Efficient Online Quantum Bayesian Inference

Nathan Wiebe[†], Christopher Granade[*], and $\cdots$
[†]*Quantum Architectures and Computation Group, Microsoft Research, Redmond, WA (USA) and*
[*]*Institute for quantum computing, Waterloo, ON (Canada)*

We present a new quantum algorithm for online quantum Bayesian inference. Our algorithm has a number of advantages over existing approaches, such as sequential Monte–Carlo methods, and is exponentially faster than existing quantum algorithms in an online setting. An important feature of our method is that it only tracks a single hypothesis as opposed to the thousands required by state of the art particle filter methods. Our method does so by sampling from the distribution over these single hypotheses to learn a simple model for the posterior distribution, which allows our algorithm to learn in an online fashion without measurement errors causing the posterior distribution to be lost. We further show that the success probability of our algorithm is optimal and also provide quantum methods for adaptive experiment design and learning in the presence of stochastically varying model parameters.

I. INTRODUCTION

Data processing has emerged within the last few years to be a problem of central importance within both academic circles and industry. The need for ever more sophisticated methods for processing data arose from the fact that modern experiments produce huge amounts of data and properly inferring information from huge data sets can require substantial computational power. A natural question to ask is whether quantum technologies can help quell the insatiable demand on computational time that statistical inference creates. We present here evidence that quantum computing can be used to speed up statistical inference relative to classical methods and show that limits exist on the extent to which such methods can be accelerated.

In particular, we focus on accelerating Bayesian inference. Bayes' rule is the heart of Bayesian inference, which gives the correct way to update a prior distribution that describes the experimentalist's initial beliefs about a system model when a piece of experimental evidence is received. If E be a piece of evidence and x denotes a candidate model for the experimental system then Bayes' rule states that the probability that the model is valid given the evidence (denoted $P(x|E)$) is

$$P(x|E) = \frac{P(E|x)P(x)}{\int_x P(E|x)P(x)\mathrm{d}x} = \frac{P(E|x)P(x)}{\langle P(E|x), P(x)\rangle}, \tag{1}$$

where $P(E|x)$ is known as the likelihood function and is assumed to either be either numerically or inferred emperically. This form of learning has a number of advantages. Firstly, it is highly robust to noise and experimental imperfections. Secondly, it is broadly applicable to almost every data processing problem. Finally, Bayesian inference provides a probability distribution rather than a point estimate of the model. This allows the uncertainty in the inferred parameter to be directly computed.

There are several features that can make Bayesian inference computationally expensive, especially for scientific applications. Perhaps the biggest contributor to the cost of these algorithms is the dimensionality of the model space. Typically the model x is parameterized by a vector in $\mathbb{R}^d$, which means that precisely performing an update requires integrating over an infinite number of hypotheses. Such integrals are often intractable, which limits the applicability of exact Bayesian inference. Another important issue is that the likelihood function is also often intractable in absentia of quantum computing. These features make exact Bayesian inference challenging.

A natural question to ask at this point is whether quantum computing could make inference tractable. This issue is has been recently discussed in [1], which uses ideas from quantum rejection sampling [2, 3] to accelerate the inference process. Their work leaves a number of important issues open. The method has success probability that shrinks exponentially with the number of updates attempted. Furthermore, the algorithm cannot be applied in an online fashion nor can it be applied to continuous problems or those with stochastically varying model parameters. We address these issues here by providing a quantum algorithm that can implement Bayesian inference in an online fashion by periodically classically caching a model of the posterior distribution. This approach to learning not only generalizes the prior Bayesian inference work, but also generalizes Grover's search to scenarios where queries do not provide definitive evidence about the marked state. We provide a complete description of this in the following.

II. QUANTUM BAYESIAN UPDATING

Quantum theory is often viewed as a generalization of classical probability theory, which suggests that the notion of a Bayesian update may require re–assessment in quantum computing. Indeed, there are several natural analogues of quantum Bayesian updating that can be thought of in a quantum setting. Our discussion will primarily focus on a form of quantum Bayesian inference that we call *semi–classical* Bayesian inference, which we introduce to address the shortcommings of quantum Bayesian inference.

Definition 1. *A quantum Bayesian update of a prior state $\sum_x \sqrt{P(x)}\,|x\rangle$ performs, for evidence E and likelihood function $P(E|x)$, the map*

$$|E\rangle \sum_x \sqrt{P(E|x)}\,|x\rangle \mapsto |E\rangle \sum_x \sqrt{P(x|E)}\,|x\rangle\,.$$

In order to formalize this notion of quantum Bayesian updating within an oracular setting we will further make a pair of assumptions.

1. There exists a self-inverse quantum oracle, O_E, that computes the likelihood function as a bit string in a quantum register: $O_E\,|x\rangle\,|y\rangle = |x\rangle\,|y \oplus P(E|x)\rangle$.
2. A constant Γ_E is known such that $P(E|x) \le \Gamma_E \le 1$ for all x.

The use of a value $\Gamma_E > 1$ can be thought of as transitioning from pure rejection sampling to importance sampling where the instrumental distribution is the distribution $\mathbb{1}/\Gamma_E$. In this vein, Γ_E couls also be taken to be a non–trivial function of x. This can reduce the posterior variance in the estimate of the true model parameter but we will not consider this case in detail as it purposefully distorts the posterior distribution.

The following lemma uses these components to implement a quantum Bayesian update. The result can be thought of as a generalization of the result of [1] to an oracular setting.

Lemma 1. *Given an initial state $\sum_x \sqrt{P(x)}\,|x\rangle$ and $\Gamma_E : P(E|x) \le \Gamma_E$ the state $\sum_x \sqrt{P(x|E)}\,|x\rangle$ can be prepared using an average number of queries to O_E that is in $O(\sqrt{\Gamma_E/\langle P(x), P(E|x)\rangle})$.*

Proof. Using a single call to O_E and adding a sufficient number of ancilla qubits, we can transform the state $\sum_x \sqrt{P(x)}\,|x\rangle$ into

$$\sum_x \sqrt{P(x)}\,|x\rangle\,|P(E|x)\rangle\,|0\rangle\,. \tag{2}$$

Then by adding a qubit and performing $R_y(2\sin^{-1}(P(E|x)/\Gamma_E))$ we can enact

$$\sum_x \sqrt{P(x)}\,|x\rangle\,|P(E|x)\rangle\,|0\rangle \mapsto \sum_x \sqrt{P(x)}\,|x\rangle\,|P(E|x)\rangle \left(\sqrt{\frac{P(E|x)}{\Gamma_E}}\,|1\rangle + \sqrt{1-\frac{P(E|x)}{\Gamma_E}}\,|0\rangle\right). \tag{3}$$

Then if the right most qubit register is measured and a result of 1 is obtained, the resultant state is

$$\frac{\sum_x \sqrt{P(x)P(E|x)}\,|x\rangle}{\sqrt{\sum_x P(x)P(E|x)}}, \tag{4}$$

which gives a probability distribution that corresponds to that expected by Bayes' rule. The probability of this occurring is $\sum_x P(x)P(E|x)/\Gamma_E$.

Since the process is heralded, amplitude amplification can be used to boost the probability of success for the successful branch quadratically [4]. Thus the average number of queries is in $O(\sqrt{\Gamma_E/\langle P(x), P(E|x)\rangle})$ as claimed. □

If the Bayesian algorithm is used solely to post–process information then one update will suffice to give the posterior distribution. In settings where experiments are chosen in an online fashion then many updates will be needed to reach the final posterior distribution. If L updates are required then the probability of success is

$$P_{\rm succ} \le \sum_x P(x)\left(\max_E \frac{P(E|x)}{\Gamma_E}\right)^L \le \sqrt{\sum_x P^2(x) \sum_x \left(\max_E \frac{P(E|x)}{\Gamma_E}\right)^{2L}} \le \sqrt{\sum_x \left(\max_E \frac{P(E|x)}{\Gamma_E}\right)^{2L}}\,. \tag{5}$$

This shows that the probability of success generically will shrink exponentially with L. The exponential decay of the success probability with L can be combatted by using amplitude amplification on the condition that all L updates are successful This reduces the expected number of updates needed to

$$O\left(\left(\sum_x P(x)\left(\min_E \frac{P(E|x)}{\Gamma_E}\right)^L\right)^{-1/2}\right), \tag{6}$$

but this strategy is obviously insufficient to rid the method of its exponentially shrinking success probability. Furthermore, we will see that there are fundamental limitations to our ability to avoid or correct such failures.

The reason why the errors in quantum rejection sampling cannot, in general, be efficiently corrected stems from the fact that quantum Bayesian inference algorithm described in Lemma 1 can be thought of as a generalization of Grover's algorithm [5]. Grover's problem (with one marked element) seeks to find $x = \text{argmax}(f(x))$ where $O(x)$ is a blackbox Boolean function that is promised to have a unique x_{mark} such that $O(x_{\text{mark}}) = 1$. The generalization to multiple marked elements is similar. The reduction between the two problems is formally proved below.

Lemma 2. *Grover's problem with m marked items reduces to Bayesian inference.*

Proof. Let O be a Boolean function that takes the value 1 iff $x \in X_m$ where $|X_m| = m$. Identifying the set X_m by querying this function is equivalent to Grover's problem. Consider the following likelihood function on a two–outcome space where 1 corresponds to finding a marked state and 0 corresponds to finding an un–marked state:

$$P(1|x) = \begin{cases} 1 & \text{if } x \in X_m \\ 0 & \text{otherwise} \end{cases}. \tag{7}$$

We also have that $P(0|x) = 1 - P(1|x)$, but this fact is not needed for the proof.

It is then easy to see that $P(1|x) = O(x)$ and thus a likelihood evaluation is equivalent to a query to O. This means that we can solve Grover's problem using the following algorithm.

1. Set the prior to be $P(x) = 1/N$.

2. Set $E = 1$, which corresponds to pretending that an experiment was performed that found a marked entry.

3. Compute $P(x|E) \propto P(E|x)P(x)$.

4. Output all x such that $P(x|E) = 1/m$.

The validity of this algorithm is easy to verify from (1) and it is clear that the posterior distribution is a uniform distribution over $x \in X_m$. Since $|X_m| = m$ and $P(x)$ is uniform, all elements in the support of the posterior distribution have probability $1/m$ and thus Grover's problem can be solved using Bayesian inference. This algorithm will succeed classically using N queries to O, rather than the $N - m$ queries required in the worst case scenario if a Bayesian framework is not adopted.

□

This reduction of Grover's problem to Bayesian inference brings with it tight lower bounds on the query complexity of solving the problem [6]. This means that we can exploit these bounds to show limitations on our ability to perform Bayesian inference on a quantum computer and also our ability to correct erroneous measurements that occur in the application of the method of Lemma 1. The following theorem states two such restrictions, which show that the method of Lemma 1 cannot be trivially improved nor can its failures be inexpensively corrected.

Theorem 1. *The following are impossible.*

1. *An algorithm capable of performing a quantum Bayesian update of an arbitrary pure quantum state of the form $\sum_{x=1}^N \alpha_x |x\rangle$ using $O(1)$ queries to O_E with success probability $\omega(\sqrt{\sum_x |\alpha_x|^2 P(E|x)/\Gamma_E})$.*

2. *An algorithm that requires $o(\sqrt{N}/\log(N))$ queries to O_E to correct a failure of an algorithm that performs a quantum Bayesian update using $O(1)$ queries with probability of success $\Omega(\sqrt{\sum_x |\alpha_x|^2 P(E|x)/\Gamma_E})$.*

Proof. Seeking a contradiction, assume that there exists a quantum algorithm that can perform a Bayesian update with probability $\omega(\sqrt{\sum_x |\alpha_x|^2 P(E|x)})$ for any likelihood function $P(E|x)$. Next, let us choose the likelihood function to be that used in (7) in the reduction proof of Lemma 2 and take $\alpha_x = 1/\sqrt{N}\ \forall\ x$. It is clear from Lemma 1 that $\Gamma_E = 1$ must be chosen for this problem. Then by assumption the state $|x_m\rangle$ where $x_m \in X_m$ can be found with probability $\omega(\sqrt{\sum_x |\alpha_x|^2 P(E|x)}) = \omega(1/\sqrt{N})$. Since each application of the algorithm requires $O(1)$ queries, $o(\sqrt{N})$ queries are needed on average to learn x_m, which violates lower bounds for the average query complexity for Grover's problem [6]. Therefore such a quantum Bayesian inference algorithm is impossible.

Again seeking a contradiction, consider the following likelihood function with outcomes $\{1, 0\}$

$$P(1|x) = \begin{cases} 2/3, & x = x_m \\ 1/3, & x \neq x_m \end{cases}, \; P(0|x) = 1 - P(E|x) \tag{8}$$

For each x, $P(1|x)$ can be computed using a single query to O and vice versa, thus a query to this likelihood function is equivalent to a call to O. Thus (1) gives that the posterior probability after measuring 1 is

$$\frac{2P(x_m)}{1 + P(x_m)} = 2P(x_m) + O(P(x_m)^2). \tag{9}$$

Therefore $O(\log(N))$ measurements of 1 suffice to amplify the probability from $1/N$ to $\Theta(1)$. In practice, it would be unlikely that $O(\log(N))$ sequential measurements would all yield 1 (i.e. give noisy information about the marked state), but the user of a quantum Bayesian updating algorithm can always pretend such evidence was obtained similar to Lemma 2. This assumption is key to our reduction of Grover's search to an online Bayesian inference problem.

If we pretend to have measured $O(\log(N))$ such values then Lemma 1 shows that there exists a quantum algorithm that can perform each such update with probability of success

$$\Omega\left(\sqrt{\sum_x |\alpha_x|^2 P(1|x)}\right) \in \Omega(1), \tag{10}$$

since $P(1|x) \geq 1/3$ and $\sum_x |\alpha_x|^2 = 1$. If we were not able to correct errors then (10) shows that the probability of successfully inferring the marked state is $O(\mathrm{poly}(1/N))$ since $O(\log(N))$ updates are needed; however, by assumption each failure can be corrected using $o(\sqrt{N}/\log(N))$ queries. Therefore by attempting quantum Bayesian updates, correcting any errors that might occur and repeating until success, a successful update can be obtained with an average number of queries that is in

$$o\left(\frac{\sqrt{N}}{\log(N)\sqrt{\sum_x |\alpha_x|^2 P(Y|x)}}\right) \in o\left(\frac{\sqrt{N}}{\log(N)}\right). \tag{11}$$

Since there are $O(\log(N))$ such queries, the marked state can be inferred within probability $p \in \Theta(1)$ using $o(\sqrt{N})$ queries to the likelihood function. Since a query to the likelihood function is equivalent to a query to Grover's oracle, an inexpensive error correction algorithm is also impossible for algorithms with complexity similar to that of Lemma 1. □

These impossibility results show that quantum Bayesian updating cannot be made efficient without making assumptions about the underlying prior distributions or likelihood functions. This inneficiency is perhaps unsurprising as exact Bayesian inference is also classically inefficient. In particular, an efficient sampling algorithm from a distribution that is a close approximation to the posterior distribution would imply $\mathsf{P} = \mathsf{NP}$ [7]. A quantum algorithm capable of efficient Bayesian inference for general models would similarly imply that $\mathsf{NP} \subseteq \mathsf{BQP}$, which is false under reasonable complexity theoretic conjectures.

Although it may not be surprising that quantum Bayesian updating is not generically efficient, it is perhaps surprising that both it and classical updating fail to be efficient for different reasons. Classical Bayesian updating fails to be efficient because it needs to store prior and posterior probabilities for an exponentially large number of hypotheses; however, its cost scales linearly with the number of updates used. In contrast, quantum Bayesian updating scales polynomially with the number of hypotheses considered but scales exponentially with the number of updates. This begs the question of whether it is possibile to combine the best features of quantum and classical Bayesian updating. We do so in the subsequent section, wherein we show how a classical model can be stored for the system that can be reverted to in the event that a failure is observed in a quantum Bayesian update.

III. SEMI–CLASSICAL BAYESIAN UPDATING

Approximations are therefore often needed to make both classical as well as quantum Bayesian inference tractable. However, the purpose of these approximations is very different. Classical methods struggle when dealing with probability distributions in high–dimensional spaces, and sophisticated methods like sequential Monte–Carlo approximations are often employed to reduce the effective dimension [8–10]. However, the non–linear nature of the update rule and the problem of extracting information from the posterior distribution are not issues. Our quantum algorithm has the exact opposite strengths and weaknesses: it can easily cope with exponentially large spaces but struggles emulating the non-linear nature of the update rule.

We attack the problem by making our quantum algorithm a little more classical, meaning that through out the learning process we aim to learn an approximate classical model for the posterior alongside the quantum algorithm. This classical model allowsus to approximately re–prepare the state should an update fail throughout the updating process. This removes the exponential scaling, but results in an approximate inference. We refer to this procedure as *quantum resampling* as it is reminiscent of resampling in sequential Monte–Carlo algorithms or other particle filter methods such as ADF [9]. In order to prepare the distribution The posterior distribution is modeled as a Gaussian distribution with mean and covariance equal to that of the true posterior. This choice is sensible because once the Gaussian distribution is specified, the Grover–Rudolph state preparation method can be used to prepare such states as their cummulative distribution functions can be efficiently computed. Alternatively, for one–dimensional problems, such states could be manufactured by approximate cloning.

We define a semi–classical Bayesian update below

Definition 2. *A semi–classical Bayesian update of a prior state, for a piece of experimental evidence E and likelihood function $P(E|x)$ and family of probability distributions $F(x;\rho)$ parameterized by the vector ρ, maps*

$$|E\rangle \sum_x \sqrt{P(E|x)}\,|x\rangle \mapsto \rho : F(x;\rho) \approx P(x|\rho).$$

We call this process semi–classical updating because the processs yields an approximate classical model for the posterior distribution. This model can take many forms in principle, but for example we could consider this model to be a Gaussian distribution that has the same mean and standard deviation as the posterior distribution. Semi–classical Bayesian updating will be discussed in more detail in the following section, but for now we will focus on quantum Bayesian updating.

We need a means to measure the expectation values and components of the covariance matrix of the posterior for this method to work. We provide such a method, based on the Hadamard test, below.

Lemma 3. *Given a unitary operator U such that $U|0\rangle = \sum_x \sqrt{P(x)}\,|x\rangle$, an observable $\Lambda = \sum_x \lambda_x |x\rangle\langle x|$ and an estimate $\lambda_0 : \max_x |\lambda_x - \lambda_0| \le \Delta\lambda$, there exists a quantum algorithm to estimate $(\langle\psi|\Lambda|\psi\rangle - \lambda_0)$ within error ϵ with probability at least $8/\pi^2$ using $\tilde{O}(\Delta\lambda/\epsilon)$ applications of U and queries to an oracle O_λ such that $O_\lambda |x\rangle |0\rangle = |x\rangle |\lambda_x\rangle$.*

Proof. By following the reasoning in the prior lemma, we can prepare the following state using one query to O_λ and one application of U:

$$\sum_x \sqrt{\frac{P(x)}{2}}\,|x\rangle\,|\lambda_x\rangle \left(\sqrt{1+\frac{\lambda_x-\lambda_0}{\Delta\lambda}}\,|1\rangle + \sqrt{1-\frac{\lambda_x-\lambda_0}{\Delta\lambda}}\,|0\rangle\right). \tag{12}$$

The probability of measuring 1 is

$$\sum_x \frac{P(x)}{2}\left(1+\frac{\lambda_x-\lambda_0}{\Delta\lambda}\right) = \frac{1}{2} + \frac{\langle\psi|\Lambda|\psi\rangle - \lambda_0}{2\Delta\lambda}. \tag{13}$$

This probability can be learned within additive error δ using $O(1/\delta^2)$ samples and hence $\langle\psi|\Lambda|\psi\rangle - \lambda_0$ can be learned within error ϵ using $O(\Delta\lambda^2/\epsilon^2)$ samples.

This probability can also be learned using the amplitude estimation algorithm. Amplitude estimation requires that we mark a set of states in order to estimate the probability of measuring a state within that set. Here we mark all states in (13) where the rightmost qubit is 1. The amplitude estimation algorithm then requires $O(1/\delta)$ queries to U and the above state preparation method to estimate the probability to within error δ and store it in a qubit register [4]. Amplitude estimation has a probability of success of at least $8/\pi^2$, hence $\tilde{O}(1/\delta)$ queries are needed to achieve arbitrarily large success probability. The result then follows from taking $\delta = \epsilon/\Delta\lambda$. □

Lemma 3 is crucial for our arguments because it provides a method for learning not just the mean of the posterior distribution but also the standard deviation. This allows us to infer a two parameter model for the posterior distribution in cases where the model is one–dimensional. In particular, using $\lambda_x = x$ in Lemma 3 yields the mean x_0. The variance can then be computed by taking $\lambda_x = x^2$ and using the fact that $\sigma^2 = \langle x^2\rangle - \langle x\rangle^2$. The generalization to higher dimensions is straight forward.

Corollary 1. *Assume that the components of the mean of the prior distribution over model vectors are* $\{\lambda_{0,j} : j = 1 \ldots D\}$ *and* $\Delta\lambda \geq \max_j |\langle \Lambda_j \rangle - \lambda_{0,j}|$. *The mean and covariance matrix can be computed within error* ϵ *using* $\tilde{O}\left(\frac{D^2 \Delta\lambda \Gamma_E}{\epsilon \langle w, P(E|x) \rangle}\right)$ *queries to* O_E.

Proof. In order to apply the reasoning of Lemma 3 to estimate the posterior mean and variance, we need to prepare the following state conditioned on evidence E

$$\sum_x \sqrt{\frac{P(x|E)}{2}} |x\rangle |\lambda_x\rangle \left(\sqrt{1 + \frac{\lambda_x - \lambda_0}{\Delta\lambda}} |1\rangle + \sqrt{1 - \frac{\lambda_x - \lambda_0}{\Delta\lambda}} |0\rangle\right), \tag{14}$$

where $\lambda_x = x$ or $\lambda_x = x^2$.

Let $P(1) = \sum_x P(x)P(E|x)/\Gamma_E$ be the probability of performing the Bayesian update. Let $P(11)$ be the probability of both performing the update and measuring the right most qubit in (12) to be 1. This probability is

$$P(11) = \frac{P(1)}{2} \left(1 + \frac{\langle \Lambda_k \rangle - \lambda_{0,k}}{\Delta\lambda}\right), \tag{15}$$

where Λ_k is an observable that reports either the k^{th} mean or the k^{th} element of the covariance matrix and $\langle \cdot \rangle$ refers to the expectation of a quantity in the posterior state. Therefore $\langle \Lambda_k \rangle$ can be computed from $P(11)$ and $P(1)$ via

$$\langle \Lambda_k \rangle = \left(2\frac{P(11)}{P(1)} - 1\right) \Delta\lambda + \lambda_{0,k}, \tag{16}$$

Therefore if we estimate $P(1)$ and $P(11)$ within error $O(\delta)$ then the error in $\langle \Lambda_k \rangle$ is from calculus $O(\delta/P(1))$ since $P(1) \geq P(11)$. Therefore $\langle \Lambda_k \rangle$ can be estimated to within error ϵ if $\delta \in O(\epsilon P(1)/\Delta\lambda)$. Bounds on the cost of amplitude estimation give the cost of this to be [4]

$$\tilde{O}\left(\frac{\Gamma_E \Delta\lambda}{\epsilon \sum_x P(x)P(E|x)}\right). \tag{17}$$

The result then follows from noting that there are $O(D^2)$ different values of k that need to be computed. □

This shows that if we require modest relative error (i.e. $\Delta\lambda \in O(\epsilon)$) and Γ_E is reasonably tight then this process is highly efficient. In contrast, previous results that do not use these factors that incorporate apriori knowledge of the scale of these terms may not be efficient under such assumptions.

Below we combine these ideas to construct an online quantum algorithm that is capable of efficiently processing a series of L pieces of data before outputting a classical model for the posterior distribution in the quantum device. This result is key to our argument because it provides a result that one can fall back on if an update fails, thereby removing the exponentially shrinking success probability from existing quantum Bayesian inference methods.

Theorem 2. *Let* $F(x; \mu, \Sigma)$ *be a family of approximations to the posterior distribution parameterized by the posterior mean* μ *and the posterior covariance matrix* Σ *and* $\{E_k : k = 1, \ldots L\}$ *be a sequence of* L *pieces of evidence. Then a semi–classical update of a quantum state* $\sum_x \sqrt{P(x)} |x\rangle$ *can be performed using a number of queries to* O_E *that is in*

$$\tilde{O}\left(\frac{LD^2 \Delta\lambda}{\epsilon \langle P(x), \prod_{k=1}^L P(E_k|x)/\Gamma_{E_k} \rangle}\right).$$

Proof. The algorithm is simple.

1. Perform L quantum Bayesian updates, but without measuring the qubits that determine whether the updates succeed or fail.

2. Use the method of Corollary 1 to learn the mean and covariance matrix of the quantum posterior distribution.

3. Return these quantities, which give a parameterization of the approximation to the posterior distribution.

After step 1, we have from the independence of the successes that the probability of all L updates succeeding is $\sum_x P(x) \prod_{k=1}^L P(E_k|x)/\Gamma_{E_k}$. Thus step 2 can be performed using $\tilde{O}\left(\frac{D^2 \Delta\lambda}{\epsilon \langle w, \prod_k P(E_k|x)/\Gamma_{E_k} \rangle}\right)$ preparations of the posterior state from Corollary 1. Since each such preparation requires L queries to O_E the total query complexity required to learn the posterior mean and variance is

$$\tilde{O}\left(\frac{LD^2\Delta\lambda}{\epsilon\langle P(x), \prod_{k=1}^{L} P(E_k|x)/\Gamma_{E_k}\rangle}\right) \tag{18}$$

Finally, since the algorithm outputs the mean and covariance matrix of the posterior distribution, it outputs a function $F(x;\mu,\sigma)$ that captures (to within error ϵ) the first two moments of the posterior distribution. Thus the algorithm clearly performs a semi–classical update as per Definition 2. □

Comparing this to SMC methods, we see that if we are given a likelihood function as a classical oracle then the number of queries needed is

$$\tilde{O}(N_{\rm part}L), \tag{19}$$

where $N_{\rm part}$ is the number of particles in the SMC cloud approximating the prior distribution. It is known that $N_{\rm part}$ scales sub–exponentially with D, however in practice tens of thousands of particles are often needed for simple problems. In the quantum case, only one particle is needed so this often will constitute a significant speed advantage over SMC if the remaining terms are $\max_k \frac{\Gamma_{E_k}\Delta\lambda}{\epsilon\sum_x P(x)P(E_k|x)} \in O(1)$, which we expect for many realistic problems. Also, the quantum algorithm yields the expectation value over the true posterior distribution without resorting to the approximate posteriors yielded by SMC methods.

As a final note, since amplitude estimation is used liberally in these protocols it is important to reduce the cost of this algorithm as much as possible. Although amplitude estimation is near-optimal it does not by default use prior information that can be used to accelerate the learning process. Below we formally prove a theorem, which is proposed informally in [11].

Theorem 3. *Let U be a unitary operator such that $U|0\rangle = \sqrt{a}|\phi\rangle + \sqrt{1-a}|\phi^\perp\rangle$ where $\langle\phi|\phi^\perp\rangle = 0$ for $0 < a \le a_0 < 1$, $1 - a_0 \in \Theta(1)$ and let G be a projector such that $S|\phi\rangle = -|\phi\rangle$ and $S|\phi^\perp\rangle = |\phi^\perp\rangle$. Then a can be estimated to within error ϵ using $\tilde{O}(\sqrt{a_0}/\epsilon)$ applications of S with high probability.*

Proof. Our proof follows the same intuition as that of the proof of amplitude estimation in [4] except rather than performing amplitude estimation on $U|0\rangle$ we use amplitude amplification to first boost the probability and then use amplitude estimation to learn the boosted probability. The actual value of a is then inferred from the amplified value of a learned in the amplitude estimation step.

First by following Lemma 1 in [4] we can apply a sequence of reflection operators that contains m applications of S to form a unitary operation V such that performs, up to a global phase,

$$V|0\rangle = \sin((2m+1)\sin^{-1}(\sqrt{a}))|\phi\rangle + e^{i\theta}\cos((2m+1)\sin^{-1}(\sqrt{a}))|\phi^\perp\rangle. \tag{20}$$

Since V is a unitary operation, amplitude estimation can be used to learn $\sin^2((2m+1)\sin^{-1}(\sqrt{a}))$ to within error ϵ by using Theorem 12 of [4] with probability at least $8/\pi^2$ using $O(1/\delta)$ applications of V. Thus using the Chernoff bound, $\sin^2((2m+1)\sin^{-1}(\sqrt{a}))$ can be estimated within the same error tolerance using $\tilde{O}(1/\delta)$ operations with high probability.

Since V contains m S operators, the total number of applications of S needed to infer this is $O(m/\delta)$. However, although $\sin^2((2m+1)\sin^{-1}(\sqrt{a}))$ is inferred within error δ, this does not imply that a is. If we define this estimated value to be y and assume that $0 \le (2m+1)\sin^{-1}(\sqrt{a}) \le \pi/2$ then

$$a = \sin^2\left(\frac{\sin^{-1}(\sqrt{y})}{2m+1}\right). \tag{21}$$

If there is an error of $O(\delta)$ in y then Taylor analysis implies that

$$a = \sin^2\left(\frac{\sin^{-1}(\sqrt{y})}{2m+1}\right) + O\left(\frac{\delta}{m^2\sqrt{1-a}}\right). \tag{22}$$

Since $a < 1$ the error is $O(\delta/m^2)$. Hence if we desire error ϵ in a then it suffices to take $\delta \in O(\epsilon m^2)$. Thus $\tilde{O}(1/\epsilon m)$ applications of G are needed to infer a to within error ϵ.

Although this may seem to suggest that taking large m always leads to a better inference of a, this is not necessarily true for this inversion process. This is because if

$$m > \frac{1}{2}\left(\frac{\pi}{2\sin^{-1}(\sqrt{a})} - 1\right) \tag{23}$$

then (21) no longer holds. Ergo $m \in O(1/\sin^{-1}(\sqrt{a})) \in O(1/\sqrt{a})$ for small a. Since the user does not know a, the best that can be done is to take $m \in \Theta(1/\sqrt{a_0})$ since taking $a = a_0$ also guarantees (23) does not hold for a. Therefore the number of applications of G, for $m \in \Theta(1/\sqrt{a_0})$, needed to learn a within error ϵ with high probability scales as

$$\tilde{O}\left(\frac{1}{\epsilon m}\right) \in \tilde{O}\left(\frac{\sqrt{a_0}}{\epsilon}\right), \tag{24}$$

as claimed. □

IV. APPROXIMATING THE POSTERIOR

There are many methods for preparing states that represent probability distributions using a quantum computer. By the linearity of quantum mechanics, any method of doing so which is coherent can then be used to prepare a mixture distribution such as is used in the SMC approximation. As a result, each such method then corresponds to a quantum resampling procedure.

Perhaps the most straightforward is the method of Grover and Rudolph which provides an efficient algorithm for preparing a probability distribution that is efficiently integrable. This work is further elaborated by the result of Kitaev and Webb [12] which shows that Gaussian states can be efficiently prepared using a quantum computer. Here we focus on preparing a simpler class of functions to use within the resampler that can be easily constructed using Fourier transforms.

Lemma 4. *Let k be a non–negative integer then a quantum computer can prepare a quantum state of the form $\sum_{x=0}^{2^n-1} e^{i\phi(x)}\sqrt{P(x)}\,|x\rangle$ where*

$$P(x) = \frac{\sin^2(\pi 2^k x/2^n)}{2^{k+n}\sin^2(\pi x/2^n)},$$

using at most $O(n\log(n/\epsilon))$ one and two–quibit gates taken from the set $\{H, P(\theta), \Lambda(P(\theta))\}$ where $P(\theta) : |x\rangle \mapsto e^{2\pi i\theta x}|x\rangle$ for $\theta \in \{y/2^n : y \in \mathbb{Z}_{2^n}\}$ and $\Lambda(P(\theta))$ is its controlled counterpart.

Proof. This probability distribution can be prepared using the following steps.

1. Prepare for integer $k > 0$ the state $|\psi\rangle = \frac{1}{\sqrt{2^k}}\sum_{j=0}^{2^k-1}|j\rangle$ using k Hadamard gates.

2. Apply the Pauli operator $Z = P(1/2)$ to the least significant bit in $|\psi\rangle$.

3. Apply QFT to the result.

This produces the required state because applying Z to the least significant qubit in $|\psi\rangle$ gives

$$\frac{1}{\sqrt{k}}\sum_{j=0}^{k-1} e^{-i\pi j}\,|j\rangle\,, \tag{25}$$

using the shift property of DFTs it is clear that this phase shift displaces the Fourier transform to the middle of the spectrum. Then using the shift property of the discrete Fourier transforms again along with the formula for the DFT of a window function gives that

$$P(x) = \frac{\sin^2(\pi 2^k x/2^n)}{2^{k+n}\sin^2(\pi x/2^n)}. \tag{26}$$

This formula can also be easily proved using the discrete Fourier transform of the Kronecker–delta function and the linearity of the Fourier transform. Since $k \le n$ the cost of the circuit is dominated by that of the Fourier transform, which is $O(n\log(n/\epsilon))$ using this gate set according to [13]. □

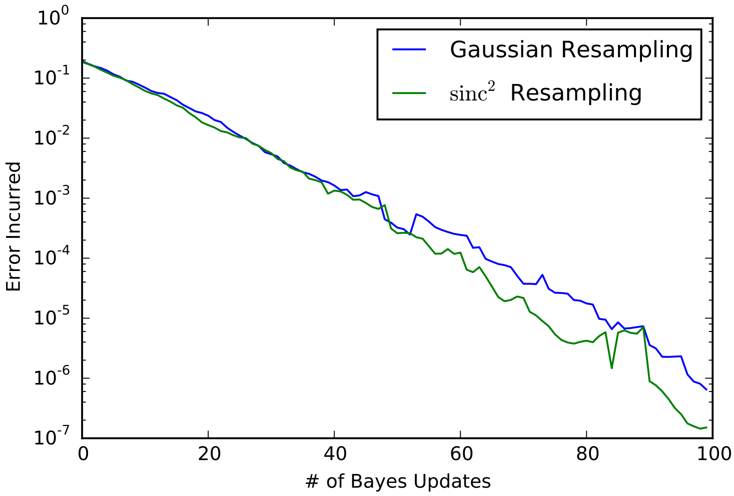


FIG. 1: Performance of classical SMC with Gaussian and sinc^2 resampling kernels for the periodic likelihood (27) and averaged over 1,000 trials.

Note that there is residual phase present in the quantum state in Lemma 4 because the initial window is not centered at $x = 0$. This can be corrected, if necessary, by using an adder or by applying a phase rotation in Fourier space. Also, the result of [13] shows that the depth of the resulting circuits is actually much smaller: it is $O(\log(n) + \log\log(1/\epsilon))$. These costs can also be translated into circuits over the Clifford + T gate library by incurring a multiplicative cost that is at most $O(\log(n/\epsilon))$ using [14] or related methods.

Unlike the method of [12], the above method is not capable of modeling correlations in many–variable systems. It instead can be used to model the posterior distribution as a product of independent distributions. This tends not to be a major issue though because for most well posed learning problems the posterior distribution tends to a unimodal distribution, but potentially can cause problems in cases where the distribution is bimodal or has non–trivial correlations. The latter issue can be addressed, in part, by using principal component analysis (PCA) which re-expresses the problem in the eigenbasis of the covariance matrix where no correlations exist. Since the model dimension is generically low, PCA can be considered to be efficient. Quantum techniques can also be used to sample efficiently from the principal components of the distribution, under certain circumstances [15], although it is uncertain whether this capability will be of use here.

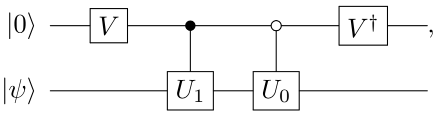
A potential flaw of using sinc–based resampling is that it has many nodes which will be ascribed zero probability in the initial prior. If the true model happens to reside at, or near, one of these nodes then the updating procedure may accidentally omit the true model during a quantum resampling step. This can be corrected through many approaches, such as resorting to the more expensive Gaussian state preparation methods of [12, 16].

We examine whether this poses an issue for the periodic likelihood

$$\begin{aligned} P(1|\omega;\omega_-,t) &= \cos^2((\omega-\omega_-)t), \\ P(0|\omega;\omega_-,t) &= \sin^2((\omega-\omega_-)t), \end{aligned} \tag{27}$$

where ω_- and t are experimental parameters. We choose ω_- according to the *particle guess heuristic* of [17] , which randomly chooses $\omega_- \sim P(\omega)$ and $t = 1/(\omega_- - \omega')$ where $\omega' \sim P(\omega)$. (27). Although our algorithm is hard to exactly simulate classically, we show that classical Liu-West resampling continues to work using sinc^2 as a resampling kernel, such that we can reasonably expect that sinc^2 will be effective in the quantum SMC case as well. In particular, after 100 Bayes updates with classical SMC and the sinc^2 resampling kernel, the error incurred is comparable to that incured by resampling with Gaussian kernels. We therefore do not suspect that sinc–based resampling will typically degrade the performance of the inference algorithm.

Although we do not expect the nodes of the sinc^2 function to substantially impede inference, the impact of these nodes can be reduced by using a linear combination of such states. In particular, we can prepare linear combinations of step functions as our initial states and Fourier transform the result. Linear combinations of states can be prepared using the circuit



where for some real valued $A \geq 0$ [18] we have

$$V = \begin{pmatrix} \sqrt{\frac{A}{A+1}} & -\sqrt{\frac{1}{A+1}} \\ \sqrt{\frac{1}{A+1}} & \sqrt{\frac{A}{A+1}} \end{pmatrix}. \tag{28}$$

This is relevant because if the top-most qubit is measured to be 0 then the circuit performs

$$|\psi\rangle \to \frac{(AU_1 + U_0)\,|\psi\rangle}{|(AU_1 + U_0)\,|\psi\rangle\,|}. \tag{29}$$

This allows linear combinations of initial states to be prepared from $|0\rangle$ with probability

$$P_{\rm succ} = \left|\frac{(AU_1 + U_0)\,|0\rangle}{A+1}\right|^2. \tag{30}$$

Assuming $U_1\,|0\rangle = \frac{1}{\sqrt{2^{k_1}}}\sum_{j=0}^{2^{k_1}-1}|j\rangle$ and $U_2\,|0\rangle = \frac{1}{\sqrt{2^{k_2}}}\sum_{j=0}^{2^{k_2}-1}|j\rangle$ for $k_2 \geq k_1$ then it is straight forward to see from (30) we have that a linear combination of two states can be formed with probability

$$P_{\rm succ} = \frac{\left(\frac{A}{\sqrt{2^{k_1}}} + \frac{1}{\sqrt{2^{k_2}}}\right)^2 k_2 + \left(\frac{A}{\sqrt{2^{k_1}}}\right)^2 (2^{k_1} - 2^{k_2})}{(A+1)^2}. \tag{31}$$

Since the probability of successfully combining the two unitaries via measurement and postselection is known, amplitude amplification can be used to make the process deterministic [19]. The probability of success for the case where a polynomial number of terms are summed can also be computed and is furthermore also efficient.

V. FILTERING DISTRIBUTIONS FOR TIME-DEPENDENT MODELS

In practice a physical system whose properties we want to infer is seldom time independent. Moreover, by demanding time-invariance, we preclude applications to many interesting problem domains outside of physics, such as in financial modeling and computer vision. In such cases, the likelihood function $P(E|x)$ is replaced by $P(E|x;\tau)$ where τ is the time in the experimental system. Thus, the techniques developed do not directly apply to time-dependent cases, but rather to the model that results from marginalizing over this time- dependence.

There are several ways of dealing with problems involving a time-dependent likelihood. The most natural way is by introducing new parameters, called *hyperparameters*, that allow the variation of the likelihood function to be modeled. Estimation and inference then proceed on the hyperparameters, rather than on the model parameters directly. For instance, letting ω in the periodic likelihood (27) be drawn from a stationary Gaussian process and then marginalizing over the history of that process results in a new hyperparameterized likelihood

$$P(1|\mu,\sigma;\omega_-,t) = \frac{1}{2}\left(e^{-\frac{1}{2}\sigma^2 t^2}\cos(\mu t) + 1\right), \tag{32}$$

where μ and σ^2 are the mean and variance of the Gaussian process. Using hyperparameters works well for modeling the distribution of the dynamics of a model and can be directly implemented using the previously discussed methods, but it does little to help track the *instantaneous* model parameters of the system.

To extend to track the instantaneous states of an unknown process, we note that a major application of classical SMC algorithms is tracking the positions or time-dependent properties of an object. This application can be challenging because as Bayesian inference proceeds the certainty in the position of the SMC particles tends to increase, but if the true hypothesis drifts, then this will not be represented by Bayes updates alone. This means that the true model can easily drift into a region that the is not supported by the a prior obtained by previous updates that neglected the stochasticity of the model parameters. This in turn will cause the update to fail, such that the algorithm will no longer be able to track the position of the object.

SMC can be made to track stochastically varying model parameters, by incorporating a prediction step that diffuses the model parameters of each particle [20]. Here, we extend this technique to the our quantum algorithm by performing Bayes updates on QFT-transformed posterior states. This will allow our algorithm to continue enjoying dramatic advantages in space complexity even in the presence of time-dependence.

In particular, by convolving the prior with a filter function such as a Gaussian, the width of the resultant distribution can be increased without affecting the prior mean. This means that the Bayes estimate of the true model will remain identical while granting the prior the ability to recover from time-variation of the true model parameters. In particular, if we assume that at each step Bayesian inference causes the posterior variance to contract by a factor of α and convolution with a filter function causes the variance to expand by β then the viariance of the resulting distribution asymptotes to $\beta/(1-\alpha)$. Thus we can combat $\sigma(x)$ from becoming unrealistically small by applying such filtering strategies.

The convolution property of the Fourier transform gives for any two functions P and Q

$$P \star Q \propto \mathcal{F}^{-1}\left(\mathcal{F}(P)\cdot\mathcal{F}(Q)\right), \tag{33}$$

where $\star$ is the circular convolution operation. The quantum Fourier transform can therefore be used to convolve an unknown P with a known distribution Q that has an efficiently computable Fourier transform $\hat{Q}$. This convolution allows us to filter the prior distribution.

Theorem 4. *Let $O_{\hat{Q}}$ be a quantum oracle such that $O_{\hat{Q}}\,|x\rangle\,|y\rangle = |x\rangle\left|y\oplus\sin^{-1}(\hat{Q}(x)/\Gamma_E)\right\rangle$ where $\hat{Q} := \mathcal{F}(Q)$ and $\hat{Q}(x)\le\Gamma_E$ and $Q(x)\in\mathbb{C}^{2^n}$. Then given access to a unitary oracle $O_{\rm in}$ that prepares the state $\sum_x\sqrt{P(x)}\,|x\rangle$, the state $\sum_x\sqrt{(P\star Q)(x)}\,|x\rangle$ can be prepared using a number of queries that has an average-case query complexity of $O(\sqrt{\Gamma_E/\langle\mathcal{F}(P),\mathcal{F}(Q)\rangle})$ and requires on average $O(n\log(n/\epsilon)\sqrt{\Gamma_E/\langle\mathcal{F}(P),\mathcal{F}(Q)\rangle})$ one– and two–qubit gates from the set $\{H, P(\theta), \Lambda(P(\theta))\}$ where $P(\theta) : |x\rangle\mapsto e^{2\pi i\theta x}\,|x\rangle$ for $\theta\in\{y/2^n : y\in\mathbb{Z}_{2^n}\}$ and $\Lambda(P(\theta))$ is its controlled counterpart.*

Proof. Notice, however, that Bayes updating the quantum SMC state consists of pointwise multiplication. As a result, applying Lemma 1 in the Fourier domain, we can implement the convolution described above. Doing so involves the following process

1. Fourier transform the current posterior, $|P\rangle := \sum_x P(x)\,|x\rangle \mapsto \hat{\mathcal{F}}\left(\sum_x P(x)\,|x\rangle\right) := \sum_k \omega_k\,|k\rangle$.
2. Prepare the Fourier-domain representation of the convolution kernel,
$\sum_k \omega_k\,|k\rangle \mapsto \sum_k \omega_k\,|k\rangle\left|\sin^{-1}(\sqrt{\hat{Q}(k)/\Gamma_E})\right\rangle$.
3. Update by the convolution kernel and transform back,
$\sum_k \omega_k\,|k\rangle\left|\sin^{-1}(\sqrt{\hat{Q}(k)/\Gamma_E})\right\rangle \mapsto \hat{\mathcal{F}}^{-1}\left(\sum_k \omega_k\,|k\rangle\,|0\rangle\left(\sqrt{\hat{Q}(k)/\Gamma_E}\,|1\rangle + \sqrt{1-\hat{Q}(k)/\Gamma_E}\,|0\rangle\right)\right)$.

If 1 is measured then the result will implement the circular convolution $P\star Q$ according to (33) and Plancherel's theorem.

First, the query complexity of this algorithm is easy to estimate. The initial state preparation requires a query to $O_{\rm in}$ and the calculation of $\hat{Q}(k)$ requires a query to $P_{\hat{Q}}$. By using amplitude amplification on the 1 result, we have that on average $O(\sqrt{\Gamma_E/\langle\mathcal{F}(P),\mathcal{F}(Q)\rangle})$ queries are required to prepare the state.

The number of non–query operations is the number of operations needed per attempt at rejection sampling multiplied by the number of attempts made. Step 1 requires $O(n\log(n/\epsilon))$ gate operations [13] to implement the Fourier transform and its inverse. Step 2 only requires query operations. Step 3 requires $O(\log(1/\epsilon))$, controlled rotations each of which can be implemented using a constant number of one– and two–qubit operations to ensure that the rotation on the ancilla qubit is successfully implemented within error $O(\epsilon)$. It also requires performing the inverse quantum Fourier transform which involves performing $O(n\log(n/\epsilon))$ gates. Summing these costs gives $O(n\log(n/\epsilon))$. The quoted scaling of the circuit size then follows by multiplying this by the number of attempts needed to achieve a successful result using amplitude amplification. □

VI. ADAPTIVE EXPERIMENT DESIGN

Bayesian methods can also be used in an online fashion to design new experiments to perform, given the current knowledge that one has about a system of interest. Here, we show that our algorithm allows for quantum computing to be used to perform Bayesian experiment design with significant advantages over classical methods.

In practice, Bayesian experiment design is often posed in terms of finding experiments which maximize a *utility function* such as the information gain or the reduction in a loss function. Once a utility function is chosen, the argmax can be found by gradient ascent methods provided that the derivatives of the utility can be efficiently computed. In particular, since the reduction in the *quadratic loss* is given by the posterior variance, our algorithm allows for computing gradients of the corresponding utility function.

Formally, we need to define two quantities: the loss function and the Bayes risk. In doing so, we will assume without loss of generality that the model parameters are renormalized such that all components of x lie in $[0, 1]$. The loss function represents a penalty assigned to errors in the in our estimates of x. We consider here the multiparameter generalization of the mean-squared error, the quadratic loss. For an estimate $\hat{x}$,

$$\mathcal{L}(x, \hat{x}) = (x - \hat{x})^{\mathrm{T}} (x - \hat{x}) \,. \tag{34}$$

Letting $\hat{x}$ be the Bayesian mean estimator for the posterior $P(x|d, c)$ and considering the single-parameter case,

$$\mathcal{L}(x, P(x|d, c)) = \left(x - \int P(x'|d, c) x' \mathrm{d}x'\right)^2 . \tag{35}$$

Having defined the loss function, the risk is the expectation of the loss over experimental data, $\mathbb{E}_d\{\mathcal{L}(x, \hat{x})\}$, where $\hat{x}$ is taken to depend on the experimental data. The Bayes risk is then the expectation of risk over both the prior distribution and the outcomes,

$$\begin{aligned} \mathcal{R}(x, P(x)) &= \mathbb{E}_{d, x\sim P(x)}\{\mathcal{L}(x, P(x|d, c))\} \\ &= \int P(x) \int P(d|x, c) \left(x - \int P(x'|d, c) x' \mathrm{d}x'\right)^2 \mathrm{d}x \mathrm{d}d. \end{aligned} \tag{36}$$

The Bayes risk for the quadratic loss function is thus the trace of the posterior covariance matrix, averaged over possible experimental outcomes. We want to find c that minimizes the Bayes risk, so that a reasonable utility function to optimize for is the negative posterior variance,

$$\mathcal{U}(P(x), c) = -\int P(x) \int P(d|x, c) \left(x - \int P(x'|d, c) x' \mathrm{d}x'\right)^2 \mathrm{d}x \mathrm{d}d. \tag{37}$$

The application of our algorithm is now made clear: like classical particle filtering methods, our algorithm efficiently computes expectation values over posterior distributions. Thus, $\mathcal{U}$ can be calculated using quantum resources, including in cases where classical methods alone fail. In the finite dimensional setting that we're interested in we simply replace these integrals by sums over the corresponding variables. The derivatives of $\mathcal{U}$ can then be approximated for small but finite h as

$$\frac{\partial \mathcal{U}(P(x), c)}{\partial c_j} = \frac{\mathcal{U}(P(x), c + h\hat{c_j}) - \mathcal{U}(P(x), c)}{h} + O(h^2). \tag{38}$$

Thus if c consists of C different components then $O(C)$ calculations of the utility function are needed to estimate the gradient for a finite value of h. This is the intuition behind our method, the performance of which is given in the following theorem.

Theorem 5. *Assume that the prior distribution $P(x)$ has support only on the interval $x \in [0, 1]$ there are D experimental outcomes then each component of the gradient of $\mathcal{U}$ can be computed within error ϵ using on average $\tilde{O}\left(D\sqrt{\max_{c,j}\left|\frac{\partial^3 U(c)}{\partial c_j^3}\right|}/\epsilon^{3/2}\right)$ queries to the likelihood function and the prior, for $\epsilon \le \min_d \int P(d|x)P(x)\mathrm{d}x/2$.*

Proof. The utility function can be directly computed on a quantum computer, but doing so is challenging because of the need to coherently store the posterior means of the distribution. We simplify this by expanding the square in (37) to find

$$\begin{aligned} \mathcal{U}(P(x), c) = &-\iint P(x)P(d|x, c)x^2 \mathrm{d}x \mathrm{d}d \\ &+ 2\iiint P(x)P(d|x, c)P(x'|d, c)xx' \mathrm{d}x' \mathrm{d}d \mathrm{d}x \\ &- \iiiint P(x)P(d|x, c)P(x'|d, c)P(x''|d, c)x'x'' \mathrm{d}x'' \mathrm{d}x' \mathrm{d}d \mathrm{d}x. \end{aligned} \tag{39}$$

We then compute each of these terms individually and combine the results classically to obtain an estimate of $\mathcal{U}$.

The double integral term in (39) is the easiest to compute. It can be computed by preparing the state

$$\sum_x \sqrt{P(x)}\,|x\rangle \frac{1}{\sqrt{D}}\sum_{d=1}^{D}|d\rangle\left(\sqrt{P(D|x,c)x^2}\,|1\rangle + \sqrt{1-P(D|x,c)x^2}\,|0\rangle\right). \tag{40}$$

The probability of measuring the right most qubit to be 1 is

$$\sum_x\sum_d P(x)P(D|x,c)x^2/D \le 1/D.$$

Therefore the desired probability can be found by estimating the likelihood of observing 1 divided by the total number of outcomes D. A direct application of amplitude estimation gives that the expectation value can be learned within error ϵ using $\tilde{O}(D/\epsilon)$ preparations of the initial state and evaluations of the likelihood function.

Since the probability of success is known to be bounded above by $1/D$, Theorem 3 implies that $\tilde{O}(\sqrt{D}/\epsilon_0)$ state preparations are needed to estimate the integral if we define S to be a reflection operator that imparts a phase if and only if the ancilla qubit equals 1.

The numerator can be estimated in exactly the same fashion, by preparing the state

$$\sum_x \sqrt{P(x)}\,|x\rangle \sum_{x'}\sqrt{P(x')}\,|x'\rangle\left(\sqrt{P(d|x',c)P(d|x,c)xx'}\,|1\rangle + \sqrt{1-P(d|x',c)P(d|x,c)xx'}\,|0\rangle\right), \tag{41}$$

Note that the numerator, $N(d|c)$, is not $\Theta(1)$: it is in fact $O(P^2(d))$ as seen by the Cauchy–Schwarz inequality and $x\in[0,1]$

$$\sum_x\sum_{x'}P(x)P(x')P(d|x,c)P(d|x',c)xx' \le \left(\sum_x P(x)P(d|x,c)\right)^2 = P_d^2. \tag{42}$$

The triple integral in (39) is much more challenging. It can be expressed as

$$\iiint P(x)P(d|x,c)P(x'|d,c)xx'\mathrm{d}x'\mathrm{d}d\mathrm{d}x = \iiint P(x)P(d|x,c)\frac{P(d|x',c)P(x')}{\int P(d|x',c)P(x')\mathrm{d}x'}xx'\mathrm{d}x'\mathrm{d}d\mathrm{d}x.$$

The integral over d in this expression is difficult to compute in superposition. So instead, we forgo directly integrating over d using the quantum computer and instead compute the integrand quantumly and classically integrate over d. In many models D will be small ($D=2$ is not uncommon) hence a polynomial reduction in the scaling with D will often not warrant the additional costs of amplitude amplification.

For fixed d, the first step is to compute $P(d) := \int P(d|x,c)P(x)\mathrm{d}x$, which can be estimated by preparing the state

$$\sum_x\sqrt{P(x)}\,|x\rangle\left(\sqrt{P(d|x,c)}\,|1\rangle + \sqrt{1-P(d|x,c)}\,|0\rangle\right), \tag{43}$$

and estimating, $P(d)$, the probability that the right–most qubit is 1, which is the required probability. This can be learned within error ϵ using amplitude estimation, which requires $\tilde{O}(1/\epsilon)$ queries to the initial state and the likelihood oracle [19].

For simplicity let us define the integral to be $N(d|c)/P(d)$ and the approximation to the integral as $\tilde{N}(D|c)/\tilde{P}(d)$. We then see from the triangle inequality that if we estimate the denominator to within error $\epsilon_0 \le P(d)/2$ then

$$\begin{aligned}
\left|\frac{\tilde{N}(d|c)}{\tilde{P}(d)} - \frac{N(d|c)}{P(d)}\right| &\le \left|\frac{\tilde{N}(d|c)}{\tilde{P}(d)} - \frac{N(d|c)}{\tilde{P}(d)}\right| + \left|\frac{N(d|c)}{\tilde{P}(d)} - \frac{N(d|c)}{P(d)}\right|.\\
&\le \frac{1}{P(d)-\epsilon_0}\left|\tilde{N}(d|c)-N(d|c)\right| + P(d)^2\left|\frac{1}{\tilde{P}(d)}-\frac{1}{P(d)}\right|\\
&\le \frac{2}{P(d)}\left|\tilde{N}(d|c)-N(d|c)\right| + P(d)\left|\frac{1}{1-\epsilon_0/P(d)}-1\right|\\
&\le \frac{2}{P(d)}\left|\tilde{N}(d|c)-N(d|c)\right| + \epsilon_0.
\end{aligned} \tag{44}$$

Therefore under these assumptions it is necessary to estimate $N(d|c)$ to within error $O(\epsilon_0/P(d))$ to achieve error $O(\epsilon_0)$. We can accelerate this inference process by observing that

$$N(d|c) \le (P(d)+\epsilon_0)^2 \in O(P^2(d)), \tag{45}$$

since $\epsilon_0 \le P(d)/2$. Theorem 3 can then be used to estimate $N(d|c)$ within error δ using $\tilde{O}(P(d)/\delta)$ queries. Since we need error $\epsilon_0 P(d)$ the number of query operations needed to infer $N(d|c)$ within error ϵ_0 is in $\tilde{O}(1/\epsilon_0)$. This process needs to be repeated classically D times so the total cost is $\tilde{O}(D/\epsilon_0)$ for this step as well. Thus we see from (44) that the total error can be made less than ϵ_0, with high probability, using a number of queries that scales as $\tilde{O}(D/\epsilon)$.

The analysis of the quadrouple integral is exactly the same and requires $\tilde{O}(D/\epsilon)$ queries on average. Thus the cost of evaluating the utility function to within error ϵ with high probability is $\tilde{O}(D/\epsilon)$.

Given an algorithm that can compute $U(c)$ using a number of queries that scales as $\tilde{O}(D/\epsilon)$, we can estimate the derivative using a centered difference formula. In particular we know that

$$\left|\frac{\partial U(c)}{\partial c_j} - \frac{U(c+\delta_j)-U(c-\delta_j)}{2\delta}\right| \le \max_{c,j}\left|\frac{\partial^3 U(c)}{\partial c_j^3}\right|\frac{\delta^2}{6}, \tag{46}$$

where $|\delta_j| = \delta$ and δ_j is a vector parallel to the unit vector c_j. Since we cannot compute $U(c\pm\delta_j)$ exactly, the error we want to bound is

$$\begin{aligned}&\left|\frac{\partial U(c)}{\partial c_j} - \frac{\tilde{U}(c+\delta_j)-\tilde{U}(c-\delta_j)}{2\delta}\right| \\ &\qquad\le \left|\frac{\partial U(c)}{\partial c_j} - \frac{U(c+\delta_j)-U(c-\delta_j)}{2\delta}\right| + \left|\frac{U(c+\delta_j)-U(c-\delta_j)}{2\delta} - \frac{\tilde{U}(c+\delta_j)-\tilde{U}(c-\delta_j)}{2\delta}\right|,\end{aligned} \tag{47}$$

where $\tilde{U}$ is the approximation to the utility function that has error at most ϵ_0. The error is then

$$O\left(\max_{c,j}\left|\frac{\partial^3 U(c)}{\partial c_j^3}\right|\delta^2 + \frac{\epsilon_0}{\delta}\right). \tag{48}$$

Since δ is a free parameter that we will choose to make both sources of error equivalent. This corresponds to $\delta = \epsilon_0^{1/3}/\max_{c,j}\left|\frac{\partial^3 U(c)}{\partial c_j^3}\right|^{1/3}$. This gives an overall error of

$$O\left(\epsilon_0^{2/3}\max_{c,j}\left|\frac{\partial^3 U(c)}{\partial c_j^3}\right|^{1/3}\right). \tag{49}$$

If we wish to make this error ϵ then it suffices to take $\epsilon_0 = \epsilon^{3/2}/\sqrt{\max_{c,j}\left|\frac{\partial^3 U(c)}{\partial c_j^3}\right|}$. Since the cost of computing $U(c\pm\delta_j)$ within error ϵ_0 with high probability is $\tilde{O}(D/\epsilon_0)$ the cost estimates follow. □

This shows that we can use quantum techniques to achieve a polynomial speedup over classical methods for computing the gradient using sampling, which would require $O(\Lambda CD/\epsilon^2)$ queries. Another interesting feature of this approach is that we do not explicitly use the qubit string representation for the likelihood to prepare states such as (43). Similar states could therefore also be prepared for problems such as quantum Hamiltonian learning [22] by eschewing a digital oracle and instead using a quantum simulation circuit that marks parts of the quantum state that correspond to measurement outcome d being observed. This means that these algorithms can be used in concert with quantum Hamiltonian learning ideas to efficiently optimize experimental design, whereas no efficient classical method exists to do so because of the expense of simulation.

VII. CONCLUSION

Here we have shown a new approach to quantum Bayesian inference that addresses the problems faced by earlier quantum methods: namely their inefficiency in cases where $P(E) = \langle P(x), P(E|x)\rangle$ is exponentially small. We have shown that by using an approximate form of inference where we learn a concise model of the posterior distribution, rather than the whole distribution, we can continue to learn in spite of failures in quantum rejection sampling because we always have a known prior distribution to fall back on. We further show a class of unimodal distributions that are trivial to construct using a quantum computer that can serve as a model for the prior and illustrate how to generalize these approaches to time dependent data. Finally we show that quantum computers can be used to speed up adaptive experiment design by showing that they can speed up evaluating derivatives of the loss function, allowing for efficient experiment optimization for quantum Hamiltonian learning.

On a conceptual level, and important remaining question that our work shines light on is the question of what learning means in a quantum setting. In particular, we are able to prepare a state that is mathematically equivalent to the posterior distribution but because we are constrained to sample from such a distribution we cannot be said to know the posterior distribution. Classical methods do give an approximation to the posterior distribution that can be introspected without destroying the posterior distribution. Thus an important question to address is what learning actually means for quantum systems and also whether it makes sense to think about inference in absentia of classical memory. Answering this question may not only shed light on the structure of quantum machine learning algorithms, but also may lead to a deeper understanding of the nature of inference and measurement.

APPENDIX A: ASYMPTOTIC STABILITY OF UPDATING

Interestingly, this process of classically learning a model for the posterior need not be repeated forever. If the true model has sufficient support in the final posterior then classical feedback is irrelevant because the quantum algorithm will converge to the true model as $L \to \infty$ if $P(E|x) \neq P(E|y)$ for all $x \neq y$, regardless whether success or failure is observed. This is summarized in the following theorem.

Theorem 6. *There exists $\delta > 0$ such that if $|\,|\psi\rangle - |x\rangle\,| \leq \delta$ then the method of Lemma 1 converges to $|x\rangle$ if the failure and success branches are treated equivalently and $P(E|x) \neq P(E|y)$ for all $x \neq y$.*

Proof. The algorithm that results from ignoring whether success or failure is measured in the method of Lemma 1 can be studied by examining the map that results from tracing over the success or failure register. First, let us assume that the likelihood function is non–degenerate, meaning that $P(E|x)$ is unique for all x. Then applying (3) we see that

$$|x\rangle\,|P(E|x)\rangle\,|0\rangle \mapsto \sqrt{P(x)}\,|x\rangle\,|P(E|x)\rangle \left(\sqrt{\frac{P(E|x)}{\Gamma_E}}\,|1\rangle + \sqrt{1-\frac{P(E|x)}{\Gamma_E}}\,|0\rangle\right). \tag{A1}$$

Because the state is not entangled, measuring the right most qubit does not affect the remaining state. Therefore the transformation given by Lemma 1 has each computational basis state as an eigenvector with eigenvalue 1.

Now let us assume that we apply the algorithm to the state $|\psi\rangle = |x\rangle + \delta\,|x_2\rangle + O(\delta^2)$ for $\delta \ll 1$. It then follows from tracing over the register that the resultant state is

$$|x\rangle\!\langle x| + \delta\left(\sqrt{\frac{P(E|x)P(E|x_2)}{\Gamma_E^2}} + \sqrt{\left(1-\frac{P(E|x)}{\Gamma_E}\right)\left(1-\frac{P(E|x_2)}{\Gamma_E}\right)}\right)(|x\rangle\!\langle x_2| + |x_2\rangle\!\langle x|) + O(\delta^2). \tag{A2}$$

It is straightforward to see from calculus that the $O(\delta)$ term is maximized when $P(E|x) = P(E|x_2)$, which is forbidden under our assumptions. Furthermore, the coefficient is at most 1, ergo the resultant state can be expressed as

$$(|x\rangle + c\delta\,|x_2\rangle)(\langle x| + c\delta\langle x_2|) + O(\delta^2), \tag{A3}$$

for $0 \leq c < 1$. Therefore the resulting state is equivalent to the initial state, but with the component orthogonal to it reduced by a factor of c. This means that the algorithm converges to $|x\rangle$ after a sufficient number of repetitions given that $\delta \ll 1$.

Now let us imagine that an initial state of the form $|x\rangle + \delta\sum_{y\neq x} a_y\,|y\rangle + O(\delta^2)$ is prepared. The density operator that results from applying the mapping in (3) and tracing over the last qubit is

$$|x\rangle\!\langle x| + \delta\sum_{y\neq x} a_y\left(\sqrt{\frac{P(E|x)P(E|y)}{\Gamma_E^2}} + \sqrt{\left(1-\frac{P(E|x)}{\Gamma_E}\right)\left(1-\frac{P(E|y)}{\Gamma_E}\right)}\right)(|x\rangle\!\langle y| + |y\rangle\!\langle x|) + O(\delta^2). \tag{A4}$$

Following the same argument it is clear that there exist $0 \leq c_y < 1$ such that the resultant state is

$$\left(|x\rangle + \sum_{y\neq x} a_y c_y \delta\, |y\rangle\right) \left(\langle x| + \sum_{y\neq x} c_y \delta \langle y|\right) + O(\delta^2), \tag{A5}$$

It is clear that the resultant state can be written in the form is $|x\rangle + c\delta\, |\psi\rangle + O(\delta^2)$ where $0 \leq c \leq \max_y c_y < 1$. This shows that the algorithm converges to $|x\rangle$ even if the initial perturbation is a superposition of basis states. □

APPENDIX B: DISCRETIZATION ERRORS

Apart from the quantum resampling step, the only source of error that emerges in our inference algorithm is from the discretization of the problem. We assume here that the underlying probability distribution $P(x)$ and the likelihood function $P(E|x)$ are differentiable functions of x and assume without loss generality that $x \in [0,1]^D$. We furthermore assume that the mesh used to approximate the probability distribution is uniform and a gridspacing of Δx is used in each direction. This means that the number of points is

$$N = (\Delta x)^{-D}. \tag{B1}$$

For notational simplicity, we take

$$\langle P(x), P(E|x)\rangle := \int P(x)P(E|x)\mathrm{d}^D x. \tag{B2}$$

We then give our main theorem below using this notation.

Theorem 7. *Let $P(E|x)$ be a differentiable function of $x \in [0,1]^D$ such that $0 < \max_E |\nabla P(E|x)|_{\max} \leq \Lambda$ and assume $\langle P(E|x), P(x)\rangle \neq 0$. A component of the posterior mean, $[x]_k$, can then be approximated for any $k \in \{1,\ldots,K\}$ within error ϵ by simulating a Bayes update of $P(x)$ on a uniform mesh of $[0,1]^D$ with mesh spacing Δx where*

$$\Delta x \leq \min_E \frac{\epsilon \langle P(E|x), P(x)\rangle^2}{\langle P(E|x), P(x)\rangle^2 + 3D\Lambda},$$

and

$$\epsilon \leq \frac{\langle P(E|x), P(x)\rangle^2 + 3D\Lambda}{2D\Lambda \langle P(E|x), P(x)\rangle}.$$

Proof. We employ the following approximation scheme. Let V_j be a hypercube of volume Δx^D with centroid $\bar{x}_j$. We then approximate the prior distribution within the hypercube as $P(x) \approx \delta(x - \bar{x}_j) \int_{V_j} P(x)\mathrm{d}^D x$. Our goal is to bound the error that this approximation incurs in the posterior mean.

We first analyze the error in approximating the probability assigned to each hypercube V_j after a Bayesian update

$$\begin{aligned}&\left|\frac{\int_{V_j} P(E|x)P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x} - \frac{P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\sum_j P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}\right| \\ &\quad\leq \left|\frac{\int_{V_j} P(E|x)P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x} - \frac{P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x}\right| + \left|\frac{P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x} - \frac{P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\sum_j P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}\right|.\end{aligned} \tag{B3}$$

From Taylor's remainder theorem and the triangle inequality, we then see that

$$\int_{V_j} (P(E|x) - P(E|\bar{x}_j))P(x)\mathrm{d}^D x \leq D\Lambda\Delta x \int_{V_j} P(x)\mathrm{d}^D x, \tag{B4}$$

which implies that

$$\left|\frac{\int_{V_j} P(E|x)P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x} - \frac{P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x}\right| \leq \frac{D\Lambda\Delta x \int_{V_j} P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x}. \tag{B5}$$

Now looking at the remaining term in (B3) we see that setting

$$\sum_j \int_{V_j} \Delta P(E|x)P(x)\mathrm{d}^D x := \sum_j P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x - \int P(E|x)P(x)\mathrm{d}x,$$

Using this definition, we can upper bound

$$\begin{aligned}&\left|\frac{1}{\int P(E|x)P(x)\mathrm{d}^D x} - \frac{1}{\sum_j P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}\right| \\ &\qquad \le \frac{1}{\int P(E|x)P(x)\mathrm{d}^D x}\max\left|1 - \frac{1}{1-\sum_j \Delta P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x/\int P(E|x)P(x)\mathrm{d}^D x}\right| \end{aligned} \tag{B6}$$

For the moment, let us assume that Δx is chosen such that

$$|\Delta P(E|x)| \le D\Lambda\Delta x \le \int P(E|x)P(x)\mathrm{d}^D x/2. \tag{B7}$$

We will see that this is a consequence of the bound on ϵ in the theorem statement. Then, using the fact that for all $|z| \le 1/2$, $|1/(1+z) - 1| \le 2|z|$

$$\frac{1}{\int P(E|x)P(x)\mathrm{d}^D x}\max\left|1 - \frac{1}{1-\sum_j \int_{V_j} \Delta P(E|\bar{x}_j)P(x)\mathrm{d}^D x/\int P(E|x)P(x)\mathrm{d}^D x}\right| \le \frac{2D\Lambda\Delta x}{\left(\int P(E|x)P(x)\mathrm{d}^D x\right)^2}. \tag{B8}$$

This implies that

$$\left|\frac{P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x} - \frac{P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\sum_j P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}\right| \le \frac{2D\Lambda\Delta x P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\left(\int P(E|x)P(x)\mathrm{d}^D x\right)^2} \tag{B9}$$

Thus from (B5), (B8) and (B3)

$$\left|\frac{\int_{V_j} P(E|x)P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x} - \frac{P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\sum_j P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}\right| \le \frac{3D\Lambda\Delta x\int_{V_j} P(x)\mathrm{d}^D x}{\left(\int P(E|x)P(x)\mathrm{d}^D x\right)^2}. \tag{B10}$$

Now let $[x]_k$ be the k–th component of the vector x. It then follows that the posterior mean of that component of the model vector obeys

$$\begin{aligned}\left|\int P(x|E)x_k\mathrm{d}^D x - \sum_j \frac{[\bar{x}_j]_k P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\sum_j P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}\right| &\le \left|\int P(x|E)x_k\mathrm{d}^D - \sum_j \int_{V_j} P(x|E)[\bar{x}_j]_k\mathrm{d}^D x\right| \\ &\quad + \left|\sum_j \int_{V_j} P(x|E)[\bar{x}_j]_k\mathrm{d}^D x - \sum_j \frac{[\bar{x}_j]_k P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\sum_j P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}\right|.\end{aligned} \tag{B11}$$

Since $0 \le [\bar{x}_j]_k \le 1$ and the sum of the prior probability is 1, (B10) implies

$$\left|\sum_j \int_{V_j} P(x|E)[\bar{x}_j]_k\mathrm{d}^D x - \sum_j \frac{[\bar{x}_j]_k P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\sum_j P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}\right| \le \frac{3D\Lambda\Delta x}{\left(\int P(E|x)P(x)\mathrm{d}^D x\right)^2}. \tag{B12}$$

Similarly,

$$\left|\int P(x|E)x_k\mathrm{d}^D - \sum_j \int_{V_j} P(x|E)[\bar{x}_j]_k\mathrm{d}^D x\right| \le \sum_j \int_{V_j} P(x|E)\mathrm{d}^D x\Delta x = \Delta x. \tag{B13}$$

Therefore (B11), (B12) and (B13) imply that the error in the posterior mean is

$$\left|\int P(x|E)x_k \mathrm{d}^D x - \sum_j \frac{[\bar{x}_j]_k P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\sum_j P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}\right| \le \Delta x\left(1+\frac{3D\Lambda}{\left(\int P(E|x)P(x)\mathrm{d}^D x\right)^2}\right). \tag{B14}$$

Simple algebra then shows that the error in the approximate posterior mean is at most ϵ if

$$\Delta x \le \max_E \frac{\epsilon\langle P(E|x), P(x)\rangle^2}{\langle P(E|x), P(x)\rangle^2 + 3D\Lambda}. \tag{B15}$$

Eq. (B7) is a key assumption behind (B15). It is then easy to see from algebra that the assumption is implied by (B15) if

$$\epsilon \le \frac{\langle P(E|x), P(x)\rangle^2 + 3D\Lambda}{2D\Lambda\langle P(E|x), P(x)\rangle}, \tag{B16}$$

as claimed. □

This theorem shows that if the derivatives of the likelihood function are large or the inner product between the prior and the likelihood function is small then the errors incurred by updating can be potentially large. These errors can be combated by making Δx small. This is potentially expensive since $\Delta x = N^{-D}$ where N is the number of points in the mesh approximating the posterior.

Corollary 2. *Given the likelihood function satisfies the assumptions of Theorem 7, the number of qubits needed to represent the prior distribution using a uniform mesh of $[0,1]^D$ to sufficient precision to guarantee that the error in the posterior mean after an update is at most ϵ is bounded above by*

$$D\left\lceil \log_2\left(\max_E \frac{\langle P(E|x), P(x)\rangle^2 + 3D\Lambda}{\epsilon\langle P(E|x), P(x)\rangle^2}\right)\right\rceil.$$

Proof. Proof is an immediate consequence of substituting $\Delta x = 1/N^{1/D}$ into Theorem 7 and solving for N. □

This shows that the number of qubits needed is at most logarithmic in the error. Furthermore, if L updates are required then the total cost is increased by at most an additive factor of $\log(L)$. We do not include this in our cost estimates since this error estimate is needlessly pessimistic as Bayesian inference is insensitive to the initial prior according to the Bernstein von-Mises theorem and consequently such errors are unlikely to be additive.

APPENDIX C: CLASSICAL LIMIT

Another approach to incorporating the robustness to updating errors that classical Bayesian inference provides is to examine quantum Bayesian inference in the classical limit. The way that the classical limit is reached depends strongly on the correspondance taken between the quantum and classical dynamics of the system. In this case, it makes no sense to examine the classical limit of quantum Bayesian inference as resulting as a consequence of decoherence. Instead we assume that we wish an algorithm that, in the limit of an infinite number of copies, allows the means and covariances of the quantum distribution to be extracted without disturbing the underlying posterior distribution. We achieve this in effect by using a repetition code. We focus on the one-dimensional case, but generalization to the multi-dimensional case is straight forward.

The repetition code that allows us to reach the classical limit of the quantum algorithm is trivial:

$$\sum_j \sqrt{P(x_j)}\,|x_j\rangle \mapsto |\psi\rangle := \left(\sum_j \sqrt{P(x_j)}\,|x_j\rangle\right)^{\otimes K}. \tag{C1}$$

In order to learn the mean from such a state without destroying it, we need to add an additional register that stores an estimate of the mean-value to a fixed number of bits of precision. This can be achieved using the method of (CITE GROVER). We denote this state as

$$\sum_{x_1,\ldots,x_K} \sqrt{P(x_1)\cdots P(x_K)}\,|x_1\ldots x_K\rangle\,|\bar{x}(x_1\ldots x_K)\rangle, \tag{C2}$$

where $\bar{x}$ is an approximation to the mean that is truncated to give error $\Delta \le \mu$. For simplicity, we drop the explicit dependence of $\bar{x}$ on x in the following.

Let $\mu = \sum_j P(x_j)x_j$ be the true mean. Then as each of the distributions over the constituent x_j is independent and assuming that $x_j \le X_{\max}$, the chernoff bound states that

$$P\left(|\bar{x} - \mu| \ge \Delta\right) \le e^{-\frac{\Delta^2 K}{3\mu X_{\max}}}. \tag{C3}$$

Thus the probability of measuring a mean that deviates more than Δ from μ is at most ϵ if

$$K \ge \frac{3\mu X_{\max}}{\Delta^2} \ln\left(\frac{1}{\epsilon}\right). \tag{C4}$$

This implies that for every $\epsilon > 0$ and every discretization error Δ there exists a value of K such that the probability of measuring the discretized mean to be μ is at least $1 - \epsilon$.

Let $|\phi\rangle = (\mathbb{1} \otimes |\mu\rangle\langle\mu|)\,|\psi\rangle / |(\mathbb{1} \otimes |\mu\rangle\langle\mu|)\,|\psi\rangle\,|$ then

$$|\langle\psi|\phi\rangle|^2 = \frac{|\langle\psi|(\mathbb{1} \otimes |\mu\rangle\langle\mu|)\,|\psi\rangle\,|^2}{|(\mathbb{1} \otimes |\mu\rangle\langle\mu|)\,|\psi\rangle\,|^2} \ge \frac{1-\epsilon}{|(\mathbb{1} \otimes |\mu\rangle\langle\mu|)\,|\psi\rangle\,|^2} \ge 1 - \epsilon. \tag{C5}$$

Thus up to error $O(\epsilon)$, we can treat the state after learning the mean as identical to the state that existed before learning μ. Ergo despite the fact that the x_i used in the distribution are no longer identically distributed, we can treat them as if they were while incurring an error of at most ϵ in the estimate of $P(|\bar{x} - \mu| \le \Delta)$. From the triangle inequality, it is then straight forward to see that after L such steps that the total error incurred in the final state (as measured by the trace distance) is at most $L\sqrt{\epsilon}$, which can be made at most $\sqrt{\epsilon}$ by choosing

$$K \ge \frac{3\mu X_{\max}}{\Delta^2} \ln\left(\frac{L^2}{\epsilon}\right). \tag{C6}$$

This in turn means that the error in the inference of μ after L steps is at most $X_{\max}\epsilon$.

[1] G. H. Low, T. J. Yoder, and I. L. Chuang, Physical Review A **89**, 062315 (2014).

[2] A. W. Harrow, A. Hassidim, and S. Lloyd, Physical review letters **103**, 150502 (2009).

[3] M. Ozols, M. Roetteler, and J. Roland, ACM Transactions on Computation Theory (TOCT) **5**, 11 (2013).

[4] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, Contemporary Mathematics **305**, 53 (2002).

[5] L. K. Grover, in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (ACM, 1996), pp. 212–219.

[6] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, arXiv preprint quant-ph/9605034 (1996).

[7] P. Dagum and M. Luby, Artificial intelligence **60**, 141 (1993).

[8] J. Liu and M. West, in *Sequential Monte Carlo Methods in Practice*, edited by D. Freitas and N. Gordon (Springer-Verlag, New York, 2001), URL http://ftp.stat.duke.edu/WorkingPapers/99-14.html.

[9] T. P. Minka, in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence* (Morgan Kaufmann Publishers Inc., 2001), pp. 362–369.

[10] R. Van Der Merwe, A. Doucet, N. De Freitas, and E. Wan, in *NIPS* (2000), pp. 584–590.

[11] D. Wecker, M. B. Hastings, N. Wiebe, B. K. Clark, C. Nayak, and M. Troyer, arXiv preprint arXiv:1506.05135 (2015).

[12] A. Kitaev and W. A. Webb, arXiv preprint arXiv:0801.0342 (2008).

[13] R. Cleve and J. Watrous, in *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on* (IEEE, 2000), pp. 526–536.

[14] V. Kliuchnikov, D. Maslov, and M. Mosca, Quantum Information & Computation **13**, 607 (2013).

[15] S. Lloyd, M. Mohseni, and P. Rebentrost, Nature Physics **10**, 631 (2014).

[16] L. Grover and T. Rudolph, arXiv preprint quant-ph/0208112 (2002).

[17] N. Wiebe, C. Granade, C. Ferrie, and D. Cory, Physical Review Letters **112**, 190501 (2014), URL http://link.aps.org/doi/10.1103/PhysRevLett.112.190501.

[18] A. M. Childs and N. Wiebe, Quantum Information & Computation **12**, 901 (2012).

[19] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, arXiv preprint quant-ph/0005055 (2000).

[20] M. Isard and A. Blake, International Journal of Computer Vision **29**, 5 (1998), ISSN 0920-5691, 1573-1405, URL http://link.springer.com/article/10.1023/A%3A1008078328650.

[21] N. Wiebe, A. Kapoor, and K. M. Svore, arXiv preprint arXiv:1412.3489 (2014).

[22] N. Wiebe, C. Granade, C. Ferrie, and D. Cory, Physical review letters **112**, 190501 (2014).