{\text{inc}} + x$

$V_{\text{inc}} \leftarrow V_{\text{inc}} + x^2$

$V'_{\text{inc}} \leftarrow V'_{\text{inc}} + x'^2$

$N_a \leftarrow N_a + 1$.

end if

end for

▷ Return mean, variance of the posterior using accumulators.

$\mu' \leftarrow \mu_{\text{inc}}/N_a$

$\sigma' \leftarrow \min\left(\sqrt{\frac{1}{N_a-1}(V_{\text{inc}} - \mu_{\text{inc}}^2)}, \sqrt{\frac{1}{N_a-1}(V'_{\text{inc}} - \mu'^2_{\text{inc}})}\right)$

return (μ', σ')

Algorithm 2 Restarting algorithm

Input: Prior RejS state, records of all previous models found in the phase estimation algorithm μ, $\vec{\sigma}$, initial standard deviation σ_{init}, a counter CNT, Γ and τ.
Output: CNT, σ

```
function RESTART(μ, σ⃗, T_2)
    D ← derivative of log σ.
    if CNT < 5 then                ▷ Checks to see if enough points have been considered to accurately estimate gradient.
        CNT ← CNT + 1
        return CNT, σ.
    else if D ≥ Γ then
        Perform experiment with M = τ/σ and θ = μ.
        if Outcome is 0 then                ▷ Test concludes state estimate is valid
            return CNT, σ.
        else                                ▷ Test concludes state estimate is invalid
            CNT ← 0
            σ ← σ_init
            return CNT, σ
        end if
    end if
end function
```

Algorithm 3 PGH for decoherent phase estimation using RejS

Input: Prior RejS state μ, Σ. Resampling kernel F.
Output: An experiment (M, θ).

```
function PGH_RejS (μ, Σ, T_2)
    M ← 1.25/√Tr(Σ)
    if M ≥ T_2 then
        M ~ f(x; 1/T_2)            ▷ Draw M from an exponential distribution with mean T_2.
    end if
    (−θ/M) ~ F(μ, Σ)
    return (M, θ).
end function
```