# Quantum inference on Bayesian networks

Guang Hao Low, Theodore J. Yoder, and Isaac L. Chuang
*Massachusetts Institute of Technology, 77 Massachusetts Avenue, Cambridge, Massachusetts 02139, USA*

(Received 28 February 2014; published 13 June 2014)

Performing exact inference on Bayesian networks is known to be $\#P$-hard. Typically approximate inference techniques are used instead to sample from the distribution on query variables given the values $e$ of evidence variables. Classically, a single unbiased sample is obtained from a Bayesian network on $n$ variables with at most $m$ parents per node in time $O(nmP(e)^{-1})$, depending critically on $P(e)$, the probability that the evidence might occur in the first place. By implementing a quantum version of rejection sampling, we obtain a square-root speedup, taking $O(n2^m P(e)^{-\frac{1}{2}})$ time per sample. We exploit the Bayesian network's graph structure to efficiently construct a quantum state, a q-sample, representing the intended classical distribution, and also to efficiently apply amplitude amplification, the source of our speedup. Thus, our speedup is notable as it is unrelativized—we count primitive operations and require no blackbox oracle queries.

DOI: 10.1103/PhysRevA.89.062315 PACS number(s): 03.67.Ac, 02.50.Tt

## I. INTRODUCTION

How are rational decisions made? Given a set of possible actions, the logical answer is the one with the largest corresponding utility. However, estimating these utilities accurately is the problem. A rational agent endowed with a model and partial information of the world must be able to evaluate the probabilities of various outcomes, and such is often done through inference on a Bayesian network [1], which efficiently encodes joint probability distributions in a directed acyclic graph of conditional probability tables. In fact, the standard model of a decision-making agent in a probabilistic time-discretized world, known as a partially observable Markov decision process, is a special case of a Bayesian network. Furthermore, Bayesian inference finds application in processes as diverse as system modeling [2], model learning [3,4], data analysis [5], and decision making [6], all falling under the umbrella of machine learning [1].

Unfortunately, despite the vast space of applications, Bayesian inference is difficult. To begin with, exact inference is $\#P$-hard in general [1]. It is often far more feasible to perform approximate inference by sampling, such as with the Metropolis-Hastings algorithm [7] and its innumerable specializations [8], but doing so is still NP-hard in general [9]. This can be understood by considering rejection sampling, a primitive operation common to many approximate algorithms that generates unbiased samples from a target distribution $P(\mathcal{Q}|\mathcal{E})$ for some set of query variables $\mathcal{Q}$ conditional on some assignment of evidence variables $\mathcal{E} = e$. In the general case, rejection sampling requires sampling from the full joint distribution $P(\mathcal{Q},\mathcal{E})$ and throwing away samples with incorrect evidence. In the specific case in which the joint distribution is described by a Bayesian network with $n$ nodes each with no more than $m$ parents, it takes time $O(nm)$ to generate a sample from the joint distribution, and so a sample from the conditional distribution $P(\mathcal{Q}|\mathcal{E})$ takes average time $O(nmP(e)^{-1})$. Much of the computational difficulty is related to how the marginal $P(e) = P(\mathcal{E} = e)$ becomes exponentially small as the number of evidence variables increases, since only samples with the correct evidence assignments are recorded.

One very intriguing direction for speeding up approximate inference is in developing hardware implementations of sampling algorithms, for which promising results such as natively probabilistic computing with stochastic logic gates have been reported [10]. In this same vein, we could also consider physical systems that already describe probabilities and their evolution in a natural fashion to discover whether such systems would offer similar benefits.

Quantum mechanics can in fact describe such naturally probabilistic systems. Consider an analogy: If a quantum state is like a classical probability distribution, then measuring it should be analogous to sampling, and unitary operators should be analogous to stochastic updates. Though this analogy is qualitatively true and appealing, it is inexact in ways yet to be fully understood. Indeed, it is a widely held belief that quantum computers offer a strictly *more* powerful set of tools than classical computers, even probabilistic ones [11], though this appears difficult to prove [12]. Notable examples of the power of quantum computation include exponential speedups for finding prime factors with Shor's algorithm [13], and square-root speedups for generic classes of search problems through Grover's algorithm [14]. Unsurprisingly, there is a ongoing search for ever more problems amenable to quantum attack [15–17].

For instance, the quantum rejection sampling algorithm for approximate inference was only developed quite recently [18], alongside a proof, relativized by an oracle, of a square-root speedup in runtime over the classical algorithm. The algorithm, just like its classical counterpart, is an extremely general method of doing approximate inference, requiring preparation of a quantum pure state representing the joint distribution $P(\mathcal{Q},\mathcal{E})$ and amplitude amplification to amplify the part of the superposition with the correct evidence. Owing to its generality, the procedure assumes access to a state-preparation oracle $\hat{A}_P$, and the runtime is therefore measured by the query complexity [19], the number of times the oracle must be used. Unsurprisingly, such oracles may not be efficiently implemented in general, as the ability to prepare arbitrary states allows for witness generation to QMA-complete problems, the quantum analog to classically deterministic NP-complete problems [18,20]. This also corresponds consistently to the NP-hardness of classical sampling.

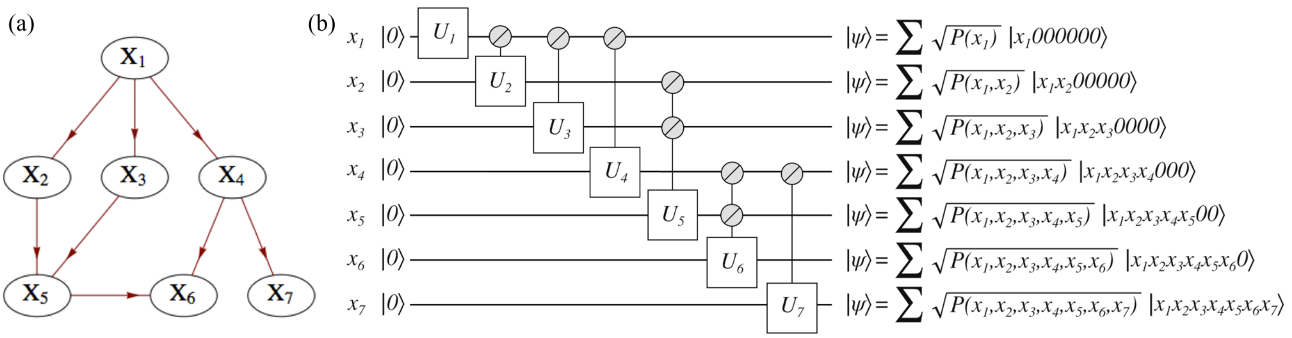
(a)
X1
X2
X3
X4
X5
X6
X7
(b)
x1 |0⟩
x2 |0⟩
x3 |0⟩
x4 |0⟩
x5 |0⟩
x6 |0⟩
x7 |0⟩
U1
U2
U3
U4
U5
U6
U7
|ψ⟩ = Σ √P(x1) |x1000000⟩
|ψ⟩ = Σ √P(x1,x2) |x1x200000⟩
|ψ⟩ = Σ √P(x1,x2,x3) |x1x2x30000⟩
|ψ⟩ = Σ √P(x1,x2,x3,x4) |x1x2x3x4000⟩
|ψ⟩ = Σ √P(x1,x2,x3,x4,x5) |x1x2x3x4x500⟩
|ψ⟩ = Σ √P(x1,x2,x3,x4,x5,x6) |x1x2x3x4x5x60⟩
|ψ⟩ = Σ √P(x1,x2,x3,x4,x5,x6,x7) |x1x2x3x4x5x6x7⟩

FIG. 1. (Color online) (a) An example of a directed acyclic graph which can represent a Bayesian network by associating with each node a conditional probability table. For instance, associated with the node $X_1$ is the one value $P(X_1 = 1)$, while that of $X_5$ consists of four values, the probabilities $X_5 = 1$ given each setting of the parent nodes, $X_2$ and $X_3$. (b) A quantum circuit that efficiently prepares the q-sample representing the full joint distribution of panel (a). Notice in particular how the edges in the graph are mapped to conditioning nodes in the circuit. The $|\psi_j\rangle$ represent the state of the system after applying the operator sequence $U_1,\dots,U_j$ to the initial state $|0000000\rangle$.

In this paper, we present an unrelativized (i.e., no oracle) square-root, quantum speedup to rejection sampling on a Bayesian network. Just as the graphical structure of a Bayesian network speeds up classical sampling, we find that the same structure allows us to construct the state-preparation oracle $\hat{A}_P$ efficiently. Specifically, quantum sampling from $P(\mathcal{Q}|\mathcal{E}=e)$ takes time $O(n2^m P(e)^{-1/2})$, compared with $O(nmP(e)^{-1})$ for classical sampling, where $m$ is the maximum number of parents of any one node in the network. We exploit the structure of the Bayesian network to construct an efficient quantum circuit $\hat{A}_P$ composed of $O(n2^m)$ controlled-NOT gates and single-qubit rotations that generates the quantum state $|\psi_P\rangle$ representing the joint $P(\mathcal{Q},\mathcal{E})$. This state must then be evolved to $|\mathcal{Q}\rangle$ representing $P(\mathcal{Q}|\mathcal{E}=e)$, which can be done by performing amplitude amplification [21], the source of our speedup and heart of quantum rejection sampling in general [18]. The desired sample is then obtained in a single measurement of $|\mathcal{Q}\rangle$.

We better define the problem of approximate inference with a review of Bayesian networks in Sec. II. We discuss the sensible encoding of a probability distribution in a quantum state axiomatically in Sec. III. This is followed by an overview of amplitude amplification in Sec. IV. The quantum rejection sampling algorithm is given in Sec. V. As our main result, we construct circuits for the state preparation operator in Secs. VI A and VI B and circuits for the reflection operators for amplitude amplification in Sec. VI C. The total time complexity of quantum rejection sampling in Bayesian networks is evaluated in Sec. VI D, and we present avenues for further work in Sec. VII.

## II. BAYESIAN NETWORKS

A Bayesian network is a directed acyclic graph structure that represents a joint probability distribution over $n$ bits. A significant advantage of the Bayesian network representation is that the space complexity of the representation can be made much less than the general case by exploiting conditional dependencies in the distribution. This is achieved by associating with each graph node a conditional probability table for each random variable, with directed edges representing conditional dependencies, such as in Fig. 1(a).

We adopt the standard convention of capital letters (e.g., $X$) representing random *variables* while lowercase letters (e.g., $a$) are particular fixed *values* of those variables. For simplicity, the random variables are taken to be binary. Accordingly, probability vectors are denoted $P(X)=\{P(X=0),P(X=1)\}$ while $P(x)\equiv P(X=x)$. Script letters represent a set of random variables $\mathcal{X}=\{X_1,X_2,\dots,X_n\}$.

An arbitrary joint probability distribution $P(x_1,x_2,\dots,x_n)$ on $n$ bits can always be factored by recursive application of Bayes's rule $P(X,Y)=P(X)P(Y|X)$,

$$P(x_1,x_2,\dots,x_n)=P(x_1)\prod_{i=2}^{n}P(x_i|x_1,\dots,x_{i-1}). \qquad (1)$$

However, in most practical situations a given variable $X_i$ will be dependent on only a few of its predecessors' values, those we denote by parents $(X_i)\subseteq\{x_1,\dots,x_{i-1}\}$ [see Fig. 1(a)]. Therefore, the factorization above can be simplified to

$$P(x_1,x_2,\dots,x_n)=P(x_1)\prod_{i=2}^{n}P(x_i|\,\text{parents}(X_i)). \qquad (2)$$

A Bayes net diagrammatically expresses this simplification, with a topological ordering on the nodes $X_1\preceq X_2\preceq\cdots\preceq X_n$ in which parents are listed before their children. With a node $x_i$ in the Bayes net, the conditional probability factor $P(x_i=1|\,\text{parents}(X_i))$ is stored as a table of $2^{m_i}$ values [1], where $m_i$ is the number of parents of node $X_i$, also known as the indegree. Letting $m$ denote the largest $m_i$, the Bayes net data structure stores at most $O(n2^m)$ probabilities, a significant improvement over the direct approach of storing $O(2^n)$ probabilities [1].

A common problem with any probability distribution is inference. Say we have a complete joint probability distribution on $n$ bits, $P(\mathcal{X})$. Given the values $e=e_{|\mathcal{E}|},\dots,e_2e_1$ for a set $\mathcal{E}\subseteq\mathcal{X}$ of random variables, the task is to find the distribution over a collection of query variables $\mathcal{Q}\subseteq\mathcal{X}\setminus\mathcal{E}$. That is, the exact inference problem is to calculate $P(\mathcal{Q}|\mathcal{E}=e)$. Exact inference is #$P$-hard [1], since one can create a Bayes net encoding the $n$ variable $k$-SAT problem, with nodes for each variable, each clause, and the final verdict—a count of the satisfying assignments.

Approximate inference on a Bayesian network is much simpler, thanks to the graphical structure. The procedure for sampling is as follows: Working from the top of the network, generate a value for each node, given the values already generated for its parents. Since each node has at most $m$ parents that we must inspect before generating a value, and there are $n$ nodes in the tree, obtaining a sample $\{x_1,x_2\ldots,x_n\}$ takes time $O(nm)$. For unbiased sampling from the full joint distribution $P(\mathcal{X})$ this runtime is classically optimal [1] as it follows exactly the decomposition in Eq. (2).

Having sampled from $P(\mathcal{X})$, to complete the classical approximate inference algorithm, we must now postselect on the correct evidence values $\mathcal{E}=e$, leaving us with an average time per sample of $O(nmP(e)^{-1})$, which suffers when the probability $P(e)$ becomes small, typically exponentially small with the number of evidence variables $|\mathcal{E}|$. Quantum rejection sampling, however, will improve the factor of $P(e)^{-1}$ to $P(e)^{-1/2}$, while preserving the linear scaling in the number of variables $n$, given that we use an appropriate quantum state to represent the Bayesian network.

## III. QUANTUM SAMPLING FROM $P(\mathcal{X})$

This section explores the analogy between quantum states and classical probability distributions from first principles. In particular, for a classical probability distribution function $P(\mathcal{X})$ on a set of $n$ binary random variables $\mathcal{X}$ what quantum state $\rho_P$ (possibly mixed, $d$ qubits) should we use to represent it? The suitable state, which we call a quantum probability distribution function (qpdf), is defined with three properties.

*Definition 1.* A qpdf for the probability distribution $P(\mathcal{X})$ has the following three properties:

(1) Purity: In the interest of implementing quantum algorithms, we require that the qpdf be a pure state $\rho_P=|\Psi_P\rangle\langle\Psi_P|$.

(2) Q-sampling: A single qpdf can be measured to obtain a classical $n$-bit string, a sample from $P(\mathcal{X})$. Furthermore, for any subset of variables $\mathcal{W}\subset\mathcal{X}$, a subset of qubits in the qpdf can be measured to obtain a sample from the marginal distribution $P(\mathcal{W})$. We call these measurement procedures q-sampling.

(3) Q-stochasticity: For every stochastic matrix $T$ there is a unitary $U_T$ such that whenever $T$ maps the classical distribution $P(\mathcal{X})$ to $P'(\mathcal{X})$, $U_T$ maps the qpdf $|\Psi_P\rangle$ to $|\Psi_{P'}\rangle=U_T|\Psi_P\rangle$.

The motivation for property 3 is for implementing Markov chains, Markov decision processes, or even sampling algorithms such as Metropolis-Hastings, on quantum states [22]. The question we pose, and leave open, is whether a qpdf exists.

The simplest way to satisfy the first two criteria, but not the third, is to initialize a single qubit for each classical binary random variable. This leads to what is called the q-sample, defined in prior work [23] as

*Definition 2.* The q-sample of the joint distribution $P(x_1,\ldots,x_n)$ over $n$ binary variables $\{X_i\}$ is the $n$-qubit pure state $|\psi_P\rangle=\sum_{x_1,\ldots,x_n}\sqrt{P(x_1,\ldots,x_n)}|x_1,\ldots,x_n\rangle$.

The q-sample possesses property 1 and the eponymous property 2 above. However, it does not allow for stochastic updates as per property 3, as a simple single-qubit example shows. In that case, property 3 requires

$$\begin{pmatrix}U_{11} & U_{12}\\ U_{21} & U_{22}\end{pmatrix}\begin{pmatrix}\sqrt{p}\\ \sqrt{1-p}\end{pmatrix}=\begin{pmatrix}\sqrt{pT_{11}+(1-p)T_{12}}\\ \sqrt{pT_{21}+(1-p)T_{22}}\end{pmatrix}, \tag{3}$$

for all $p\in[0,1]$. Looking at Eq. (3) for $p=0$ and $p=1$ constrains $U$ completely, and it is never unitary when $T$ is stochastic. Thus, the q-sample fails to satisfy property 3.

Yet, the q-sample satisfies properties 1 and 2 in a very simplistic fashion, and various more complicated forms might be considered. For instance, relative phases could be added to the q-sample giving $\sum_x e^{i\phi(x)}\sqrt{P(x)}|x\rangle$, though this alone does not guarantee property 3, which is easily checked by adding phases to the proof above. Other extensions of the q-sample may include ancilla qubits, different measurement bases, or a postprocessing step including classical randomness to translate the measurement result into a classical sample. It is an open question whether a more complicated representation satisfying all three properties exists, including q-stochasticity, so that we would have upgraded the q-sample $|\psi_P\rangle$ into a qpdf $|\Psi_P\rangle$ possessing all the defining properties.

Nevertheless, although the q-sample is not a qpdf by our criteria, it will still be very useful for performing quantum rejection sampling. The property that a sample from a marginal distribution is obtained by simply measuring a subset of qubits means that, using conditional gates, we can form a q-sample for a conditional distribution from the q-sample for the full joint distribution, as we show in Sec. VI. This corresponds to the classical formula $P(\mathcal{V}|\mathcal{W})=P(\mathcal{V},\mathcal{W})/P(\mathcal{W})$, which is the basis behind rejection sampling. The way it is actually done quickly on a q-sample is through amplitude amplification, reviewed next, in the general case.

## IV. AMPLITUDE AMPLIFICATION

Amplitude amplification [21] is a well-known extension of Grover's algorithm and is the second major concept in the quantum inference algorithm. Given a quantum circuit $\hat{A}$ for the creation of an $n$-qubit pure state $|\psi\rangle=\alpha|\psi_t\rangle+\beta|\bar{\psi}_t\rangle=\hat{A}|0\rangle^{\otimes n}$, where $\langle\psi_t|\bar{\psi}_t\rangle=0$, the goal is to return the target state $|\psi_t\rangle$ with high probability with as few queries to state preparation $\hat{A}$ as possible [24]. To make our circuit constructions more explicit, we assume target states are marked by a known evidence bit string $e=e_{|\mathcal{E}|}\ldots e_2e_1$, so that $|\psi_t\rangle=|\mathcal{Q}\rangle|e\rangle$ lives in the tensor product space $\mathcal{H}_\mathcal{Q}\otimes\mathcal{H}_\mathcal{E}$ and the goal is to extract $|\mathcal{Q}\rangle$.

Just like in Grover's algorithm, a pair of reflection operators are applied repetitively to rotate $|\psi\rangle$ into $|\psi_t\rangle$. Reflection about the evidence is performed by $\hat{S}_e=\hat{I}\otimes(\hat{I}-2|e\rangle\langle e|)$ followed by reflection about the initial state by $\hat{S}_\psi=(\hat{I}-2|\psi\rangle\langle\psi|)$. Given $\hat{A}$, then $\hat{S}_\psi=\hat{A}\hat{S}_0\hat{A}^\dagger$, where $\hat{S}_0=(\hat{I}-2|0\rangle\langle 0|^{\otimes n})$.

The analysis of the amplitude amplification algorithm is elucidated by writing the Grover iterate $\hat{G}=-\hat{S}_\psi\hat{S}_e=-\hat{A}\hat{S}_0\hat{A}^\dagger\hat{S}_e$ in the basis of $\frac{\alpha}{|\alpha|}|\psi_t\rangle\equiv\binom{1}{0}$ and $\frac{\beta}{|\beta|}|\bar{\psi}_t\rangle\equiv\binom{0}{1}$ [19],

$$\hat{G}=\begin{pmatrix}1-2|\alpha|^2 & 2|\alpha|\sqrt{1-|\alpha|^2}\\ -2|\alpha|\sqrt{1-|\alpha|^2} & 1-2|\alpha|^2\end{pmatrix}. \tag{4}$$

In this basis, the Grover iterate corresponds to a rotation by small angle $\theta=\cos^{-1}(1-2|\alpha|^2)\approx 2|\alpha|$. Therefore, applying the iterate $N$ times rotates the state by $N\theta$. We conclude that $\hat{G}^N|\psi\rangle$ is closest to $\frac{\alpha}{|\alpha|}|\psi_t\rangle$ after $N=O(\frac{\pi}{4|\alpha|})$ iterations.

Usually, amplitude amplification needs to be used without knowing the value of $|\alpha|$. In that case, $N$ is not known. However, the situation is remedied by guessing the correct number of Grover iterates to apply in exponential progression. That is, we apply $\hat{G}$ $2^k$ times, with $k = 0,1,2,\dots$, measure the evidence qubits $|\mathcal{E}\rangle$ after each attempt, and stop when we find $\mathcal{E} = e$. It has been shown [21] that this approach also requires on average $O(\frac{1}{|\alpha|})$ applications of $\hat{G}$.

## V. THE QUANTUM REJECTION SAMPLING ALGORITHM

The quantum rejection sampling algorithm [18], which we review now, is an application of amplitude amplification on a q-sample. The general problem, as detailed in Sec. II, is to sample from the $n$-bit distribution $P(\mathcal{Q}|\mathcal{E}=e)$. We assume that we have a circuit $\hat{A}_P$ that can prepare the q-sample $|\psi_P\rangle = \hat{A}_P|0\rangle^{\otimes n}$. Now, permuting qubits so the evidence lies to the right, the q-sample can be decomposed into a superposition of states with correct evidence and states with incorrect evidence,

$$|\psi_P\rangle = \sqrt{P(e)}\,|\mathcal{Q}\rangle\,|e\rangle + \sqrt{1-P(e)}|\overline{\mathcal{Q},e}\rangle, \qquad (5)$$

where $|\mathcal{Q}\rangle$ denotes the q-sample of $P(\mathcal{Q}|\mathcal{E}=e)$, our target state. Next perform the amplitude amplification algorithm from the last section to obtain $|\mathcal{Q}\rangle$ with high probability. Note that this means the state preparation operator $\hat{A}_P$ must be applied $\mathcal{O}(P(e)^{-1/2})$ times. Once obtained, $|\mathcal{Q}\rangle$ can be measured to get a sample from $P(\mathcal{Q}|\mathcal{E}=e)$, and we have therefore done approximate inference. Pseudocode is provided as an algorithm 1.

However, we are so far missing a crucial element. How is the q-sample preparation circuit $\hat{A}_P$ actually implemented, and can this implementation be made efficient, that is, polynomial in the number of qubits $n$? The answer to this question removes the image of $\hat{A}_P$ as a featureless black box and is addressed in the next section.

## VI. CIRCUIT CONSTRUCTIONS

While the rejection sampling algorithm from Sec. V is entirely general for any distribution $P(\mathcal{X})$, the complexity of q-sample preparation, in terms of the total number of CNOTs and single qubit rotations involved, is generally exponential in the number of qubits, $O(2^n)$. We show this in Sec. VI A, providing the complexity analysis of the general state preparation algorithm [25], which uses conditional qubit rotations based on conditional probabilities [26,27]. The difficulty of such a preparation is not surprising, since arbitrary q-sample preparation encompasses witness generation to QMA-complete problems [18,20].

However, there are cases in which the q-sample can be prepared efficiently [23,27]. The main result of this paper adds to those cases—for probability distributions resulting from a Bayesian network $\mathcal{B}$ with $n$ nodes and maximum indegree $m$, the circuit complexity of the q-sample preparation circuit $\hat{A}_\mathcal{B}$ is $O(n2^m)$. We show this in Sec. VI B. The circuit constructions for the remaining parts of the Grover iterate, the phase flip operators, are given in Sec. VI C. Finally, we evaluate the complexity of our constructions as a whole in Sec. VI D and find that approximate inference on Bayesian networks can be done with a polynomially sized quantum circuit.

Throughout this section we denote the circuit complexity of a circuit $\hat{C}$ as $Q_{\hat{C}}$. This complexity measure is the count of the number of gates in $\hat{C}$ after compilation into a complete, primitive set. The primitive set we employ includes the CNOT gate and all single-qubit rotations.

**Algorithm 1** Quantum rejection sampling algorithm: generate one sample from $P(\mathcal{Q}|\mathcal{E}=e)$ given a q-sample preparation circuit $\hat{A}_P$

```
$k \leftarrow -1$
while evidence $\mathcal{E} \neq e$ do
    $k \leftarrow k+1$
    $|\psi_P\rangle \leftarrow \hat{A}_P|0\rangle^{\otimes n}$   //prepare a q-sample of $P(\mathcal{X})$
    $|\psi_P'\rangle \leftarrow \hat{G}^{2^k}|\psi_P\rangle$   //where $\hat{G} = -\hat{A}_P\hat{S}_0\hat{A}_P^\dagger\hat{S}_e$
    Measure evidence qubits $\mathcal{E}$ of $|\psi_P'\rangle$
Measure the query qubits to obtain a sample $\mathcal{Q} = q$
```

### A. Q-sample preparation

If $P(x)$ lacks any kind of structure, the difficulty of preparing the q-sample $|\psi_P\rangle = \hat{A}_P|0\rangle^{\otimes n}$ with some unitary $\hat{A}_P$ scales at least exponentially with the number of qubits $n$ in the q-sample. Since $P(x)$ contains $2^n - 1$ arbitrary probabilities, $\hat{A}_P$ must contain at least that many primitive operations. In fact, the bound is tight—we can construct a quantum circuit preparing $|\psi_P\rangle$ with complexity $\mathcal{O}(2^n)$.

*Theorem 1.* Given an arbitrary joint probability distribution $P(x_1,\dots,x_n)$ over $n$ binary variables $\{X_i\}$, there exists a quantum circuit $\hat{A}_P$ that prepares the q-sample $\hat{A}_P|0\rangle^{\otimes n} = |\psi_P\rangle = \sum_{x_1,\dots,x_n}\sqrt{P(x_1,\dots,x_n)}|x_1\cdots x_n\rangle$ with circuit complexity $\mathcal{O}(2^n)$.

*Proof.* Decompose $P(x) = P(x_1)\prod_{i=2}^n P(x_i|x_1\dots x_{i-1})$ as per Eq. (1). For each conditional distribution $P(X_i|x_1\dots x_{i-1})$, let us define the $i$-qubit uniformly controlled rotation $\hat{U}_i$ such that given an $(i-1)$ bit string assignment $x_c \equiv x_1\dots x_{i-1}$ on the control qubits, the action of $\hat{U}_i$ on the $i$th qubit initialized to $|0\rangle_i$ is a rotation about the $y$ axis by angle $2\tan^{-1}(\sqrt{P(x_i=1|x_c)/P(x_i=0|x_c)})$ or $\hat{U}_i|0\rangle_i = \sqrt{P(x_i=0|x_c)}|0\rangle_i + \sqrt{P(x_i=1|x_c)}|1\rangle_i$. With this definition, the action of the single-qubit $\hat{U}_1$ is $\hat{U}_1|0\rangle_1 = \sqrt{P(x_1=0)}|0\rangle_1 + \sqrt{P(x_1=1)}|1\rangle_1$. By applying Bayes's rule in reverse, the operation $\hat{A}_P = \hat{U}_n\dots\hat{U}_1$ then produces $|\psi_P\rangle = \hat{A}_P|0\rangle$. As each $k$-qubit uniformly controlled rotation is decomposable into $O(2^k)$ CNOTs and single-qubit rotations [28], the circuit complexity of $\hat{A}_P$ is $Q_{\hat{A}_P} = \sum_{i=1}^n O(2^{i-1}) = O(2^n)$. ■

The key quantum compiling result used in this proof is the construction of Bergholm *et al.* [28] that decomposes $k$-qubit uniformly controlled gates into $O(2^k)$ CNOTs and single-qubit operations. Each uniformly controlled gate is the realization of a conditional probability table from the factorization of the joint distribution. We use this result again in Bayesian q-sample preparation.

### B. Bayesian q-sample preparation

We now give our main result, a demonstration that the circuit $\hat{A}_\mathcal{B}$ that prepares the q-sample of a Bayesian network is exponentially simpler than the general q-sample preparation circuit $\hat{A}_P$. We begin with a Bayesian network with, as usual, $n$ nodes and maximum indegree $m$ that encodes a distribution $P(\mathcal{X})$. As a minor issue, because the Bayesian network may have nodes reordered, the indegree $m$ is actually a function of the specific parentage of nodes in the tree. This nonuniqueness of $m$ corresponds to the nonuniqueness of the decomposition $P(x_1,\ldots,x_n) = P(x_1)\prod_{i=2}^{n} P(x_i|x_1\ldots x_{i-1})$ due to permutations of the variables. Finding the variable ordering minimizing $m$ is unfortunately an NP-hard problem [29], but typically the variables have real-world meaning and the natural causal ordering often comes close to optimal [30]. In any case, we take $m$ as a constant much less than $n$.

*Definition 3.* If $P(\mathcal{X})$ is the probability distribution represented by a Bayesian network $\mathcal{B}$, the Bayesian q-sample $|\psi_{\mathcal{B}}\rangle$ denotes the q-sample of $P(\mathcal{X})$.

*Theorem 2.* The Bayesian q-sample of the Bayesian network $\mathcal{B}$ with $n$ nodes and bounded indegree $m$ can be prepared efficiently by an operator $\hat{A}_{\mathcal{B}}$ with circuit complexity $\mathcal{O}(n2^m)$ acting on the initial state $|0\rangle^{\otimes n}$.

*Proof.* As a Bayesian network is a directed acyclic graph, let us order the node indices topologically such that for all $1 \leqslant i \leqslant n$, we have parents$(x_i) \subseteq \{x_1,x_2,\ldots,x_{i-1}\}$, and $\max_i|\text{parents}(x_i)| = m$. Referring to the construction from the proof of theorem 1, the state preparation operator $\hat{A} = \hat{U}_n \ldots \hat{U}_1$ then contains at most $m$-qubit uniformly controlled operators, each with circuit complexity $O(2^m)$, again from Bergholm *et al.* [28]. The circuit complexity of $\hat{A}_{\mathcal{B}}$ is thus $Q_{\hat{A}_{\mathcal{B}}} = \sum_{i=1}^{n} O(2^m) = O(n2^m)$. ■

Figure 1(b) shows the circuit we have just described. We also note here that, although the complexity of $\hat{A}_{\mathcal{B}}$ is in general $O(n2^m)$, there are Bayesian networks where the complexity is reduced to the classical value of $O(nm)$. This reduced bound arises when the conditional gates in Fig. 1(b) factor into several conditional gates, each conditioned on a single qubit. For example, the Bayesian network in which every child depends on its parents by the binary addition operation, $x_i = y_1 \oplus y_2 \oplus \cdots \oplus y_m$ for $y_k \in$ parents$(x_i)$, is an explicit instance saturating this reduced bound.

The Bayesian q-sample preparation we have covered in this section forms part of the Grover iterate required for amplitude amplification. The rest is comprised of the reflection, or phase flip, operators.

### C. Phase flip operators

Here we show that the phase flip operators are also efficiently implementable, so that we can complete the argument that amplitude amplification on a Bayesian q-sample is polynomial time. Note first that the phase flip operators $\hat{S}_e$ acting on $k = |\mathcal{E}| \leqslant n$ qubits can be implemented with a single $k$-qubit controlled $Z$ operation along with at most $2k$ bit flips. The operator $\hat{S}_0$ is the special case $\hat{S}_{e=0^n}$. A Bayesian q-sample can be decomposed exactly as in Eq. (5):

$$|\psi_{\mathcal{B}}\rangle = \sqrt{P(e)}\,|\mathcal{Q}\rangle\,|e\rangle + \sqrt{1-P(e)}|\overline{\mathcal{Q},e}\rangle. \tag{6}$$

Recall $|\mathcal{Q}\rangle$ is the q-sample of $P(\mathcal{Q}|\mathcal{E}=e)$ and $|\overline{\mathcal{Q},e}\rangle$ contains all states with invalid evidence $\mathcal{E} \neq e$. We write the evidence as a $k$-bit string $e = e_k\ldots e_2e_1$ and $\hat{X}_i$ as the bit flip on the $i$th evidence qubit. The controlled phase, denoted $\hat{Z}_{1\ldots k}$, acts on all $k$ evidence qubits symmetrically, flipping the phase if and only if all qubits are 1. Then $\hat{S}_e$ is implemented by

$$\hat{S}_e = \hat{B}\hat{Z}_{1\ldots k}\hat{B} \tag{7}$$

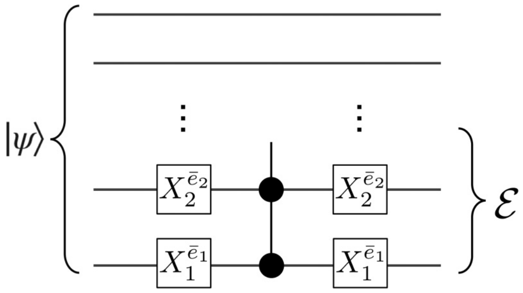

FIG. 2. Quantum circuit for implementing the phase flip operator $\hat{S}_e$. The $k = |\mathcal{E}|$ qubit controlled phase operator acts on the evidence qubits $\mathcal{E}$. It may be compiled into $O(k)$ CNOTs and single-qubit operators given $O(k)$ ancillas [16]. The evidence values $e = e_k\ldots e_2e_1$ control the bit flips through $\bar{e}_i \equiv 1 - e_i$.

where $\hat{B} = \prod_{i=1}^{k}\hat{X}_i^{\bar{e}_i}$ with $\bar{e}_i \equiv 1 - e_i$. Explicitly,

$$\begin{aligned}\hat{S}_e|\psi_{\mathcal{B}}\rangle &= \hat{B}\hat{Z}_{1\ldots k}\hat{B}[\sqrt{P(e)}|\mathcal{Q}\rangle|e\rangle + \sqrt{1-P(e)}|\overline{\mathcal{Q},e}\rangle]\\ &= \hat{B}\hat{Z}_{1\ldots k}[\sqrt{P(e)}|\mathcal{Q}\rangle|1^n\rangle + \sqrt{1-P(e)}|\overline{\mathcal{Q},1^n}\rangle]\\ &= \hat{B}[-\sqrt{P(e)}|\mathcal{Q}\rangle|1^n\rangle + \sqrt{1-P(e)}|\overline{\mathcal{Q},1^n}\rangle]\\ &= [-\sqrt{P(e)}|\mathcal{Q}\rangle|e\rangle + \sqrt{1-P(e)}|\overline{\mathcal{Q},e}\rangle].\end{aligned} \tag{8}$$

The circuit diagram representing $\hat{S}_e$ is shown in Fig. 2. The $k$-qubit controlled phase can be constructed from $O(k)$ CNOTs and single-qubit operators using $O(k)$ ancillas [16] or, alternatively, $O(k^2)$ CNOTs and single-qubit operators using no ancillas [31].

### D. Time complexity

The circuit complexities of the various elements in the Grover iterate $\hat{G} = -\hat{A}\hat{S}_0\hat{A}^\dagger\hat{S}_e$ are presented in Table I. As the circuit complexity of the phase flip operator $S_0$ ($S_e$) scales linearly with number of qubits $n$ ($|\mathcal{E}|$), $Q_{\hat{G}}$ is dominated by the that of the state preparation operator $\hat{A}$. Although $Q_{\hat{A}_P}$ scales exponentially with the number of nodes $n$ for general q-sample preparation, Bayesian q-sample preparation on a network of bounded indegree $m$ is efficient. Namely, $Q_{\hat{A}_{\mathcal{B}}} = O(n2^m)$ scales linearly with $n$ as in classical sampling from a Bayesian network. It takes $O(P(e)^{-1/2})$ applications of $\hat{A}_{\mathcal{B}}$ to perform the rejection sampling algorithm from Sec. V and, thus, a single sample from $P(\mathcal{Q}|\mathcal{E})$ can be obtained by a

TABLE I. Circuit complexity $Q_{\hat{U}}$ of implementing the operators $\hat{U}$ discussed in the text. The Grover iterate $\hat{G}$ for amplitude amplification of a Bayesian q-sample (general q-sample) consists of two instances of the preparation circuit $\hat{A}_{\mathcal{B}}$ ($\hat{A}_P$) and one instance each of $\hat{S}_0$ and $\hat{S}_e$. The time to collect one sample from $P(\mathcal{Q}|\mathcal{E}=e)$ is $O(Q_{\hat{G}}P(e)^{-1/2})$.

| $\hat{U}$ | $Q_{\hat{U}}$ | Comments |
|---|---|---|
| $\hat{A}_P$ | $O(2^n)$ | Q-sample preparation |
| $\hat{A}_{\mathcal{B}}$ | $O(n2^m)$ | Bayesian state preparation |
| $\hat{S}_0$ | $O(n)$ | $O(n)$ ancilla qubits |
| $\hat{S}_e$ | $O(\lvert\mathcal{E}\rvert)$ | $O(\lvert\mathcal{E}\rvert)$ ancilla qubits |

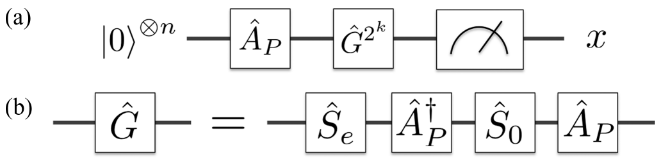

FIG. 3. (a) Quantum Bayesian inference on Bayes net $\mathcal{B}$ for evidence $\mathcal{E} = e$ is done by repetition of the circuit shown, with $k$ incrementing $k = 0, 1, \dots,$ stopping when the measurement result $x$ contains evidence bits $e$. Then $x$ can be recorded as a sample from the conditional distribution $P(\mathcal{Q}|\mathcal{E})$. This corresponds to Algorithm 1. (b) The constituents of the Grover iterate $\hat{G}$, the state preparation $\hat{A}_{\mathcal{B}}$, and phase flip operators $\hat{S}_e$ and $\hat{S}_0$. The state preparation operator is constructed from Theorem 2, and an example is shown in Fig. 1(b). The phase flip operators are constructed as shown in Fig. 2.

quantum computer in time $O(n2^m P(e)^{-1/2})$. In Sec. II, we saw that classical Bayesian inference takes time $O(nmP(e)^{-1})$ to generate a single sample. Thus, quantum inference on a Bayesian network provides a square-root speedup over the classical case. The quantum circuit diagram for Bayesian inference is outlined in Fig. 3.

## VII. CONCLUSION

We have shown how the structure of a Bayesian network allows for a square-root, quantum speedup in approximate inference. We explicitly constructed a quantum circuit from CNOT and single-qubit rotations that returns a sample from $P(\mathcal{Q}|\mathcal{E} = e)$ using just $O(n2^m P(e)^{-\frac{1}{2}})$ gates. For more general probability distributions, the Grover iterate would include a quantity of gates exponential in $n$, the number of random variables, and thus not be efficient. This efficiency of our algorithm also implies experimental possibilities. As a proof of principle, one could experimentally perform inference on a two-node Bayesian network with only two qubits with current capabilities of ion trap qubits [32].

We also placed the idea of a q-sample into the broader context of an analogy between quantum states and classical probability distributions. If a qpdf can be found that is pure, can be q-sampled, and allows q-stochastic updates, the quantum machine learning subfield would greatly benefit. Algorithms for many important routines, such as Metropolis-Hastings, Gibbs sampling, and even Bayesian learning, could find square-root speedups in a similar manner to our results here.

Artificial intelligence and machine learning tasks are often at least NP-hard. Although exponential speedups on such problems are precluded by the BBBV result [33], one might hope for square-root speedups, as we have found here, for a variety of tasks. For instance, a common machine learning environment is online or interactive, in which the agent must learn while making decisions. Good algorithms in this case must balance exploration, finding new knowledge, with exploitation, making the best of what is already known. The use of Grover's algorithm in reinforcement learning has been explored [34], but much remains to be investigated. One complication is that machine learning often takes place in a classical world; a robot is not usually allowed to execute a superposition of actions. One might instead focus on learning tasks that take place in a purely quantum setting. For instance, quantum error-correcting codes implicitly gather information on what error occurred in order to correct it. Feeding this information back into the circuit could create an adaptive, intelligent error-correcting code [35].

## ACKNOWLEDGMENT

We are grateful for the support of ARO Project W911NF1210486, the NSF IGERT program, and the NSF CUA grant.

[1] S. J. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 3rd ed. (Pearson Education, Upper Saddle River, NJ, 2010).

[2] M. Bensi, A. D. Kiureghian, and D. Straub, Reliab. Eng. Syst. Safety **112**, 200 (2013).

[3] R. E. Neapolitan *et al.*, *Learning Bayesian Networks* (Prentice Hall, Upper Saddle River, NJ, 2004), Vol. 1.

[4] G. F. Cooper and E. Herskovits, Mach. Learn. **9**, 309 (1992).

[5] N. Friedman, M. Goldszmidt, and A. Wyner, in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* (Morgan Kaufmann, San Francisco, CA, 1999), pp. 196–205.

[6] T. D. Nielsen and F. V. Jensen, *Bayesian Networks and Decision Graphs* (Springer, Berlin, Heidelberg, 2009).

[7] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, J. Chem. Phys. **21**, 1087 (1953).

[8] S. Chib and E. Greenberg, Am. Statistician **49**, 327 (1995).

[9] P. Dagum and M. Luby, Artif. Intell. **60**, 141 (1993).

[10] V. K. Mansinghka, Ph.D. thesis, Massachusetts Institute of Technology, 2009.

[11] E. Bernstein and U. Vazirani, in *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, 1993), pp. 11–20.

[12] S. Aaronson, in *Proceedings of the Forty-Second ACM Symposium on Theory of Computing* (ACM, New York, NY, 2010), pp. 141–150.

[13] P. W. Shor, SIAM J. Comput. **26**, 1484 (1997).

[14] L. K. Grover, in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, 1996), pp. 212–219.

[15] A. Galindo and M. A. Martin-Delgado, Rev. Mod. Phys. **74**, 347 (2002).

[16] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information* (Cambridge University Press, Cambridge, UK, 2000).

[17] S. Jordan, http://math.nist.gov/quantum/zoo/

[18] M. Ozols, M. Roetteler, and J. Roland, ACM Trans. Comput. Theory **5**, 11 (2013).

[19] L. K. Grover, in *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, 2000), pp. 618–626.

[20] A. D. Bookatz, Quantum Inf. Comput. **14**, 5 (2014).

[21] G. Brassard, P. Hoyer, M. Mosca, and A. Tapp, arXiv:quant-ph/0005055.

[22] K. Temme, T. Osborne, K. Vollbrecht, D. Poulin, and F. Verstraete, Nature (London) **471**, 87 (2011).

[23] D. Aharonov and A. Ta-Shma, in *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing* (ACM, New York, NY, 2003), pp. 20–29.

[24] T. Lee, R. Mittal, B. W. Reichardt, R. Spalek, and M. Szegedy, in *Foundations of Computer Science (FOCS)* (IEEE, Piscataway, NJ, 2011), pp. 344–353.

[25] P. Kaye and M. Mosca, in *Proceedings of the International Conference on Quantum Information* (Optical Society of America, Washington, DC, 2002).

[26] C. Zalka, Proc. R. Soc. London A **454**, 313 (1998).

[27] L. Grover and T. Rudolph, arXiv:quant-ph/0208112.

[28] V. Bergholm, J. J. Vartiainen, M. Möttönen, and M. M. Salomaa, Phys. Rev. A **71**, 052330 (2005).

[29] D. M. Chickering, D. Heckerman, and C. Meek, J. Mach. Learn. Res. **5**, 1287 (2004).

[30] M. J. Druzdzel and H. A. Simon, in *Proceedings of the Ninth International Conference on Uncertainty in Artificial Intelligence* (Morgan Kaufmann, Upper Saddle River, NJ, 1993), pp. 3–11.

[31] M. Saeedi and M. Pedram, Phys. Rev. A **87**, 062318 (2013).

[32] D. Hanneke, J. Home, J. Jost, J. Amini, D. Leibfried, and D. Wineland, Nat. Phys. **6**, 13 (2010).

[33] C. H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani, SIAM J. Comput. **26**, 1510 (1997).

[34] D. Dong, C. Chen, H. Li, and T. Tarn, IEEE Trans. Syst., Man, Cybern., B **38**, 1207 (2008).

[35] J. Combes, H. Briegel, C. Caves, C. Cesare, C. Ferrie, G. Milburn, and M. Tiersch, in *APS March Meeting* (American Physical Society, College Park, MD, 2014).