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 direcly 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 algorithms. The simplicity and low memory footprint of our method also make it amenable to embedded or FPGA applications.

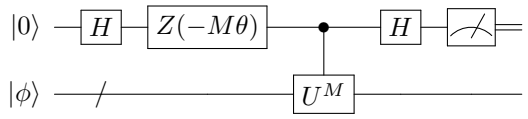
I. INTRODUCTION

Eigenvalue estimation has been a cornerstone of physics since the dawn of spectroscopy. In recent years, ideas from quantum information have revolutionized the ways that we estimate these values by providing methods that require exponentially fewer experiments than statistical sampling and total experimental time that saturates the Heisenberg limit, which is the best possible scaling allowed by quantum mechanics. This quantum approach, known as phase estimation (PE), is crucial to acheive many of the celebrated speedups promised by quantum computing [1–5] and as such optimizing these algorithms is especially important for present day experimental demonstrations of such algorithms because the number of quantum gates that can be performed is often severely limitted by decoherence.

The most popular approach to PE is known as iterative phase estimation (IPE). Its popularity stems from the fact that it forgoes the use of quantum resources to infer the eigenvalues of unitary matrix in favor of using a classical inference algorithm to estimate the eigenvalues from experimental data [6–9]. In recent years there has been a revolution in machine learning that has lead to a plethora of new classical inference procedures. This raises an important question: can these ideas be used to speed up phase estimation or make it more robust to experimental error?

We provide a new classical inference method, inspired by recent work on particle filter methods, that is tailored to the phase estimation problems. This method not only provides performance advantages over existing ITPE methods [6, 7], but also is robust to experimental imperfections such as depolarizing noise and systematic errors in the PE circuit. Before going into detail about our algorithm we will first review the phase estimation circuit and Bayesian approaches to PE. We then show that rejection sampling can be used to efficiently approximate Bayesian inference thereby allowing PE to inherit the speed and robustness of Bayesian approaches while retaining the efficiency of traditional PE methods.

IPE infers the eigenvalue of a given eigenvector of a unitary matrix U from a set of experiments that are performed on the following circuit.



where $U|\phi\rangle = \mathrm{e}^{\mathrm{i}\phi}|\phi\rangle$ for an unknown eigenphase $\phi \in \mathbb{R}$, and where $Z(M\theta) = \mathrm{e}^{\mathrm{i}M\theta Z}$ 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.

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

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

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

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

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

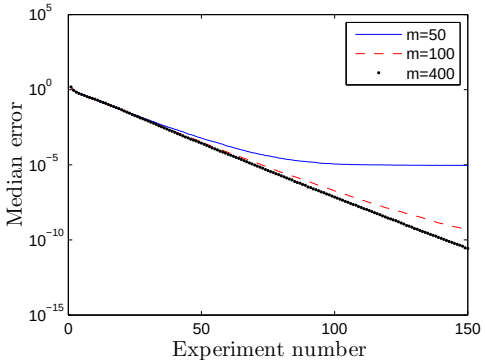


FIG. 1. Median errors in phase estimation for 10 000 random initial choices of the true eigenphase. 1 000 samples were used.

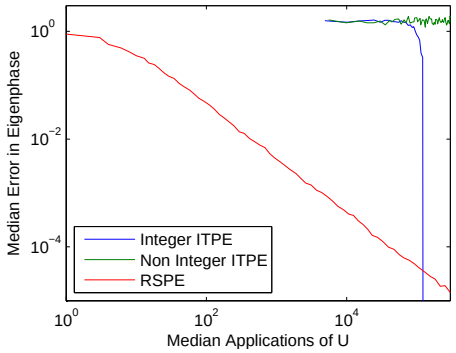


FIG. 2. Comparison of RFPE to ITPE for $t = 10\,000$ with 100 samples for $\phi_{\text{true}} = 2\pi k/t$ for integer and non-integer k.

and standard deviation of this distribution 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]. These estimates of the uncertainty are crucial for our method because they not only allow the user to prematurely end the protocol when a fixed accuracy has been reached, but also because they allow us to design highly informative experiments for finding the true eigenvalue.

III. APPROXIMATE BAYESIAN PHASE ESTIMATION

Exact Bayesian inference is impossible in cases where the eigenphase is continuous. As a result approximations to Bayesian inference are needed to make the procedure tractable. Rather than naively discritizing the prior distribution, modern methods use sampling methods to discretize the prior and then perform Bayesian inference on the discrete set of samples (often called "particles"). These particle filter methods methods are quite powerful and have become a mainstay in computer vision and machine learning [11, 12].

Despite the power of these methods, they are often hard to implement and even harder to deploy in a memory limited environment such as on an embedded controller or an FPGA. With the increasing use of FPGAs in the control of quantum information experiments, [13**?** **?**], embedding inference algorithms presents a significant advantage to modern experiment design. As a result, we propose using a much simpler approach that we call Rejection Filtering Phase Estimation (RFPE). 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, 14]. We then further optimize this method by approximating the Bayesian inference step using rejection sampling. This can reduce the memory required for the inference step by a factor of 1 000 or more. 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 $\mathcal{N}(\phi|\mu, \sigma^2)$.
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 resultant samples are equivalent to those drawn from 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 $\mathcal{N}(\phi|\mu, \sigma^2) = P(\phi)$. Eqn (1) then implies

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

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

Although it is difficult to concretely predict the value of m needed to make the error in the inference small, we show in the appendix that m must scale at least as the inverse square of the relative fluctuations in the likelihood function. Similarly, Markov's inequality shows that m must scale at least as $\kappa_E / \int P(E|\phi; \theta, M)P(\phi)\mathrm{d}\phi$ to ensure that, with high probability, the number of accepted samples will be large enough to accurately estimate the mean. We introduce the term κ_E to help compensate for small expected likelihoods.

The main issue that remains is how to optimally choose the parameters θ and M. This issue can be solved by

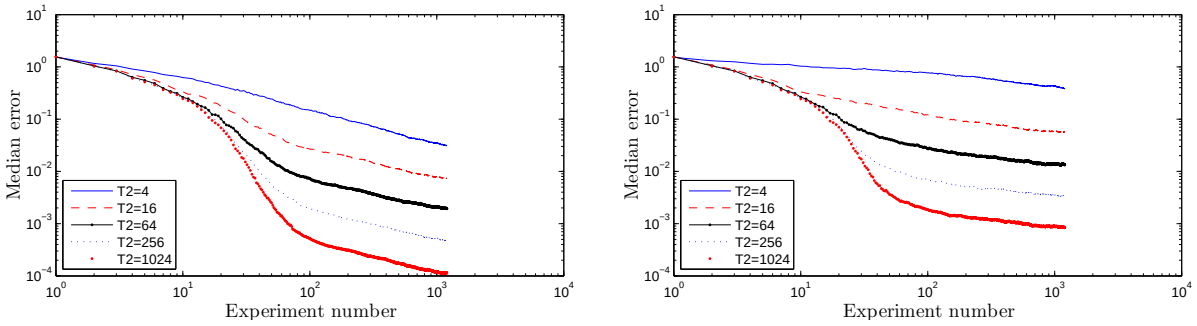


FIG. 3. Median errors for phase estimation in decohering systems for experiments constrained to use $M \leq T_2$ (left) and $M \leq T_2/10$ (right). We take $m = 12\,000$ and use 1 000 random samples to generate the data above. The initial state is taken to be a randomly chosen eigenstate in all cases.

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 of tens of megahertz or faster. Fortunately, there is an expedient guess heuristic, known as the particle guess heuristic (PGH) that can give a near optimal experiment for this class of likelihood functions [15]:

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

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

Figure 1 shows the error incurred using RFPE. 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]. To make this comparison concrete, the data from [9] shows roughly 48 000 measurements are needed to achieve 1 000 bits of precision with probability 0.5 using [6, 7]. We find by extrapolation that RFPE requires roughly 4 100 experiments to achieve the same accuracy.

Although the number of experiments needed is small, the experimental time need not be. If the error shrinks as $\epsilon \in \Theta(e^{-\lambda N})$ where N is the experiment number then $T_{\text{exp}} \in O(\sum_{N=1}^{N_{\max}} e^{\lambda N}) \in O(e^{\lambda N_{\max}}) \in O(1/\epsilon)$. The time required saturates the Heisenberg limit, up to a multiplicative constant. Specifically, $\lambda \approx 0.17$ for RFPE.

Figure 2 compares RFPE to the information-theory PE (ITPE) method of Svore *et al.* Although ITPE is inefficient and is non–adaptive, it is exact and as a result it is a natural benchmark to compare RFPE 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 RFPE requires 55 experiments if k is an integer. RFPE is handicapped in this comparison because it uses a prior that is ignorant of the fact k is an integer. On the other hand, if the true value of k is real valued and ITPE is left unmodified then Figure 2 also shows no evidence of learning for non–integer k in contrast to RFPE. The ITPE algorithm fails to learn in these circumstances because the hypotheses in the support of the prior will never match the true hypothesis, causing ITPE to become confused in the face of contradictory results that can arise from random experiments. Intelligent experimental designs and approximations to the prior are therefore 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 [16, 17]. 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

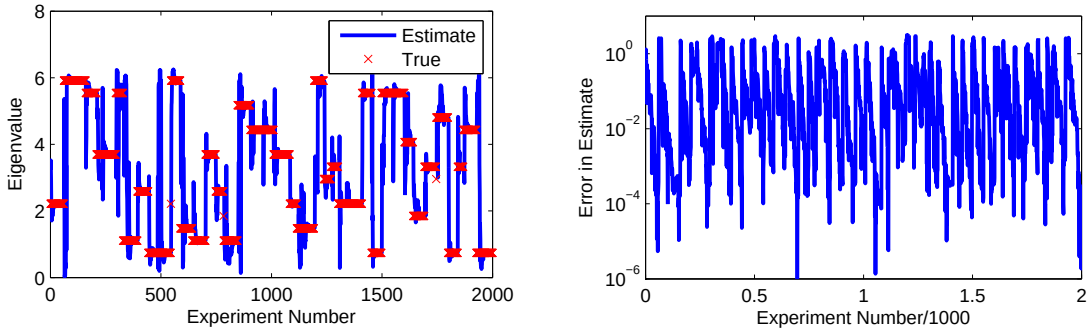


FIG. 4. Instantaneous estimate of the eigenphase for a system with 16 eigenvalues, $\Delta = 0$, $\tau = 0.1$ and $T_2 = 10^4$.

assuming the existence of a *decoherence time* T_2 such that

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

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 simulation. For simplicity, in the following we assume that T_2 is known.

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

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

The experiments yielded by (6) are qualitatively similar to locally optimized experiments for frequency estimation which are chosen to saturate the Cramér-Rao bound [18]; however, (6) requires nearly 100 fold less computing time to select an experiment. We choose M to cut off at T_2 in part because the locally optimized solutions choose to saturate at this value which can be understood using the following argument. The Cramér-Rao bound for frequency estimation scales as $O(M^{-2})$ [19] in the absence of decoherence. Eqn. (5) suggests that decoherence causes the posterior variance to increase as $O(\exp(2M/T_2))$. As in the frequency estimation case [18], calculus reveals that $M = T_2$ optimally trades off these tendencies, hence $M \leq T_2$ is optimal as also observed in Figure 3.

Figure 3 shows that RFPE smoothly transitions between the exponential scaling expected at short times and the polynomial scaling expected 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 RFPE 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 RFPE 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 RFPE 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 those of the closest eigenvalue.

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

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

If $\tau = 0.1$ the probability of measuring 0 is approximately 0.998 and hence measuring 1 implies that the hypothesis

that the prior is correct can be rejected at $p \leq 0.002$. Alternatively, as discussed in Appendix D, we can also use the Bayes factor to determine when to apply the reset rule.

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 when the algorithm is restarted. 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 if such an identification is made. This allows the algorithm to resume learning once the depolarized state is projected onto a known eigenstate.

This process also permits the eigenvalue of an eigenstate in a decohering system to be estimated in real time as seen in Figure 4. The plot shows that the restarting 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 RFPE 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), quant-ph/0005055.
[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] A. Smith, A. Doucet, N. de Freitas, and N. Gordon, *Sequential Monte Carlo methods in practice* (Springer Science & Business Media, 2013).
[13] M. D. Shulman, S. P. Harvey, J. M. Nichol, S. D. Bartlett, A. C. Doherty, V. Umansky, and A. Yacoby, Nature Communications **5**, 5156 (2014).
[14] M. Opper and O. Winther, On-line Learning in Neural Networks , 363 (1998).
[15] N. Wiebe, C. Granade, C. Ferrie, and D. Cory, Physical Review Letters **112**, 190501 (2014).
[16] 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**, 4213 (2014).
[17] J. R. McClean, R. Babbush, P. J. Love, and A. Aspuru-Guzik, Journal of Physical Chemistry Letters **5**, 4368 (2014).
[18] C. Ferrie, C. E. Granade, and D. G. Cory, Quantum Information Processing **12**, 611 (2013).
[19] N. Wiebe, C. Granade, and D. G. Cory, New Journal of Physics **17**, 022005 (2015).
[20] N. Wiebe, C. Granade, C. Ferrie, and D. Cory, Physical Review A **89**, 042314 (2014).
[21] S. Goodman, Seminars in Hematology Interpretation of Quantitative Research, **45**, 135 (2008).
[22] R. Hoekstra, R. D. Morey, J. N. Rouder, and E.-J. Wagenmakers, Psychonomic Bulletin & Review , 1 (2014).

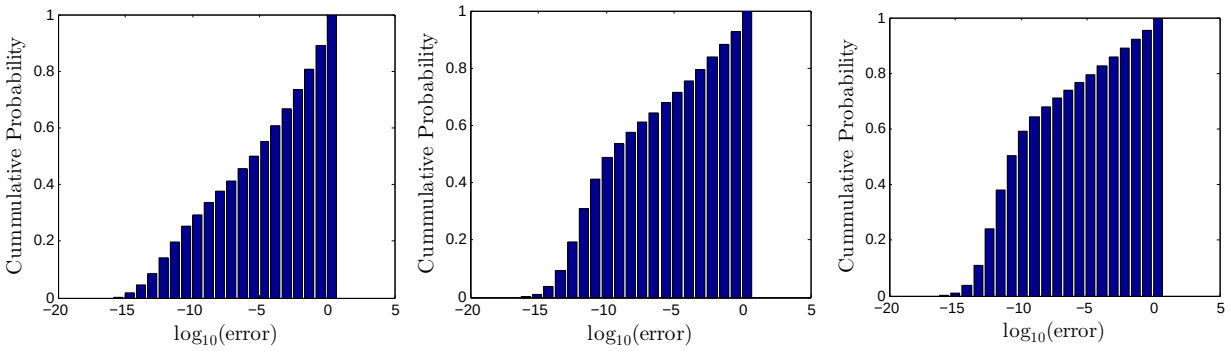


FIG. 5. Cumulative distribution function of probability that PE error is less than x after 150 updates for $m = 50$ (left) $m = 100$ (middle) $m = 200$ (right).

Appendix A: Variance Reduction Strategies

An important drawback of RFPE is the tails for the distribution of errors in phase estimation can be quite fat in typical applications, 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. Alternatively, less aggressive experiments can be chosen in the guess heuristic or multi–modal models for the prior distribution (such as Gaussian mixture models) can be used. The later approach can be quite valuable in computer vision and machine learning, however here we have an additional freedom not typically enjoyed in those fields: we can perform adaptive experiments to test to see if our current hypothesis is correct.

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 semi–log 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. The restarting portion of this algorithm is given in Algorithm 2.

When a restarting strategy like this is employed, we need to modify the model selection criteria. Previously, we 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. To our knowledge, this data represents the first time that a heuristic method has been successful in reducing the mean error in frequency estimation to an acceptable level of uncertainty [10]. Previous approaches that succeeded in making the mean–error small used costly numerically optimized experiments, which render such approaches impractical for online phase estimation [10, 15, 18–20].

Appendix B: Stability Against Errors in the Likelihood Function

Our algorithm is not only robust against well known depolarizing noise but is 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%. Although this may seem qualitatively similar to the numerical examples for depolarizing noise considered in the main body of the paper, this case is much more pathological because it is only used to generate the experimental data and is not permitted in the model of the system used in the likelihood function. This raises the possibility that the algorithm could become

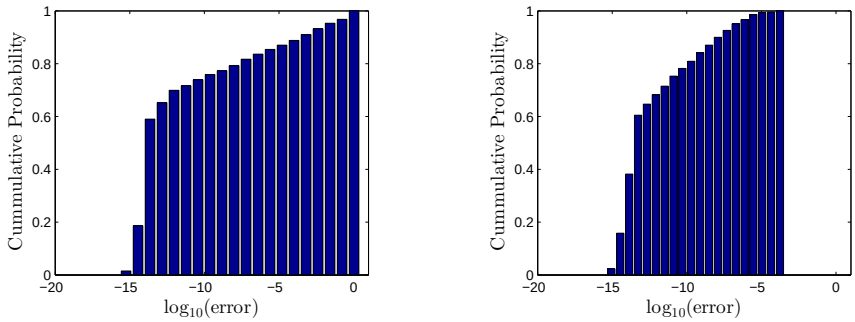


FIG. 6. Cumulative distribution function of probability that PE error is less than x after 200 updates for $m = 2000$ for $\Gamma = \infty$ (left) $\Gamma = 0.1$ (right) and $\tau = 0.1$. Estimate of CDF consists of 1000 randomly sampled eigenphase inference problems with $T_2 = \infty$.

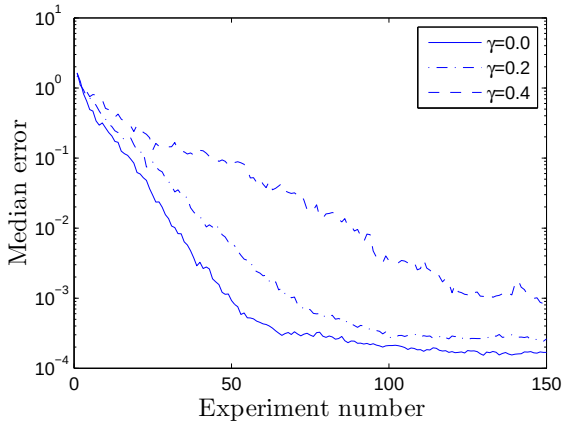


FIG. 7. Errors in inference for phase estimation with different levels of un-modeled noise, γ, for $T_2 = 1000$. In each instance the initial state was taken to be a fixed, but unknown, eigenstate of the Hamiltonian. The median was found for 100 randomly chosen values of the eigenphase of this fixed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

confused due to experimental results that are fundamentally inconsistent with the assumed likelihood function for the system.

Figure 7 shows that the inclusion of uncharacterized depolarizing noise does not actually prevent the eigenvalue from being estimated. Rather it reduces the number of bits per experiment that the algorithm can infer. We see this by fitting the exponential part of the data in Figure 7 (along with similar data for $\gamma = 0.1$, $\gamma = 0.3$, $\gamma = 0.5$) and find that the error decay exponent, λ, to shrink as roughly $\lambda \approx 0.17e^{-3.1\gamma}$ it does not prevent our algorithm from learning at an exponential rate (until depolarizing noise becomes significant). Thus RFPE continue to work even in the presence of depolarizing noise sources that are both strong and uncharacterized.

Appendix C: Stability of Rejection Filtering 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, m, drawn from the prior distribution. This breakdown occurs because the sample variance in the posterior mean is much greater than the difference between the prior mean and posterior 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.

In practice, the flatness of the distribution can be reduced by batching several experiments together and processing them. This process is in general inefficient, but can be made efficient if an appropriate value of κ_E is known for each datum recoreded.

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 $m \in \Omega(\alpha^2/\delta^2)$. □

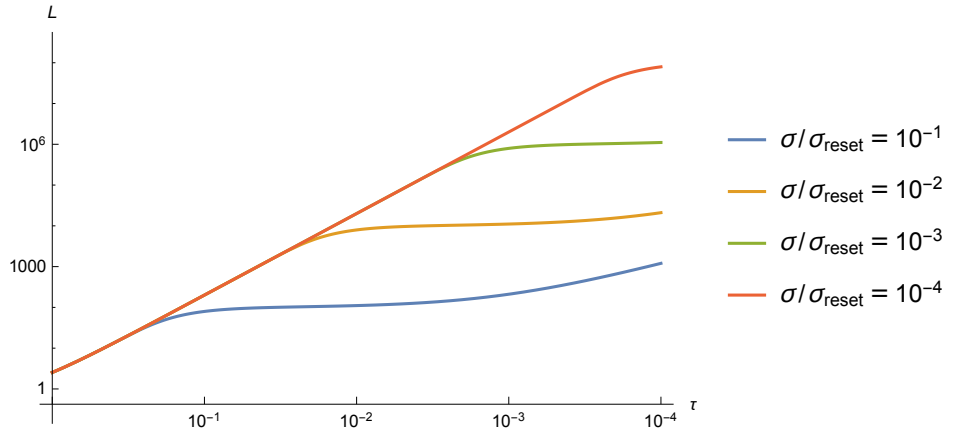


FIG. 8. Likelihood ratio test values for various settings of the parameter τ, and for various prior variances $\sigma/\sigma_{\text{reset}}$. In this example, we suppose that $\mu = \mu_{\text{reset}}$ and that $T_2 = 100$.

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: Bayes Factors for Reset Rule

Though in the main body, we have chosen to present the reset step in terms of p-values for familiarity, we note that p-values are difficult to correctly interpret and can lead to misunderstandings [21, 22]. As an alternative, one may prefer to use the Bayes factor. For a uniform prior over whether the rejection filter has failed, this reduces to the likelihood ratio test

$$L = \frac{\Pr(\text{result} = 1|\text{prior wrong})}{\Pr(\text{result} = 1|\text{prior correct})} \tag{D1a}$$

$$= \frac{1 - \mathrm{e}^{-(\tau^2\sigma_{\text{reset}}^2/2\sigma^2 + \sigma\tau/T_2)} \cos\left(\tau\left(\mu - \mu_{\text{reset}}\right)/\sigma\right)}{1 - \mathrm{e}^{-\tau^2/2}} \tag{D1b}$$

where μ_{reset} and σ_{reset} are the values of μ and σ immediately following a reset. Using this test, the degree by which $L > 1$ informs as to the degree to which we should prefer the model proposed by the reset rule. Again under the assumption of a uniform prior over the validity of the reset rule,

$$\Pr(\text{prior wrong}|\text{result} = 1) = L \Pr(\text{prior right}|\text{result} = 1). \tag{D2}$$

For instance, if the variance has been reduced by a factor of 100 from its initial value ($\sigma = \sigma_{\text{reset}}/100$) and the current mean is correct ($\mu = \mu_{\text{reset}}$), then assuming $T_2 = 100$ and an initial standard deviation of $\sigma_{\text{reset}} = \pi/\sqrt{3}$, $L \approx 8000$ for a result of 1. That is, the initial prior is 8,000 times as probably correct as the current prior in this example.

Importantly, this example rests on the assumption that one take a uniform prior over whether the current prior is valid. This assumption corresponds to that which an impartial observer might take in evaluating whether the numerical approximations made by our rejection filter algorithm have failed for the observed data. That is, the reset rule proposed corresponds performing an intervention without relying on the full experimental data record. Choosing a threshold other than $L = 1$ represents placing a different prior, as could be informed by observing the failure modalities of many runs of the rejection filter method. As demonstrated in Figure 8, because our reset rule resets with probability 1 if a 1 is observed, choosing τ effectively sets the threshold for L. In practice, however, because $\Pr(\text{result} = 1|\text{prior correct} \approx 0$, the reset rule is only weakly dependent on the specific threshold one places on L.

Appendix E: 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.

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 [15]. 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 RejF

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.

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

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

return (μ', σ')

Algorithm 2 Restarting algorithm

Input: Prior RejF state, records of all previous models found in the phase estimation algorithm μ, $\vec{\sigma}$, initial standard deviation σ_{init}, M, T_2, a counter CNT, Γ and τ.
Output: CNT, σ

```
function RESTART(μ, σ⃗, σ_init, M, T_2, CNT, Γ, τ)
    D ← derivative of log σ.
    if D ≥ Γ and CNT < 5 or rand()> exp(−M/T_2) then        ▷ Checks to see if the eigenstate is suspect.
        Perform experiment with M = τ/σ and θ = μ.
        if Outcome is 0 then                                 ▷ Test concludes state estimate is valid
            CNT ← CNT + 1.
            return CNT, σ.
        else                                                 ▷ Test concludes state estimate is invalid
            CNT ← 0
            σ ← σ_init
            return CNT, σ
        end if
    else                                                     ▷ Does not restart if state is not suspect
        CNT ← CNT + 1
        return CNT, σ.
    end if
end function
```

Algorithm 3 PGH for decoherent phase estimation using RejF

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

```
function PGH_REJF (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
```