Electronic Colloquium on Computational Complexity, Report No. 154 (2013)

Simulating Quantum Circuits with Sparse Output Distributions

Martin Schwarz*[1] and Maarten Van den Nest†[2]

[1]Vienna Center for Quantum Science and Technology, Faculty of Physics, University of Vienna, Austria
[2]Max Planck Institut für Quantenoptik, Hans-Kopfermann-Str. 1, D-85748 Garching, Germany

Abstract

We show that several quantum circuit families can be simulated efficiently classically if it is promised that their output distribution is approximately sparse i.e. the distribution is close to one where only a polynomially small, a priori unknown subset of the measurement probabilities are nonzero. Classical simulations are thereby obtained for quantum circuits which—without the additional sparsity promise—are considered hard to simulate. Our results apply in particular to a family of Fourier sampling circuits (which have structural similarities to Shor's factoring algorithm) but also to several other circuit families, such as IQP circuits. Our results provide examples of quantum circuits that cannot achieve exponential speed-ups due to the presence of too much destructive interference i.e. too many cancelations of amplitudes. The crux of our classical simulation is an efficient algorithm for approximating the significant Fourier coefficients of a class of states called computationally tractable states. The latter result may have applications beyond the scope of this work. In the proof we employ and extend sparse approximation techniques, in particular the Kushilevitz-Mansour algorithm, in combination with probabilistic simulation methods for quantum circuits.

1 Introduction

In this paper we present classical algorithms for the simulation of several related classes of quantum circuits containing blocks of Quantum Fourier Transforms (QFTs). In particular, we consider n-qubit circuits with a QFT-Toffoli-QFT^{-1} block structure followed by a (partial) measurement immediately after the final QFT. Circuits of this kind are used in various quantum algorithms, most notably Shor's factoring algorithm. Whereas the circuits considered in this paper are unlikely to have an efficient classical simulation in general, the aim of this work is to analyze under which additional conditions an efficient classical simulation becomes possible. This provides an approach to identify features which are essential in the (believed) superpolynomial speed-ups achieved by, say, the factoring algorithm. In this paper we will in particular place restrictions on the *output distribution* of the circuit. In short, our results are as follows: given the promise that the output distribution is *approximately sparse* (or "*peaked*")—in the sense that only $O(\text{poly}(n))$ of the $O(2^n)$ probabilities have significant magnitude of $\Omega(1/\text{poly}(n))$—then an efficient classical simulation algorithm is provided. Not unexpectedly, Shor's algorithm does not satisfy such sparseness promise i.e. its output distribution is "superpolynomially flat". Our results thus imply that the approximate sparseness promise alone suffices to bring down the (believed) superpolynomial speed-up achieved by the factoring algorithm to the realm of a classically simulatable quantum computation. Below we provide a discussion of how our findings shed light on the factoring algorithm (see Section 2).

The implications of our results are twofold. First, they pose restrictions on the design of fast quantum algorithms. For example, our results show that any *exact* quantum algorithm adopting the QFT-Toffoli-QFT^{-1} block structure (or more generally the structures considered in Theorems 1-4) which has as its output state a single computational

*m.schwarz@univie.ac.at
†maarten.vandennest@mpq.mpg.de

ISSN 1433-8092

basis state containing the answer of the problem, can never achieve an exponential quantum speed-up. Given the generality of the class of circuits considered, we believe that these classical simulation results may provide useful insights for the quantum algorithms community. Second, the present results have conceptual implications as follows: the exponential speed-up found in quantum algorithms is often related to the availability of interference of probability amplitudes in this model. Indeed in several quantum algorithms, first a superposition of states is created using a QFT, then amplitudes are manipulated in some nontrivial way using reversible (classical) gates, such that in a final QFT, by means of interference, only desired basis states survive whereas the amplitudes for undesired states cancel out. Our results imply that this qualitative picture has to be refined, since too much cancelation leading to only a few classical output states (let alone a single one!) can in fact be simulated efficiently classically, and thus cannot offer exponential speed-up. Indeed, our results imply that the final probability distribution must *necessarily have super-polynomially large support* (e.g. in the same order as the full state space), in order to allow for exponential speed-up. Finally, since only polynomially many measurements can be performed efficiently on the output state—and thus only a small fraction of the necessarily large number of states can be sampled—the output distribution must have a special structure such that meaningful information can be recovered from just a few measurements. Notably, the coset state produced by Shor's algorithm (and its generalizations) has group structure which is indeed exploited in the classical post-processing step to recover the entire state space from just a few measurements (cf. Section 2).

The proof techniques we use to obtain our results are twofold. First, we use randomized classical simulation methods for Computationally Tractable (CT) states as developed in [VdN11]. Furthermore the latter methods are combined with algorithms for sublinear sparse Fourier transforms (SFTs), which have been pioneered in seminal work by Goldreich-Levin [GL89] and Kushilevitz-Mansour [KM91] and which have been refined throughout the last two decades [Man95, GGI+02, AGS03, GMS05, AGGM06, Iwe10, Aka10, HIKP12b, HIKP12a]. Our work also provides further extensions of the above sparse approximation techniques.

Whereas to our knowledge this is the first paper which analyzes the effect of (approximate) sparseness of the output distribution on the classical simulability of quantum circuits, from a more general point of view several works are related to the present paper (e.g. in terms of the class of quantum circuits considered or in terms of the techniques used). For example, a relevant series of papers regards [YS07, ALM06, Bro07], that all focus on efficient classical simulation of the QFT with the aim of understanding better the workings of Shor's factoring algorithm. In the latter context, see also [VdN12, BVN12] for classical simulations of a class of circuits involving QFTs over finite abelian groups supplemented with a particular family of group-theoretic operations (Normalizer circuits). Classical simulation of CT states were considered in [VdN11] by one of us. In the latter work, the algorithms from Goldreich-Levin [GL89] and Kushilevitz-Mansour [KM91] were applied in the context of classical simulation, albeit in a rather different context compared to the present paper, namely to analyze the role of the classical postprocessing for quantum speed-ups (more particularly in Simon's algorithm). Further work on CT states is done in [Sta13]; the latter work also analyzes the role of interference effects in quantum speed-ups (although from a different perspective then the present paper). Below we will also make statements about classical simulability of IQP (Instanteneous Quantum Polynomial-time) circuits. In [BJS11] it was shown (roughly speaking) that general IQP circuits cannot be simulated efficiently, unless the polynomial hierarchy collapses. In contrast, here we show that IQP circuits with an additional sparseness promise on the output distribution, are efficiently simulable classically. Finally, in [MO10] the authors consider and generalize prior work on SFTs in a different direction i.e. unrelated to classical simulation issues; they prove a quantum Goldreich-Levin theorem and use it for efficient quantum state tomography for quantum states that are approximately sparse in the Pauli product operator basis.

2 Main results: statements and discussion

We prove four theorems, all similar in spirit, about efficient classical simulability of classes of quantum circuits with a promise on the (approximate) sparseness of the output distributions and/or the output states. We call a probability distribution over 2^n events *t-sparse*, if only t probabilities are nonzero, and *ε-approximately t-sparse* if the probability distribution is ε-close in ℓ_1-distance to a t-sparse one. Throughout this paper we will work with qubit systems and sometimes indicate where generalizations of definitions and results to d-level systems are possible. The computational basis states of an n-qubit system are denoted by $|x\rangle$ where $x = x_1 \cdots x_n$ is an bit string. The set of n-bit strings will be denoted by B_n.

A key concept we build upon in this work are *computationally tractable* states introduced in [VdN11], which capture two key properties of simulable quantum states:

Definition 1 (Computationally Tractable (CT) states)**.** *An n-qubit state $|\psi\rangle$ is called 'computationally tractable' (CT) if the following conditions hold:*

1. *it is possible to sample in poly(n) time with classical means from the probability distribution $\mathcal{P} = \{p_x : x \in B_n\}$ defined by $p_x = |\langle x|\psi\rangle|^2$, and*
2. *upon input of any bit string x, the coefficient $\langle x|\psi\rangle$ can be computed in poly(n) time on a classical computer.*

The definition of CT states is straightforwardly generalized to states of systems of qudits. Several important state families are CT: matrix product states with polynomial bond dimension, states generated by poly-size Clifford circuits, states generated by poly-size nearest-neighbor matchgate circuits, states generated by bounded tree-width circuits (where all aforementioned circuits act on standard basis inputs). For definitions of these classes and proofs that they are CT states, we refer to [VdN11]. Further examples of CT states are states generated by normalizer circuits over finite Abelian groups (acting on coset states) [VdN12, BVN12].

Example 1. *For our purposes it will be especially useful to point out that the following classes of states are CT [VdN11].*

(i) Let $|x\rangle$ be an arbitrary n-qubit computational basis state, let $\mathcal{F}$ denote the quantum Fourier transform over $\mathbb{Z}_{2^k}$ for some $k \leq n$ (acting on any subset of k qubits) and let $\mathcal{T}$ be a poly-size circuit of classical reversible gates (e.g. Toffoli gates), then the state $\mathcal{T}\mathcal{F}|x\rangle$ is CT.

(ii) Let $f : B_n \to \{1, -1\}$ be a classically efficiently computable function, then the state $|\psi_f\rangle = \frac{1}{\sqrt{2^n}} \sum f(x)|x\rangle$, where the sum is over all n-bit strings x, is CT.

One may also consider a notion of CT states in the presence of oracles (see also [BH13]). We say that an n-qubit state $|\psi\rangle$ is f-CT given access to an oracle $f : \{0,1\}^m \to \{0,1\}$ (with $m = \text{poly}(n)$) if conditions (a)-(b) in Definition 1 hold when allowing, instead of poly-time classical computations, poly-many queries to the oracle. For example, if the function f in (ii) is given as an oracle, the state $|\psi_f\rangle$ in Example 1 is trivially f-CT.

Based on these definitions, we are now ready to state our main results.

Sparse output distributions

Theorem 1. *Consider a unitary n-qubit quantum circuit composed of two blocks $\mathcal{C} = U_2U_1$ with input state $|\psi_{in}\rangle$. Suppose that the following conditions are fulfilled:*

(a) the state $U_1|\psi_{in}\rangle$ obtained after applying the first block is CT;

(b) the second block U_2 is a QFT (or QFT^{-1}) modulo 2^k, for some $k \leq n$, applied to any subset S of k qubits. The circuit is followed by a measurement of the qubits in S in the computational basis, giving rise to a probability distribution $\mathcal{P}$.

(c) The distribution $\mathcal{P}$ is promised to be ε-approximately t-sparse for some $\varepsilon \leq 1/6$ and for some t (and otherwise no information about $\mathcal{P}$ is available).

Then there exists a randomized classical algorithm with runtime poly$(n, t, 1/\varepsilon, \log \frac{1}{\delta})$ which outputs (by means of listing all nonzero probabilities) an s-sparse probability distribution $\mathcal{P}'$ where $s = O(t/\varepsilon)$; with probability at least $1 - \delta$, the distribution $\mathcal{P}'$ is $O(\varepsilon)$-close to $\mathcal{P}$. Furthermore, it is possible to sample $\mathcal{P}'$ on a classical computer in poly$(n, t, 1/\varepsilon)$ time.

Thus, if the sparseness t is at most polynomially large in n, if the error ε is at worst polynomially small in n, and if $\delta = 2^{-\text{poly(n)}}$, then the classical simulation is efficient i.e. it runs in poly(n) time, and the probability of failure is exponentially small.

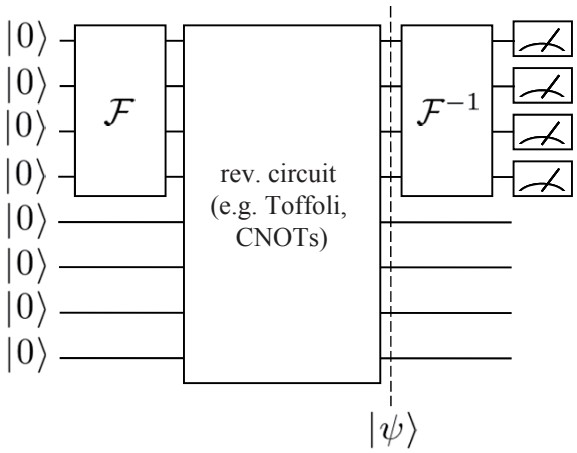

Figure 1: *Shor's algorithm [Sho99] consists of (1) a quantum Fourier transform (QFT) on a subset of qubits, (2) a block of reversible gates (a modular exponentiation circuit), and (3) an inverse QFT on the same subset qubits. Note that the state $|\psi\rangle$ obtained after the first QFT is a computationally tractable (CT) state. Thus conditions (a) and (b) of Theorem 1 are satisfied. However the output distribution of Shor's algorithm is* not *sparse in general, as required by our algorithm (cf. condition (c)).*

We emphasize that, apart from the promise (c), no information about the structure of $\mathcal{P}$ is a priori available. For example, suppose that $\mathcal{P}$ is promised to be approximately 1-sparse, where a distribution is 1-sparse if there exists a single bit string x^* which occurs with probability 1 and all other bit strings have probability 0. Then, crucially, we do not assume knowledge of the bit string x^*, i.e a priori all (potentially exponentially many in n!) bit strings are equally likely. Perhaps surprisingly, Theorem 1 implies that a good approximation of $\mathcal{P}$ can nevertheless be efficiently computed.

Since several circuit families satisfy condition (a) (recall examples above and see [VdN11]), Theorem 1 yields an efficient classical simulation of various types of circuits. For example, letting $|\psi_{\text{in}}\rangle$ be an arbitrary computational basis input, the block U_1 may be e.g. any poly-size Clifford circuit, nearest-neighbor matchgate circuit or bounded-treewidth circuit. A particularly interesting class of circuits, denoted by $\mathcal{A}_{\text{Shor}}$, is depicted in Figure 1. Note that Shor's factoring algorithm belongs to the class $\mathcal{A}_{\text{Shor}}$. It is easily verified that, for any $\mathcal{A}_{\text{Shor}}$ circuit, the state of the quantum register immediately before the second QFT is CT (recall Example 1 (i) from above). Thus any circuit in $\mathcal{A}_{\text{Shor}}$ *which, in addition, satisfies the sparseness condition (c) of Theorem 1* can be simulated efficiently classically. Upon closer inspection of Shor's factoring algorithm, one finds that its output distribution $\mathcal{P}_{\text{Shor}}$ generally contains super-polynomially many nonzero probabilities and thus (non-surprisingly) Theorem 1 does not yield an efficient classical simulation of the factoring algorithm. More precisely, the size of the support of the flat distribution $\mathcal{P}_{\text{Shor}}$ equals the multiplicative order r of a randomly chosen integer x modulo N. For a general integer N, the order is conjectured to be $\Omega(N/\log(N))$ on average over all N [Arn05, KP13]. In the case of RSA, with $N = pq$, the primes p and q might be *chosen* such that w.h.p. $r \approx N/4$ [Sho11]. Nevertheless it is interesting that the mere promise of (approximate) sparsity of the output distribution suffices to arrive at an efficient classical simulation for all $\mathcal{A}_{\text{Shor}}$ circuits, without otherwise restricting the allowed operations. This implies that the feature that $\mathcal{P}_{\text{Shor}}$ is sufficiently flat is an essential ingredient in the (believed) superpolynomial speed-up achieved by Shor's factoring algorithm.

Another observation is the following. Any quantum circuit $\mathcal{A}$ satisfying (a)-(b) in Theorem 1 (for example any $\mathcal{A}_{\text{Shor}}$ circuit) which, when implemented on a quantum computer, aspires to deliver a superpolynomial speed-up over classical computers, must generate a distribution $\mathcal{P}$ which cannot be well-approximated by a poly(n)-sparse distribution. At the same time, at most poly(n) repetitions of $\mathcal{A}$ are allowed if the total computational cost is to be polynomially bounded, yielding only poly(n) samples of $\mathcal{P}$. In other words, one only has access to 'few' samples of a distribution which has support on a 'large' number of outputs. Yet somehow these few samples should contain sufficient infor-

mation to extract the final result of the computation with high probability (working within the standard bounded-error setting). This point is nicely illustrated by considering again the factoring algorithm (or more generally the abelian hidden subgroup algorithm). Here the output distribution is (close to) the uniform distribution over an unknown *group* H (and determining this group is essentially the goal of the algorithm) and the final measurement only yields a small set of $O(\log|H|)$ randomly chosen elements of H. However, since such a small set of randomly generated group elements is with high probability a generating set of the group, a small number of measurements indeed suffices to determine the entire group H.

Theorem 1 can be extended by allowing the block U_2 to comprise tensor product operations, instead of the QFT:

Theorem 2. *The conclusions of Theorem 1 also apply if condition (b) is replaced by*

(b') the second block U_2 is an arbitrary tensor product unitary operation $U_2 = u_1 \otimes \cdots \otimes u_n$. The circuit is followed by a measurement of an arbitrary subset of qubits S in the computational basis, giving rise to a probability distribution $\mathcal{P}$.

In addition, the conclusions of Theorem 1 also apply when U_2 is a tensor product operation as in (b'), but now for quantum algorithms operating on the Hilbert space $\mathcal{H} = \mathbb{C}_{d_1} \otimes \cdots \otimes \mathbb{C}_{d_n}$ with $d_i = O(1)$ but otherwise arbitrary, i.e. $\mathcal{H}$ is a system of n qudits of possibly different dimensions.

A first example of the setting considered in Theorem 2 regards the family of IQP circuits (Instantaneous Quantum Polynomial time [SB09]). Here the input is an n-qubit computational basis state $|x\rangle$ and the circuit consists of gates of the form $\exp[i\theta T]$ where θ is an arbitrary real parameter and where T is a tensor product of the form $T = T_1 \otimes \cdots \otimes T_n$ with $T_i \in \{I, X\}$. Since $X = HZH$, every IQP circuit $\mathcal{C}$ can be written as $\mathcal{C} = H^{\otimes n}\mathcal{C}'H^{\otimes n}$ where $\mathcal{C}'$ is obtained by replacing each gate $\exp[i\theta T]$ by $\exp[i\theta T']$ where $T' = T'_1 \otimes \cdots \otimes T'_n$ with $T'_i = HT_iH$. Thus T' is a tensor product of Z operators and identity gates and hence each gate $e^{i\theta T'}$ is diagonal in the computational basis. Setting $U_1 := \mathcal{C}'H^{\otimes n}$ and $U_2 := H^{\otimes n}$ we find that conditions (a)-(b') of Theorem 2 are fulfilled; indeed it is straightforward to show that $\mathcal{C}'H^{\otimes n}|x\rangle$ is a CT state. Thus Theorem 2 shows that any IQP circuit with an approximately sparse output distribution can be simulated efficiently classically. This result is particularly interesting when compared to a hardness-of-simulation result obtained for general IQP circuits (i.e. without sparseness promise) in [BJS11]. In the latter work it was shown that an efficient, approximate classical simulation of IQP circuits (w.r.t. a certain multiplicative approximation) would imply a collapse of the polynomial hierarchy.

A second example of the setting considered in Theorem 2 is the following. Consider a finite, possibly non-abelian group G given as a direct product of n individual groups, $G = G_1 \times \cdots \times G_n$ where the order of each G_i is $O(1)$. Define a Hilbert space $\mathcal{H}_G$ with computational basis vectors $|g\rangle = |g_1\rangle \otimes \cdots \otimes |g_n\rangle$ labeled by group elements $g = (g_1, \ldots, g_n) \in G$. The space $\mathcal{H}_G$ is naturally associated with a tensor product of n individual spaces, each of constant dimension. We may now consider quantum circuits of the following kind. The total Hilbert space is $\mathcal{H}_G \otimes \mathcal{H}_n$ where $\mathcal{H}_n$ is an n-qubit system. In analogy to Figure 1, we consider circuits of the block structure $\mathcal{C} = A_3A_2A_1$ where A_1 is the QFT over G acting on the register $\mathcal{H}_G$, A_2 is an arbitrary poly-size circuit of classical reversible gates acting on the entire system and A_3 is the inverse QFT over G. The input is $|1_G, 0_n\rangle$ where 1_G is the neutral element in G and 0_n denotes the all-zeros n-bit string; the circuit is followed by measurement of the system $\mathcal{H}_G$ in the basis $\{|g\rangle\}$. Circuits of this kind are are of interest in the context of quantum algorithms for the (non-abelian) Hidden subgroup problem (see e.g. [AMR07, Lom04]). For a definition of the QFT over a finite group we refer to e.g. [MRR06]; here it suffices to mention that the QFT over a product group $G = G_1 \times \cdots \times G_n$ is a tensor product operator. Furthermore it is easily verified (recall also the discussion on CT states above) that condition (a) in Theorem 1 is satisfied with $U_1 \equiv A_2A_1$. Thus Theorem 2 implies that any quantum circuit of this kind which has an approximately sparse output distribution can be simulated classically. This gives an example of a quantum circuit family comprising non-abelian QFTs (albeit of a restricted kind) which can be simulated classically. For other examples of simulations of non-Abelian QFTs we refer to [BV11].

Sparse output states

Let us present two more results regarding quantum circuits of the kinds considered in Theorem 1 and Theorem 2, when promised that the output *state* is approximately sparse. In this case we show how an approximation of the latter output state can be efficiently determined by means of a classical randomized algorithm.

An n-qubit state $|\varphi\rangle$ is called ε-approximately t-sparsee if there exists a state $|\varphi'\rangle$ which is ε-close to $|\psi\rangle$ and for which at most t amplitudes $\langle x|\varphi'\rangle$ (with $|x\rangle$ computational basis states) are nonzero (see also section 4).

Theorem 3. *Consider a unitary n-qubit quantum circuit composed of two blocks $\mathcal{C} = U_2U_1$ with input state $|\psi_{in}\rangle$. Suppose that the following conditions are fulfilled:*

(a) the state $U_1|\psi_{in}\rangle$ obtained after applying the first block is CT;

(b) the second block U_2 is the QFT modulo 2^n or its inverse.

(c) The final state $|\psi_{out}\rangle = \mathcal{C}|\psi_{in}\rangle$ is promised to be $\sqrt{\varepsilon}$-approximately t-sparse for some $\varepsilon \leq 1/6$ and some t.

Then there exists a randomized classical algorithm with runtime poly$(n, t, 1/\varepsilon, \log\frac{1}{\delta})$ which outputs (by means of listing all nonzero amplitudes) an s-sparse state $|\psi\rangle$ which, with probability at least $1-\delta$, is $O(\sqrt{\varepsilon})$-close to $|\psi_{out}\rangle$, where $s = O(t/\varepsilon)$.

Theorem 4. *The conclusions of Theorem 3 also apply if condition (b) is replaced by*

(b') the second block U_2 is an arbitrary tensor product unitary operation $U_2 = u_1 \otimes \cdots \otimes u_n$.

In addition, the conclusions of Theorem 3 also apply when U_2 is a tensor product operation as in (b'), but now for quantum algorithms operating on the Hilbert space $\mathcal{H} = \mathbb{C}_{d_1} \otimes \cdots \otimes \mathbb{C}_{d_n}$ with $d_i = O(1)$ but otherwise arbitrary.

Theorem 3 and Theorem 4 are closely connected to an important result in theoretical computer science, namely the Kushilevitz-Mansour (KM) algorithm [KM91]: if one has oracle access to a Boolean function $f : B_n \to \{1, -1\}$ which is promised to have an approximately sparse Fourier spectrum, it is possible to compute a sparse approximation of f in polynomial time. We connect our result to Kushilevitz-Mansour by considering Theorem 4 for an n-qubit system where

$$|\psi_{\text{in}}\rangle \equiv |\psi_f\rangle = \frac{1}{2^{n/2}} \sum_x f(x)|x\rangle \tag{1}$$

is a CT state, $U_1 \equiv I$ and $U_2 \equiv H^{\otimes n}$ where H is the Hadamard gate. Then Theorem 4 implies that if $H^{\otimes n}|\psi_f\rangle$ is promised to be approximately sparse, then a sparse approximation of the latter state can be computed efficiently. This is effectively (a version of) the KM result, stated in the language of quantum computing. Similarly, Theorem 3 relates to a version of the KM result [Man95] considered for transformations of Boolean functions under the Fourier transform over $\mathbb{Z}_{2^n}$. The proof method of the KM theorem, suitably generalized to our setting at hand, will be an important tool for us.

Computing significant weights

Whereas Theorems 1 to 4 involve a promise about the approximate sparseness of the output distributions/states, our final result does not. The following theorem asserts that, for CT states expanded in the Fourier basis, it is possible to efficiently determine (in a suitable approximate and probabilistic sense) all Fourier coefficients which are larger than some threshold value; a similar result also holds for CT states expanded in product bases. The result is in the present paper mainly used as a technique in the proof of Theorems 3 and 4 (similar to the proof of Kushilevitz-Mansour). However we believe it may be of independent interest, given the broadness of the class of CT states and the frequent usage of Fourier transforms.

Let $\mathbb{Z}_{2^n}$ denote the cyclic group of integers modulo 2^n. Any n-bit string x is identified with an element of $\mathbb{Z}_{2^n}$ via the binary expansion. Recall that the quantum Fourier transform over $\mathbb{Z}_{2^n}$ is the following n-qubit unitary operator:

$$\mathcal{F}_{2^n} = \frac{1}{\sqrt{2^n}} \sum_{x,y \in \mathbb{Z}_{2^n}} \exp\left(\frac{2\pi i x y}{2^n}\right) |x\rangle\langle y|. \tag{2}$$

and the *Fourier basis* is simply the orthonormal basis $\{|F_x\rangle : x \in B_n\}$ defined by $|F_x\rangle = \mathcal{F}_{2^n}|x\rangle$.

Theorem 5. *Let $|\psi\rangle$ be an n-qubit CT state and consider its expansion in the Fourier basis:*

$$|\psi\rangle = \sum \hat{\psi}_x |F_x\rangle. \tag{3}$$

There exists a randomized classical algorithm with runtime poly$(n, \frac{1}{\theta}, \log \frac{1}{\pi})$ which outputs a list $L = \{x^1, \ldots, x^l\}$ where $l \leq 2/\theta$ and where each x^i is an n-bit string such that, with probability at least $1 - \pi$:

(a) for all $y \in L$, it holds that $|\hat{\psi}_x|^2 \geq \frac{\theta}{2}$;

(b) every k-bit string x satisfying $|\hat{\psi}_x|^2 \geq \theta$ belongs to the list L;

Furthermore, given any $x \in B_n$, there exists a classical algorithm with runtime poly$(n, 1/\varepsilon, \log \frac{1}{\delta})$ which, with probability at least $1 - \delta$, outputs an ε-approximation of $\hat{\psi}_x$. Finally, the above results also holds if the Fourier basis is replaced by a product basis $\{U|x\rangle\}$ where $U = U_1 \otimes \cdots \otimes U_n$ is an arbitrary tensor product unitary operator.

3 Proof outline and organization of the paper

In Section 4 we discuss ε-approximately t-sparse distributions and states. A key property will be Lemma 7 where we show that the large probabilities contain most of the information of an approximately sparse distribution i.e. discarding the small probabilities does not introduce too much error.

It will be a key point in our proofs that the output distributions of the quantum circuits considered in Theorems 1 to 4, as well as a suitable subset of their marginal distributions, are what will be called here *additively approximable*. The latter are distributions whose individual probabilities can be efficiently approximated with a randomized classical algorithm with a performance in terms of error and success probability which is similar to the one given by the Chernoff bound. Our analysis of additively approximable distributions (Section 5 and Section 6), which is a significant component in the proofs of our main results, will not make reference to quantum computing (the latter is done as of Section 7). In Section 5, we introduce the notion of additively approximable distributions and develop their properties. An important feature will be established in Theorem 10 where we show that, for any probability distribution which is itself additively approximable and for which a designated subset of its marginals are additively approximable as well, it is possible to efficiently determine (in a suitable approximate sense) those probabilities which are larger than some given, sufficiently large, threshold value. This lemma, in combination with Lemma 7 mentioned above, will yield an efficient algorithm to (approximately) sample any ε-approximately t-sparse distribution which is additively approximable and whose marginals are as well; this algorithm is given in Section 6 (Theorem 11). The results developed in Section 5 to Section 6 will follow the general proof idea of the Kushilevitz-Mansour theorem [KM91, GL89].

In Section 7 we recall classical simulation properties of CT states. Finally, in Section 8 the proofs of our main results are given: the main strategy is to show that the output distributions of the circuits considered in our main theorems, as well as their marginals, are additively approximable.

4 Approximate sparseness

4.1 Basic definitions

We call a quantum state $|\varphi\rangle$ *t-sparse* (relative to the computational basis), if at most t amplitudes $\langle x|\varphi\rangle$ are nonzero. We will use the standard ℓ_2-norm as as the natural distance measure for two pure states. Thus we will call two quantum states $|\varphi\rangle, |\psi\rangle$ *ε-close*, if $\||\varphi\rangle - |\psi\rangle\|_2 \leq \varepsilon$. We call a normalized pure state $|\varphi\rangle$ *ε-approximately t-sparse* if there exists a, not necessarily normalized, t-sparse vector which is ε-close to $|\varphi\rangle$. In this paper we will mostly be interested in a sparseness t which scales at most polynomially with the number of qubits n, and in an error ε which is worst polynomially small in n. Note that in the definition of approximate sparseness we allow the t-sparse vector to be an unnormalized state (this will be a convenient definition in our proofs). However, if $|\varphi\rangle$ is ε-approximately t-sparse and if ε is sufficiently small (namely $\varepsilon \leq 0.5$), there always exists a *normalized* t-sparse state $|\varphi'\rangle$ which is $O(\varepsilon)$-close to $|\varphi\rangle$ as well (see Section 4.2).

Similar to sparse quantum states, we call a probability distribution $\mathcal{P} = \{p_x : x \in B_n\}$ on the set of n-bit strings t-sparse if at most t of its probabilities p_x are nonzero. The distance between two probability distributions $\mathcal{P}$ and $\mathcal{P}'$ will be measured in terms of the total variation distance, defined by

$$\|\mathcal{P} - \mathcal{P}'\|_1 = \sum |p_x - p'_x|. \tag{4}$$

We say that $\mathcal{P}$ is ε-approximately t-sparse if there exists a t-sparse vector $v = (v_x : x \in B_n)$ such that $\sum |p_x - v_x| \leq \varepsilon$. The entries v_x may a priori be arbitrary complex numbers. However, similar to above, if $\mathcal{P}$ is ε-approximately t-sparse and if ε is sufficiently small, there always exists a *normalized probability distribution* $\mathcal{P}'$ which is t-sparse and such that $\|\mathcal{P} - \mathcal{P}'\|_1 \leq O(\varepsilon)$ (see Section 4.2).

The support of a probability distribution $\mathcal{P} = \{p_x : x \in B_n\}$ is the set of all x for which $p_x \neq 0$. If $A \subseteq B_n$, the restriction of $\mathcal{P}$ to A is the subnormalized distribution $\{q_x : x \in B_n\}$ defined by

$$q_x = \begin{cases} p_x & \text{if } x \in A \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

Similarly, the support of an n-qubit state is the set of all x for which $\langle x|\varphi\rangle \neq 0$. If $A \subseteq B_n$, the restriction of $|\varphi\rangle$ to A is the subnormalized state

$$\sum_{x \in A} \langle x|\varphi\rangle |x\rangle. \tag{6}$$

4.2 Basic properties

Let $\mathcal{P} = \{p_x : x \in B_n\}$ be an arbitrary probability distribution. Let $A_t \subseteq B_n$ be a subset which, roughly speaking, contains t bit strings corresponding to the t largest probabilities of $\mathcal{P}$. More formally, A_t satisfies the properties (i) $|A_t| = t$ and (ii) $p_x \geq p_y$ for all $x \in A_t$ and $y \notin A_t$. Note that there may be more than one set A_t with this property (e.g. if multiple probabilities happen to be equal). For our purposes the particular choice of A_t will however be irrelevant. Let $\mathcal{P}[t]$ denote the restriction of $\mathcal{P}$ to A_t. Note that $\mathcal{P}[t]$ is t-sparse. Furthermore it is straightforward to show that, for any t-sparse vector $v = (v_x : x \in B_n)$ (where the v_x may be arbitrary complex numbers), one has $\|\mathcal{P}[t] - \mathcal{P}\|_1 \leq \|v - \mathcal{P}\|_1$ i.e. $\mathcal{P}[t]$ has minimal distance to $\mathcal{P}$ among all such t-sparse v's. It follows that $\mathcal{P}$ is ε-approximately t-sparse iff

$$\|\mathcal{P} - \mathcal{P}[t]\|_1 \leq \varepsilon. \tag{7}$$

Next we show that, for any ε-approximately t-sparse distribution $\mathcal{P}$ with $\varepsilon \leq 0.5$ there always exists a t-sparse *normalized* distribution $\mathcal{P}'$ which is $O(\varepsilon)$-close to $\mathcal{P}$. To see this, set $\mathcal{P}' := \mathcal{P}[t]/\|\mathcal{P}[t]\|_1$. Owing to eq. (7) we have

$$1 - \varepsilon \leq \|\mathcal{P}[t]\|_1 \leq 1. \tag{8}$$

We then find

$$\begin{aligned} \|\mathcal{P}' - \mathcal{P}\|_1 &= \frac{\|\mathcal{P}[t] - \|\mathcal{P}[t]\|_1 \cdot \mathcal{P}\|_1}{\|\mathcal{P}[t]\|_1} \leq \frac{\|\mathcal{P}[t] - \|\mathcal{P}[t]\|_1 \cdot \mathcal{P}\|_1}{1 - \varepsilon} \\ &\leq \frac{\|\mathcal{P}[t] - \mathcal{P}\|_1}{1 - \varepsilon} + \frac{(1 - \|\mathcal{P}[t]\|_1) \cdot \|\mathcal{P}\|_1}{1 - \varepsilon} \leq \frac{2\varepsilon}{1 - \varepsilon}. \end{aligned} \tag{9}$$

Here in the equality we used the definition of $\mathcal{P}[t]$; in the first inequality we used eq. (8); in the second inequality we used the triangle inequality; finally we used eq. (7) and eq. (8). Then, if $\varepsilon \leq 0.5$, we have $\|\mathcal{P}' - \mathcal{P}\|_1 \leq 4\varepsilon$.

Let $|\varphi\rangle$ be an n-qubit state. In analogy with above, let $A_t \subseteq B_n$ be a subset which, roughly speaking, contains t bit strings corresponding to the t largest amplitudes of $|\varphi\rangle$. More formally, A_t satisfies (i) $|A_t| = t$ and (ii) $|\langle x|\varphi\rangle| \geq |\langle y|\varphi\rangle|$ for all $x \in A_t$ and $y \notin A_t$. Letting $|\varphi[t]\rangle$ denote the restriction of $|\varphi\rangle$ to A_t, it is straightforward to show that $|\varphi[t]\rangle$ has minimal ℓ_2-distance to $|\varphi\rangle$ among all t-sparse vectors. It follows that $|\varphi\rangle$ is ε-approximately t-sparse iff

$$\||\varphi\rangle - |\varphi[t]\rangle\|_2 \leq \varepsilon. \tag{10}$$

Fully analogous to above, for any ε-approximately t-sparse state $|\varphi\rangle$ with $\varepsilon \leq 0.5$ there always exists a t-sparse *normalized* state $|\varphi'\rangle$ which is $O(\varepsilon)$-close to $|\varphi\rangle$. The state $|\varphi'\rangle := |\varphi[t]\rangle / \||\varphi[t]\rangle\|_2$ does the job.

Let $|\varphi\rangle$ be an n-qubit pure state and let $\mathcal{P}$ be the probability distribution arising from measuring all qubits of $|\varphi\rangle$ in the computational basis. We may then ask whether $\mathcal{P}$ is approximate sparse or whether the full state $|\varphi\rangle$ is approximately sparse, where in the former case closeness is measured w.r.t. total variation distance and in the latter case it is measured w.r.t. ℓ_2 distance. Next we show that both notions of approximate sparseness are equivalent up to a square-root rescaling of the accuracy ε (which is mostly harmless if one is ultimately interested in $\varepsilon = 1/\mathrm{poly}(n)$, as we will mostly be in this paper).

Lemma 6. *Let $|\varphi\rangle$ be an n-qubit pure state and let $\mathcal{P}$ be the probability distribution arising from measuring all qubits of $|\varphi\rangle$ in the computational basis. Then $|\varphi\rangle$ is $\sqrt{\varepsilon}$-approximately t-sparse (relative to the ℓ_2-distance, as above) iff $\mathcal{P}$ is ε-approximately t-sparse (relative to the total variation distance, as above).*

Proof. Define $p_x = |\langle x|\varphi\rangle|^2$ for all x. As above, let A_t be a set of t n-bit string satisfying $p_x \geq p_y$ for all $x \in A_t$ and $y \notin A_t$. This is (trivially) equivalent to $|\langle x|\varphi\rangle| \geq |\langle y|\varphi\rangle|$ for all $x \in A_t$ and $y \notin A_t$. Let $\mathcal{P}[t]$ denote the restriction of $\mathcal{P}$ to A_t and similarly $|\varphi[t]\rangle$ is the restriction of $|\varphi\rangle$ to A_t. Recall that $|\varphi\rangle$ is $\sqrt{\varepsilon}$-approximately t-sparse iff $\||\varphi\rangle - |\varphi[t]\rangle\|_2 \leq \sqrt{\varepsilon}$ and that $\mathcal{P}$ is ε-approximately t-sparse iff $\|\mathcal{P} - \mathcal{P}[t]\|_1 \leq \varepsilon$. A straightforward application of definitions now shows that $\||\varphi\rangle - |\varphi[t]\rangle\|^2 = \|\mathcal{P} - \mathcal{P}[t]\|$, since both expressions coincide with

$$\sum_{x \notin A_t} |\langle x|\varphi\rangle|^2. \tag{11}$$

This shows that $\||\varphi\rangle - |\varphi[t]\rangle\|_2 \leq \sqrt{\varepsilon}$ iff $\|\mathcal{P} - \mathcal{P}[t]\|_1 \leq \varepsilon$. □

4.3 Sparse distributions have large coefficients

The next lemma shows that, for an approximately sparse probability distribution, the 'small' probabilities can be ignored without introducing much error. This property will be important in the proof of our main results, in combination with Theorem 10 which states that the large probabilities can be efficiently computed for certain distributions. The following lemma is also closely related to [KM91, Lemma 3.11]

Lemma 7. *Let $\mathcal{P} = \{p_x : x \in B_n\}$ be an ε-approximately t-sparse probability distribution. Define $B_{\varepsilon,t}$ to be the subset of all bit strings x such that $p_x \geq \varepsilon/t$. Define the subnormalized distribution $\mathcal{Q}_{\varepsilon,t}$ to be the restriction of $\mathcal{P}$ to $B_{\varepsilon,t}$. Then $\mathcal{Q}_{\varepsilon,t}$ is $O(\varepsilon)$-close to $\mathcal{P}$. More precisely $\|\mathcal{Q}_{\varepsilon,t} - \mathcal{P}\|_1 \leq 2\varepsilon$.*

Proof. Let $A_t \subseteq B_n$ and $\mathcal{P}[t]$ be defined as in Section 4.2. Recall that $\|\mathcal{P}[t] - \mathcal{P}\|_1 \leq \varepsilon$ owing to the approximate sparseness of $\mathcal{P}$. Furthermore construct $\mathcal{P}'$ as follows: start from $\mathcal{P}[t]$ and set all probabilities with magnitudes $\leq \frac{\varepsilon}{t}$ to zero; let C denote the support of $\mathcal{P}'$. Note that $C \subseteq A_t$ and thus $|A_t \setminus C| \leq t$. Furthermore $p_x \leq \varepsilon/t$ for every $x \in A_t \setminus C$. Then

$$\begin{aligned} \|\mathcal{P} - \mathcal{P}'\|_1 &\leq \|\mathcal{P} - \mathcal{P}[t]\|_1 + \|\mathcal{P}[t] - \mathcal{P}'\|_1 \\ &\leq \varepsilon + \sum_{x \in A_t \setminus C} p_x \leq \varepsilon + t \cdot \frac{\varepsilon}{t} = 2\varepsilon. \end{aligned} \tag{12}$$

Note also that $C \subseteq B_{\varepsilon,t}$ since $p_x \geq \varepsilon/t$ for all $x \in C$. Thus both $\mathcal{P}'$ and $\mathcal{Q}_{\varepsilon,t}$ are restrictions of $\mathcal{P}$, and that the support $B_{\varepsilon,t}$ of $\mathcal{Q}_{\varepsilon,t}$ contains the support C of $\mathcal{P}'$. This implies that $\|\mathcal{P} - \mathcal{Q}_{\varepsilon,t}\|_1 \leq \|\mathcal{P} - \mathcal{P}'\|_1$. Together with (12) this proves the result. □

An analogous result holds for approximately sparse quantum states. We do not make it explicit here since it will not be needed in our proofs of the main results.

5 Additively approximable probability distributions

5.1 Definition and basic properties

The Chernoff-Hoeffding bound is a basic tool in probability theory which will be used in this work. Whereas the bound is usually stated for real-valued random variables, here we state a simple generalization to the complex-valued case, which follows from the real-valued case by bounding real and imaginary parts of independently.

Lemma 8 (Chernoff-Hoeffding bound)**.** *Let $X_1, \ldots, X_T$ be i.i.d. complex-valued random variables with $E := \mathbf{E}X_i$ and $|X_i| \leq 1$ for every $i = 1, \ldots, T$. Then with $T = \frac{4}{\varepsilon^2}\log(\frac{4}{\delta})$ we have*

$$\Pr\left\{\left|\frac{1}{T}\sum_{i=1}^{T} X_i - E\right| \leq \varepsilon\right\} \geq 1 - \delta$$

A proof of Lemma 8 can be found in Appendix A. The main application of the Chernoff bound used in this work will be in the following context. Let $F : B_n \to \mathbb{C}$ be an efficiently computable complex function (i.e. computable in polynomial time on a deterministic classical computer) satisfying $|F(x)| \leq 1$ for all $x \in B_n$ and let $\mathcal{P} := \{p_x : x \in B_n\}$ be a probability distribution on the set of n-bit strings which can be sampled in poly(n) time on a randomized classical computer. Then a direct application of the Chernoff-Hoeffding bound shows that there exists a classical randomized algorithm to estimate

$$\langle F\rangle := \sum p_x F(x) \tag{13}$$

with error ε and probability at least $1 - \delta$ in poly$(n, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$ time. This means that in poly(n) time it is possible to achieve an accuracy $\varepsilon = 1/\text{poly}(n)$ and exponentially small failure probability $\delta = 2^{-\text{poly(n)}}$.

Next we introduce a definition for functions that are approximable with randomized classical algorithms having a performance in terms of error ε and failure probability δ that is analogous to those obtained by applying the Chernoff bound (see also [BFLW05] for a related notion of additive approximations).

Definition 2. *A function $f : B_n \to \mathbb{C}$ is said to be additively approximable if their exists a randomized classical algorithm with runtime poly$(n, 1/\varepsilon, \log\frac{1}{\delta})$ which, on input of an n-bit bit string x, outputs with probability at least $1 - \delta$ an ε-approximation of $f(x)$. A probability distribution $\mathcal{P} = \{p_x\}$ on the set of n-bit strings is said to be additively approximable if the function $x \to p_x$ is additively approximable.*

Note that any $\mathcal{P}$ which can be sampled classically in poly(n) time is additively approximable since each individual probability can essentially be computed by sampling the distribution. More precisely, to estimate p_x, write $p_x = \sum \delta(x, y)p_y$ where $\delta(x, y)$ equals 1 if $x = y$ and 0 otherwise. We have thus rewritten p_x as the expectation value of $F \equiv \delta(x, \cdot)$ which is a poly(n)-time computable function satisfying $|F(x)| \leq 1$ for all $x \in B_n$. The discussion above Definition 2 then immediately implies that $\mathcal{P}$ is additively approximable.

In the example discussed in eq. (13) we found that $\langle F\rangle$ can be efficiently approximated provided that F was efficiently computable on a deterministic computer. In the following lemma it is shown that the same performance in estimating $\langle F\rangle$ can be achieved even when F is only additively approximable. The argument is a basic application of the Chernoff bound.

Lemma 9. *Let $F : B_n \to \mathbb{C}$ be an additively approximable function and let $\mathcal{P} := \{p_x : x \in B_n\}$ be a probability distribution which can be sampled in poly(n) time on a classical computer. Then there exists a classical randomized algorithm to estimate $\langle F\rangle := \sum p_x F(x)$ with error ε and probability $1 - \delta$ in poly$(n, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$ time.*

Proof. By generating $K = O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ bit strings $x^1, \ldots, x^K$ from the distribution $\mathcal{P}$, the inequality

$$\left|\frac{1}{K}\sum_{i=1}^{K} F(x^i) - \langle F\rangle\right| \leq \varepsilon/2 \tag{14}$$

holds with probability at least $1 - \delta/2$, owing to the Chernoff bound. Then, for each x^i we compute a complex number c_i satisfying $|c_i - F(x^i)| \leq \frac{\varepsilon}{2}$ with probability at least $1 - \delta/(2K)$. Since F is additively approximable, each c_i can

be computed in time

$$T = \text{poly}(n, \tfrac{2}{\varepsilon}, \log \tfrac{2K}{\delta}) = \text{poly}(n, \tfrac{1}{\varepsilon}, \log \tfrac{1}{\delta}). \tag{15}$$

Thus the total runtime of computing all values c_i is $KT = \text{poly}(n, \frac{1}{\varepsilon}, \log \frac{1}{\delta})$. The total probability that each c_i is $\frac{\varepsilon}{2}$-close to $F(x^i)$ *and* that (14) holds is at least

$$(1 - \tfrac{\delta}{2}) \cdot (1 - \tfrac{\delta}{2K})^K \geq (1 - \tfrac{\delta}{2}) \cdot (1 - \tfrac{\delta}{2}) \geq 1 - \delta \tag{16}$$

where we have repeatedly used that $(1 - a)^r \geq 1 - ra$ for all positive integers r and for all $a \in [0, 1]$. It follows that, with probability at least $1 - \delta$, we have

$$|\frac{1}{K} \sum_{i=1}^{K} c_i - \langle F \rangle| \leq \varepsilon \tag{17}$$

by using the triangle inequality.

□

5.2 Estimating large coefficients

The following theorem contains the property of additive approximations which is most important for our purposes. It is a statement that, for distributions which are additively approximable and for which also (a designated subset of) the marginals are additively approximable, there exists an efficient algorithm to determine those probabilities which are larger than some given threshold value. The proof technique is a type of binary search algorithm which is a direct generalization of the proof of the Kushilevitz-Mansour algorithm [KM91].

Theorem 10. *Let $\mathcal{P} = \{p_x : x \in B_k\}$ be a probability distribution. Let $\mathcal{P}_m$ denote the marginal probability distribution of the first m bits, for every m ranging from 1 to k (with $\mathcal{P}_k \equiv \mathcal{P}$). Suppose that all distributions $\mathcal{P}_m$ are additively approximable. Then the following holds: given $\theta, \pi > 0$, there exists a randomized classical algorithm with runtime poly$(k, \frac{1}{\theta}, \log \frac{1}{\pi})$ which outputs a list $L = \{x^1, \ldots, x^l\}$ where $l \leq 2/\theta$ and where each x^i is an k-bit string such that, with probability at least $1 - \pi$:*

(a) for all $y \in L$, it holds that $p(y) \geq \frac{\theta}{2}$;

(b) every k-bit string x satisfying $p(x) \geq \theta$ belongs to the list L;

Proof. For any integer $m \leq k$ we denote by $p(x_1 \cdots x_m)$ the marginal probability of the bit string $x_1 \cdots x_m$. We point out the basic fact that

$$p(x_1 \cdots x_{m-1}) \geq p(x_1 \cdots x_{m-1} x_m) \tag{18}$$

for all m and for all x_j's.

The algorithm will consist of k steps. In each step we construct a list L_m containing a certain collection of m-bit strings, where m ranges from 1 to k. The final list L_k will satisfy (a)-(b) with probability at least $1 - \pi$. In the algorithm we will repeatedly invoke that each $\mathcal{P}_m$ is additively approximable; whenever an additive approximation of any $\mathcal{P}_m$ will be considered, we will set the required probability of success to be at least $1 - \delta$ with $\delta := \theta\pi/2k$ and the accuracy to be $\varepsilon := \theta/4$. Each single estimate of such a probability can be done in time

$$N_{\text{single}} = \text{poly}(k, \frac{1}{\varepsilon}, \log \frac{1}{\delta}) = \text{poly}(k, \frac{1}{\theta}, \log \frac{1}{\pi}). \tag{19}$$

Step 1. The list $L_1 \subseteq B_1 \equiv \{0, 1\}$ is computed as follows. We use that $\mathcal{P}_1$ is additively approximable and compute $p(0)$ (i.e. the probability of the outcome 0 on the first bit). More formally, we compute a number $c(0)$ satisfying

$$|c(0) - p(0)| \leq \theta/4 \tag{20}$$

with probability at least $1-\delta$. If $c(0) \geq 3\theta/4$ then define the bit 0 to belong to the list L_1. Analogously we compute $c(1)$ as an approximation of $p(1)$ and add the bit 1 to L_1 if $c(1) \geq 3\theta/4$.

Step 2. To compute the list $L_2 \subseteq B_2 \equiv \{00, 01, 10, 11\}$ we use that $\mathcal{P}_2$ is additively approximable as follows. For every $x \in L_1$ and $u \in \{0,1\}$ we compute an $\theta/4$-approximation of $p(xu)$ with probability at least $1-\delta$, yielding a number $c(xu)$ in analogy to Step 1. If $c(xu) \geq 3\theta/4$ then we add the bit pair xu to the list L_2.

Steps 3-k. The above procedure is continued for all $m = 3 \cdots k$ where in the m-th step we use that $\mathcal{P}_m$ is additively approximable. To compute the list $L_m \subseteq B_m$, for every $x_1 \cdots x_{m-1} \in L_{m-1}$ and $u \in \{0,1\}$ we compute $c(x_1 \cdots x_{m-1}u)$, which is an $\theta/4$-approximation of $p(x_1 \cdots x_{m-1}u)$ with probability at least $1-\delta$. If $c(x_1 \cdots x_{m-1}u) \geq 3\theta/4$ then we add the bit string $x_1 \cdots x_{m-1}u$ to the list L_m.

Finally, if at some point in the above algorithm one of the lists L_m contains strictly more than $2/\theta$ elements, the algorithm is halted and all subsequent lists $L_{m+1}, \ldots, L_k$ are defined to be empty. With this extra constraint, we ensure that at most $2k/\theta$ probabilities are estimated. It follows that the total runtime of the algorithm is

$$\frac{2k}{\theta} \cdot N_{\text{single}} = \text{poly}(k, \frac{1}{\theta}, \log \frac{1}{\pi}). \tag{21}$$

Furthermore, since at most $2k/\theta$ probabilities are estimated, each succeeding with probability $1-\delta$, the probability that all estimates succeed is at least $(1-\delta)^{\frac{2k}{\theta}} \geq 1 - \frac{2k}{\theta}\delta = 1 - \pi$.

From this point on we consider the case that all estimates succeed, and claim that in this case the list L_k satisfies (a)-(b). We make the following observations. First, for every m we prove property (a'): *For all $x_1 \cdots x_m \in L_m$ it holds that $p(x_1 \cdots x_m) \geq \theta/2$.* This is true since $c(x_1 \cdots x_m)$ is an $\frac{\theta}{4}$-approximation of $p(x_1 \cdots x_m)$ and since $x_1 \cdots x_m$ was only added to L_m if $c(x_1 \cdots x_m) \geq 3\theta/4$. Property (a') implies that the list L_k satisfies (a). Furthermore, property (a') implies that every list L_m contains at most $2/\theta$ bit strings (since probability distributions are normalized to sum up to 1). This shows that, as long as all estimates of the probabilities are successful, the halting procedure described above need never be applied (indeed, the latter is only incorporated in the algorithm to ensure that successive failed estimations of probabilities do not result in an (exponentially) long runtime).

Second, we argue that each L_m satisfies property (b'): *If $p(x_1 \cdots x_m) \geq \theta$ then $x_1 \cdots x_m \in L_m$.* To see this, we argue by induction on m. For $m = 1$, property (b') follows immediately from the definition of L_1. Furthermore suppose that $y = y_1 \cdots y_m$ satisfies $p(y) \geq \theta$. Then, using eq. (18) we have $p(y_1 \cdots y_{m-1}) \geq \theta$ and thus, by induction, we have $y_1 \cdots y_{m-1} \in L_{m-1}$. The definition of L_m now immediately implies that $y_1 \cdots y_m \in L_m$. This shows that property (b') holds for all L_m, so that L_k satisfies (b) as desired. □

6 Algorithm for additively approximable, approximately sparse distributions

We now arrive at an efficient algorithm which, on input of a probability distribution $\mathcal{P}$ which is promised to be approximately sparse *and* which satisfies the conditions of Theorem 10, outputs an (exactly) sparse distribution $\mathcal{P}'$ which is close to $\mathcal{P}$. In addition, the distribution $\mathcal{P}'$ can be sampled efficiently. The proof will be obtained by combining Theorem 10 and Lemma 7. The argument is straightforward but somewhat tedious since some care is required in choosing suitable epsilons and deltas. We also note that Theorem 11 is closely related to theorem 3.11 in [KM91], which provides a randomized classical algorithm for computing representations of Boolean functions which are promised to be approximately sparse.

Theorem 11. *Let $\mathcal{P}$ be a distribution on B_k which satisfies the following conditions:*

(i) $\mathcal{P}$ is promised to be ε-approximately t-sparse, where $\varepsilon \leq 1/6$.

(ii) $\mathcal{P}$ and its marginals $\mathcal{P}_m$ ($m = 1, \ldots, k$) are additively approximable as in Theorem 10.

Then there exists a randomized classical algorithm with runtime poly$(k, t, \frac{1}{\varepsilon}, \log \frac{1}{\delta})$ which outputs (by means of listing all nonzero probabilities) an s-sparse probability distribution $\mathcal{P}' = \{p'_x\}$ where $s = O(t/\varepsilon)$ such that, with probability at least $1-\delta$, $\mathcal{P}'$ is $O(\varepsilon)$-close to $\mathcal{P}$ (more precisely $\|\mathcal{P} - \mathcal{P}'\|_1 \leq 12\varepsilon$). Furthermore, $p'_x \geq \varepsilon/8t$ for all p'_x which are nonzero. Finally, it is possible to sample $\mathcal{P}'$ on a classical computer in poly$(k, t, 1/\varepsilon)$ time.

Proof. First we invoke Theorem 10 with $\theta := \varepsilon/t$ and

$$\pi := \tfrac{\delta}{2t/\varepsilon+1}. \tag{22}$$

This yields, with probability at least $1-\pi$, a list L of k-bit strings satisfying conditions (a)-(b), within a runtime

$$N_1 = \text{poly}(k, \tfrac{1}{\theta}, \log \tfrac{1}{\pi}) = \text{poly}(k, t, \tfrac{1}{\varepsilon}, \log \tfrac{1}{\delta}). \tag{23}$$

Note that $|L| \leq 2t/\varepsilon$. Second, since $\mathcal{P}$ is additively approximable, each individual probability p_x with $x \in L$ can be computed with success probability at least $1-\pi$ and with an error ε' set to

$$\varepsilon' := \min\{\varepsilon/|L|, \varepsilon/4t\} \tag{24}$$

in time

$$N_2 = \text{poly}(k, \tfrac{1}{\varepsilon'}, \log \tfrac{1}{\pi}) = \text{poly}(k, t, \tfrac{1}{\varepsilon}, \log \tfrac{1}{\delta}). \tag{25}$$

This yields a list of numbers $\{c_x : x \in L\}$ such that $|p_x - c_x| \leq \varepsilon'$ for all $x \in L$ if all evaluations were successful. Up to this point, the runtime of the algorithm is $N = N_1 + |L|N_2$ which scales as $\text{poly}(k, t, \frac{1}{\varepsilon}, \log \frac{1}{\delta})$, and the total success probability is at least

$$(1-\pi)^{|L|+1} \geq 1 - (|L|+1)\pi \geq 1 - \delta \tag{26}$$

where we have used (22) and the property $|L| \leq 2t/\varepsilon$. From this point on, the entire algorithm proceeds deterministically.

Define c_x to be 0 for all $x \notin L$ and let $\mathcal{C} = \{c_x : x \in B_k\}$ denote the resulting list of 2^k coefficients. Now let $\mathcal{Q}_{\varepsilon,t} = \{q_x\}$ be the restriction of $\mathcal{P}$ to $B_{\varepsilon,t}$, where $B_{\varepsilon,t}$ is the set of strings satisfying $p_x \geq \varepsilon/t$, as defined in Lemma 7. Note that $B_{\varepsilon,t} \subseteq L$ (recall condition (b) of Theorem 10 and the fact that here $\theta = \varepsilon/t$). Then

$$\begin{aligned} \|\mathcal{C} - \mathcal{P}\|_1 &= \sum_{x \in L} |c_x - p_x| + \sum_{x \notin L} p_x \leq |L| \cdot \varepsilon' + \sum_{x \notin L} p_x \\ &\leq \varepsilon + \sum_{x \notin L} p_x \leq \varepsilon + \sum_{x \notin B_{\varepsilon,t}} p_x = \varepsilon + \|\mathcal{P} - \mathcal{Q}_{\varepsilon,t}\|_1 \leq 3\varepsilon. \end{aligned} \tag{27}$$

Here in the first inequality we used that $|c_x - p_x| \leq \varepsilon'$ for all $x \in L$; in the second, we used the definition of ε'; in the third, we used $B_{\varepsilon,t} \subseteq L$; in the equality, we used the definition of $\mathcal{Q}_{\varepsilon,t}$; finally, we used Lemma 7.

Since $|c_x - p_x| \leq \varepsilon' \leq \varepsilon/4t$ (recall the definition of ε') and since $p_x \geq \varepsilon/2t$ owing to condition (a) of Theorem 10, we have $c_x \geq \varepsilon/4t$ for every $x \in L$; in particular, all c_x are nonnegative. Finally, we set $\mathcal{P}'$ to be $\mathcal{C}$ divided by its 1-norm $\|\mathcal{C}\|_1 = \sum |c_x|$, so that $\mathcal{P}'$ is a proper probability distribution. Since $\mathcal{P}'$ is $|L|$-sparse, computing $\mathcal{P}'$ from $\mathcal{C}$ can be done in $O(|L|) = \text{poly}(t, 1/\varepsilon)$ time. Putting everything together, the total runtime for computing $\mathcal{P}'$ scales as $\text{poly}(k, t, \frac{1}{\varepsilon}, \log \frac{1}{\delta})$. We now show that $\mathcal{P}'$ is also $O(\varepsilon)$-close to $\mathcal{P}$. The argument is straightforward and fully analogous to the one in Section 4.2, cf. (8)-(9). Since $\|\mathcal{C} - \mathcal{P}\|_1 \leq 3\varepsilon$ and $\|\mathcal{P}\|_1 = 1$ we have

$$1 - 3\varepsilon \leq \|\mathcal{C}\|_1 \leq 1 + 3\varepsilon. \tag{28}$$

We then find

$$\begin{aligned} \|\mathcal{P}' - \mathcal{P}\|_1 &= \frac{\|\mathcal{C} - \|\mathcal{C}\|_1 \cdot \mathcal{P}\|_1}{\|\mathcal{C}\|_1} \leq \frac{\|\mathcal{C} - \|\mathcal{C}\|_1 \cdot \mathcal{P}\|_1}{1 - 3\varepsilon} \\ &\leq \frac{\|\mathcal{C} - \mathcal{P}\|_1}{1 - 3\varepsilon} + \frac{|1 - \|\mathcal{C}\|_1| \cdot \|\mathcal{P}\|_1}{1 - 3\varepsilon} \leq \frac{6\varepsilon}{1 - 3\varepsilon}. \end{aligned} \tag{29}$$

Then, for $\varepsilon \leq 1/6$, we have $\|\mathcal{P}' - \mathcal{P}\|_1 \leq 12\varepsilon$. Note also that $p'_x \geq \varepsilon/8t$ for all $x \in L$ follows by combining the inequalities $c_x \geq \varepsilon/4t$ and $\|\mathcal{C}\| \leq 1 + 3\varepsilon$ and $\varepsilon \leq 1/6$.

Finally, we show how to sample $\mathcal{P}'$. For a bit string $x_1 \cdots x_m$ with m between 1 and k, let $p'(x_1 \cdots x_m)$ denote the marginal probability of $\mathcal{P}'$ for obtaining $x_1 \cdots x_m$ on the first m bits. Since $\mathcal{P}$ is s-sparse with $s = O(t/\varepsilon)$, each $p'(x_1 \cdots x_m)$ can be computed from $\mathcal{P}'$ in $\text{poly}(s) = \text{poly}(t, 1/\varepsilon)$ time on input of $x_1 \cdots x_m$. By a standard argument, the property that all such marginals can be computed, allows to sample $\mathcal{P}'$ in $\text{poly}(k, t, 1/\varepsilon)$ time [JVV86, Val02, TD04]. □

7 Classical simulation of CT states

Here we review two classical simulation results for CT states which will be used in the proofs of our results. An n-qubit unitary operator U is said to be efficiently computable basis-preserving if there exist efficiently computable functions $f, f' : B_n \to B_n$ and $g, g' : B_n \to \mathbb{C}$ where $|g(x)| = 1 = |g'(x)|$ for all $x \in B_n$, such that, for every computational basis state $|x\rangle$, one has

$$U|x\rangle = g(x)|f(x)\rangle \quad \text{and} \quad U^\dagger|x\rangle = g'(x)|f'(x)\rangle \tag{30}$$

A notable example of efficiently computable basis preserving operations is given by operators comprising tensor products of Pauli matrices $\mathbb{1}, X, Y, Z$.

Lemma 12 ([VdN11]). *Let $|\psi\rangle$ and $|\varphi\rangle$ be CT n-qubit states and let A be an efficiently computable basis-preserving n-qubit operation. Then there exists a randomized classical algorithm with runtime poly$(n, 1/\varepsilon, \log \frac{1}{\delta})$ which outputs an approximation of $\langle\psi|A|\varphi\rangle$ with accuracy ε and success probability at least $1 - \delta$.*

Lemma 13 ([VdN11]). *Let $|\psi\rangle$ and $|\varphi\rangle$ be CT n-qubit states, let $|\xi\rangle$ and $|\chi\rangle$ be CT k-qubit states with $k \leq n$. Then there exists a randomized classical algorithm with runtime poly$(n, 1/\varepsilon, \log \frac{1}{\delta})$ which outputs an approximation of $\langle\varphi|[|\xi\rangle\langle\chi| \otimes \mathbb{1}]|\psi\rangle$ with accuracy ε and success probability at least $1 - \delta$.*

The above results are slightly more detailed then the corresponding results in [VdN11] since the latter reference does not provide explicit information about the scaling with ε and δ. For completeness, proofs of Lemma 12 and Lemma 13 (which are straightforward extensions of the proofs in [VdN11]) are given in Appendix B.

8 Proofs of main results

8.1 Proof of Theorem 1

The proof will be obtained by showing that the output distribution of any quantum circuit considered in Theorem 1 satisfies the conditions of Theorem 11. We introduce some further basic definitions. For any positive integer d, let X_d, Z_d be *generalized Pauli operators* (also known as *Weyl operators*) [Got99], which act on the d-level computational basis states $|x\rangle$ (with $x \in \mathbb{Z}_d$) as follows

$$X_d|x\rangle = |x + 1\rangle \tag{31}$$

$$Z_d|x\rangle = e^{\frac{2\pi i}{d}x}|x\rangle \tag{32}$$

where $x + 1$ is defined modulo d. Note that the order of both X_d is d (i.e. is the smallest integer $r \geq 2$ satisfying $X_d^r = I$ is precisely d), as is the order of Z_d. Let $\mathcal{F}_d$ denote the Fourier transform over $\mathbb{Z}_d$. A straightforward application of definitions [Got99] shows that

$$\mathcal{F}_d^\dagger Z_d \mathcal{F}_d = X_d. \quad \text{and} \quad \mathcal{F}_d Z_d \mathcal{F}_d^\dagger = X_d^\dagger. \tag{33}$$

Theorem 1 now follows immediately from Theorem 11 in combination with the following result:

Lemma 14. *Let $\mathcal{P}$ be a probability distribution on B_k arising from a quantum circuit satisfying conditions (a)-(b) in Theorem 1. Let $\mathcal{P}_m$ denote the marginal distributions arising from measurement of the first m qubits, for $m = 1, \ldots, k$ (with $\mathcal{P} \equiv \mathcal{P}_m$). Then each $\mathcal{P}_m$ is additively approximable.*

Proof. Without loss of generality we let S be the set of first k qubits. For a k-bit string $x = (x_1, \ldots, x_k)$, consider the associated k-bit integer $\hat{x} := x_1 2^0 + x_2 2 + \cdots + x_k 2^{k-1}$. The standard basis states of a k-qubit system will be labeled both by the set of k-bit strings x and the associated integers $\hat{x}$ depending on which formulation is most convenient. Below we will use the basic fact that, for any $m = 1, \ldots, k$,

$$\hat{x} \mod 2^m = x_1 2^0 + \cdots + x_m 2^{m-1}. \tag{34}$$

Let $m \in \{1, \ldots, k\}$. For an m-bit string $y = y_1 \cdots y_m$, consider the projector (acting on k qubits)

$$|y_1 \cdots y_m\rangle\langle y_1 \cdots y_m| \otimes I \equiv P(y) \tag{35}$$

where I denotes the identity on the last $k - m$ qubits. Thus $P(y)$ is the projector onto those k-qubit computational basis states $|x\rangle$ where the first m bits of x coincide with y. Owing to (34), this means that $P(y)$ is the projector on those computational $|x\rangle$ satisfying $\hat{x} \bmod 2^m = \hat{y}$, where $\hat{y} := y_1 2^0 + \cdots + y_m 2^{m-1}$. Let $Z_{2^k} \equiv Z$ and $X_{2^k} \equiv X$ denote the generalized Pauli operators acting on $\mathbb{C}^{2^k}$. A straightforward application of the definition of Z shows that

$$\hat{x} \mod 2^m = \hat{y} \quad \text{iff} \quad \alpha^{\hat{y}} Z^{2^{k-m}} |\hat{x}\rangle = |\hat{x}\rangle \quad \text{with } \alpha := e^{-\frac{2\pi i}{2^m}}. \tag{36}$$

This implies that $P(y)$ coincides with the projector onto the eigenspace of $M := \alpha^{\hat{y}} Z^{2^{k-m}}$ with eigenvalue 1. This projector can be obtained by averaging over all powers of M; since the order of M is 2^m (recall that the order of Z is 2^k), this implies that

$$P(y) = \frac{1}{2^m} \sum_{u=0}^{2^m-1} M^u. \tag{37}$$

Let $\mathcal{F} \equiv \mathcal{F}_{2^k}$ denote the Fourier transform modulo 2^k. We consider the scenario where $\mathcal{F}$ is applied in the block U_2; the case where $\mathcal{F}^\dagger$ is applied is treated in full analogy and is omitted here. Denoting $N := \alpha^{\hat{y}} X^{2^{k-m}}$ (i.e. we replace Z by $X \equiv X_{2^k}$ in the definition of M) and recalling the first identity of eq. (32) we find

$$\mathcal{F}^\dagger P(y) \mathcal{F} = \frac{1}{2^m} \sum_{u=0}^{2^m-1} N^u. \tag{38}$$

Now denote the n-qubit CT state generated after application of the block U_1 by $|\mathrm{CT}\rangle$. Furthermore denote the marginal probability of obtaining the bit string y when measuring the first m qubits at the end of the circuit by $p(y)$. Then

$$p(y) = \langle \mathrm{CT}| [\mathcal{F}^\dagger P(y) \mathcal{F}] \otimes I |\mathrm{CT}\rangle \tag{39}$$

where I denotes the identity acting on the last $n - k$ qubits. Using Lemma 14 we find

$$p(y_1 \cdots y_m) = \frac{1}{2^m} \sum_{u=0}^{2^m-1} \langle \mathrm{CT}| N^u \otimes I |\mathrm{CT}\rangle. \tag{40}$$

It easily follows from the definition of N that each $N^u \otimes I$ is efficiently computable basis-preserving (as defined in section 7). Together with Lemma 12 this implies that the function $u \in \mathbb{Z}_{2^m} \to \langle \mathrm{CT}| N^u \otimes I |\mathrm{CT}\rangle$ is additively approximable. But then Lemma 9 implies that $y \to p(y)$ is additively approximable as well. □

8.2 Proof of Theorem 2

Similar to the proof of Theorem 1, also the proof of Theorem 2 follows immediately by showing that the output distribution of any quantum circuit considered in Theorem 2 satisfies the conditions of Theorem 11. The latter is done next.

Lemma 15. *Let $\mathcal{P}$ be a probability distribution on B_k arising from a quantum circuit satisfying conditions (a)-(b') in Theorem 2. Let $\mathcal{P}_m$ denote the marginal distributions arising from measurement of the first m qubits, for $m = 1, \ldots, k$ (with $\mathcal{P} \equiv \mathcal{P}_m$). Then each $\mathcal{P}_m$ is additively approximable.*

Proof. We prove the result for qubit systems; the proof will carry over straightforwardly to systems of qudits of potentially different dimensions. Without loss of generality we let S be the set of first k qubits. For an m-bit string $y = y_1 \cdots y_m$ with $m \leq k$, let $p(y)$ denote the marginal probability of the outcome $y_1 \cdots y_m$ when measuring the first m qubits at the end of the circuit. We need to show that the function $y \to p(y)$ is additively approximable. Denote the

CT state generated after application of the block U_1 by $|\text{CT}\rangle$. Since $U_2 = u_1 \otimes \cdots \otimes u_n$ is a tensor product operator and since $|y\rangle$ is a product state, we have

$$p(y) = \langle \text{CT}|U^\dagger[|y\rangle\langle y| \otimes \mathbb{1}]U|\text{CT}\rangle = \langle \text{CT}||\alpha\rangle\langle\alpha| \otimes \mathbb{1}|\text{CT}\rangle \tag{41}$$

for some m-qubit tensor product state $|\alpha\rangle$ (with efficiently computable description). Since product states are CT, Lemma 13 immediately implies that $y \to p(y)$ is additively approximable. □

8.3 Proof of Theorem 3 and Theorem 4

Lemma 16. *Let $|CT\rangle$ be an n-qubit CT state, let $U = U_1 \otimes \cdots \otimes U_n$ be a unitary tensor product operator and let $\mathcal{F}$ denote the Fourier transform modulo 2^n. Then the following functions are additively approximable (where $x = x_1 \cdots x_n$ is an n-bit string):*

$$x \to \langle x|\mathcal{F}|CT\rangle \tag{42}$$
$$x \to \langle x|\mathcal{F}^\dagger|CT\rangle \tag{43}$$
$$x \to \langle x|U|CT\rangle. \tag{44}$$

The last function is still additively approximable when generalized to tensor product operators acting on n qudit systems with potentially different dimensions.

Proof. A straightforward application of definitions shows that the states $\mathcal{F}|x\rangle$, $\mathcal{F}^\dagger|x\rangle$ and $U|x\rangle$ are CT. The result then immediately follows from Lemma 12 (with A being the identity). □

Lemma 17. *Let c, c' be two complex numbers satisfying $c \neq 0$ and $|c - c'| \leq \alpha$ for some $\alpha > 0$. Let $c = \theta|c|$ where θ is the phase of c and similarly $c' = \theta'|c'|$. Then $|\theta - \theta'| \leq 2\alpha/|c|$.*

Proof. Since $|c - c'| \leq \alpha$, we have $||c| - |c'|| \leq \alpha$. Then

$$|\theta - \theta'||c| = |c - \theta'|c|| \leq |c - c'| + |c' - \theta'|c|| = |c - c'| + ||c'| - |c|| \leq 2\alpha. \tag{45}$$

□

Next we prove Theorem 3 and Theorem 4. Let $|\psi_{\text{out}}\rangle$ denote the final state in any of the settings considered in Theorem 3 and Theorem 4. We write $\langle x|\psi_{\text{out}}\rangle = \gamma_x\sqrt{p_x}$ where γ_x is the phase and p_x the modulus squared, so that $\mathcal{P} = \{p_x\}$ is the probability distribution arising from measuring all qubits of $|\psi_{\text{out}}\rangle$ in the computational basis. Since $|\psi_{\text{out}}\rangle$ is $\sqrt{\varepsilon}$-approximately t-sparse, $\mathcal{P}$ is ε-approximately t-sparse by Lemma 6. Recalling Lemma 14 and Lemma 15, we find that all conditions of Theorem 11 are fulfilled. Thus there exists a randomized classical algorithm with runtime $\text{poly}(n, t, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$ which outputs an s-sparse probability distribution $\mathcal{P}' = \{p'_x\}$ where $s = O(t/\varepsilon)$ such that, with probability at least $1 - \delta$, $\|\mathcal{P}' - \mathcal{P}\|_1 \leq 12\varepsilon$. Let L be the list of bit strings as in the proof of Theorem 11. Recall from the latter proof also the following properties: $|L| \leq 2t/\varepsilon$; L is precisely the support of $\mathcal{P}'$; $p_x \geq \varepsilon/2t$ for every $x \in L$.

Thus far we have computed an approximation $\mathcal{P}'$ of the probability distribution $\mathcal{P}$. Next we will also approximately compute the amplitudes of $|\psi_{\text{out}}\rangle$ by employing Lemma 16. For every $x \in L$ we compute a complex number a_x satisfying

$$|a_x - \langle x|\psi_{\text{out}}\rangle| \leq \sqrt{\varepsilon^3/8t}. \tag{46}$$

Owing to Lemma 16, the function $x \to \langle x|\psi_{\text{out}}\rangle$ is additively approximable. Therefore each individual a_x can be computed with success probability at least $1 - \delta/|L|$ in time $N = \text{poly}(n, t, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$. Thus the total runtime for computing all a_x is $|L|T = \text{poly}(n, t, \frac{1}{\varepsilon}, \log\frac{1}{\delta})$ and the total success probability is at least $1 - \delta$. We then compute the complex phase θ_x of each a_x (which requires $O(|L|)$ computational steps in total) and define the state

$$|\varphi\rangle := \sum_{x \in L} \theta_x\sqrt{p'_x}|x\rangle. \tag{47}$$

Note that $|\varphi\rangle$ has 2-norm equal to 1: indeed $\||\varphi\rangle\|_2^2$ coincides with $\sum_{x\in L} p'_x$ which equals 1 since L coincides with the support of $\mathcal{P}'$. Next we prove that $|\varphi\rangle$ is $O(\sqrt{\varepsilon})$-close to $|\psi_{\text{out}}\rangle$. The idea of the argument is rather straightforward but the details will be somewhat tedious.

First we show that the phase θ_x is close to γ_x for every $x \in L$ (recall that the latter is the phase of $\langle x|\psi_{\text{out}}\rangle$): using Lemma 17 and recalling that $p_x \geq \varepsilon/2t$, we have

$$|\theta_x - \gamma_x| \leq 2 \cdot \sqrt{\frac{\varepsilon^3}{8t}} \cdot \frac{1}{\sqrt{p_x}} \leq \varepsilon. \tag{48}$$

This implies that

$$\| \sum_{x\in L} (\theta_x - \gamma_x)\sqrt{p'_x}|x\rangle\|_2^2 = \sum_{x\in L} |\theta_x - \gamma_x|_2^2 p'_x \leq \varepsilon^2 \sum_{x\in L} p'_x \leq \varepsilon^2. \tag{49}$$

For every two numbers $a, b \geq 0$ we have $|a-b|^2 \leq |a^2 - b^2|$. This implies that

$$\sum |\sqrt{p'_x} - \sqrt{p_x}|^2 \leq \sum |p'_x - p_x| = \|\mathcal{P}' - \mathcal{P}\|_1 \leq 12\varepsilon \tag{50}$$

where the sums are over all $x \in B_n$. Hence

$$\begin{aligned} \||\psi_{\text{out}}\rangle - \sum_{x\in L} \gamma_x \sqrt{p'_x}|x\rangle\|_2^2 &= \sum_{x\in L} |\gamma_x\sqrt{p_x} - \gamma_x\sqrt{p'_x}|^2 + \sum_{x\notin L} p_x \\ &= \sum_{x\in L} |\sqrt{p_x} - \sqrt{p'_x}|^2 + \sum_{x\notin L} p_x \\ &= \sum_{x\in B_n} |\sqrt{p_x} - \sqrt{p_x}'|^2 \leq 12\varepsilon \end{aligned} \tag{51}$$

where in the last equality we used that $p'_x = 0$ for all $x \notin L$. Writing

$$|\varphi\rangle = \sum_{x\in L} \gamma_x \sqrt{p'_x}|x\rangle + \sum_{x\in L} (\theta_x - \gamma_x)\sqrt{p'_x}|x\rangle \tag{52}$$

and using the triangle inequality, we then find

$$\begin{aligned} \||\psi_{\text{out}}\rangle - |\varphi\rangle\|_2 &\leq \||\psi_{\text{out}}\rangle - \sum_{x\in L} \gamma_x \sqrt{p'_x}|x\rangle\|_2 + \| \sum_{x\in L} (\theta_x - \gamma_x)\sqrt{p'_x}|x\rangle\|_2 \\ &\leq \sqrt{12\varepsilon} + \varepsilon \leq 5\sqrt{\varepsilon}. \end{aligned} \tag{53}$$

8.4 Proof of Theorem 5

Denote by $\mathcal{P} = \{p_x : x \in B_n\}$ the probability distribution arising from a standard basis measurement of all n qubits performed on the state $\mathcal{F}_{2^n}^\dagger|\psi\rangle$. Then $p_x = |\hat{\psi}_x|^2$. It follows from Lemma 14 that $\mathcal{P}$ and its marginals $\mathcal{P}_m$ fulfill all conditions of Theorem 10. The latter result then immediately implies the existence of a classical algorithm with runtime $\text{poly}(k, \frac{1}{\theta}, \log\frac{1}{\pi})$ which outputs a list $L = \{x^1, \ldots, x^l\}$ where $l \leq 2/\theta$ such that, with probability at least $1-\pi$, conditions (a) and (b) in Theorem 5 are fulfilled. Furthermore, Lemma 16 implies that, given any $x \in B_n$, there exists a classical algorithm with runtime $\text{poly}(n, 1/\varepsilon, \log\frac{1}{\delta})$ which, with probability at least $1-\delta$, outputs an ε-approximation of $\hat{\psi}_x$, since $\hat{\psi}_x = \langle x|\mathcal{F}_{2^n}^\dagger|\psi\rangle$.

Fully analogously, for $U = U_1 \otimes \cdots \otimes U_n$ let $\mathcal{P} = \{p_x\}$ be the probability distribution arising from a standard basis measurement of all n qubits performed on the state $U^\dagger|\psi\rangle$. The extension of Theorem 5 to the product basis $\{U|x\rangle\}$ is now obtained by combining Lemma 15, Theorem 10, and Lemma 16.

9 Further research

In the classical simulation algorithms given in this paper, we have not optimized the degree or constants involved in the polynomial-time simulation. While our algorithm is a generalization of [KM91, GL89], for optimal performance one could try to adapt the more advanced, query-optimal algorithm of [HIKP12a] to our setting.

References

[AGGM06] A. Akavia, O. Goldreich, S. Goldwasser, and D. Moshkovitz. On basing one-way functions on NP-hardness. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 701–710. ACM, 2006.

[AGS03] A. Akavia, S. Goldwasser, and S. Safra. Proving hard-core predicates using list decoding. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, volume 44, pages 146–157, Oct. 2003.

[Aka10] A. Akavia. Deterministic sparse fourier approximation via fooling arithmetic progressions. In *Proceedings of the 2010 Conference on Learning Theory, AT Kalai and M. Mohri, eds., Omnipress*, pages 381–393, 2010.

[ALM06] D. Aharonov, Z. Landau, and J. Makowsky. The quantum fft can be classically simulated. *quant-ph/0611156*, 2006.

[AMR07] Gorjan Alagic, Cristopher Moore, and Alexander Russell. Quantum algorithms for simon's problem over general groups. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1217–1224. Society for Industrial and Applied Mathematics, 2007.

[Arn05] V Arnold. Number-theoretical turbulence in fermat–euler arithmetics and large young diagrams geometry statistics. *Journal of Mathematical Fluid Mechanics*, 7:S4–S50, 2005.

[BFLW05] M Bordewich, M Freedman, L Lovász, and D Welsh. Approximate counting and quantum computation. *Combinatorics Probability and Computing*, 14(5):737–754, 2005.

[BH13] Fernando G.S.L. Brandao and Michal Horodecki. Exponential Quantum Speed-ups are Generic. *Quantum Information and Computation*, 13:0901–0924, 2013.

[BJS11] Michael J Bremner, Richard Jozsa, and Dan J Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 467(2126):459–472, 2011.

[Bro07] Daniel E Browne. Efficient classical simulation of the quantum fourier transform. *New Journal of Physics*, 9(5):146, 2007.

[BV11] Juan Bermejo-Vega. Classical simulations of non-abelian quantum fourier transforms. Master's thesis, Technische Universität München, 2011.

[BVN12] Juan Bermejo-Vega and Maarten Van den Nest. Classical simulations of Abelian-group normalizer circuits with intermediate measurements. *arXiv preprint arXiv:1210.3637*, 2012.

[GGI+02] A.C. Gilbert, S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss. Near-optimal sparse fourier representations via sampling. In *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, pages 152–161. ACM, 2002.

[GL89] O. Goldreich and LA Levin. A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 25–32. ACM, 1989.

[GMS05] A. Gilbert, S. Muthukrishnan, and M. Strauss. Improved time bounds for near-optimal sparse fourier representations. In *Proceedings of SPIE*, volume 5914, page 59141A, 2005.

[Got99] Daniel Gottesman. Fault-tolerant quantum computation with higher-dimensional systems. In *Quantum Computing and Quantum Communications*, pages 302–313. Springer, 1999.

[HIKP12a] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Nearly optimal sparse fourier transform. In *Proceedings of the 44th symposium on Theory of Computing*, pages 563–578. ACM, 2012.

[HIKP12b] H. Hassanieh, P. Indyk, D. Katabi, and E. Price. Simple and practical algorithm for sparse fourier transform. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1183–1194. SIAM, 2012.

[Iwe10] MA Iwen. Combinatorial sublinear-time fourier algorithms. *Foundations of Computational Mathematics*, 10(3):303–338, 2010.

[JVV86] Mark R Jerrum, Leslie G Valiant, and Vijay V Vazirani. Random generation of combinatorial structures from a uniform distribution. *Theoretical Computer Science*, 43:169–188, 1986.

[KM91] E. Kushilevitz and Y. Mansour. Learning decision trees using the Fourier spectrum. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 455–464. ACM, 1991.

[KP13] Pär Kurlberg and Carl Pomerance. On a problem of Arnold: The average multiplicative order of a given integer. *Algebra & Number Theory*, 7(4):981–999, 2013.

[Lom04] Chris Lomont. The hidden subgroup problem-review and open problems. *arXiv preprint quant-ph/0411037*, 2004.

[Man95] Y. Mansour. Randomized interpolation and approximation of sparse polynomials. *SIAM Journal on Computing*, 24(2):357–368, 1995.

[MO10] Ashley Montanaro and Tobias J Osborne. Quantum boolean functions. *Chicago Journal OF Theoretical Computer Science*, 1:1–45, 2010.

[MRR06] Cristopher Moore, Daniel Rockmore, and Alexander Russell. Generic quantum fourier transforms. *ACM Transactions on Algorithms (TALG)*, 2(4):707–723, 2006.

[SB09] Dan Shepherd and Michael J Bremner. Temporally unstructured quantum computation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 465(2105):1413–1439, 2009.

[Sho99] P.W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999.

[Sho11] Peter Shor. Lower bounds on the period in integer factorization? http://cstheory.stackexchange.com/questions/7043/lower-bounds-on-the-period-in-integer-factorization, 2011.

[Sta13] Dan Stahlke. Quantum interference as a resource for quantum speedup. *arXiv preprint arXiv:1305.2186*, 2013.

[TD04] Barbara M Terhal and David P DiVincenzo. Adptive quantum computation, constant depth quantum circuits and arthur-merlin games. *Quantum Information & Computation*, 4(2):134–145, 2004.

[Val02] Leslie G Valiant. Quantum circuits that can be simulated classically in polynomial time. *SIAM Journal on Computing*, 31(4):1229–1254, 2002.

[VdN10] Maarten Van den Nest. Classical simulation of quantum computation, the Gottesman-Knill theorem, and slightly beyond. *Quantum Information and Computation*, 10(3-4):0258–0271, 2010.

[VdN11] Maarten Van den Nest. Simulating quantum computers with probabilistic methods. *Quantum Information and Computation*, 11(9-10):784–812, 2011.

[VdN12] Maarten Van den Nest. Efficient classical simulations of quantum fourier transforms and normalizer circuits over abelian groups. *arXiv preprint arXiv:1201.4867*, 2012.

[YS07] Nadav Yoran and Anthony J. Short. Efficient classical simulation of the approximate quantum fourier transform. *Phys. Rev. A*, 76:042321, Oct 2007.

A Proof of lemma 8

We recall the standard Chernoff-Hoeffding bound for real-valued random variables.

Theorem 18 (Chernoff-Hoeffding bound)**.** *Let $X_1, \ldots, X_T$ be i.i.d. real random variables. Assume that $|X_i| \leq 1$ and denote $E := \mathbf{E}X_i$. Then*

$$Prob\left\{\left|\frac{1}{T}\sum_{i=1}^{T} X_i - E\right| \leq \varepsilon\right\} \geq 1 - 2e^{-\frac{T\varepsilon^2}{2}}. \tag{54}$$

The proof of the complex-valued version of the Chernoff-Hoeffding bound as given in lemma 8 is an immediate corollary of the real-valued version, as follows. For complex-valued random variables $X_1, \ldots, X_T$ we apply Theorem 18 independently to the real and imaginary parts of the X_i, where we choose $\tilde{\varepsilon} = \frac{\varepsilon}{\sqrt{2}}$. Denoting $Y := \frac{1}{T}\sum_{i=1}^{T} X_i - E$, this yields lower bounds for the probabilities that $Re(Y) \leq \tilde{\varepsilon}$ and $Im(Y) \leq \tilde{\varepsilon}$. Putting things together we find

$$\text{Prob}\left\{\left|\frac{1}{T}\sum_{i=1}^{T} X_i - E\right| \leq \varepsilon\right\} \geq 1 - 4e^{-\frac{T\varepsilon^2}{4}}. \tag{55}$$

B Proofs of lemmas 12 and 13

In this section we give explicit quantitative versions of the definition and theorems about CT states, which were only stated implicitly in [VdN10].

Definition 3 (Computationally Tractable (CT) states)**.** *An n-qubit state $|\psi\rangle$ is called 'computationally tractable' (CT) if the following conditions hold:*

1. *[Sample] it is possible to sample in time $s_{|\psi\rangle} = O(poly(n))$ with classical means from the probability distribution $Prob(x) = |\langle x|\psi\rangle|^2$ on the set of n-bit strings x.*
2. *[Query] upon input of any bit string x, the coefficient $\langle x|\psi\rangle$ can be computed in $c_{|\psi\rangle} = O(poly(n))$ time on a classical computer.*

The proof of lemma 12 will follow immediately from the following result:

Lemma 19. *Let $|\psi\rangle$ and $|\varphi\rangle$ be two CT n-qubit states and let $s = s_{|\psi\rangle} + s_{|\varphi\rangle}$, $c = c_{|\psi\rangle} + c_{|\varphi\rangle}$. Then there exists a randomized classical algorithm to compute μ such that $|\langle\varphi|\psi\rangle - \mu| \leq \varepsilon$ in time $O(\frac{s+c}{\varepsilon^2}\log(\frac{4}{\delta}))$ with error probability δ.*

Proof. Denote $p_x := |\langle x|\psi\rangle|^2$ and $q_x := |\langle x|\varphi\rangle|^2$. Since $|\psi\rangle$ and $|\varphi\rangle$ are CT states, it is possible to sample from the probability distributions $\{p_x\}$ and $\{q_x\}$ in time s (Definition 3, Item 1). Define the function $\alpha : \{0,1\}^n \mapsto \{0,1\}$ by $\alpha(x) = 1$ if $p_x \geq q_x$ and $\alpha(x) = 0$ otherwise, for every n-bit string x, and define the function β by $\beta(x) := 1 - \alpha(x)$. Then α and β can be computed in time $O(c)$ since p_x and q_x can be computed in time c each by Item 2 in Definition 3. The overlap $\langle\varphi|\psi\rangle$ is equal to

$$\langle\varphi|\psi\rangle = \sum\langle\varphi|x\rangle\langle x|\psi\rangle\alpha(x) + \sum\langle\varphi|x\rangle\langle x|\psi\rangle\beta(x) \tag{56}$$

where the sums are over all n-bit strings x. Defining the functions F and G by

$$F(x) = \frac{\langle\varphi|x\rangle\langle x|\psi\rangle}{p_x}\alpha(x), \quad G(x) = \frac{\langle\varphi|x\rangle\langle x|\psi\rangle}{q_x}\beta(x) \tag{57}$$

we have $\langle\varphi|\psi\rangle = \langle F\rangle + \langle G\rangle$, where $\langle F\rangle = \sum p_x F(x)$ and $\langle G\rangle = \sum p_x G(x)$. It follows from the query property (Definition 3, Item 2) of CT states, that F and G can be evaluated in time $O(c)$. Furthermore, both $|F(x)|$ and $|G(x)|$ are not greater than 1. It thus follows from Lemma 8, that both $\langle F\rangle$ and $\langle G\rangle$ can be approximated with accuracy $\varepsilon/2$ and error probability at most $\delta/2$ by estimating the averages over samples from the distributions p_x and q_x, respectively. More precisely, let X_i, $1 \leq i \leq T$, be samples drawn from distribution $\{p_x\}$ with $T = \frac{16}{\varepsilon^2}\log(\frac{8}{\delta})$, and let $\mu_F = \frac{1}{T}\sum_{i=1}^{T} F(X_i)$, (and similarly for samples Y_i drawn from $\{q_x\}$, $\mu_G = \frac{1}{T}\sum_{i=1}^{T} G(Y_i)$), then it follows from Lemma 8 that

$$\Pr\{|\mu_F - \langle F\rangle| \leq \varepsilon/2\} \geq 1 - \delta/2 \tag{58}$$

$$\Pr\{|\mu_G - \langle G\rangle| \leq \varepsilon/2\} \geq 1 - \delta/2 \tag{59}$$

Thus we conclude that $\langle\varphi|\psi\rangle$ can be approximated by $\mu = \mu_F + \mu_G$ in time $O(\frac{s+c}{\varepsilon^2}\log(\frac{4}{\delta}))$ such that

$$\Pr\{|\mu - \langle\varphi|\psi\rangle| \leq \varepsilon\} \geq 1 - \delta \tag{60}$$

□

The proof of lemma 13 is obtained by noting that any partial overlap of n-qubit CT states (as considered in lemma 13) can be re-expressed (via a poly(n) time classical reduction) as a complete overlap $\langle\phi|\phi'\rangle$ where $|\phi\rangle$ and $|\phi'\rangle$ are CT states on $O(n)$ qubits. Invoking lemma 12 then proves the result.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de