Low-space Quantum Algorithms for Estimate-Mark-Amplify Tasks

Debajyoti Bera[1,2], Tharrmashastha SAPV[1,*]

[1]Department of Computer Science, IIIT-Delhi, New Delhi, India.
[2]Center for Quantum Technologies, IIIT-Delhi, New Delhi, India.

Abstract

Amplitude filtering is concerned with identifying basis-states in a superposition whose amplitudes are greater than a specified threshold; probability filtering is defined analogously for probabilities. Given the scarcity of qubits, the focus of this work is to design log-space algorithms for them.

Both algorithms follow a similar pattern of estimating the amplitude (or the probability for the latter problem) of each state in superposition, then comparing each estimate against the threshold for marking a flag qubit upon success, finally followed by amplitude amplification of states in which the flag is set. We show how to implement each step using very few qubits. The main technical ingredient is an amplitude amplification algorithm that amplifies the "good state" even when the "good state" operator has a small probability of being incorrect. We provide an algorithm to perform this amplification, and we improve upon the space complexity of the previously known algorithms.

As an immediate reward, the above algorithms for the filtering problems directly improve the upper bounds on the space-bounded query complexity of problems such as non-linearity estimation of Boolean functions and a version of k-distinctness.

In addition, we present the query lower bounds of the amplitude and probability filtering problems where we show that our algorithms are tight with respect to each of the individual parameters.

Keywords

Quantum Algorithm, Quantum Complexity, Amplitude Estimation, Amplitude Amplification

1. Introduction

A quantum circuit is always associated with a distribution, say $\mathcal{D}$, over the observed outcomes that can, in principle, encode complex information. Given a threshold τ, and a blackbox to run the circuit, it may be useful to know if there is an outcome with a probability of at least τ. We call this problem Probability Filtering (denoted ProFil). We also introduce Amplitude Filtering (denoted AmpFil) that determines if the absolute value of the amplitude of any basis state is above a given threshold; even though this problem appears equivalent to ProFil, an annoying difference crawls in if we allow absolute or relative errors with respect to the threshold. We are unaware of prior algorithms for these problems. The most interesting takeaway from this work is $\tilde{O}(1)$-qubit algorithms for the ProFil and the AmpFil problems whose query complexities, measured as the number of calls to the circuit, are independent of the domain size of $\mathcal{D}$. Here by $\tilde{O}(1)$ we mean $O(\log^c n)$ for some constant $c > 0$.

ICTCS'24: Italian Conference on Theoretical Computer Science, September 11–13, 2024, Torino, Italy
*Corresponding author.
dbera@iiitd.ac.in (D. Bera); tharrmashasthav@iiitd.ac.in (T. SAPV)

The framework offered by these problems supports interesting tasks. For example, a binary search over τ (tweaked to handle the above annoyance) can be a way to compute the largest probability among all the outcomes and can be used to find the modal outcome. We have observed that several combinatorial problems can be reduced to finding the mode of a distribution or identifying if the mode is greater than something. For instance, the k-DISTINCTNESS problem generalizes the well-studied element-distinctness problem: whether an array has at least k repetitions of any element. Consider the distribution over the domain of the array elements. If any element appears at least k times, then its mode will be at least k/N (N denoting the size of the array), and vice versa. Our algorithm for probability filtering can be used to design an algorithm for k-DISTINCTNESS that makes an optimal number of queries (up to logarithmic factors) when $k = \Omega(N)$, and that too using $\tilde{O}(1)$ qubits. Previous quantum algorithms for large k have an exponential query complexity and require a larger number of qubits [1]. We hope the PROFIL and AMPFIL will be useful for designing more quantum algorithms.

When space is not a constraint, the query complexity of a discrete problem with n-sized inputs is $O(n)$, achievable by querying and caching the entire input at the beginning. However, this is not feasible when space is limited. In contrast, our algorithms are allowed only constant many logarithmic-sized registers. Thus, it should not appear as a surprise that we sometime end up making super-linear queries in an attempt to restrict the number of qubits to $\tilde{O}(1)$.

1.1. Summary of Results

The PROFIL and AMPFIL problems can be solved using an intuitively simple idea of doing amplitude estimation for each basis state in superposition, using the threshold as a marking function, and then doing amplitude amplification with respect to the marking. Doing this while ensuring that the errors inherent in the estimation step do not increase significantly in the amplification step can be reduced to the problem of biased-oracle amplitude amplification.

Biased Amplitude Amplification in Log-space (section 2) In the above mentioned amplification problem, the oracle to mark "good" states is allowed to err with some probability $1 - p$. Hoyer et al. [2] studied this problem earlier for $p = 9/10$. They proposed an algorithm that uses $O(\sqrt{1/\lambda})$ queries to obtain a marked element with high probability where λ is the probability of obtaining any marked element out of N elements. This algorithm performs an "error reduction step" after each amplification step, which uses one new qubit to reduce the error. However, in the worst case, the number of qubits required is as much as $O(\sqrt{1/\lambda})$.

To reduce the qubit footprint of our algorithms, we designed our own algorithm for biased oracle amplitude amplification based on Grover's algorithm which has a space complexity of $O(\log(N))$ qubits. For arbitrary $p > 1/2$, our algorithm uses $O\Big(\frac{p}{(p-\frac{1}{2})^2\sqrt{\lambda}} \log\big(\frac{1}{\lambda\delta}\big)\Big)$ queries and just $\log(N) + O\Big(\frac{2p}{(p-1/2)^2} \log(1/\lambda\delta)\Big)$ qubits (which is $\tilde{O}(1)$ for $p > 2/3$ and $\lambda = 1/poly(N)$). It is to be noted that the performance of our algorithm worsens when $p \approx 1/2$. In addition to use in algorithms for PROFIL and AMPFIL, our biased amplitude amplification algorithm can be used in the NISQ era, where the marking oracles are generally erroneous; this could be of independent interest to the community.

Algorithms for ProFil and AmpFil (section 3) Our objective was to design a $\tilde{O}(\frac{1}{\epsilon\tau})$-query algorithm to decide if there is any state with amplitude (rather, its absolute value) crosses the threshold τ, given the promise that either there is some such state or all states have amplitude at most $\tau - \epsilon$ for some $\epsilon > 0$. We designed our AmpFil algorithm by combining the idea of parallel estimation with our algorithm for biased amplitude amplification, where we first use amplitude estimation in parallel to estimate the amplitude and follow it with a biased-oracle amplitude amplification. We show that this algorithm uses $\tilde{O}(\frac{1}{\epsilon\tau})$ queries. While the quantum amplitude estimation algorithm is widely known for obtaining an ϵ-estimate of a probability in $O(1/\epsilon)$ queries [1], a closer look at its analysis reveals that it directly returns an ϵ-estimate of the absolute value of an amplitude using $O(1/\epsilon)$ queries. The $\tilde{O}(\frac{1}{\epsilon\sqrt{\tau}})$-complexity algorithm for ProFil was obtained by reducing it to AmpFil. One should note that the ProFil and AmpFil problems can also be solved if we were to replace our algorithm for biased amplitude amplification with the one proposed by Hoyer et al; however, the space complexity increases significantly.

For the ProFil problem, we show that our algorithm is tight with respect to the parameters ϵ and τ individually, i.e., we show a lower bound of $\Omega(\frac{1}{\epsilon} + \frac{1}{\sqrt{\tau}})$ queries. Further, we show an almost tight lower bound of $\Omega(\frac{1}{\epsilon} + \frac{1}{\tau})$ queries for the AmpFil problem. Both the lower bounds use standard approaches like the adversary method [3] and reduction from a counting problem [4]. The details on the lower bounds are presented in Section 4.

Applications of ProFil and AmpFil (section 5) The results in this work can be used to design low-space algorithms for several problems which have received recent attention. These problems can now be solved using a logarithmic number of qubits — often exponentially less compared to the existing approaches, and have a better query complexity, thus leading to better space-time complexities. The reductions are mostly straightforward, and some have been omitted due to space constraints, but the implications are interesting, as discussed below.

- Our algorithm for k-Distinctness makes an optimal number of queries (up to logarithmic factors) when $k = \Omega(n)$, and that too using $\tilde{O}(1)$ qubits (see section 5). Previous quantum algorithms for large k have an exponential query complexity in limited space [1] or require a polynomial number of qubits [5, 6, 7].
- Our algorithm for ProFil can be used to identify the presence of high-frequency items in an array (those above a given threshold — this problem is also known as "heavy hitters") using $\tilde{O}(\log \frac{1}{\epsilon})$ qubits; it also generates a superposition of such items along with estimates of their frequencies. The best algorithms for identifying heavy hitters in low space classical algorithms are of streaming nature but require $\tilde{O}(\frac{1}{\epsilon})$ space [8]. Here $\epsilon \in (0, 1]$ indicates the inaccuracy in frequency estimation.
- Our ProFil and AmpFil algorithms can be used to binary search for the largest threshold, which essentially yields the largest probability and the largest amplitude, respectively.
- Valiant and Valiant showed that $\tilde{O}(\frac{m}{\epsilon^2})$ samples of an m-valued array are sufficient to classically estimate common statistical properties of the distribution of values in the array [9]. Recently it was shown that samples of the order of $\tilde{O}(\frac{1}{g^2})$ could be used if we

[1] If this is used indirectly to estimate the absolute value of the amplitudes, then the query complexity would scale as $\frac{1}{\epsilon^2}$ instead of the desired $\frac{1}{\epsilon}$.

want to identify the item with the largest probability (denoted $p_{\max}$) [10]; here g denotes the gap between $p_{\max}$ and the largest probability strictly less than p_{max}. A binary search using ProFil makes only $\tilde{O}(\frac{1}{g\sqrt{p_{\max}}})$ queries and can locate such an item.

- The non-linearity of a Boolean function can also be calculated in terms of the amplitude with the largest norm in the output state of the Deutsch-Jozsa circuit. We present an algorithm based on AmpFil for non-linearity estimation with query complexity $O(\tilde{O}(\frac{1}{\lambda\hat{f}_{max}}))$, improving a previous work of Bera et al. with complexity $\tilde{O}(\frac{1}{\lambda^2\hat{f}_{max}})$, where $\hat{f}_{max}$ denotes the largest absolute value of any Walsh coefficient of the function. It should be noted that the best known lower bound for non-linearity estimation is $\Omega(\frac{1}{\lambda})$ (Appendix H).

To summarize, our techniques yield the current best algorithms for non-linearity estimation of Boolean functions, k-Distinctness for $k = \Omega(n)$, and k-Distinctness for constant k with $\tilde{O}(1)$-qubits; further, these algorithms almost match their lower bounds.

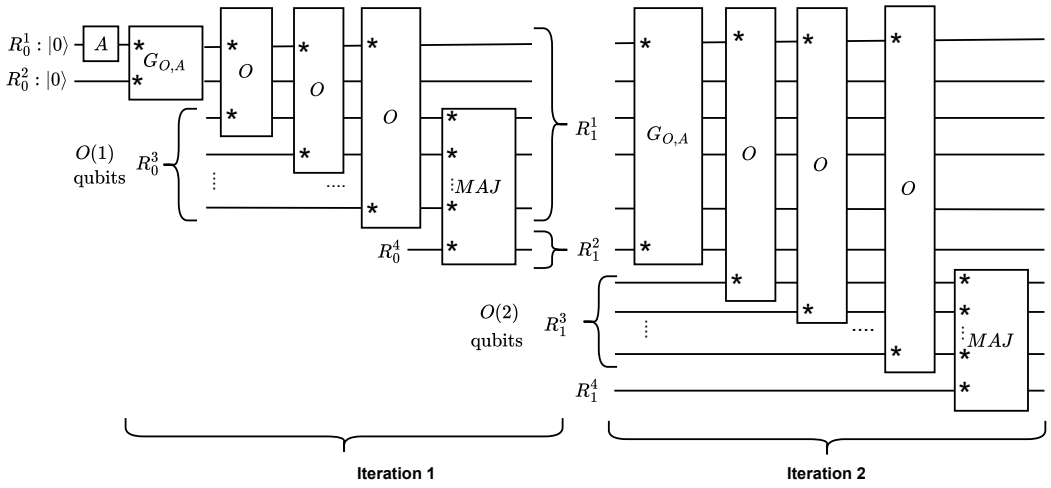
2. Amplitude Amplification using Biased Oracle

Given an oracle $\mathcal{O}$ that marks a state of interest (say $|x\rangle$) and an algorithm A such that $A|0\rangle = |\psi\rangle$, we know that the amplitude amplification algorithm allows us to obtain $|x\rangle$ *w.h.p.* from $|\psi\rangle$ quadratically faster as compared to classical approaches using A as a black-box. We use $AA_{A,\mathcal{O}}$ to denote such an amplitude amplification algorithm, the key ingredient of which is the Grover iterator $G_{A,\mathcal{O}} = -AR_{|0\rangle}A^\dagger\mathcal{O}$. Here, $R_{|i\rangle}$ denotes the reflection operator $2|i\rangle\langle i| - \mathbb{I}$. The standard assumption is that, with probability 1, the oracle $\mathcal{O}$ marks only the 'good' state $|x\rangle$ with probability 1, i.e., $\mathcal{O}|x\rangle|b\rangle = |x\rangle|b \oplus 1\rangle$ if x is 'good' and $\mathcal{O}|x\rangle|b\rangle = |x\rangle|b\rangle$ if x is 'bad'.

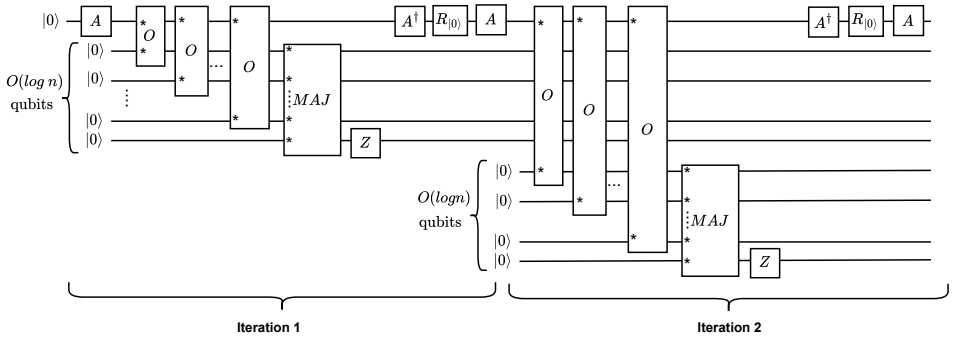
However, if we replace the oracle $\mathcal{O}$ with a bounded-error oracle $\hat{\mathcal{O}}_p$ which marks $|x\rangle$ but with some probability $p \in (1/2, 1)$, then the naive amplitude estimation algorithm does not work as intended since $\hat{\mathcal{O}}_p$ would also mark the 'bad' states with probability at most $1 - p$. With each iteration of the amplification algorithm, the probability of these false positives will also increase, thus potentially giving an erroneous output.

Previous Works: Hoyer et al. [2] first investigated this setting for $p = 9/10$ and proposed two different algorithms. We outline them below. The central idea of the first algorithm is to interleave the amplitude amplification and error reduction recursively. They showed that by following each amplification step with an error reduction step, which uses $O(k)$ extra qubits in the k^{th} iteration, it is possible to solve the bounded-error search problem using $O(\sqrt{N})$ queries to oracle $\hat{\mathcal{O}}_p$ for a search of 1 good element over N elements. However, the space complexity increases with each iteration. At the end of each iteration, the qubits in the first and second registers are highly entangled due to computing the majority. So it is impossible to cleanly uncompute the $O(k)$ qubits to get $|0\rangle$. This blows up the space complexity after each iteration. Although the algorithm is query-optimal, i.e., $O(\sqrt{N})$, its space complexity shoots to $O(\sqrt{N})$.

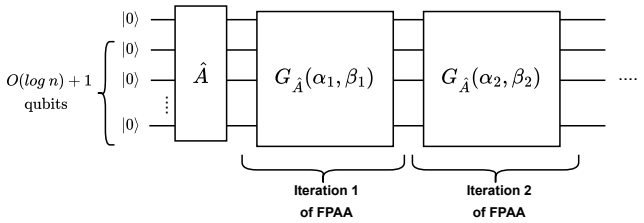
Hoyer et al. hinted at another approach to solve the bounded-error search problem. In this approach, a single call to the oracle $\hat{\mathcal{O}}_p$ in the Grover iterator is replaced by $O(\log(1/\delta))$-many independent calls to $\hat{\mathcal{O}}_p$ and using the majority value over those copies for marking the state of interest. For any $p > 1/2$ that is at least constant away from $\frac{1}{2}$, the majority value of $O(\log(1/\delta))$ outputs of $\hat{\mathcal{O}}_p$ would reduce the marking error to δ. Thus, the error accumulates to



(a) The interleaving algorithm proposed by Hoyer et al. in [2].



(b) The biased oracle amplitude amplification algorithm suggest by Hoyer et al. in [2].



(c) Our Algorithm.

Figure 1: Comparison of the biased amplitude amplification algorithms. Please refer to [11] for the exact description of $G_{\hat{A}}(\alpha_i, \beta_i)$ operators. Note that the number of working qubits increases with each iteration in the algorithms proposed in [2], whereas the number of qubits is fixed in our algorithm.

$O(\delta\sqrt{N})$ after $O(\sqrt{N})$ iterations of the Grover iterate. $O(\log(1/\delta))$ ancillæ qubits are required for performing the majority in each iteration. However, due to their entanglement with the workspace qubits, it is not possible to uncompute all the ancillæ qubits cleanly to $|0\rangle$. Hence, the space complexity would asymptotically remain $O(\sqrt{N})$. We present a detailed description

Algorithm 1 Constructing the algorithm $\hat{A}_{A,\hat{\mathcal{O}}_p,k}$

Require: Bounded-error oracle $\hat{\mathcal{O}}_p$, the initial algorithm A and k.
1: Initialize R_1 to $|0^n\rangle$. Next $k+1$ registers $R_{21}R_{22}\cdots R_{2k}R_{maj}$ are initialized to $|0\rangle$.
2: Apply A to R_1.
3: **for** i in 1 to k **do**
4: Apply $\hat{\mathcal{O}}_p$ to R_1R_{2i}.
5: **end for**
6: Apply a conditional majority gate, using R_1 as the control, using the registers $R_{21}R_{22}\ldots R_{2k}$ as inputs to the majority circuit, and storing the majority value in R_{maj}.

of these two algorithms in Appendix B.

Our Work: Now we describe our technique that uses just log-space to solve the bounded-error search problem using $O(\sqrt{N}\log(1/\delta))$ queries to $\hat{\mathcal{O}}_p$ [2]. The idea is to replace the operator A in the amplification iterator with a newly constructed operator $\hat{A}$ that internally uses $\hat{\mathcal{O}}_p$ to enhance A. The role of $\hat{A}$ will be to generate a state in which the good and the bad states are explicitly marked using an additional register whose state is $|1\rangle$ or $|0\rangle$, accordingly, and furthermore, the probability of marking a bad state can be made arbitrarily low. The algorithm for constructing such an $\hat{A}$ is presented as Algorithm 1.

Lemma 1. *Suppose that we are given an algorithm A that generates the initial state $A|0^n\rangle = \sum_x \alpha_x|x\rangle$, a bounded-error oracle $\hat{\mathcal{O}}_p$ as defined above and an error parameter δ; further, let G denote the set of good states, and B the set of bad states. Choose an appropriate $k = O\Big(\frac{2p}{(p-1/2)^2}\log\big(\frac{1}{\delta}\big)\Big)$, and construct a quantum circuit $\hat{A}$ as described in Algorithm 1. Then,*

$$\hat{A}\left|0^{n+k+1}\right\rangle = \sum_{x\in G}\alpha_x|x\rangle\left[\eta^g_{x0}|\ldots\rangle|0\rangle + \eta^g_{x1}|\ldots\rangle|1\rangle\right] + \sum_{x\in B}\alpha_x|x\rangle\left[\eta^b_{x0}|\ldots\rangle|0\rangle + \eta^b_{x1}|\ldots\rangle|1\rangle\right],$$

such that $|\eta^g_{x0}|^2 \le \delta$ and $|\eta^b_{x1}|^2 \le \delta$ (we have ignored the ancillæ).

The result can be understood by taking $\delta \to 0$ and analysing the observation upon measuring the output of $\hat{A}\left|0^{n+k+1}\right\rangle$. For $x \in G$, we are more likely to observe $|x\rangle|\ldots\rangle|1\rangle$ as compared to $|x\rangle|\ldots\rangle|0\rangle$, and for $x \in B$, $|x\rangle|\ldots\rangle|0\rangle$ is the more likely outcome, i.e., the information about x being good or bad is encoded in the final qubit, *w.h.p.*. We present the proof of Lemma 1 in Appendix C.

The next step is straightforward. We run one of the amplification routines using $\hat{A}$ as the state preparation oracle and amplifying the probability of states of the form $|x\rangle|\ldots\rangle|1\rangle$ as presented in Algorithm 2. Note that, apart from amplifying states corresponding to $x \in G$, this would also amplify states corresponding to $x \in B$. However, if we choose δ sufficiently small, we can ensure that the probability of states of the form $|x\rangle|\ldots\rangle|1\rangle$ for $x \in B$, would be extremely small, and hence, would be within tolerable limits as the algorithm terminates. We present a pictorial comparison between the three algorithms in Figure 1. One can easily note that the number of qubits in the workspace increases with each iteration for the algorithms proposed in [2] in contrast to our algorithm, where the number of qubits is fixed.

[2] Careful readers will observe a logarithmic overhead in the query complexity.

The details of Algorithm 2 and the proof of Theorem 2 are included in Appendix D; here, we briefly discuss its query complexity. Let λ be the probability of obtaining some good state on measuring $|\psi\rangle$ if some good state is present in $|\psi\rangle$; formally, $\lambda = \min_x(|\alpha_x|^2)$ over all "good" x. We will use the fact that FPAA [11] employed by the algorithm can amplify an unknown success probability, lower bounded by λ, to any desired $1-\delta$ within $O(\frac{1}{\sqrt{\lambda}}\log\frac{1}{\delta})$ iterations of Line 2. When p is a constant, the number of queries is $\tilde{O}(\frac{1}{\sqrt{\lambda}})$ and $\tilde{O}(1)$ additional qubits are used. We use FPAA as our choice of amplification algorithm because of its ability to output the correct answer with at most δ error by running the algorithm once. However, one can use the naive amplification algorithm ([12]) instead of FPAA. In that case, the naive amplification algorithm might have to be run multiple times to obtain the correct answer with error at most δ for any $0 < \delta < 1/2$.

Theorem 2. *Given an n-qubit algorithm A that generates the initial state $A|0\rangle = |\psi\rangle = \sum_{x\in\{0,1\}^n}\alpha_x|x\rangle$, a bounded-error oracle $\hat{\mathcal{O}}_p$ as defined above and an error parameter δ, there exists an algorithm that uses $O\Big(\frac{p}{(p-\frac{1}{2})^2\sqrt{\lambda}}\log\big(\frac{1}{\lambda\delta}\big)\Big)$ queries to $\hat{\mathcal{O}}_p$ along with $n + O\Big(\frac{2p}{(p-1/2)^2}\log(1/\lambda\delta)\Big)$ qubits and outputs a good state with probability at least $1-\delta$, if one exists and outputs "No Solution" with probability at least $1-\delta$ if there is no good state in $|\psi\rangle$.*

Algorithm 2 Amplitude amplification using a biased oracle

Require: Bounded-error oracle $\hat{\mathcal{O}}_p$, the initial algorithm A and k.

1: Initialize $k+2$ registers such that the first register R_1 is initialize to $|0^n\rangle$ and the next $k+1$ registers $R_{21}R_{22}\cdots R_{2k}R_{maj}$ are initialized to $|0\rangle$.
2: Use Algorithm 1 to construct $\hat{A}_{A,\hat{\mathcal{O}}_p,k}$. Then, apply $\hat{A}_{A,\hat{\mathcal{O}}_p,k}$ on $R_1R_{21}R_{22}\cdots R_{2k}R_{maj}$.
3: Apply the fixed point amplitude amplification algorithm (FPAA) on R_{maj} using the good state as $|1\rangle$ and with error at most $\delta/2$. Stop if the number of iterations crosses the limit of $\tilde{O}(\frac{1}{\sqrt{\lambda}})$ set by the FPAA algorithm.
4: Measure R_{maj} as m. If $m = |0\rangle$, output "No Solution". Else, measure R_1 as y and output y.

3. Probability and Amplitude Filtering

The filtering problems are formally defined as follows:

Problem 1 (Amplitude and Probability Filtering)**.** *Suppose that we are given a quantum algorithm O_D that generates a distribution $D : (p_x = |\alpha_x|^2)_{x=1}^m$ on measuring the first $\log(m)$ qubits of*

$$O_D\left|0^{\log(m)+a}\right\rangle = \sum_{x\in\{0,1\}^{\log(m)}}\alpha_x|x\rangle|\psi_x\rangle = |\Psi\rangle \text{ (say)}$$

in the standard basis. In addition, we are also provided a threshold τ and a parameter $0 < \epsilon < \tau$.

- *The amplitude filtering problem, denoted AmpFil(D,τ,ϵ), is to decide if $|\alpha_x| < \tau - \epsilon$ for all $x \in \{0,1\}^{\log(m)}$ or there exists some x such that $|\alpha_x| \geq \tau$ given the promise that it is one of the two cases.*

- *The probability filtering problem, denoted* ProFil(D, τ, ϵ)*, is to decide if* $p_x = |\alpha_x|^2 < \tau - \epsilon$ *for all* $x \in \{0,1\}^{\log(m)}$ *or there exists some* x *such that* $p_x \geq \tau$ *given the promise that it is one of the two cases.*

We first present an algorithm for the amplitude filtering problem. The first step is to design an appropriate biased oracle, AmpFilBOrcl, for amplitude filtering as described in algorithm 3. The oracle is used for marking a basis state to be good if its amplitude, as required, is more than τ. Then, use our amplitude amplification algorithm for biased-oracle (see Section 2) to amplify the probability of finding a marked state if one exists. We summarise the behaviour of the amplitude filtering algorithm in the following theorem.

Theorem 3 (Additive-error algorithm for AmpFil)**.** *For any choice of parameters* $0 < \epsilon < \tau$ *for additive accuracy and* δ *for error, there exists a quantum algorithm that uses* $O\big((\log(m) + \log\big(\frac{1}{\epsilon}\big) + a)\log\big(\frac{1}{\delta\tau}\big)\big)$ *qubits and makes* $O(\frac{1}{\epsilon\tau}\log\frac{1}{\delta\tau})$ *queries to* O_D *such that when its final state is measured in the standard basis, we observe the following.*

1. *If* $|\alpha_x| < \tau - \epsilon$ *for all* x *then the output register is observed in the state* $|0\rangle$ *with probability at least* $1 - \delta$*.*
2. *If* $|\alpha_x| \geq \tau$ *for any* x*, then with probability at least* $1 - \delta$ *the output register is observed in the state* $|1\rangle$ *and some* x *such that* $|\alpha_x| \geq \tau$ *is returned as output.*

Algorithm 3 Constructing biased-oracle AmpFilBOrcl for amplitude filtering

Require: Oracle O_D (with parameters m, a), threshold τ, and accuracy ϵ.
Require: Input register R_1 set to some basis state $|x\rangle$ and output register R_5 set to $|0\rangle$.
1: Set $r = \log(m) + a$, $\tau' = \frac{1}{2}(1 + \tau - \frac{\epsilon}{8})$, $q = \lceil\log\big(\frac{1}{\epsilon}\big)\rceil + 5$ and $l = q + 3$.
2: Set $\tau_1 = \left\lfloor \frac{2^l}{\pi}\sin^{-1}(\tau')\right\rfloor$.
3: Initialize ancillæ registers $R_2R_3R_4$ of lengths r, l and 1, respectively, and set $R_3 = |\tau_1\rangle$.
4: **Stage 1:** Apply EQAmpEst (sans measurement) with R_2 as the input register, R_4 as the output register and O_D is used as the state preparation oracle. R_1 is used in EQ to determine the "good state". EQAmpEst is called with error at most $1 - \frac{8}{\pi^2}$ and additive accuracy $\frac{1}{2^q}$.
5: **Stage 2:** Set R_5 to 1 if the estimate, calculated using R_4, is at least τ_1.
6: Use HD_l on R_3 and R_4 individually.
7: Use CMP on $R_3 = |\tau_1\rangle$ and R_4 as input registers and R_5 as output register.
8: Use $\mathrm{HD}_l^\dagger$ on R_3 and R_4 individually.

Now, we explain how we implemented the biased oracle (listed in Algorithm 3). The role of the oracle is to mark the basis states whose amplitude is at least τ, with at most some small error probability. Its construction involves two stages: an estimation stage followed by a marking stage. For the AmpFilBOrcl, we use EQAmpEst to estimate the amplitudes of each basis state in superposition. EQAmpEst is an extension of the well-known amplitude estimation where the basis state whose amplitude (or probability) we want to estimate is provided in a separate register instead of as an oracle. We have presented a more detailed treatment of EQAmpEst in Appendix A.2. In the marking stage, the algorithm uses straight-forward quantum operations to compare the estimate in one register with τ, hardcoded in a suitable encoding in another register. Since EQAmpEst performs an estimation with error probability that is at most $1 - \frac{8}{\pi^2}$, AmpFilBOrcl marks the good basis states with probability at least $8/\pi^2$.

The query complexity arising from biased-oracle amplitude amplification (see Section 2) scales as $\tilde{O}(\frac{1}{\sqrt{\lambda}})$ where $\lambda = \min |\alpha_x|^2$ among all x that are good. For amplitude filtering, we want to amplify any state $|x\rangle$ such that $|\alpha_x| \geq \tau$, so, $\lambda \geq \tau^2$. Thus, amplitude amplification will call AmpFilBOrcl $\tilde{O}(\frac{1}{\tau})$ times, each of which requires $O(\frac{1}{\epsilon})$ calls to O_D. The details of the AmpFilBOrcl algorithm are discussed in Appendix G.

Probability filtering can be easily reduced to amplitude filtering with very little overhead.

Lemma 4. *Any instance of ProFil(D, τ, ϵ) can be reduced to an instance of AmpFil$(D, \sqrt{\tau}, \epsilon/2)$.*

A proof is explained in Appendix F. This reduction gives us an algorithm for ProFil.

Corollary 5 (Additive-error algorithm for ProFil)**.** *For any choice of parameters $0 < \epsilon < \tau$ for additive accuracy and δ for error, there exists a quantum algorithm that uses $O\big((\log(m) + \log(\frac{1}{\epsilon}) + a)\log(\frac{1}{\delta\tau})\big)$ qubits and makes $O(\frac{1}{\epsilon\sqrt{\tau}}\log\frac{1}{\delta\tau})$ queries to O_D such that when its final state is measured in the standard basis, we observe the following.*

1. *If $p_x < \tau - \epsilon$ for all x then the output register is observed in the state $|0\rangle$ with probability at least $1 - \delta$.*
2. *If $p_x \geq \tau$ for any x, then with probability at least $1 - \delta$ the output register is observed in the state $|1\rangle$ and some x such that $|\alpha_x| \geq \tau$ is returned as output.*

With access to unbounded space, it is easy to see that one can estimate the distribution $\mathcal{D}_p$ with ϵ additive accuracy using $O(1/\epsilon^2)$ and $O(1/\epsilon)$ queries to the oracle O_D in the classical and quantum settings respectively which can then be used to solve the ProFil problem. However, in the case of $\tilde{O}(1)$ space, classically, one would be required to make $O(n/\epsilon^2)$ queries to answer the ProFil problem. In contrast, our algorithm solves the same in $\tilde{O}(1)$ space using just $\tilde{O}(1/\epsilon\sqrt{\tau})$ queries to O_D. Notice that for constant τ, our algorithm is optimal up to some $\log$ factors.

4. Lower bounds for ProFil and AmpFil

The CountDecision problem takes as input a binary string of length n and decides if the number of ones in X, denoted $|X|$, is l_1 or $l_2 > l_1$, given a promise that one of the two cases is true. Nayak and Wu proved that any quantum algorithm takes $\Omega(\sqrt{n/\Delta} + \sqrt{(l_2 - \Delta)(n - (l_2 - \Delta))/\Delta})$ queries to solve the CountDecision problem [4] in which $\Delta = \frac{1}{2}(l_2 - l_1)$.

Theorem 6. *Any quantum algorithm that solves ProFil $(\mathcal{D}, \epsilon, \tau)$ requires $\Omega(\frac{1}{\epsilon} + \frac{1}{\sqrt{\tau}})$ queries.*

To prove that ProFil requires $\Omega(\frac{1}{\epsilon})$ queries, we reduce an instance of CountDecision on a n-bit string X with $l_1 = \frac{n}{2}$ and $l_2 = \frac{n}{2} + \varepsilon n$ to ProFil. Observe that the frequencies of 0 and 1 in the string X induce a distribution $\mathcal{D}$. So, an oracle to X can be used to implement an oracle O_D to the distribution $\mathcal{D}$. By Corollary 1.2 of [4], the query complexity to decide the above CountDecision instance is $\Omega(\frac{1}{\varepsilon})$. If $|X| = \frac{n}{2}$ then $\Pr_D[0] = \Pr_D[1] = \frac{1}{2}$, and if $|X| = \frac{n}{2} + \varepsilon n$, then $\Pr_D[1] = \frac{1}{2} + \varepsilon$. Thus, the output of a ProFil algorithm with $\tau = \frac{1}{2} + \varepsilon$ and additive accuracy $\epsilon = \varepsilon$ can be used to decide our CountDecision instance, proving the $\Omega(\frac{1}{\varepsilon})$ bound.

Next we prove a bound of $\Omega(\frac{1}{\sqrt{\tau}})$. For proving this lower bound, instead of the CountDecision problem, we use the quantum adversary method. We first use this method to obtain a lower

bound on the mode decision problem: Given an array A of size n and a threshold $\tau' \in [1, n]$ we have to decide if there exists any element whose frequency is greater than τ'. We then show a reduction from mode decision problem to ProFil to get a lower bound on it.

The main theorem of the quantum adversary method can be stated as below [3]:

Theorem 7. *Let F be a n-bit Boolean function and X and Y be two sets of inputs such that $F(x) \neq F(y)$ for any $x \in X$ and $y \in Y$. Let $R \subseteq X \times Y$ be a relation such that*

1. *for every $x \in X$, $\exists$ at least m different $y \in Y$ and for every $y \in Y$, $\exists$ at least m' different $x \in X$ such that $(x, y) \in R$.*
2. *for each $i \in \{1, ..., n\}$, for every $x \in X$, $\exists$ at most l different $y \in Y$ and for every $y \in Y$, $\exists$ at most l' different $x \in X$ such that $x_i \neq y_i$ and $(x, y) \in R$.*

Then any quantum algorithm uses $\Omega\left(\sqrt{\frac{m \cdot m'}{l \cdot l'}}\right)$ queries to compute F on $X \bigcup Y$.

Consider the mode decision problem. Let $\tau' \in [1, n]$ be a threshold and set $t = \frac{n}{\tau' - 1}$. Let F be a Boolean function such that $F(x) = 1$ if x is an array whose modal value is greater than or equal to τ' and $F(y) = 0$ if the modal value of y is strictly less than τ'.

Let Y be the set containing one array B such that B contains all unique elements with frequency $\tau' - 1$. Let the unique elements be denoted $b_1, b_2, \cdots, b_t$. Let X be the set that contains the arrays A_i for all $2 \leq i \leq t$ where A_i is the array that is exactly the same as B except that the first occurrence of b_i is changed to a_1. Notice that the modal element in any A_i is b_1. Define relation R as $R = X \times Y$.

For any $A \in X$, we can see that there is exactly one element $B \in Y$ such that $(A, B) \in R$ since $|Y| = 1$. For $B \in Y$, there is exactly $t - 1$ elements $A \in X$ such that $(A, B) \in R$ as $|X| = t - 1$. Similarly, for any $A \in X$ and any $i \in [n]$, there is at most one element $B \in Y$ such that $A[i] \neq B[i]$ and $(A, B) \in R$. For $B \in Y$ and any $j \in [n]$, there is at most one element $A \in X$ such that $A[j] \neq B[j]$ and $(A, B) \in R$.

From these, we can derive that the quantum query complexity of computing F is $\Omega(\sqrt{\frac{t-1 \cdot 1}{1 \cdot 1}}) = \Omega(\sqrt{t}) = \Omega(\sqrt{n/\tau' - 1})$.

Now, the reduction from the mode decision problem to ProFil can be trivially done by setting the threshold of ProFil τ as $\tau = \tau'/n$. This would imply that the quantum query lower bound of ProFil is $\Omega(1/\sqrt{\tau})$ for $\epsilon = \frac{1}{n}$.

We know from Theorem 4 that any instance of ProFil(D, τ, ϵ) can be reduced to an instance of AmpFil$(D, \sqrt{\tau}, \epsilon/2)$. We obtain the following theorem using this reduction and Theorem 6.

Theorem 8. *Any quantum algorithm that solves AmpFil $(\mathcal{D}, \epsilon, \tau)$ requires $\Omega(\frac{1}{\epsilon} + \frac{1}{\tau})$ queries.*

From these lower bounds, we can note that our algorithms for AmpFil and ProFil, with complexities $\tilde{O}(\frac{1}{\epsilon\tau})$ and $\tilde{O}(\frac{1}{\epsilon\sqrt{\tau}})$ are tight with respect to the parameters τ and ϵ individually.

5. Applications of ProFil and AmpFil

5.1. The k-Distinctness Problem

The ElementDistinctness problem [13, 1, 14] is being studied for a long time both in the classical and the quantum domain. It is a special case of the k-Distinctness problem [1, 5]

with $k = 2$ which too has received a fair attention.

Problem 2 (k-DISTINCTNESS)**.** *Given an oracle to an n-sized m-valued array A, decide if A has k distinct indices with identical values.*

An m-valued array means one whose entries are from $\{0, \ldots, m-1\}$. Observe that k-DISTINCTNESS can be reduced to PROFIL with $\tau = \frac{k}{n}$, assuming the ability to uniformly sample.

The best-known classical algorithm for k-DISTINCTNESS uses sorting and has a time complexity of $O(n \log(n))$ with a space complexity $O(n)$. In the quantum domain, apart from $k = 2$, the $k = 3$ setting has also been studied earlier [6, 7]. The focus of all these algorithms has been primarily to reduce their query complexities. As a result, their space requirement is significant (polynomial in the size of the list). Recently, Li et al. [15] reduced the problem of estimating the min-entropy to k-DISTINCTNESS with a very large k, making this case additionally important. The F_∞ problem [16, 17], the problem of estimating the modal frequency, can also be reduced to the same, along with a promise on the gap of this frequency.

Table 1
Results for the k-DISTINCTNESS problem

k-DISTINCTNESS		
	Prior upper bound [1]	Our upper bound
$k \in \{2, 3, 4\}$	Setting $r = k$, $O((\frac{n}{k})^{k/2})$ queries, $O(\log(m) + \log(n))$ space	$\tilde{O}(n^{3/2}/\sqrt{k})$ queries, $O\big((\log(m) + \log(n)) \log\big(\frac{n}{\delta k}\big)\big)$ space
$k = \omega(1)$ and $k \geq 4$	$O(\frac{n^2}{k})$ queries, $O(\log(m) + \log(n))$ space for $r \geq k$	
$k = \Omega(n)$	$O(n^{n/2})$ queries, $O(n \log(m) + \log(n))$ space	$\tilde{O}(n)$ queries, $O\big((\log(m) + \log(n)) \log\big(\frac{n}{\delta k}\big)\big)$ space

The $k = 2$ version is the ELEMENTDISTINCTNESS problem, which was first solved by Buhrman et al. [13]; their algorithm makes $O(n^{3/4} \log(n))$ queries (with roughly the same time complexity), but requires the entire array to be stored using qubits. Ambainis [1] proposed the current best algorithm for k-DISTINCTNESS with general k. Their quantum-walk algorithm uses $\tilde{O}(r)$ qubits and $O(r + (n/r)^{k/2}\sqrt{r})$ queries (with roughly the same time complexity) for any $r \geq k$. Later Belovs designed a learning-graph for the k-DISTINCTNESS problem, but only for constant k, and obtained a tighter bound of $O(n^{\frac{3}{4} - \frac{1}{2^{k+2}-4}})$. However, it is not clear whether the bound holds for non-constant k.

Thus, it appears that even though efficient algorithms may exist for small values of k, the situation is not very pleasant for large k, especially $k = \Omega(n)$ — the learning graph idea may not work (even if the corresponding algorithm could be implemented in a time-efficient manner) and the quantum walk algorithm uses $\Omega(k)$ space.

We propose to use PROFIL to solve k-DISTINCTNESS by (a) implementing an oracle O_D from the array A (this is straightforward) and then calling our algorithm for probability filtering using $\tau = k/n$ (see Theorem 5), and $\epsilon = 1/n$ to ensure that estimates (which are always of the form t/n) are well-separated.

Lemma 9. *There exists a bounded-error algorithm for k-DISTINCTNESS, for any $k \in [n]$, that uses $O(\frac{n^{3/2}}{\sqrt{k}} \log\big(\frac{1}{\delta \cdot k}\big))$ queries and $O\big((\log(m) + \log(n)) \log\big(\frac{1}{\delta \cdot k}\big)\big)$ qubits.*

See Table 1 for a comparison of our method with respect to the others. This algorithm has a few attractive features. It is specifically designed to use $\tilde{O}(1)$ qubits, and as an added benefit, it works for any k. Further, it improves upon the algorithm proposed by Ambainis for $k \geq 4$ when we require that $\tilde{O}(1)$ space be used, and moreover, its query complexity does not increase with k. Remember that the query complexity of k-DISTINCTNESS (or any other problem) with unbounded space is trivially n but need not be so with bounded space.

For k that is large, e.g. $\Omega(n)$, the query complexity of Ambainis' algorithm is exponential in n, and that of ours is $O(n^{3/2})$. Montanaro used a reduction from the COUNTDECISION problem [4] to prove a lower bound of $\Omega(n)$ queries for $k = \Omega(n)$ — of course, assuming unrestricted space [16]. Our algorithm matches this lower bound but with only $\tilde{O}(1)$ space.

5.2. The Non-linearity Estimation Problem

Non-linearity is an important cryptographic measure of a Boolean function. Non-linearity of a function $f : \{0,1\}^n \to \{0,1\}$ is defined in terms of the largest absolute-value of its Walsh-Hadamard coefficient [18] as $\eta(f) = \frac{1}{2} - \frac{1}{2}\hat{f}_{max}$ where $\hat{f}_{max} = \max_x |\hat{f}(x)|$ and $\hat{f}(x)$ is the Walsh-Hadamard coefficient of f at the point x. Boolean functions with low non-linearity can be easily approximated by linear functions.

Ab initio, the non-linearity can be estimated from an estimate of $\hat{f}_{max}$. Recall that the output state of the Deutsch-Jozsa circuit is $\sum_x \hat{f}(x)\,|x\rangle$, i.e., the probability of observing $|x\rangle$ is $\hat{f}(x)^2$. It immediately follows that we can utilize the PROFIL algorithm in conjunction with a binary search on the interval $(0, 1]$ to estimate $\hat{f}^2_{max}$, and hence, non-linearity, with additive inaccuracy. This approach is presented as Algorithm 1 in [18]. However, this would lead to a query complexity of $\tilde{O}(1/\lambda^2\hat{f}_{max})$ [3] to estimate non-linearity to within λ additive accuracy.

Alternately, we can replace PROFIL with AMPFIL to estimate $\hat{f}_{max}$ instead of $\hat{f}^2_{max}$ in the algorithm presented in [18]. This reduces the number of queries since to estimate $\hat{f}_{max}$ within $\pm\lambda$; it now suffices to call AMPFIL with inaccuracy λ, instead of calling PROFIL with inaccuracy λ^2. Given that the query complexity of AMPFIL is $\tilde{O}(1/\lambda)$, this leads to a quadratic improvement in the query complexity in the form of $\tilde{O}(\frac{1}{\lambda\hat{f}_{max}})$.

Lemma 10. *Given an oracle to an n-bit Boolean function, an accuracy parameter λ and an error parameter δ, there exists an algorithm that returns an estimate $\tilde{\eta}_f$ such that $|\eta_f - \tilde{\eta}_f| \leq \lambda$ with probability at least $1 - \delta$ using $O(\frac{1}{\lambda\hat{f}_{max}} \log(\frac{1}{\lambda}) \log\left(\frac{1}{\delta\hat{f}_{max}}\right))$ queries to the oracle.*

Bera et al. ([18]) also showed a lower bound of $\Omega(1/\sqrt{\lambda})$ for the non-linearity estimation. This can be further improved to $\Omega(1/\lambda)$ via a reduction from the COUNTDECISION problem to the non-linearity problem. The proof of the next lemma is given in Appendix H.

Lemma 11. *Any quantum algorithm uses $\Omega(1/\lambda)$ queries to estimate the non-linearity of any given Boolean function.*

This shows that our non-linearity estimation algorithm based on AMPFIL is close to optimal.

[3] Although the query complexity of this algorithm has been proved to be $\tilde{O}(1/\lambda^3)$ in [18], the query complexity can be reduced to $\tilde{O}(1/\lambda^2\hat{f}_{max})$ with a slightly tighter analysis of their Algorithm 1.

References

[1] A. Ambainis, Quantum Walk Algorithm for Element Distinctness, SIAM Journal on Computing 37 (2007) 210–239. doi:10.1137/S0097539705447311.

[2] P. Høyer, M. Mosca, R. d. Wolf, Quantum search on bounded-error inputs, in: International Colloquium on Automata, Languages, and Programming, Springer, 2003, pp. 291–299.

[3] A. Ambainis, Quantum lower bounds by quantum arguments, Journal of Computer and System Sciences 64 (2002) 750–767.

[4] A. Nayak, F. Wu, Quantum query complexity of approximating the median and related statistics, Conference Proceedings of the Annual ACM Symposium on Theory of Computing (1999) 384–393. doi:10.1145/301250.301349.

[5] A. Belovs, Learning-Graph-Based Quantum Algorithm for k-Distinctness, in: 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science, IEEE, 2012, pp. 207–216. doi:10.1109/FOCS.2012.18.

[6] A. Belovs, Applications of the adversary method in quantum query algorithms, arXiv preprint arXiv:1402.3858 (2014).

[7] A. M. Childs, S. Jeffery, R. Kothari, F. Magniez, A time-efficient quantum walk for 3-distinctness using nested updates, arXiv preprint arXiv:1302.7316 (2013).

[8] G. Cormode, S. Muthukrishnan, An improved data stream summary: The count-min sketch and its applications, J. Algorithms 55 (2005) 58–75. doi:10.1016/j.jalgor.2003.12.001.

[9] G. Valiant, P. Valiant, Estimating the unseen: an n/log (n)-sample estimator for entropy and support size, shown optimal via new clts, in: Proceedings of the forty-third annual ACM symposium on Theory of computing, 2011, pp. 685–694.

[10] S. Dutta, A. Goswami, Mode estimation for discrete distributions, Mathematical Methods of Statistics 19 (2010) 374–384. doi:10.3103/S1066530710040046.

[11] T. J. Yoder, G. H. Low, I. L. Chuang, Fixed-point quantum search with an optimal number of queries, Physical review letters 113 (2014) 210501.

[12] G. Brassard, P. Hoyer, M. Mosca, A. Tapp, Quantum amplitude amplification and estimation, Contemporary Mathematics 305 (2002) 53–74.

[13] H. Buhrman, C. Dürr, M. Heiligman, P. Høyer, F. Magniez, M. Santha, R. de Wolf, Quantum Algorithms for Element Distinctness, SIAM Journal on Computing 34 (2005) 1324–1330. doi:10.1137/S0097539702402780.

[14] S. Aaronson, Y. Shi, Quantum lower bounds for the collision and the element distinctness problems, Journal of the ACM 51 (2004) 595–605. doi:10.1145/1008731.1008735.

[15] T. Li, X. Wu, Quantum Query Complexity of Entropy Estimation, IEEE Transactions on Information Theory 65 (2019) 2899–2921. doi:10.1109/TIT.2018.2883306.

[16] A. Montanaro, The quantum complexity of approximating the frequency moments, Quantum Information and Computation 16 (2016) 1169–1190.

[17] M. Bun, R. Kothari, J. Thaler, The polynomial method strikes back: Tight quantum query bounds via dual polynomials, in: Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, 2018, pp. 297–310.

[18] D. Bera, S. Tharrmashastha, Quantum and randomised algorithms for non-linearity estimation, ACM Transactions on Quantum Computing 2 (2021). doi:10.1145/3456509.

A. Amplitude amplification, amplitude estimation and majority

In this section, we present details on the quantum amplitude amplification subroutine and the MAJ operator which are used as part of our algorithms.

A.1. Amplitude amplification

The amplitude amplification algorithm (AA) is a generalization of the novel Grover's algorithm. Given an n-qubit algorithm A that outputs the state $|\phi\rangle = \sum_k \alpha_k |k\rangle$ on $|0^n\rangle$ and a set of basis states $G = \{|a\rangle\}$ of interest, the goal of the amplitude amplification algorithm is to amplify the amplitude α_a corresponding to the basis state $|a\rangle$ for all $|a\rangle \in G$ such that the probability that the final measurement output belongs to G is close to 1. In the most general setting, one is given access to the set G via an oracle O_G that marks all the states $|a\rangle \in G$ in any given state $|\phi\rangle$; i.e., O_G acts as

$$O_G \sum_k \alpha_k |k\rangle |0\rangle \to \sum_{a \notin G} \alpha_a |a\rangle |0\rangle + \sum_{a \in G} \alpha_a |a\rangle |1\rangle .$$

Now, for any G, any state $|\phi\rangle = \sum_k \alpha_k |k\rangle$ can be written as

$$|\phi\rangle = \sum_k \alpha_k |k\rangle = \sin(\theta) |\nu\rangle + \cos(\theta) |\overline{\nu}\rangle$$

where $\sin(\theta) = \sqrt{\sum_{a \in G} |\alpha_a|^2}$, $|\nu\rangle = \frac{\sum_{a \in G} \alpha_a |a\rangle}{\sqrt{\sum_{a \in G} |\alpha_a|^2}}$ and $|\overline{\nu}\rangle = \frac{\sum_{a \notin G} \alpha_a |a\rangle}{\sqrt{\sum_{a \notin G} |\alpha_a|^2}}$. Notice that the states $|\nu\rangle$ and $|\overline{\nu}\rangle$ are normalized and are orthogonal to each other. The action of the amplitude amplification algorithm can then be given as

$$AA\Big(\sum_k \alpha_k |k\rangle |0\rangle\Big) = AA\big(\sin(\theta) |\nu\rangle + \cos(\theta) |\overline{\nu}\rangle\big) |0\rangle \to \sqrt{(1-\beta)} |\nu\rangle |1\rangle + \sqrt{\beta} |\overline{\nu}\rangle |0\rangle$$

where β satisfies $|\beta| < \delta$ and δ is the desired error probability. This implies that on measuring the final state of AA, the measurement outcome $|a\rangle$ belongs to G with probability $|1 - \beta|$ which is at least $1 - \delta$.

A.2. Quantum Amplitude Estimation (QAE)

Consider a quantum circuit A on n qubits whose final state is $|\psi\rangle$ on input $|0^n\rangle$. Let $|x\rangle$ be some basis state (in the standard basis — this can be easily generalized to any arbitrary basis). Given an accuracy parameter $\epsilon \in (0, 1)$, the amplitude estimation problem is to estimate the probability p of observing $|x\rangle$ upon measuring $|\psi\rangle$ in the standard basis, up to an additive accuracy ϵ. Brassard et al. [12] proposed a quantum amplitude estimation circuit, which we call QAEAlgo$_A$, that acts on two registers of size n and m qubits and makes $2^m - 1$ calls to controlled-A to output an estimate $\tilde{p} \in [0, 1]$ of p that behaves as follows.

Theorem 12. *The amplitude estimation algorithm returns an estimate $\tilde{p}$ that has a confidence interval $|p - \tilde{p}| \leq 2\pi k \frac{\sqrt{p(1-p)}}{2^m} + \pi^2 \frac{k^2}{2^{2m}}$ with probability at least $\frac{8}{\pi^2}$ if $k = 1$ and with probability at least $1 - \frac{1}{2(k-1)}$ if $k \geq 2$. It uses exactly $2^m - 1$ evaluations of the oracle. If $p = 0$ or 1 then $\tilde{p} = p$ with certainty.*

The following corollary is obtained from the above theorem by setting $k = 1$ and $m = q + 3$.

Corollary 13. *The amplitude estimation algorithm returns an estimate $\tilde{p}$ that has a confidence interval $|p - \tilde{p}| \leq \frac{1}{2^q}$ with probability at least $\frac{8}{\pi^2}$ using $q + 3$ qubits and $2^{q+3} - 1$ queries. If $p = 0$ or 1 then $\tilde{p} = p$ with certainty.*

Setting $\frac{1}{2^q} = \epsilon$, the amplitude estimation algorithm returns an estimate $\tilde{p}$ that has a confidence interval $|p - \tilde{p}| \leq \epsilon$ with probability at least $\frac{8}{\pi^2}$ using $O\Big(\log\big(\frac{1}{\epsilon}\big)\Big)$ qubits and $O\big(\frac{1}{\epsilon}\big)$ queries.

We use the subscript in QAEAlgo_A to remind the reader that the circuit for amplitude estimation depends on the algorithm A that generates the state $|\psi\rangle$ from $|0^n\rangle$.

Now, let p_x be the probability of obtaining the basis state $|x\rangle$ on measuring the state $|\psi\rangle$. The amplitude estimation circuit referred to above uses an oracle, denoted O_x, to mark the "good state" $|x\rangle$, and involves measuring the output of the QAEAlgo_A circuit in the standard basis; actually, it suffices to only measure the second register. We can summarise the behaviour of the QAEAlgo_A circuit (without the final measurement) in the following lemma.

Lemma 14. *Given an oracle O_x that marks $|x\rangle$ in some state $|\psi\rangle$ and the algorithm A that acts as $A|0^n\rangle = |\psi\rangle$, QAEAlgo on an input state $|00\ldots0\rangle|0^m\rangle$ generates the following state.*

$$\textit{QAEAlgo}_{A,O_x}|00\ldots0\rangle|0^m\rangle \to \beta_{x,s}|\psi\rangle|\hat{p_x}\rangle + \beta_{x,\bar{s}}|\psi\rangle|E_x\rangle$$

Here, $|\beta_{x,s}|^2$, the probability of obtaining the good estimate, is at least $\frac{8}{\pi^2}$, and $|\hat{p}_x\rangle$ is an m-qubit normalized state of the form $|\hat{p}_x\rangle = \gamma_+|\hat{p}_{x,+}\rangle + \gamma_-|\hat{p}_{x,-}\rangle$ such that both $\sin^2(\pi\frac{\hat{p}_{x,+}}{2^m})$ and $\sin^2(\pi\frac{\hat{p}_{x,-}}{2^m})$ approximate p_x up to $m-3$ bits of accuracy. Further, $|E_x\rangle$ is an m-qubit error state (normalized) such that any basis state in $|E_x\rangle$ corresponds to a bad estimate, i.e., we can write $|E_x\rangle = \sum_{\substack{t\in\{0,1\}^m \\ t\notin\{\hat{p}_{x,+},\hat{p}_{x,-}\}}} \gamma_{t,x}|t\rangle$ in which $|\sin^2\left(\pi\frac{t}{2^m}\right) - p_x| > \frac{1}{2^{m-3}}$ for any such t.

In an alternate setting where the oracle O_x is not provided, QAEAlgo_A can still be performed if the basis state $|x\rangle$ is provided, say, in a different register. One can construct a quantum circuit, say EQ, that acts on basis states as $|x\rangle|y\rangle \mapsto (-1)^{\delta_{x,y}}|x\rangle|y\rangle$. Now perform QAEAlgo_A in which we replace all calls to O_x by EQ whose first input is set to $|x\rangle$ from the new register. We name this circuit as EQAmpEst_A that implements the following operation.

$$\text{EQAmpEst}_A\big(|x\rangle|00\ldots0\rangle|0^m\rangle\big) \to |x\rangle\left(\beta_{x,s}|\psi\rangle|\hat{p}_x\rangle + \beta_{x,\bar{s}}|\psi\rangle|E_x\rangle\right)$$

Further, since EQAmpEst_A is a quantum circuit, we could replace the state $|x\rangle$ by any superposition $\sum_x \alpha_x|x\rangle$. We would be using EQAmpEst_A in this mode in this work.

$$\text{EQAmpEst}_A\Big(\sum_x \alpha_x|x\rangle|00\ldots0\rangle|0^m\rangle\Big) \to \sum_x \alpha_x\beta_{x,s}|x\rangle|\psi\rangle|\hat{p}_x\rangle + \sum_x \alpha_x\beta_{x,\bar{s}}|x\rangle|\psi\rangle|E_x\rangle\,.$$

Let p_x denote the probability of observing the basis state $|x\rangle$ when the state $|\psi\rangle$ is measured. Notice that on measuring the first and the third registers of the output, with probability $|\alpha_x\beta_{x,s}|^2 \geq \frac{8}{\pi^2}|\alpha_x|^2$ we would obtain as measurement outcome a pair $|x\rangle|\hat{p}_x\rangle$ where

$\sin^2(\pi\frac{\hat{p}_x}{2^m}) = \tilde{p}_x$ is within $\pm\frac{1}{2^{m-3}}$ of p_x. Observe in this setting that the subroutine essentially estimates the probabilities p_x corresponding to *all* the basis states $|x\rangle$ according to the distribution implicit in the superposition. This shows how amplitude estimation can be parallelized to identify all the probabilities in a *single* distribution.

Like probability, one could be interested in estimating the absolute value of an amplitude $|\alpha_x|$ of a basis state $|x\rangle$ in $|\psi\rangle$ with an accuracy of ϵ. Naively, one can estimate the probability $p_x = |\alpha_x|^2$ with ϵ^2 accuracy as $\tilde{p}_x$ and then return $\sqrt{\tilde{p}_x}$. It is easy to show that $\sqrt{\tilde{p}_x}$ is an ϵ estimate of $|\alpha_x|$. The query complexity of this process would scale as $\frac{1}{\epsilon^2}$. However, this can be improved to $O(\frac{1}{\epsilon})$. A close inspection of the quantum amplitude estimation algorithm reveals that the output of the algorithm is an angle $\tilde{\theta}$. Moreover, $|\theta - \tilde{\theta}| \leq \epsilon/3$ implies $|\sin^2\tilde{\theta} - p_x| = |\sin^2\tilde{\theta} - \sin^2\theta| \leq \epsilon$. Nonetheless, it can be shown that $|\theta - \tilde{\theta}| \leq \epsilon/3$ also implies $|\sin\tilde{\theta} - \sin\theta| = \left|\sin\tilde{\theta} - |\alpha_x|\right| \leq \epsilon$, thus also providing an ϵ-estimate of $|\alpha_x|$ with $O(\frac{1}{\epsilon})$ queries. This suggests that any extension of the original amplitude amplification algorithm, like EQAmpEst, can also be used to estimate the absolute value of the amplitude of interest.

A.3. `MAJ` operator

Let $X_1 \ldots X_k$ be Bernoulli random variables with success probability $p > 1/2$. Let Maj denote their majority value (that appears more than $k/2$ times). Using Hoeffding's bound[4], it can be easily proved that Maj has a success probability at least $1 - \delta$, for any given δ, if we choose $k \geq \frac{2p}{(p-1/2)^2}\ln\frac{1}{\delta}$. We require a quantum formulation of the same.

Suppose we have k copies of the quantum state $|\psi\rangle = |\psi_0\rangle|0\rangle + |\psi_1\rangle|1\rangle$ in which we define "success" as observing $|0\rangle$ (without loss of generality) and k is chosen as above. Let $p = \| |\psi_0\rangle \|^2$ denote the probability of success. Suppose we measure the final qubit after applying $(\mathbb{I}^k \otimes MAJ)$ in which the MAJ operator acts on the second registers of each copy of $|\psi\rangle$. Then it is easy to show, essentially using the same analysis as above, that

$$(\mathbb{I}^k \otimes MAJ)\,|\psi\rangle^{\otimes k}|0\rangle = |\Gamma_0\rangle|0\rangle + |\Gamma_1\rangle|1\rangle$$

in which $\| |\Gamma_0\rangle \|^2 \geq 1 - \delta$.

The `MAJ` operator can be implemented without additional queries and with $poly(k)$ gates and $\log(k)$ qubits.

B. Previous works related to Biased Amplitude Amplification

Hoyer et al., in [2], introduced an algorithm for the biased-oracle amplitude amplification problem. By smartly interleaving error reduction between each amplitude amplification step, they showed that the bounded-error search can be solved using $O(\sqrt{N})$ queries to the marking oracle. The algorithm works as follows: Each iteration in the algorithm consists of two phases — the amplification phase and the error reduction phase. Say, $\mathcal{G}$ is the set of 'good' states, and $\mathcal{B}$ is the set of 'bad' states. Let $|\psi_g\rangle = \frac{1}{\sqrt{|\mathcal{G}|}}\sum_{x\in\mathcal{G}}|x\rangle$ and $|\psi_b\rangle = \frac{1}{\sqrt{|\mathcal{B}|}}\sum_{y\in\mathcal{B}}|y\rangle$.

[4] $\Pr[\sum X_i - E[\sum X_i] \geq t] \leq \exp\left(-\frac{2t^2}{n}\right)$

Then, the initial state can be given as $|\psi_0\rangle = \alpha_0 |\psi_g\rangle |0\rangle + \beta_0 |\psi_b\rangle |0\rangle$, where the first register R_0^1 contains the superposition, the second register R_0^2 is the ancilla qubit for the oracle operation, and $|\alpha_0|^2$ is the probability of obtaining a 'good' state when measuring $|\psi\rangle$ in the standard basis. After the amplification phase, the state evolves to $|\psi_0'\rangle = \alpha_0' |\Gamma_0\rangle |1\rangle + \beta_0' |\overline{\Gamma}_0\rangle |1\rangle + \eta_0' |\xi_0\rangle |0\rangle$, where $|\Gamma\rangle_0$ is a superposition of $|x\rangle$ such that $x \in \mathcal{G}$ and $|\overline{\Gamma}_0\rangle$ is that of $|x\rangle$ such that $x \in \mathcal{B}$.

Next, in the error reduction step, a register R_0^3 of $O(k)$ many qubits are attached to $|\psi_0'\rangle$ first, where k denotes the iteration number. Then, conditioned on the state of R_0^2 being $|1\rangle$, oracle is invoked on $O(k)$ qubits, and the majority of these qubits is computed and stored in a new register R_0^4. For the next iteration, R_0^1, R_0^2 and R_0^3 are considered together as the first register R_1^1 and R_0^4 is considered as the second register R_1^2 to get the state $|\psi_1\rangle = \alpha_1 |\Gamma_1\rangle |1\rangle + \beta_1 |\overline{\Gamma}_1\rangle |1\rangle + \eta_1 |\xi_1\rangle |0\rangle$.

For any k, the state after the k^{th} iteration can be given as $|\psi_k\rangle = \alpha_k |\Gamma_k\rangle |1\rangle + \beta_k |\overline{\Gamma}_k\rangle |1\rangle + \eta_k |\xi_k\rangle |0\rangle$. Performing this for $O(\sqrt{N})$ iterations yields a 'good' state with a high probability. Note that at the end of each iteration, the qubits in the first and second registers are highly entangled due to computing the majority. So it is impossible to cleanly uncompute the $O(k)$ qubits to get $|0\rangle$. This blows up the space complexity after each iteration. Despite the algorithm being query-optimal, i.e., $O(\sqrt{N})$, its space complexity shoots to $O(\sqrt{N})$.

In addition to the above discussed algorithm, Hoyer et al. presented another approach to solve the bounded-error search problem. In this approach, a single call to the oracle $\hat{\mathcal{O}}_p$ in the Grover iterator is replaced by the following sub-circuit for marking: make $O(\log(1/\delta))$-many independent calls to $\hat{\mathcal{O}}_p$, then compute the majority over those copies, and finally use the majority value for marking the state of interest. By taking the majority value of $O(\log(1/\delta))$ outputs of $\hat{\mathcal{O}}_p$, one can reduce the marking error of the oracle to δ, for any $p > 1/2$ that is at least constant away from $\frac{1}{2}$. When this sub-circuit is replaced for the bounded-error oracle $\hat{\mathcal{O}}_p$ in Grover's algorithm, the error accumulates as $O(\delta\sqrt{N})$, which can be reduced to any desired error by tweaking δ appropriately. Naturally, the ancillæ qubits required for performing the majority in each of the $O(\sqrt{N})$ calls is $O(\log(1/\delta))$. However, not all the ancillæ qubits can be cleaned up for reuse due to their entanglement with the workspace qubits. Therefore, the space complexity is still $O(\sqrt{N})$.

C. Proof of Lemma 1

Lemma 1. *Suppose that we are given an algorithm A that generates the initial state $A|0^n\rangle = \sum_x \alpha_x |x\rangle$, a bounded-error oracle $\hat{\mathcal{O}}_p$ as defined above and an error parameter δ; further, let G denote the set of good states, and B the set of bad states. Choose an appropriate $k = O\Big(\frac{2p}{(p-1/2)^2}\log\big(\frac{1}{\delta}\big)\Big)$, and construct a quantum circuit $\hat{A}$ as described in Algorithm 1. Then,*

$$\hat{A}\left|0^{n+k+1}\right\rangle = \sum_{x\in G} \alpha_x |x\rangle \left[\eta_{x0}^g |\ldots\rangle |0\rangle + \eta_{x1}^g |\ldots\rangle |1\rangle\right] + \sum_{x\in B} \alpha_x |x\rangle \left[\eta_{x0}^b |\ldots\rangle |0\rangle + \eta_{x1}^b |\ldots\rangle |1\rangle\right],$$

such that $|\eta_{x0}^g|^2 \le \delta$ and $|\eta_{x1}^b|^2 \le \delta$ (we have ignored the ancillæ).

Proof. We use the construction in algorithm 1 to prove this. After initializing the registers, on applying A on R_1, we can see that the state in R_1 is $|\psi\rangle = \sum_{x\in\{0,1\}^n} |x\rangle$. Next, on apply $\hat{\mathcal{O}}_p$

on all the R_{2i} registers, we obtain the state of the complete system as

$$|\psi_1\rangle = \sum_{x\in\{0,1\}^n} |x\rangle \left(\sqrt{p}\,|f(x)\rangle + \sqrt{1-p}\left|\overline{f(x)}\right\rangle\right)^{\otimes k} |0\rangle$$

Now, using Chernoff bounds, it is straightforward that on on computing majority over $k \geq \frac{2p}{(p-1/2)^2}\log\left(\frac{1}{\delta}\right)$ independent states of the form $\sqrt{p}\,|f(x)\rangle + \sqrt{1-p}\left|\overline{f(x)}\right\rangle$, the probability of obtaining the majority as $f(x)$ is at least $1-\delta$. Since, for each $|x\rangle$, we perform a majority over all the registers R_{2i} and save it in R_{maj}, the probability obtaining $f(x)$ in R_{maj} with the condition that R_1 is 1 is at least $1-\delta$.

□

D. Proof of Theorem 2

Theorem 2. *Given an n-qubit algorithm A that generates the initial state $A|0\rangle = |\psi\rangle = \sum_{x\in\{0,1\}^n}\alpha_x|x\rangle$, a bounded-error oracle $\hat{\mathcal{O}}_p$ as defined above and an error parameter δ, there exists an algorithm that uses $O\left(\frac{p}{(p-\frac{1}{2})^2\sqrt{\lambda}}\log\left(\frac{1}{\lambda\delta}\right)\right)$ queries to $\hat{\mathcal{O}}_p$ along with $n + O\left(\frac{2p}{(p-1/2)^2}\log(1/\lambda\delta)\right)$ qubits and outputs a good state with probability at least $1-\delta$, if one exists and outputs "No Solution" with probability at least $1-\delta$ if there is no good state in $|\psi\rangle$.*

Set $k = O(\frac{2p}{(p-1/2)^2}\log(1/\delta'))$ for a $\delta' = \lambda^4\delta^2$ and construct $\hat{A}_{A,\hat{\mathcal{O}}_p,k}$. This $\hat{A}$ (dropping the subscripts) behaves as

$$\hat{A}\,|0^n\rangle\left|0^k\right\rangle|0\rangle = \sum_x \alpha_x|x\rangle\left(\eta_{x,0}\,|\phi_{x,0}\rangle\,|0\rangle + \eta_{x,1}\,|\phi_{x,1}\rangle\,|1\rangle\right) = |\Psi\rangle$$

where $|\eta_{x,f(x)}|^2 \geq 1-\delta'$ for any x; here $f(x)$ indicates the "goodness" of x.

Now, two cases can happen.

Case (i): Let $f(x) = 0$ for all x. We analyse the situation that the algorithm does not output "No Solution", in other words, R_{maj} was observed as $|1\rangle$.

Now, the output state after $\hat{A}$ would be such that $|\eta_{x,1}|^2 \leq \delta'$ for all x. So, the probability of measuring $|1\rangle$ as output is $\sum_x |\alpha_x\eta_{x,1}|^2 \leq \delta'\sum_x|\alpha_x|^2 = \delta'$. Can such a state, with final qubit as $|1\rangle$, appear with overwhelming probability after $O(1/\sqrt{\lambda})$ iterations of amplitude amplification? We argue not by lower bounding the number of iterations needed to boost the probability of such a state to almost certainty.

Let θ be the angle made by the superposition of those states of $|\Psi\rangle$ whose last qubit is in $|1\rangle$. Then, we have $\sin^2(\theta) \leq \delta' = \lambda^4\delta^2$.

For any state $|\chi\rangle$ if the probability of obtaining a good state is $\sin^2(\theta) = \delta_1$ and if we would like to boost the probability to δ_2, then it easy to show that the number of iterations needed in the amplitude amplification algorithm is $j = \left\lceil \frac{1}{2}\left(\sin^{-1}(\sqrt{\delta_2})/\sin^{-1}(\sqrt{\delta_1})\right) - \frac{1}{2}\right\rceil > \frac{1}{4}\left(\sin^{-1}(\sqrt{\delta_2})/\sin^{-1}(\sqrt{\delta_1})\right)$. Since $\theta < \sin^{-1}(\theta)$, we have

$$j > \frac{1}{4}\left(\sin^{-1}(\sqrt{\delta_2})/\sin^{-1}(\sqrt{\delta_1})\right) > \frac{1}{4}\left(\sqrt{\delta_2}/\sin^{-1}(\sqrt{\delta_1})\right).$$

In our case, we have $\delta_1 = \lambda^4\delta^2$ and $\delta_2 = \delta$. So, the number of iterations required is

$$j > \frac{1}{4}\Big(\sqrt{\delta}/\sin^{-1}(\sqrt{\lambda^4\delta^2})\Big) = \frac{1}{4}\Big(\sqrt{\delta}/\sin^{-1}(\lambda^2\delta)\Big).$$

For any $\beta \leq 0.75$, it is easy to see that $\sin^{-1}(\beta) < \sqrt{\beta}$. Since, we set $\delta < 0.5$ and since $\lambda \leq 1$, we have $\lambda^2\delta \leq 0.5 < 0.75$. Hence we have,

$$j > \frac{1}{4}\Big(\sqrt{\delta}/\sin^{-1}(\lambda^2\delta)\Big) > \frac{1}{4}\Big(\sqrt{\delta}/\sqrt{\lambda^2\delta}\Big) = \frac{1}{4\lambda}.$$

This says that the number of amplification iterations required for improving the probability of obtaining $|1\rangle$ from $\lambda^2\delta^2$ to δ is at least $1/4\lambda$. But since the maximum number of iterations performed in the amplification routine is $O(\frac{1}{\sqrt{\lambda}})$, the probability of obtaining $|1\rangle$ on measuring the last qubit of the state after amplitude amplification is at most δ (most likely quite less).

Case (ii): Let $f(x) = 1$ for some x. In this case, for all x such that $f(x) = 1$, we will have $|\eta_{x,1}|^2 \geq 1 - \delta'$. Then the probability of measuring the last qubit as $|1\rangle$ is at least $\sum_{x:f(x)=1} |\alpha_x\eta_{x,1}|^2 \geq \lambda(1-\delta') > \lambda/2$ (since $\delta < 0.5$). Now, using the fixed point amplitude amplification subroutine, in $O(\frac{1}{\sqrt{\lambda}})$ iterations, we obtain a final state post amplification such that with probability $1-\delta$ we obtain $|1\rangle$ on measuring the R_{maj} register.

Let the post-measurement state, after observing R_{maj} in the state $|1\rangle$, be denoted $|\psi_m\rangle$. We want to clarify that it is not immediately obvious that we shall observe a good state on measuring the first register of $|\psi_m\rangle$ since the biased oracle also marks the bad states with some probability. This requires an a additional analysis.

Claim 15. *Let $|\psi_m\rangle$ be the post-measurement state obtained on measuring the last qubit as $|1\rangle$. If the set of good state $\mathscr{G} = \{x : f(x) = 1\}$ is non-empty, then the probability of obtaining some $x \in \mathscr{G}$ on measuring the first register of $|\psi_m\rangle$ is at least $3/4$.*

Proof. The state just before amplification can be given as

$$|\Psi\rangle = \sum_x \alpha_x |x\rangle \left(\eta_{x,0} |\phi_{x,0}\rangle |0\rangle + \eta_{x,1} |\phi_{x,1}\rangle |1\rangle\right)$$

where $|\eta_{x,f(x)}|^2 \geq 1 - \delta'$ for any x. The probability of obtaining some good state on the condition that the R_{maj} qubit is $|1\rangle$ is

$$Pr\Big[\, |g\rangle_{R_1} \,\Big|\, |1\rangle_{R_{maj}} \,\Big] = \frac{Pr\Big[\, |g\rangle_{R_1} |1\rangle_{R_{maj}} \,\Big]}{Pr\Big[\, |1\rangle_{R_{maj}} \,\Big]} = \frac{\sum_{x\in\mathscr{G}} |\alpha_x\eta_{x,1}|^2}{\sum_{x\in\mathscr{G}} |\alpha_x\eta_{x,1}|^2 + \sum_{x\notin\mathscr{G}} |\alpha_x\eta_{x,1}|^2} = \frac{P_g}{P_g + P_b} \text{ (say)}$$

where by $Pr\Big[\, |g\rangle_{R_1} \,\Big]$ we denote the probability of obtaining some good state in R_1. We know that

$$P_b = \sum_{x\notin\mathscr{G}} |\alpha_x\eta_{x,1}|^2 = \sum_{x\notin\mathscr{G}} |\alpha_x|^2|\eta_{x,1}|^2 \leq \delta' \sum_{x\notin\mathscr{G}} |\alpha_x|^2 \leq \delta'.$$

So, we have

$$Pr\Big[\, |g\rangle_{R_1} \,\Big|\, |1\rangle_{R_{maj}} \,\Big] = \frac{P_g}{P_g + P_b} \geq \frac{P_g}{P_g + \delta'} = \frac{1}{1 + (\delta'/P_g)}.$$

Now,

$$\begin{aligned}
\frac{\delta'}{P_g} &= \frac{\delta'}{\sum_{x\in\mathscr{G}} |\alpha_x|^2 |\eta_{x,1}|^2} \\
&\leq \frac{\delta'}{(1-\delta')\sum_{x\in\mathscr{G}} |\alpha_x|^2} \quad \big(\text{ Since } |\eta_{x,1}|^2 \geq 1-\delta' \text{ for } x\in\mathscr{G}\big) \\
&\leq \frac{\delta'}{(1-\delta')\lambda} \quad \big(\text{ Since } |\alpha_x|^2 \geq \lambda \text{ for } x\in\mathscr{G}\big) \\
&= \frac{\lambda^4\delta^2}{(1-\lambda^4\delta^2)\lambda} = \frac{\lambda^3\delta^2}{1-\lambda^4\delta^2} \leq \frac{\delta^2}{1-\delta^2} \\
&\leq \frac{1/4}{1-(1/4)} = 1/3 \quad \big(\text{ Since } \delta \leq 1/2\big).
\end{aligned}$$

Using this, we get

$$Pr\Big[\, |g\rangle_{R_1} \,\Big|\, |1\rangle_{R_{maj}} \,\Big] \geq \frac{1}{1+(\delta'/P_g)} \geq \frac{1}{1+(1/3)} = \frac{3}{4}.$$

This gives us that if R_{maj} was measured as $|1\rangle$ then on measuring R_1, with probability at least 3/4, we obtain $|x\rangle$ as measurement outcome for which $f(x)=1$. □

E. Some Useful Subroutines

In this section we first present a few subroutines that are used in the construction of ProbFilBOrcl and AmpFilBOrcl oracles.

EQ_m: Given two computational basis states $|x\rangle$ and $|y\rangle$ each of k qubits, EQ_m checks if the m-sized prefix of x and that of y are equal. Mathematically, $\text{EQ}_m|x\rangle|y\rangle = (-1)^c|x\rangle|y\rangle$ where $c=1$ if $x_i = y_i$ for all $i\in[m]$, and $c=0$ otherwise.

HD_q: When the target qubit is $|0^q\rangle$, and with a $q-$bit string y in the control register, HD computes the absolute difference of y_{int} from 2^{q-1} and outputs it as a string where y_{int} is the integer corresponding to the string y. It can be represented as $\text{HD}_q|y\rangle|b\rangle = |b\oplus\tilde{y}\rangle|y\rangle$ where $y,b\in\{0,1\}^q$ and $\tilde{y}$ is the bit string corresponding to the integer $\left|2^{q-1}-y_{int}\right|$. Even though the operator HD requires two registers, the second register will always be in the state $|0^q\rangle$ and shall be reused by uncomputing (using $HD^\dagger$) after the CMP gate. For all practical purposes, this operator can be treated as the mapping $|y\rangle\mapsto|\tilde{y}\rangle$.

CMP: The CMP operator is defined as $\text{CMP}|y_1\rangle|y_2\rangle|b\rangle = |y_1\rangle|y_2\rangle|b\oplus(y_2\leq y_1)\rangle$ where $y_1,y_2\in\{0,1\}^n$ and $b\in\{0,1\}$. It simply checks if the integer corresponding to the basis state in the first register is at most that in the second register.

Cond-MAJ: The $\text{Cond}-\text{MAJ}$ operator is defined as $\prod_x\big(|x\rangle\langle x|\otimes MAJ\big)$ where $|x\rangle\langle x|\otimes MAJ$ acts on computational basis states as $MAJ|a_1\rangle\cdots|a_k\rangle|b\rangle = |a_1\rangle\cdots|a_k\rangle|b\oplus(\tilde{a}\geq k/2)\rangle$ where $\tilde{a}=\sum_k a_k$ and $a_i,b\in\{0,1\}$.

F. Reduction of ProFil to AmpFil

Here we present the proof of the fact that probability estimation is very easily reduced to amplitude estimation.

Lemma 4. *Any instance of* ProFil(D, τ, ϵ) *can be reduced to an instance of* AmpFil$(D, \sqrt{\tau}, \epsilon/2)$.

Proof. To show the reduction we prove that the following holds for any x:

- If $p_x \geq \tau$, then $|\alpha_x| \geq \sqrt{\tau}$.
- If $p_x < \tau - 2\epsilon$, then $|\alpha_x| < \sqrt{\tau} - \epsilon$.

Consider the case when $p_x \geq \tau$. This gives $|\alpha_x|^2 \geq \tau$ which implies $|\alpha_x| \geq \sqrt{\tau}$ proving the first part of the reduction. Now, let $p_x < \tau - 2\epsilon$. This gives that $|\alpha_x| < \sqrt{\tau - 2\epsilon}$. Now, see that

$$\begin{aligned} (\sqrt{\tau} - \epsilon)^2 &= \tau + \epsilon^2 - 2\epsilon\sqrt{\tau} \geq \tau - 2\epsilon\sqrt{\tau} \geq \tau - 2\epsilon \\ \implies \sqrt{\tau} - \epsilon &\geq \sqrt{\tau - 2\epsilon} \end{aligned}$$

Using this we have, $|\alpha_x| < \sqrt{\tau - 2\epsilon} \leq \sqrt{\tau} - \epsilon$ which proves the second part of the reduction. □

G. Bounded oracle for amplitude filtering

G.1. Construction of AmpFilBOrcl to mark states with large amplitude

Algorithm 4 Constructing biased-oracle AmpFilBOrcl for probability filtering

Require: Oracle O_D (with parameters m, a), threshold τ, and accuracy ϵ.
Require: Input register R_1 set to some basis state $|x\rangle$ and output register R_5 set to $|0\rangle$.
1: Set $r = \log(m) + a$, $\tau' = \frac{1}{2}(1 + \tau - \frac{\epsilon}{8})$, $q = \lceil \log(\frac{1}{\epsilon}) \rceil + 5$ and $l = q + 3$.
2: Set $\tau_1 = \left\lfloor \frac{2^l}{\pi} \sin^{-1}(\tau') \right\rfloor$
3: Initialize ancillæ registers $R_2R_3R_4$ of lengths r, l and 1, respectively, and set $R_3 = |\tau_1\rangle$.
4: **Stage 1:** Apply EQAmpEst (sans measurement) with R_2 as the input register, R_4 as the output register and O_D is used as the state preparation oracle. R_1 is used in EQ to determine the "good state". EQAmpEst is called with error at most $1 - \frac{8}{\pi^2}$ and additive accuracy $\frac{1}{2^q}$.
5: **Stage 2:** Set R_5 to 1 if the estimate of the probability, calculated using R_4, is at least τ.
6: Use $\texttt{HD}_l$ on R_3 and R_4 individually.
7: Use CMP on $R_3 = |\tau_1\rangle$ and R_4 as input registers and R_5 as output register.
8: Use $\texttt{HD}_l^\dagger$ on R_3 and R_4 individually.

The algorithm is described in Algorithm 4. It uses the following two subroutines.

$\texttt{HD}_q$**:** When the target qubit is $|0^q\rangle$, and with a $q-$bit string y in the control register, HD computes the absolute difference of y_{int} from 2^{q-1} and outputs it as a string where y_{int} is the integer corresponding to the string y. It can be represented as $\texttt{HD}_q |y\rangle |b\rangle = |b \oplus \tilde{y}\rangle |y\rangle$ where

$y, b \in \{0,1\}^q$ and $\tilde{y}$ is the bit string corresponding to the integer $\left|2^{q-1} - y_{int}\right|$. Even though the operator HD requires two registers, the second register will always be in the state $|0^q\rangle$ and shall be reused by uncomputing (using $HD^\dagger$) after the CMP gate. For all practical purposes, this operator can be treated as the mapping $|y\rangle \mapsto |\tilde{y}\rangle$.

CMP: The CMP operator is defined as $\texttt{CMP}\,|y_1\rangle\,|y_2\rangle\,|b\rangle = |y_1\rangle\,|y_2\rangle\,|b \oplus (y_2 \le y_1)\rangle$ where $y_1, y_2 \in \{0,1\}^n$ and $b \in \{0,1\}$. It simply checks if the integer corresponding to the basis state in the first register is at most that in the second register.

In Stage 1 of the algorithm, EQAmpEst estimates the absolute value of the amplitude of $|x\rangle$ in $O_D\,|0^r\rangle$ in $R4$, and in Stage 2, this estimate is compared with τ to set or unset R_5. This makes AmpFilBOrcl a biased oracle with error $1 - \frac{8}{\pi^2}$. Further, observe that EQAmpEst makes $O(1/\epsilon)$ calls to the state preparation oracle, in this case, O_D, and no one else adds to this. The overall behaviour is summarised below.

Lemma 16. *AmpFilBOrcl makes $O(1/\epsilon)$ calls to O_D. Upon measuring its output on $|x\rangle\left|0^{2l+r+1}\right\rangle|0\rangle$, we observe the following with probability at least $\frac{8}{\pi^2}$.*

$$\textit{AmpFilBOrcl}\,|x\rangle\left|0^{2l+r+1}\right\rangle|0\rangle \Longrightarrow \begin{cases} |x\rangle\,|\phi_x\rangle\,|0\rangle & \text{, if } p_x < \tau - \epsilon, \\ |x\rangle\,|\phi_x\rangle\,|1\rangle & \text{, if } p_x \ge \tau. \end{cases}$$

G.2. Analysis of AmpFilBOrcl

Lemma 16. *AmpFilBOrcl makes $O(1/\epsilon)$ calls to O_D. Upon measuring its output on $|x\rangle\left|0^{2l+r+1}\right\rangle|0\rangle$, we observe the following with probability at least $\frac{8}{\pi^2}$.*

$$\textit{AmpFilBOrcl}\,|x\rangle\left|0^{2l+r+1}\right\rangle|0\rangle \Longrightarrow \begin{cases} |x\rangle\,|\phi_x\rangle\,|0\rangle & \text{, if } p_x < \tau - \epsilon, \\ |x\rangle\,|\phi_x\rangle\,|1\rangle & \text{, if } p_x \ge \tau. \end{cases}$$

Proof. We analyse the algorithm on the input state on registers $R_1R_{21}R_{22}R_3R_4R_5$ as $|x\rangle\,|0^r\rangle\left|0^l\right\rangle\left|0^l\right\rangle|0\rangle$ where $t = 2l + r + 1$. We set $R_3 = |\tau_1\rangle$. On applying EQAmpEst_{O_D} on $R_1R_2R_4$ with R_2 as the input register and R_4 as the output register and R_1 for marking the "good" state whose amplitude we desire to estimate (using, of course, the EQ oracle), the input state transforms to

$$\begin{aligned} |\psi_1\rangle &= |x\rangle\,|\Psi\rangle\left(\beta_{x,s}\,|a_x\rangle + \beta_{x,\bar{s}}\,|E_x\rangle\right)|\tau_1\rangle\,|0\rangle \\ &= \beta_{x,s}\,|x\rangle\,|\Psi\rangle\,|a_x\rangle\,|\tau_1\rangle\,|0\rangle + \beta_{x,\bar{s}}\,|x\rangle\,|\Psi\rangle\,|E_x\rangle\,|\tau_1\rangle\,|0\rangle \\ &= \beta_{x,s}\,|\psi_{1,s}\rangle + \beta_{x,\bar{s}}\,|\psi_{1,\bar{s}}\rangle \end{aligned}$$

where $|a_x\rangle$ is a normalized state of the form $|a_x\rangle = \gamma_+\,|a_{x,+}\rangle + \gamma_-\,|a_{x,-}\rangle$ that on measurement outputs $a \in \{a_{x,+}, a_{x,+}\}$ which is an l-bit string that behaves as $\left|\sin\left(\frac{a\pi}{2^l}\right) - |\alpha_x|\right| \le \frac{1}{2^q}$, $|\beta_{x,s}|^2 \ge \frac{8}{\pi^2}$ and $|\beta_{x,s}|^2 \le 1 - \frac{8}{\pi^2}$.

We denote the set $\{a_{x,+}, a_{x,-}\}$ by S_{a_x}. Essentially, for any x, EQAmpEst stores the correct estimate of the absolute value of the amplitude of x in $|\Psi\rangle$ into R_4 with probability at least $\frac{8}{\pi^2}$.

The correctness of stage-2 is exactly the same as that in the proof for Theorem 16.

Query Complexity : All calls to O_D are made by EQAmpEst and the latter's query complexity is $O(1/\epsilon)$. □

H. Lower bound for Non-linearity Estimation

Recall that the non-linearity of a Boolean function $f : \{0,1\}^n \to \{0,1\}$ is defined as

$$\eta(f) = \frac{1}{2} - \frac{1}{2}\hat{f}_{max}$$

where $\hat{f}_{max} = \max_x |\hat{f(x)}|$ and $\hat{f}(x)$ is the Walsh coefficient of f at the point x. We define a decision problem, namely the $\hat{f}_{max}$ decision problem, as follows: given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, a threshold τ and a parameter λ, decide if $\hat{f}_{max} \geq \tau$ or if $\hat{f}_{max} < \tau - \lambda$ given the promise that one of the two cases is true. It is quite straight forward that the $\hat{f}_{max}$ problem can be directly reduced to the problem of non-linearity estimation. So, to show a lower bound for the non-linearity estimation problem, we show a reduction from the CountDecision problem to the $\hat{f}_{max}$ decision problem. First consider the following lemma which will help prove the required reduction.

Lemma 17. *The query complexity of any quantum algorithm that solves CountDecision $(N/4, N/4 - \Delta)$ is $\Omega(N/\Delta)$ for any $0 < \Delta \leq N/5$.*

Proof. Using Corrolary 1.2 of [4], we obtain that the query complexity is

$$\begin{aligned} Q_{\text{CountDecision}} &= \Omega\left(\sqrt{\frac{N}{\Delta}} + \frac{\sqrt{\left(\frac{N}{4} - \Delta\right)\left(N - \left(\frac{N}{4} - \Delta\right)\right)}}{\Delta}\right) \\ &= \Omega\left(\sqrt{\frac{N}{\Delta}} + \frac{\sqrt{\frac{3}{16}N^2 + \Delta N - \Delta^2}}{\Delta}\right) \\ &= \Omega(N/\Delta). \end{aligned}$$

□

Lemma 11. *Any quantum algorithm uses $\Omega(1/\lambda)$ queries to estimate the non-linearity of any given Boolean function.*

Proof. For simplicity let N be some power of 2. Consider the CountDecision $(N/4, N/4 - \Delta)$ problem for some $0 < \Delta \leq N/5$. The task is to decide if the Hamming weight of the given string x is $N/4$ or $N/4 - \Delta$.

Now, for a given string x, construct a Boolean function $f^{(x)} : \{0,1\}^n \to \{0,1\}$ such that $f^{(x)}(i) = x_i$ where $n = \log(N)$. We show that the problem of deciding if the Hamming weight of x is $N/4$ or $N/4 - \Delta$ can be solved by deciding if $\hat{f}^{(x)}_{max}$ is $\frac{1}{2}$ or $\frac{1}{2} + \frac{2\Delta}{N}$.

Let y be any string of Hamming weight $N/4$. Let $f^{(y)}$ be the Boolean function constructed using y. We know that the Walsh coefficient of function f at a is defined as

$$\begin{aligned}\hat{f}(a) &= \frac{1}{2^n}\sum_{x\in\{0,1\}^n}(-1)^{f(x)\oplus a\cdot x}\\ &= \frac{1}{2^n}\Big[\big|\{x\in\{0,1\}^n : f(x)=a\cdot x\}\big| - \big|\{x\in\{0,1\}^n : f(x)\neq a\cdot x\}\big|\Big].\end{aligned}$$

Intuitively, $|\hat{f}(a)|$ gives the difference in the fraction of inputs x for which the function f matches with the linear function $a\cdot x$ and the fraction of inputs for which the function does not match with $a\cdot x$. From this we can compute the Walsh coefficient of $f^{(y)}$ at 0^n to be

$$\hat{f}^{(y)}(0^n) = \frac{1}{2^n}\left(\frac{3N}{4} - \frac{N}{4}\right) = \frac{1}{2}.$$

Now, let $a\neq 0^n$ be some n-bit string. Then, $a\cdot x$ is a linear function with equal number of 0's and 1's in its output. See that, for any Boolean function whose Hamming weight[5] is $N/4$, the maximum number of inputs such that $f(x)=a\cdot x$ is bounded above by $3N/4$ where $N/2$ inputs has to be such that $a\cdot x=0=f(x)$ and $N/4$ inputs has to be such that $a\cdot x=1=f(x)$. So, we have the Walsh coefficient of $f^{(y)}$ at any $a\neq 0^n$ as

$$\hat{f}^{(y)}(a) \leq \frac{1}{2^n}\left(\frac{3N}{4} - \frac{N}{4}\right) = \frac{1}{2}.$$

So, we have that $\hat{f}^{(y)}_{max} = \frac{1}{2}$ and it occurs at 0^n.

Next, let z be a string of Hamming weight $N/4-\Delta$ and let $f^{(z)}$ be the Boolean function constructed from z. For $f^{(z)}$, we have that

$$\hat{f}^{(z)}(0^n) = \frac{1}{2^n}\left[\left(\frac{3N}{4}+\Delta\right) - \left(\frac{N}{4}-\Delta\right)\right] = \frac{1}{2} + \frac{2\Delta}{2^n}.$$

Again, for any Boolean function f of Hamming weight $N/4-\Delta$, the maximum number of inputs such that $f(x)=a\cdot x$ is $\frac{3N}{4}-\Delta$ where $N/2$ inputs has to be such that $a\cdot x=0=f(x)$ and $N/4-\Delta$ inputs has to be such that $a\cdot x=1=f(x)$. So, we get that the Walsh coefficient of $f^{(z)}$ at any $a\neq 0^n$ is

$$\hat{f}^{(z)}(a) \leq \frac{1}{2^n}\left[\left(\frac{3N}{4}-\Delta\right) - \left(\frac{N}{4}+\Delta\right)\right] = \frac{1}{2} - \frac{2\Delta}{2^n}.$$

Thus, we get that $\hat{f}^{(z)}_{max} = \frac{1}{2} + \frac{2\Delta}{2^n}$ and it occurs at 0^n.

Consequently, any algorithm that solves the $\hat{f}_{max}$ decision problem for the parameters $\tau = \frac{1}{2}+\frac{2\Delta}{N}$ and $\lambda = \frac{\Delta}{N}$ can solve the CountDecision $(\frac{N}{4}, \frac{N}{4}-\Delta)$ problem without any query overhead. Now, using Lemma 17, we get that any quantum algorithm that solves the $\hat{f}_{max}$ decision problem is $\Omega(\frac{N}{\Delta}) = \Omega(\frac{1}{\lambda})$. □

[5]By the Hamming weight of a Boolean function f, we mean the number of 1's in the output of f.