Quantum Information and Computation, Vol. 0, No. 0 (2003) 000–000

Quantum Deep Learning

NATHAN WIEBE
Microsoft Research, One Microsoft Way
Redmond, WA 98052, USA

CHRISTOPHER GRANADE
Centre for Engineered Quantum Systems, University of Sydney, Sydney, NSW, Australia
School of Physics, University of Sydney, Sydney, NSW, Australia

Received (received date)
Revised (revised date)

In recent years, deep learning has had a profound impact on machine learning and artificial intelligence. At the same time, algorithms for quantum computers have been shown to efficiently solve some problems that are intractable on conventional, classical computers. We show that quantum computing not only reduces the time required to train a deep restricted Boltzmann machine, but also provides a richer and more comprehensive framework for deep learning than classical computing and leads to significant improvements in the optimization of the underlying objective function. Our quantum methods also permit efficient training of multilayer and fully connected models.

Keywords: Quantum computing, quantum algorithms, quantum machine learning
Communicated by: to be filled by the Editorial

1 Introduction

Quantum systems have, at first glance, an incredible capacity to store vectors. Only n qubits suffice to represent a vector in $\mathbb{R}^{2^n}$. This, in part, is the origin of all of the celebrated exponential speedups that quantum computing offers [1, 2, 3, 4]. At the same time, from a computational learning perspective, quantum states can be approximately described much more efficiently [5]. Indeed, the tension between this ability of a small quantum register to store a very high–dimensional vector and the perspective offered by learning raises an important question: can a quantum agent with a logarithmically sized quantum memory efficiently learn from its surroundings? Here, we seek to shed light on this question by investigating Bayesian inference as a way to model the quantum agent inferring properties about its surroundings. We will see that lower bounds on quantum query complexity of unstructured search place severe limitations on the ability of the system to learn exactly. Nonetheless, we will also see that quantum mechanics reveals new possibilities for approximate learning that are not present in classical Bayesian inference.

Bayes' rule is the heart of Bayesian inference. It gives the correct way to update a prior distribution that describes the users' initial beliefs about a system model when a piece of experimental evidence is received. If E is a piece of evidence (an observable variable) and x denotes a candidate model for the experimental system (a latent or hidden variable), then

Bayes' rule states that the probability that the model is valid given the evidence (denoted $P(x|E)$) is

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

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 for applications in quantum information processing [6, 7]. 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 latent variable 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.

A natural question to ask at this point is whether quantum computing could make inference tractable. Quantum advantages are seen for a wealth of other machine learning protocols [8, 9, 10, 11, 12, 13], so it stands to reason that it may be able to provide advantages here as well. This issue is has been recently discussed in [14], which uses ideas from quantum rejection sampling [4, 15] 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 latent variables. 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 allows the quantum agent to revert to a previous model in the event that the update process fails, which allows the process to be implemented efficiently under reasonable assumptions on the likelihood function.

Before we delve deeper into our approach, however, it is important to clearly frame the discussion by presenting a definition of what we mean by learning. In particular, there is no unambiguous consensus even in the case of classical learners. To address this, we present a broad definition of learning that encapsulates the problem currently under consideration.

Definition 1 *Let D be a metric space, and let S and O be discrete sets representing evidence and outcomes, respectively. Consider a system comprised of an internal state $X \in D$, a learning function $L : (S, D) \to D$ and an output function $G : D \to O$. We then say that this system* learns *if*

1. *the internal state X is updated via $X \leftarrow L(Y, X)$ when the system is presented with evidence $Y \in S$,*
2. *$L(Y, X)$ is not a constant function of either Y or X, and $G(X)$ is not a constant function of X,*
3. *there exists $F : D \to D^2$ such that for all $X \in \text{range}(L)$, $\text{dist}(F(G(X)), (G(X), G(X))) \leq \epsilon$ for some $\epsilon \geq 0$,*

4. *the previous requirement can be achieved for every* $\epsilon > 0$ *using* $O(\text{polylog}(1/\epsilon))$ *applications of* L.

If we were examining classical learning, then the first two requirements often suffice to give a useful definition of learning. This is because classical information, in the form of bit strings, can be copied and acted upon without consequence. On the other hand, quantum data cannot be copied, hence we insert the remaining two requirements to emphasize that in our setting one cannot claim to have learned a concept simply by owning a quantum state. Instead, one has to be able to extract actionable understanding from such a state to claim to have learned from it. Furthermore, we require that in settings where the error in extracting understanding is non–zero that such errors can be made arbitrarily small using polynomial resources. With this definition in hand, we will now proceed to look at the challenges faced when trying to apply quantum Bayesian inference to learn about a system with and without access to classical memory.

2 Quantum Bayesian updating

In order to investigate the question of whether small quantum systems can learn efficiently, we will examine the issue through the lens of Bayesian inference. Our first objective in this section is to provide a concrete definition for what we mean by a quantum Bayesian update and show a protocol for implementing a quantum Bayesian update. We will then show that this method cannot generically be efficient and furthermore that asymptotic improvements to the algorithm or an inexpensive error correction algorithm for its faults would violate lower bounds on Grover's search. These results motivate our definition of "semi–classical" Bayesian updating in the subsequent section.

A key assumption that we make here and in the subsequent text is that the visible and latent variables are discrete. In other words, we assume that each experiment has a discrete set of outcomes and there are a discrete set of hypotheses that could explain the data. Continuous problems can, however, be approximated by discrete models and we provide error bounds for doing so in Appendix B. We then invoke these assumptions in the following definition of a quantum Bayesian update.

Definition 2 *A quantum Bayesian update of a prior state* $\sum_x \sqrt{P(x)}\,|x\rangle$ *performs, for observable variable* E *and likelihood function* $P(E|x)$*, the map*

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

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 $P(E|x) \leq \Gamma_E \leq 1$ for all x.

With respect to Definition 1, the quantum state in the second register takes the role of X and the state in the first register stores an element $E \in S$. The function L then is given by Bayes' rule. It may be tempting to look at Bayesian inference and immediately deduce

that learning is impossible by our definition according to the no-cloning theorem of quantum mechanics. However, since we have not specified the output function this learning process may not conform to the definition and clearly will not if G is the identity map. Thus we will have to look closer at Bayesian inference to decide whether general roadblocks occur when trying to learn.

An interesting consequence of the above assumptions and Definition 2 is that in general the Bayesian update is non-unitary when working on this space. This means that we cannot implement a quantum Bayesian update deterministically without dilating the space. The following lemma discusses how to implement such a non–deterministic quantum Bayesian update. It can also be thought of as a generalization of the result of [14] to an oracular setting.

Lemma 1 *Given a unitary operator, U, such that $U|0\rangle = \sum_x \sqrt{P(x)}|x\rangle$ and $\Gamma_E : P(E|x) \leq \Gamma_E$ the state $\sum_x \sqrt{P(x|E)}|x\rangle$ can be prepared using an expected 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 applying the rotation $R_y(2\sin^{-1}(P(E|x)/\Gamma_E))$ to the ancilla qubit, controlled on the register representing $|P(E|x)\rangle$, we can enact

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

Next the right most qubit register is measured and if a result of 1 is obtained then 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 $\sum_x P(x)P(E|x)/\Gamma_E = \langle P(x), P(E|x)\rangle/\Gamma_E$.

Since the process is heralded and the initial state preparation process is unitary, amplitude amplification can be used to reduce the expected number of calls to the oracles quadratically [16]. 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 online settings many updates will be needed to reach the final posterior distribution. If L updates are required then the probability of all

such updates succeeding given a sequence of observed variables $E_1, \ldots, E_L$ is at most

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

This shows that the probability of success generically will shrink exponentially with L.

The exponential decay of the success probability with L can be mitigated to some extent by applying amplitude amplification on the outcome that all measurements 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.

While the success probability in general falls exponentially, not all failures are catastrophic. We show in Appendix 1 that a radius of convergence exists such that if the prior distribution is sufficiently close to a delta–function about the true value of the latent variable, and the likelihood function is well behaved, then this updating strategy will cause it to converge to the delta–function. This convergence occurs even if the user ignores the fact that inference errors can occur and does not attempt to correct such errors. This means that, for discrete inference problems, the computational complexity of inferring the correct latent parameter need not be infinite.

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 [17]. Grover's problem (with one marked element) seeks to find $x = \text{argmax}(f(x))$ where $f(x)$ is a blackbox Boolean function that is promised to have a unique x_{mark} such that $f(x_{\text{mark}}) = 1$. The generalization to multiple marked elements is similar. The reduction between the two problems is formally proved below.

Lemma 2 *Grover's problem with N items and m marked items reduces to Bayesian inference on a prior on $\mathbb{R}^N$.*

Proof. Let O_G 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 $P(1|x) = O_G(x)$ and thus a likelihood evaluation is equivalent to a query to O_G. 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_G, 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 [18]. We can exploit these bounds to show limitations on quantum systems ability to perform Bayesian inference and 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 *Let F be a blackbox quantum update algorithm that performs a quantum Bayesian update on $U|0\rangle = \sum_{x=1}^{N} \alpha_x |x\rangle$ for a known unitary operator U using $O(1)$ queries to the oracle O_E and is heralded and has success probability $\Omega(\sum_x |\alpha_x|^2 P(E|x)/\Gamma_E)$ for $\max_x P(E|x) \leq \Gamma_E \leq 1$. The following are impossible.*

1. *A blackbox algorithm G capable of performing a quantum Bayesian update and given a unitary operator $U : |0\rangle \mapsto \sum_{x=1}^{N} \alpha_x |x\rangle$ that uses $O(1)$ queries to O_E that is heralded and has success probability $\omega(\sum_x |\alpha_x|^2 P(E|x)/\Gamma_E)$.*
2. *A blackbox algorithm H that requires $o(\sqrt{N}/\log(N))$ queries to O_E to undo the effects that F applies to the arbitrary state $\sum_{x=1}^{N} \alpha_x |x\rangle$ upon a failed update.*
3. *A blackbox algorithm I capable of performing a quantum Bayesian update of $U|0\rangle = \sum_{x=1}^{N} \alpha_x |x\rangle$ that, for all O_E, uses on average $o(\sqrt{N})$ queries to O_E.*

Proof. Seeking a contradiction, assume that there exists a quantum algorithm that can perform a quantum Bayesian update using $O(1)$ queries that succeeds with probability $\omega(\sum_x |\alpha_x|^2 P(E|x))$ 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 $\omega(\sum_x |\alpha_x|^2 P(E|x)) \in \omega(1/N)$. Since each application of the algorithm requires $O(1)$ queries and success is heralded, $o(\sqrt{N})$ queries are needed on average to learn x_m using amplitude amplification [16], which violates lower bounds for the average query complexity for Grover's problem [18]. Therefore algorithm G, which is described in 1, is impossible.

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

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

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

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

Therefore $O(\log(N))$ measurements of 1 suffice to amplify the probability from $1/N$ to $\Theta(1)$.

Similarly, $P(x_m|1) \geq P(x_m)$ unless $1 + P(x_m) > 2$ or $P(x_m) < 0$. Since $P(x_m)$ is a probability this is impossible. Therefore the posterior probability is monotonically increasing with the number of successful updates. Thus if we define $\alpha_x^{(1)} := \alpha_x$ and $\alpha_x^{(m)}$ to be the components of the quantum state after $m + 1$ quantum updates then $\sqrt{|\alpha_x^{(m)}|^2 P(1|x)}$ is a monotonically increasing function of m.

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 that this sequence of observations was obtained (similar to Lemma 2) in order to simulate the search. Given the observable variables follow this sequence, Lemma 1 shows that there exists a quantum algorithm that can perform each such update with probability of success

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

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 $O(\text{poly}(1/N))$ 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

$$o\left(\frac{\sqrt{N}}{\log(N)\sqrt{\sum_x |\alpha_x|^2 P(1|x)}}\right) \in o\left(\frac{\sqrt{N}}{\log(N)}\right), \tag{11}$$

because $\sqrt{|\alpha_x^{(m)}|^2 P(1|x)} \geq \sqrt{|\alpha_x|^2 P(1|x)}$ for any $m \geq 1$. Since $O(\log(N))$ successful quantum updates are made in the inference process, the marked state can be inferred within probability $p \in \Theta(1)$ using $o(\sqrt{N})$ queries to the likelihood function. A to the likelihood function is equivalent to a query to Grover's oracle and thus error correction method H (described in 2) is impossible.

Finally, the impossibility of method I directly follows from Lemma 2 and lower bounds on Grover's search. □.

These impossibility results show that the quantum updating procedure of [14] and Lemma 1 cannot be improved without making assumptions about the underlying prior distributions or likelihood functions. From this we conclude that quantum Bayesian updating, as per Definition 2, is inefficient in general. This means that small quantum systems that attempt to store the prior and posterior vectors as a quantum state vector cannot do so efficiently, let alone output salient properties of the state, without making such assumptions.

This inefficiency 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 $\mathsf{P} = \mathsf{NP}$ [19]. 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 invites 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.

3 Semi–classical Bayesian updating

By the arguments in the previous section, we therefore often need approximations 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 [20, 21, 22]. However, the non–linear nature of the update rule and the problem of extracting information from the posterior distribution are not issues in the classical setting. 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 throughout 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 assumed-density filtering [21]. In order to prepare the distribution, we model the posterior distribution 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 [23] can be used to prepare such states as their cumulative distribution functions can be efficiently computed. Alternatively, for one–dimensional problems, such states could be manufactured

by approximate cloning [24].

We are now equipped to define a semi-classical Bayesian update.

Definition 3 *A semi–classical Bayesian update of a prior state on $\mathbb{C}^N$, for a discrete observable variable E, likelihood function $P(E|x)$ and family of probability distributions $F(x;\rho)$ parameterized by the vector ρ, maps*

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

We call this process semi–classical updating because it yields an approximate classical model for the posterior distribution. This model can take many forms in principle; as an 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 [25], below.

Lemma 3 *Given a unitary operator $U \in \mathbb{C}^{N\times N}$ such that $U|0\rangle = |\psi\rangle := \sum_x \sqrt{P(x)}\,|x\rangle$, an observable $\Lambda = \sum_x \lambda_x |x\rangle\langle x|$ and an estimate $\lambda_0 : \max_x |\lambda_x - \lambda_0| \leq \Delta\lambda$, 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 $O(\Delta\lambda/\epsilon)$ 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 Lemma 1, 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 [16]. Amplitude estimation has a probability of success of at least $8/\pi^2$. The result then follows from taking $\delta = \epsilon/\Delta\lambda$. □.

We now turn our attention to estimating the mean and covariance of the posterior distribution that arises from quantum updating. This is not quite a trivial application of Lemma 3 because our method for performing the update is non-unitary, which violates the assumptions of the Lemma. We avoid this problem by instead estimating these moments in a two–step probability estimation process. This approach is described in the following corollary.

Corollary 1 *Assume that λ_0 is a vector containing each $\langle x_i \rangle$ and each $\langle x_i x_j \rangle$ evaluated over the posterior $P(x|E)$. Further, let $P(x|E) = \sum_x \sqrt{P(x|E)}\,|x\rangle$ and*

$$\Delta\lambda \geq \max_k |\langle P(x|E)|\Lambda_k\,|P(x|E)\rangle - \lambda_{0,k}|$$

where Λ_k is the operator corresponding to x_i or $x_i x_j$ depending on the index k. The mean and covariance matrix can be computed, with high-probability, within error ϵ in the max–norm using $\tilde{O}\left(\frac{D^2 \Delta\lambda \Gamma_E}{\epsilon \langle P(x), P(E|x)\rangle}\right)$ queries to O_E. Here $f(x) \in \tilde{O}(g(x))$ if there exists a constant $\alpha > 0$ such that $\lim_{x\to\infty} f(x)/(\log^\alpha(x) g(x)) = 0$.

Proof. Our method works by classicially looping over all the components of the λ_0 vector, which we denote $\lambda_{0,k}$. For each k, we then need to prepare the following state conditioned on evidence E to compute the corresponding probability

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

where $\lambda_x^{(k)}$ is of the form x_i or $x_i x_j$ depending on the value of k.

We cannot directly apply the previous lemma to learn the requisite values because the method for preparing the posterior probability distribution is non–unitary. We address this by breaking the parameter estimation process into two steps, each of which involves learning a separate probability; we call the probabilities learned at each step $P(1)$ and $P(11)$, respectively. Let $P(1) = \sum_x P(x)P(E|x)/\Gamma_E$ be the probability of performing the quantum Bayesian update. Let $P(11)$ be the probability of both performing the update and measuring the right most qubit in (12) to be 1. Both will be learned using amplitude estimation, which is possible because the initial state is prepared using a unitary process. 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 either of the form x_i or $x_i x_j$ depending on the index k 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

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

If we estimate $P(1)$ and $P(11)$ within error $O(\delta)$ then the error in $\langle \Lambda_k \rangle$ is from calculus $O(\delta\Delta\lambda/P(1))$ 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 [16]

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

Here the $\tilde{O}(\cdot)$ is used to remove log-factors that arise from use of the Chernoff bound to boost the probability of success to near-unity. The result then follows from noting that there are $O(D^2)$ different values of k that need to be computed to learn the expectation values needed to compute the components of the posterior mean and covariance matrix. □. 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 problem of exponentially shrinking success probability at the price of only retaining incomplete information about the posterior distribution.

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 $\{E_k : k = 1,\dots L\}$ be a set of L observable variables. 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 $\tilde{O}\left(\frac{D^2\Delta\lambda}{\epsilon\langle P(x),\prod_k P(E_k|x)/\Gamma_{E_k}\rangle}\right)$ 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 yields the parameterization of a function $f \in F(x;\mu,\sigma)$ 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 3. □.

The complexity of the above approximate quantum inference algorithm cannot be easily compared to that of exact Bayesian inference because both algorithms provide very different pieces of information. Until very recently, no natural analogue of our quantum method could easily be found in the literature. The result of [26] provides such a classical analogue. The classical query complexity of their algorithm is quadratically worse in its scaling with ϵ.

Although we obtain a quadratic advantage in the scaling with ϵ, it would be nice to obtain further algorithmic advantages using amplitude amplification. Further advantages can be obtained in cases where the probabilities $P(11)$ and $P(1)$ are small by using amplitude amplification to boost these probabilities and then work backwards to infer the non–boosted probabilities. Below we formally prove a theorem to this effect that formalizes a similar claim made informally in [27].

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 $0 < a \leq a_0 < 1$, $1 - a_0 \in \Theta(1)$ and let S 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 [16] 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 [16] 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,

$$V|0\rangle = \sin((2m+1)\sin^{-1}(\sqrt{a}))|\phi\rangle + e^{i\theta}\cos((2m+1)\sin^{-1}(\sqrt{a}))|\phi^\perp\rangle. \tag{19}$$

Since V is a unitary operation, amplitude estimation can be used to learn the quantity $\sin^2((2m+1)\sin^{-1}(\sqrt{a}))$ to within error δ by using Theorem 12 of [16] 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 $0 \leq (2m+1)\sin^{-1}(\sqrt{a}) \leq \pi/2$ then

$$a = \sin^2\left(\frac{\sin^{-1}(\sqrt{y})}{2m+1}\right). \tag{20}$$

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

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

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 S 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

$$m > \frac{1}{2}\left(\frac{\pi}{2\sin^{-1}(\sqrt{a})} - 1\right) \tag{22}$$

then (20) 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 (22) 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

$$\tilde{O}\left(\frac{1}{\epsilon m}\right) \in \tilde{O}\left(\frac{\sqrt{a_0}}{\epsilon}\right), \tag{23}$$

as claimed. □.

This lemma is significant because it shows that if the scale of a probability that you wish to estimate is known then you can use amplitude amplification to boost the probability of success to $O(1)$ prior to estimating the probability. This can lead to a quadratic reduction in the cost of estimation if a constant number of bits of significance are required.

As an additional note, this method described in this section is not limited to tracking the values of static latent variables. If the latent variable has itself explicit time dependence then the above approach can be modified to robustly track its variation in time. This is discussed in more detail in Appendix C. We further discuss how to use quantum computing to find approximately optimal experiments to perform based on the current prior in Appendix D.

4 Quantum Bayesian updating using redundancy

While we have addressed a semi–classical form of learning for quantum Bayesian inference, an interesting remaining question is whether the form of quantum Bayesian inference that we consider has a well defined classical limit. The hope would be that such a protocol would also be an approximate Bayesian method, but would not be susceptible to the probabilistic failures that plague the quantum approach. We can reach such a limit by using redundant copies of the state to perform a protocol that is similar to semi–classical updating, but does not involve storing classical information. We also focus on the one-dimensional case in the following, but generalization to the multi-dimensional case is straightforward.

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

$$\sum_j \sqrt{P(x_j)}\,|x_j\rangle \mapsto |\psi\rangle := \left(\sum_j \sqrt{P(x_j)}\,|x_j\rangle\right)^{\otimes K}. \tag{24}$$

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 a simple arithmetic circuit. We denote this state as

$$\sum_{x_1,\ldots,x_K} \sqrt{P(x_1)\cdots P(x_K)}\,|x_1\ldots x_K\rangle\,|\bar{x}(x_1\ldots x_K)\rangle\,, \tag{25}$$

where $\bar{x}$ is an approximation to the mean that is truncated to give error $\Delta \leq \mu$. 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 $x_j \leq X_{\max}$, the Chernoff bound states that

$$P\left(|\bar{x} - \mu| \geq \Delta\right) \leq e^{-\frac{\Delta^2 K}{3\mu X_{\max}}}. \tag{26}$$

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

$$K \geq \frac{3\mu X_{\max}}{\Delta^2} \ln\left(\frac{1}{\delta}\right). \tag{27}$$

This implies that for every $\delta > 0$ 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 $1 - \delta$.

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

$$|\langle\psi|\phi\rangle|^2 = \frac{|\langle\psi|(\mathbb{1} \otimes |\mu\rangle\langle\mu|) |\psi\rangle|^2}{|(\mathbb{1} \otimes |\mu\rangle\langle\mu|) |\psi\rangle|^2} \geq \frac{1-\delta}{|(\mathbb{1} \otimes |\mu\rangle\langle\mu|) |\psi\rangle|^2} \geq 1 - \delta. \tag{28}$$

Thus up to error $O(\delta)$, 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 δ in the estimate of $P(|\bar{x} - \mu| \leq \Delta)$. 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

$$K \geq \frac{3\mu X_{\max}}{\Delta^2} \ln\left(\frac{L^2}{\epsilon}\right). \tag{29}$$

This in turn means that the error in the inference of μ after L steps is at most $X_{\max}\epsilon$. The exact same argument can be applied to learn the mean–square value of x and so the standard deviation can be learned in a similar fashion.

If K is sufficiently large, then any branches that fail can be immediately repopulated by a distribution from a two–parameter family of distributions $F(x; \mu, \sigma^2)$. This further carries an advantage because it does not necessitate that the entire distribution be approximated at each iteration, unlike semi–classical updating.

This shows that a redundant encoding can be used in order to protect the low–order moments against the effects of measurement. In turn, this therefore means that even if some updates fail then these results can be erased and replaced with a Gaussian approximation to the posterior distribution (for example). An explicit classical register is not needed in this approach, although since the entanglement of the expectation value register with the remaining qubits approaches zero. In this sense, it becomes a classical register and this result can also be thought of as an examination of the classical limit of quantum Bayesian updating.

While this shows that the algorithm can proceed without classical memory, it makes substantial demands on the memory. For even modest problems, it is likely to require thousands of copies of the state in order to be able to resist the effects of measurement back action on the state. This shows that while error correction can allow quantum systems able to learn efficiently without classical memory, the resulting systems will seldom be small. This suggests that there may be a tradeoff between system size and robustness that may make learning in small quantum systems highly challenging.

5 Conclusion

Our main contribution of this paper can be thought of as an analysis of the ability of small quantum systems' capacity to learn. In it we have examined a class of quantum learning

algorithms that require only logarithmic memory to update a register that stores its beliefs about a latent variable x that it must infer from a set of observable variables E that whose likelihoods are only known through access to a quantum oracle. We show that such algorithms cannot in general be efficient and hence they cannot learn according to the definition of learning that we give. However, we provide a semi–classical algorithm that uses classical memory to circumvent this problem within the context of approximate Bayesian inference. Thus quantum systems can learn if augmented with classical memory, however the need for such memory precludes them from being small.

Our semi–classical algorithm has a number of performance advantages over classical methods. It also can leverage quantum superposition to provide quadratic advantages for experiment design and also can be used to track the motion of time–dependent latent variables. We further demonstrate the need to store the model in classical memory is in principle superflous because multiple copies of the prior state can be used to robustly encode this information in the quantum state. This suggests that while small quantum systems may not be able to efficiently learn (especially in an online setting), redundant information can be used to protect the knowledge gained by the quantum system against the potentially destructive impacts of the non–linear transformations required by Bayesian inference. This supports the conjecture that error correction is intimately linked to learning.

Although our work suggests that small quantum systems may face substantial difficulties when trying to perform Bayesian inference in an oracular setting, much more work is needed in order to provide a complete answer to the question of whether small quantum systems can learn. Further generalizations of our algorithms could be possible using fixed-point amplitude amplification [28], which could help make the update process unitary. Perhaps more evocatively, a more concrete definition of learning may be useful for deciphering when a small quantum learning agent is capable of *usefully learning* from its surroundings. Probing such questions may be essential for more than just developing new quantum machine learning algorithms, it may help us understand the fundamental limitations that quantum mechanics imposes on our ability to learn about the universe.

Acknowledgements

We would like to thank J. Combes and J. Yard for valuable feedback and discussion as well as J. Emerson for suggesting the idea of quantum learning agents.

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 A.1 *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

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

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

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

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

$$(|x\rangle + c\delta |x_2\rangle)(\langle x| + c\delta \langle x_2|) + O(\delta^2), \tag{A.3}$$

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

$$|x\rangle\langle x| + \delta \sum_{y \neq x} a_y \left(\sqrt{\frac{P(E|x)P(E|y)}{\Gamma_E^2}} + \sqrt{\left(1 - \frac{P(E|x)}{\Gamma_E}\right)\left(1 - \frac{P(E|y)}{\Gamma_E}\right)} \right) (|x\rangle\langle y| + |y\rangle\langle x|) + O(\delta^2). \tag{A.4}$$

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

$$\left(|x\rangle + \sum_{y \neq x} a_y c_y \delta |y\rangle \right) \left(\langle x| + \sum_{y \neq x} c_y \delta \langle y| \right) + O(\delta^2), \tag{A.5}$$

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

$$N = (\Delta x)^{-D}. \tag{B.1}$$

For notational simplicity, we take

$$\langle P(x), P(E|x)\rangle := \int P(x)P(E|x)\mathrm{d}^D x. \tag{B.2}$$

We then give our main theorem below using this notation.

Theorem B.1 *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

$$\begin{aligned}
&\left| \frac{\int_{V_j} P(E|x)P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x} - \frac{P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\sum_j P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x} \right| \\
&\quad \leq \left| \frac{\int_{V_j} P(E|x)P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x} - \frac{P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x} \right| \\
&\qquad + \left| \frac{P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x} - \frac{P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\sum_j P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x} \right|.
\end{aligned} \tag{B.3}$$

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

$$\int_{V_j} (P(E|x) - P(E|\bar{x}_j))P(x)\mathrm{d}^D x \leq D\Lambda\Delta x \int_{V_j} P(x)\mathrm{d}^D x, \tag{B.4}$$

which implies that

$$\left| \frac{\int_{V_j} P(E|x)P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x} - \frac{P(E|\bar{x}_j)\int_{V_j} P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x} \right| \leq \frac{D\Lambda\Delta x \int_{V_j} P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x}. \tag{B.5}$$

Now looking at the remaining term in (B.3) 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

$$\left| \frac{1}{\int P(E|x)P(x)\mathrm{d}^D x} - \frac{1}{\sum_j P(E|\bar{x}_j) \int_{V_j} P(x)\mathrm{d}^D x} \right| \leq \frac{1}{\int P(E|x)P(x)\mathrm{d}^D x} \times \max \left| 1 - \frac{1}{1 - \sum_j \Delta P(E|\bar{x}_j) \int_{V_j} P(x)\mathrm{d}^D x / \int P(E|x)P(x)\mathrm{d}^D x} \right| \quad \text{(B.6)}$$

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

$$|\Delta P(E|x)| \leq D\Lambda\Delta x \leq \int P(E|x)P(x)\mathrm{d}^D x/2. \quad \text{(B.7)}$$

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

$$\frac{1}{\int P(E|x)P(x)\mathrm{d}^D x} \max \left| 1 - \frac{1}{1 - \sum_j \int_{V_j} \Delta P(E|\bar{x}_j)P(x)\mathrm{d}^D x / \int P(E|x)P(x)\mathrm{d}^D x} \right| \leq \frac{2D\Lambda\Delta x}{\left(\int P(E|x)P(x)\mathrm{d}^D x\right)^2}. \quad \text{(B.8)}$$

This implies that

$$\left| \frac{P(E|\bar{x}_j) \int_{V_j} P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x} - \frac{P(E|\bar{x}_j) \int_{V_j} P(x)\mathrm{d}^D x}{\sum_j P(E|\bar{x}_j) \int_{V_j} P(x)\mathrm{d}^D x} \right| \leq \frac{2D\Lambda\Delta x P(E|\bar{x}_j) \int_{V_j} P(x)\mathrm{d}^D x}{\left(\int P(E|x)P(x)\mathrm{d}^D x\right)^2} \quad \text{(B.9)}$$

Thus from (B.5), (B.8) and (B.3)

$$\left| \frac{\int_{V_j} P(E|x)P(x)\mathrm{d}^D x}{\int P(E|x)P(x)\mathrm{d}^D x} - \frac{P(E|\bar{x}_j) \int_{V_j} P(x)\mathrm{d}^D x}{\sum_j P(E|\bar{x}_j) \int_{V_j} P(x)\mathrm{d}^D x} \right| \leq \frac{3D\Lambda\Delta x \int_{V_j} P(x)\mathrm{d}^D x}{\left(\int P(E|x)P(x)\mathrm{d}^D x\right)^2}. \quad \text{(B.10)}$$

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

$$\begin{aligned} &\left| \int P(x|E)x_k \mathrm{d}^D x - \sum_j \frac{[\bar{x}_j]_k P(E|\bar{x}_j) \int_{V_j} P(x)\mathrm{d}^D x}{\sum_j P(E|\bar{x}_j) \int_{V_j} P(x)\mathrm{d}^D x} \right| \\ &\leq \left| \int P(x|E)x_k \mathrm{d}^D - \sum_j \int_{V_j} P(x|E)[\bar{x}_j]_k \mathrm{d}^D x \right| \\ &\quad + \left| \sum_j \int_{V_j} P(x|E)[\bar{x}_j]_k \mathrm{d}^D x - \sum_j \frac{[\bar{x}_j]_k P(E|\bar{x}_j) \int_{V_j} P(x)\mathrm{d}^D x}{\sum_j P(E|\bar{x}_j) \int_{V_j} P(x)\mathrm{d}^D x} \right|. \end{aligned} \quad \text{(B.11)}$$

Since $0 \leq [\bar{x}_j]_k \leq 1$ and the sum of the prior probability is 1, (B.10) implies

$$\left| \sum_j \int_{V_j} P(x|E)[\bar{x}_j]_k \mathrm{d}^D x - \sum_j \frac{[\bar{x}_j]_k P(E|\bar{x}_j) \int_{V_j} P(x) \mathrm{d}^D x}{\sum_j P(E|\bar{x}_j) \int_{V_j} P(x) \mathrm{d}^D x} \right| \leq \frac{3D\Lambda\Delta x}{\left(\int P(E|x)P(x)\mathrm{d}^D x\right)^2}. \quad \text{(B.12)}$$

Similarly,

$$\left| \int P(x|E) x_k \mathrm{d}^D - \sum_j \int_{V_j} P(x|E)[\bar{x}_j]_k \mathrm{d}^D x \right| \leq \sum_j \int_{V_j} P(x|E) \mathrm{d}^D x \Delta x = \Delta x. \quad \text{(B.13)}$$

Therefore (B.11), (B.12) and (B.13) imply that the error in the posterior mean is

$$\left| \int P(x|E) x_k \mathrm{d}^D x - \sum_j \frac{[\bar{x}_j]_k P(E|\bar{x}_j) \int_{V_j} P(x) \mathrm{d}^D x}{\sum_j P(E|\bar{x}_j) \int_{V_j} P(x) \mathrm{d}^D x} \right| \leq \Delta x \left(1 + \frac{3D\Lambda}{\left(\int P(E|x)P(x)\mathrm{d}^D x\right)^2} \right). \quad \text{(B.14)}$$

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

$$\Delta x \leq \max_E \frac{\epsilon \langle P(E|x), P(x)\rangle^2}{\langle P(E|x), P(x)\rangle^2 + 3D\Lambda}. \quad \text{(B.15)}$$

Eq. (B.7) is a key assumption behind (B.15). It is then easy to see from algebra that the assumption is implied by (B.15) if

$$\epsilon \leq \frac{\langle P(E|x), P(x)\rangle^2 + 3D\Lambda}{2D\Lambda \langle P(E|x), P(x)\rangle}, \quad \text{(B.16)}$$

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 B.1 *Given the likelihood function satisfies the assumptions of Theorem B.1, 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 B.1 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: 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 latent variables directly. For instance, letting ω in the periodic likelihood

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

be drawn from a stationary Gaussian process and then marginalizing over the history of that process results in a new hyperparameterized likelihood

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

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* latent variables of the system.

Tracking such variation in the latent variables can be challenging because as Bayesian inference proceeds the certainty in the value of the latent variables tends to increase, but if these variables drift beyond the support of the posterior distribution then Bayesian inference will never be able to recover. In other words, if the true model for a system drifts into a region that the is not supported by the a prior obtained by previous updates that neglected the stochasticity of the latent variables then the algorithm will no longer be able to track the instantaneous value of the latent variable.

Fortunately, the SMC literature has already provided a solution to this problem. Approximate inference algorithms can be made to track stochastically varying latent variables, by incorporating a prediction step that diffuses the hidden variables of each particle [29]. Here, we extend this technique to the our quantum algorithm by performing Bayes updates on QFT-transformed posterior states. This allows 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 latent variables. 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

$$P \star Q \propto \mathcal{F}^{-1}\left(\mathcal{F}(P) \cdot \mathcal{F}(Q)\right), \tag{C.3}$$

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 C.1 *Let $O_{\hat{Q}}$ be a quantum oracle such that $O_{\hat{Q}}|x\rangle|y\rangle = |x\rangle\left|y \oplus \sin^{-1}(\frac{\hat{Q}(x)}{\Gamma_E})\right\rangle$ where $\hat{Q} := \mathcal{F}(Q)$ and $\hat{Q}(x) \leq \Gamma_E$ and $Q(x) \in \mathbb{C}^{2^n}$. Then given access to a unitary oracle O_{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 with error ϵ in the 2–norm using a number of queries that has an average-case query complexity of $O(\sqrt{\Gamma_E/\langle \mathcal{F}(P), \mathcal{F}(Q)\rangle})$.*

Proof. Notice 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 (C.3) and Plancherel's theorem.

First, the query complexity of this algorithm is easy to estimate. The initial state preparation requires a query to O_{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. □.

Appendix D: Adaptive experiment design

In science and engineering, inference problems frequently involve decision variables that can be set in order to optimize the performance of the algorithm. The basic idea behind designing an adaptive experiment is to choose, based on the current beliefs about a hypothesis, the most informative experiment based on these current beliefs. A natural way to do this is to select a set of experiments and then, based on the prior distribution, simulate what the expected variance in the posterior distribution would be over these experiments. A natural choice of experiment to choose would then be the one that yields the smallest expected posterior variance. While this may provide an acceptable experiment, it is unlikely to yield a locally optimal (or approximately locally optimal) experiment. In order to do so we need to be able to perform a search algorithm on top of this to find the experiment that yields the smallest expected posterior variance.

The problem is that approaches for finding locally optimal sets of parameters can be computationally expensive, even for gradient-free optimization. The reason is that many evaluations of the likelihood function are needed in the procedure, which limits its applicability in general. Here we propose using quantum techniques to accelerate finding such locally optimal experiments within the context of Bayesian experiment design.

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 $\hat{x}$,

$$\mathcal{L}(x, \hat{x}) = (x - \hat{x})^{\mathrm{T}} (x - \hat{x}) . \tag{D.1}$$

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

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

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,

$$\begin{aligned} \mathcal{R}(x, P(x)) &= \mathbb{E}_{d, x\sim P(x)}\{\mathcal{L}(x, P(x|d, c))\} \\ &= \int P(x) \int P(d|x, c) \left(x - \int P(x'|d, c) x' \mathrm{d}x' \right)^2 \mathrm{d}x \mathrm{d}d. \end{aligned} \tag{D.3}$$

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,

$$\mathcal{U}(P(x),c) = -\int P(x)\int P(d|x,c)\left(x-\int P(x'|d,c)x'\mathrm{d}x'\right)^2 \mathrm{d}x\mathrm{d}d. \tag{D.4}$$

The application of our algorithm is now made clear: like classical particle filtering methods, our algorithm can estimate expectation values over posterior distributions efficiently. 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

$$\frac{\partial\mathcal{U}(P(x),c)}{\partial c_j} = \frac{\mathcal{U}(P(x),c+h\hat{c_j})-\mathcal{U}(P(x),c)}{h}+O(h^2). \tag{D.5}$$

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 D.1 *Assume that the prior distribution $P(x)$ has support only on the interval $x \in [0,1]$ and that the observable varible E has support only on D distinct values; 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 $\epsilon \leq \min_d \int P(d|x)P(x)\mathrm{d}x/2$.*

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 (D.4) to find

$$\begin{aligned}\mathcal{U}(P(x),c) = &-\iint P(x)P(d|x,c)x^2\mathrm{d}x\mathrm{d}d\\ &+2\iiint P(x)P(d|x,c)P(x'|d,c)xx'\mathrm{d}x'\mathrm{d}d\mathrm{d}x\\ &-\iiiint P(x)P(d|x,c)P(x'|d,c)P(x''|d,c)x'x''\mathrm{d}x''\mathrm{d}x'\mathrm{d}d\mathrm{d}x.\end{aligned} \tag{D.6}$$

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 (D.6) is the easiest to compute. It can be computed by preparing the state

$$\sum_x \sqrt{P(x)}\,|x\rangle \frac{1}{\sqrt{D}}\sum_{d=1}^{D}|d\rangle\left(\sqrt{P(D|x,c)x^2}\,|1\rangle+\sqrt{1-P(D|x,c)x^2}\,|0\rangle\right). \tag{D.7}$$

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

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

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

$$\sum_x \sqrt{P(x)}\,|x\rangle \sum_{x'} \sqrt{P(x')}\,|x'\rangle \left(\sqrt{P(d|x',c)P(d|x,c)xx'}\,|1\rangle + \sqrt{1-P(d|x',c)P(d|x,c)xx'}\,|0\rangle\right), \quad \text{(D.8)}$$

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 $x \in [0,1]$

$$\sum_x \sum_{x'} P(x)P(x')P(d|x,c)P(d|x',c)xx' \leq \left(\sum_x P(x)P(d|x,c)\right)^2 = P_d^2. \quad \text{(D.9)}$$

The triple integral in (D.6) is much more challenging. It can be expressed as

$$\iiint P(x)P(d|x,c)\frac{P(d|x',c)P(x')}{\int P(d|x',c)P(x')\mathrm{d}x'}xx'\mathrm{d}x'\mathrm{d}d\mathrm{d}x. \quad \text{(D.10)}$$

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

$$\sum_x \sqrt{P(x)}\,|x\rangle \left(\sqrt{P(d|x,c)}\,|1\rangle + \sqrt{1-P(d|x,c)}\,|0\rangle\right), \quad \text{(D.11)}$$

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 [30].

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 $\epsilon_0 \leq P(d)/2$ then

$$\begin{aligned}\left|\frac{\tilde{N}(d|c)}{\tilde{P}(d)} - \frac{N(d|c)}{P(d)}\right| &\leq \left|\frac{\tilde{N}(d|c)}{\tilde{P}(d)} - \frac{N(d|c)}{\tilde{P}(d)}\right| + \left|\frac{N(d|c)}{\tilde{P}(d)} - \frac{N(d|c)}{P(d)}\right|. \\ &\leq \frac{1}{P(d)-\epsilon_0}\left|\tilde{N}(d|c) - N(d|c)\right| + P(d)^2\left|\frac{1}{\tilde{P}(d)} - \frac{1}{P(d)}\right| \\ &\leq \frac{2}{P(d)}\left|\tilde{N}(d|c) - N(d|c)\right| + P(d)\left|\frac{1}{1-\epsilon_0/P(d)} - 1\right| \\ &\leq \frac{2}{P(d)}\left|\tilde{N}(d|c) - N(d|c)\right| + \epsilon_0. \end{aligned} \tag{D.12}$$

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

$$N(d|c) \leq (P(d) + \epsilon_0)^2 \in O(P^2(d)), \tag{D.13}$$

since $\epsilon_0 \leq P(d)/2$. 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 (D.12) 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

$$\left|\frac{\partial U(c)}{\partial c_j} - \frac{U(c+\delta_j) - U(c-\delta_j)}{2\delta}\right| \leq \max_{c,j}\left|\frac{\partial^3 U(c)}{\partial c_j^3}\right| \frac{\delta^2}{6}, \tag{D.14}$$

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

$$\begin{aligned}\left|\frac{\partial U(c)}{\partial c_j} - \frac{\tilde{U}(c+\delta_j) - \tilde{U}(c-\delta_j)}{2\delta}\right| \leq &\left|\frac{\partial U(c)}{\partial c_j} - \frac{U(c+\delta_j) - U(c-\delta_j)}{2\delta}\right| + \\ &\left|\frac{U(c+\delta_j) - U(c-\delta_j)}{2\delta} - \frac{\tilde{U}(c+\delta_j) - \tilde{U}(c-\delta_j)}{2\delta}\right|, \end{aligned} \tag{D.15}$$

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

$$O\left(\max_{c,j}\left|\frac{\partial^3 U(c)}{\partial c_j^3}\right| \delta^2 + \frac{\epsilon_0}{\delta}\right). \tag{D.16}$$

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

$$O\left(\epsilon_0^{2/3} \max_{c,j} \left| \frac{\partial^3 U(c)}{\partial c_j^3} \right|^{1/3}\right). \tag{D.17}$$

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. $\square$.

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 $O(D/\epsilon_0^2)$ queries. It is also worth noting that high–order methods for estimating the gradient may be useful for further improving the error scaling.

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 (D.11). Similar states could therefore also be prepared for problems such as quantum Hamiltonian learning [31] 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.

1. Peter W Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 124–134. IEEE, 1994.
2. Seth Lloyd et al. Universal quantum simulators. *Science*, 273:1073–1077, 1996.
3. Andrew M Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 59–68. ACM, 2003.
4. Aram W Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009.
5. Scott Aaronson. The learnability of quantum states. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 463(2088):3089–3114, December 2007.
6. Christopher E Granade, Christopher Ferrie, Nathan Wiebe, and D G Cory. Robust online Hamiltonian learning. *New Journal of Physics*, 14(10):103013, October 2012.
7. Christopher Ferrie and Christopher E. Granade. Likelihood-free methods for quantum parameter estimation. *Physical Review Letters*, 112(13):130402, April 2014.
8. Esma Aïmeur, Gilles Brassard, and Sébastien Gambs. Machine learning in a quantum world. In *Advances in Artificial Intelligence*, pages 431–442. Springer, 2006.
9. Nathan Wiebe, Ashish Kapoor, and Krysta M Svore. Quantum deep learning. *arXiv preprint arXiv:1412.3489*, 2014.
10. Nathan Wiebe, Ashish Kapoor, and Krysta Svore. Quantum nearest-neighbor algorithms for machine learning. *arXiv preprint arXiv:1401.2142*, 2014.
11. Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum algorithms for supervised and unsupervised machine learning. *arXiv preprint arXiv:1307.0411*, 2013.
12. Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014.

13. Sergio Boixo, Gerardo Ortiz, and Rolando Somma. Fast quantum methods for optimization. *The European Physical Journal Special Topics*, 224(1):35–49, 2015.
14. Guang Hao Low, Theodore J Yoder, and Isaac L Chuang. Quantum inference on bayesian networks. *Physical Review A*, 89(6):062315, 2014.
15. Maris Ozols, Martin Roetteler, and Jérémie Roland. Quantum rejection sampling. *ACM Transactions on Computation Theory (TOCT)*, 5(3):11, 2013.
16. Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *Contemporary Mathematics*, 305:53–74, 2002.
17. Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM, 1996.
18. Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *arXiv preprint quant-ph/9605034*, 1996.
19. Paul Dagum and Michael Luby. Approximating probabilistic inference in bayesian belief networks is NP-hard. *Artificial Intelligence*, 60(1):141–153, 1993.
20. Jane Liu and Mike West. Combined parameter and state estimation in simulation-based filtering. In De Freitas and NJ Gordon, editors, *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, New York, 2001.
21. Thomas P Minka. Expectation propagation for approximate bayesian inference. In *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, pages 362–369. Morgan Kaufmann Publishers Inc., 2001.
22. Rudolph Van Der Merwe, Arnaud Doucet, Nando De Freitas, and Eric Wan. The unscented particle filter. In *NIPS*, pages 584–590, 2000.
23. Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002.
24. Valerio Scarani, Sofyan Iblisdir, Nicolas Gisin, and Antonio Acin. Quantum cloning. *Reviews of Modern Physics*, 77(4):1225, 2005.
25. Dorit Aharonov, Vaughan Jones, and Zeph Landau. A polynomial quantum algorithm for approximating the jones polynomial. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 427–436. ACM, 2006.
26. Nathan Wiebe, Christopher Granade, Ashish Kapoor, and Krysta M Svore. Bayesian inference via rejection filtering. *arXiv preprint arXiv:1511.06458*, 2015.
27. Dave Wecker, Matthew B Hastings, Nathan Wiebe, Bryan K Clark, Chetan Nayak, and Matthias Troyer. Solving strongly correlated electron models on a quantum computer. *arXiv preprint arXiv:1506.05135*, 2015.
28. Theodore J Yoder, Guang Hao Low, and Isaac L Chuang. Fixed-point quantum search with an optimal number of queries. *Physical review letters*, 113(21):210501, 2014.
29. Michael Isard and Andrew Blake. CONDENSATIONConditional Density Propagation for Visual Tracking. *International Journal of Computer Vision*, 29(1):5–28, August 1998.
30. Gilles Brassard, Peter Hoyer, Michele Mosca, and Alain Tapp. Quantum amplitude amplification and estimation. *arXiv preprint quant-ph/0005055*, 2000.
31. Nathan Wiebe, Christopher Granade, Christopher Ferrie, and DG Cory. Hamiltonian learning and certification using quantum resources. *Physical Review Letters*, 112(19):190501, 2014.