Quantum Chebyshev's Inequality and Applications

Yassine Hamoudi and Frédéric Magniez

IRIF, Université Paris Diderot, CNRS, France
{hamoudi,magniez}@irif.fr

Abstract

In this paper we provide new quantum algorithms with polynomial speed-up for a range of problems for which no such results were known, or we improve previous algorithms. First, we consider the approximation of the frequency moments F_k of order $k \geq 3$ in the multi-pass streaming model with updates (turnstile model). We design a P-pass quantum streaming algorithm with memory M satisfying a tradeoff of $P^2M = \widetilde{\mathcal{O}}\left(n^{1-2/k}\right)$, whereas the best classical algorithm requires $PM = \Theta(n^{1-2/k})$. Then, we study the problem of estimating the number m of edges and the number t of triangles given query access to an n-vertex graph. We describe optimal quantum algorithms that perform $\widetilde{\mathcal{O}}\left(\sqrt{n}/m^{1/4}\right)$ and $\widetilde{\mathcal{O}}\left(\sqrt{n}/t^{1/6} + m^{3/4}/\sqrt{t}\right)$ queries respectively. This is a quadratic speed-up compared to the classical complexity of these problems.

For this purpose we develop a new quantum paradigm that we call Quantum Chebyshev's inequality. Namely we demonstrate that, in a certain model of quantum sampling, one can approximate with *relative error* the mean of any random variable with a number of quantum samples that is linear in the ratio of the square root of the variance to the mean. Classically the dependency is quadratic. Our algorithm subsumes a previous result of Montanaro [52]. This new paradigm is based on a refinement of the Amplitude Estimation algorithm of Brassard et al. [13] and of previous quantum algorithms for the mean estimation problem. We show that this speed-up is optimal, and we identify another common model of quantum sampling where it cannot be obtained. For our applications, we also adapt the variable-time amplitude amplification technique of Ambainis [5] into a variable-time amplitude estimation algorithm.

1 Introduction

Motivations and background Randomization and probabilistic methods are among the most widely used techniques in modern science, with applications ranging from mathematical economics to medicine or particle physics. One of the most successful probabilistic approaches is the Monte Carlo Simulation method for algorithm design, that relies on repeated random sampling and statistical analysis to estimate parameters and functions of interest. From Buffon's needle experiment, in the eighteenth century, to the simulations of galaxy formation or nuclear processes, this method and its variations have become increasingly popular to tackle problems that are otherwise intractable. The Markov chain Monte Carlo method [39] led for instance to significant advances for approximating parameters whose exact computation is #P-hard [43, 41, 24, 40].

The analysis of Monte Carlo Simulation methods is often based on concentration inequalities that characterize the deviation of a random variable from some parameter. In particular, the Chebyshev inequality is a key element in the design of randomized methods that estimate some target numerical value. Indeed, this inequality guarantees that the arithmetic mean of Δ^2/ε^2 independent samples, from a random variable with variance σ^2 and mean μ satisfying $\Delta \geq \sigma/\mu$, is an approximation of μ under relative error ε with high probability. This basic result is at the heart of many computational problems, such as counting via Markov chains [39, 60], estimating graph parameters [20, 30, 33, 26], testing properties of classical [34, 10, 19, 16] or quantum [14, 9] distributions, approximating the frequency moments in the data stream model [4, 51, 6].

Various quantum algorithms have been developed to speed-up or generalize classical Monte Carlo methods (e.g. sampling the stationary distributions of Markov-chains [61, 56, 23, 59, 21], estimating the expected values of observables or partition functions [45, 62, 56, 52]). The mean estimation problem (as addressed by Chebyshev's inequality) has also been studied in the *quantum sampling model*. In this model, a distribution is represented by a unitary transformation (called a *quantum sampler*) preparing a superposition over the elements of the distribution, with the amplitudes encoding the probability mass function. A *quantum sample* is defined as one execution of a quantum sampler or its inverse. The number of quantum samples needed to estimate the mean of a distribution on a bounded space $[0,B]$, with *additive* error ε, was proved to be $\mathcal{O}(B/\varepsilon)$ [36, 12], or $\widetilde{\mathcal{O}}(\bar{\sigma}/\varepsilon)$ [52] given an upper-bound $\bar{\sigma}^2$ on the variance. On the other hand, the mean estimation problem with *relative* error ε can be solved with $\mathcal{O}\left(\sqrt{B}/(\varepsilon\sqrt{\mu})\right)$ quantum samples [13, 62]. Interestingly, this is a quadratic improvement over $\sigma^2/(\varepsilon\mu)^2$ if the sample space is $\{0,B\}$ (this case maximizes the variance). Montanaro [52] posed the problem of whether this speed-up can be generalized to other distributions. He assumed that one knows an upper bound[1] Δ on $1+\sigma/\mu$, and gave an algorithm using[2] $\widetilde{\mathcal{O}}\left(\Delta^2/\varepsilon\right)$ quantum samples (thus improving the dependence on ε, compared to the classical setting). This result was reformulated in [47] to show that, knowing bounds $L \leq \mu \leq H$, it is possible to use $\widetilde{\mathcal{O}}(\Delta/\varepsilon \cdot H/L)$ quantum samples. Typically, the only upper-bound known on μ is $H = B$, so it is less efficient than [13, 62].

Quantum Chebyshev Inequality Our main contribution (Theorem 3.3 and Theorem A.2) is to show that the mean μ of any distribution with variance σ^2 can be approximated with relative error ε using $\widetilde{\mathcal{O}}(\Delta \cdot \log(H/L) + \Delta/\varepsilon)$ quantum samples, given an upper bound Δ on $1+\sigma/\mu$ and two bounds L, H such that $L < \mu < H$. This is an exponential improvement in H/L compared to previous works [47]. Moreover, if $\log(H/L)$ is negligible, this is a quadratic improvement over the number of classical samples needed when using the Chebyshev inequality. If no bound L is known, we also present an algorithm using $\widetilde{\mathcal{O}}\left(\Delta/\varepsilon \cdot \log^3(H/\mu)\right)$ quantum samples in expectation (Theorem 3.5). A corresponding lower bound is deduced from [55] (Theorem 4.1). We also show (Theorem 4.3) that no such speed-up is possible if we only had access to *copies* of the quantum state representing the distribution.

Our algorithm is based on *sequential analysis*. Given a threshold $b \geq 0$, we will consider the "truncated" mean $\mu_{<b}$ defined by replacing the outcomes larger than b with 0. Using standard techniques, this mean

[1] More precisely, Δ is an upper bound on ϕ/μ where ϕ^2 is the second moment, which satisfies $\sigma/\mu \leq \phi/\mu \leq 1+\sigma/\mu$.

[2] We use the notation $\widetilde{\mathcal{O}}(x)$ to indicate $\mathcal{O}(x \cdot \operatorname{polylog} x)$.

can be encoded in the amplitude of some quantum state $\sqrt{1-\mu_{<b}/b}|\psi\rangle + \sqrt{\mu_{<b}/b}|\psi^{\perp}\rangle$ (Corollary 2.4). We then run the *Amplitude Estimation* algorithm of Brassard et al. [13] on this state for Δ steps (i.e. with Δ quantum samples), only to see whether the estimate of $\mu_{<b}/b$ it returns is nonzero (this is our *stopping rule*). A property of this algorithm (Corollary 2.4 and Remark 2.7) guarantees that it is zero with high probability if and only if the number of quantum samples is below the inverse $\sqrt{b/\mu_{<b}}$ of the estimated amplitude. The crucial observation (Lemma 3.2) is that $\sqrt{b/\mu_{<b}}$ is smaller than Δ for large values of b, and it becomes larger than Δ when $b \approx \mu\Delta^2$. Thus, by repeatedly running the amplitude estimation algorithm with Δ quantum samples, and doing $\mathcal{O}(\log(H/L))$ steps of a logarithmic search on decreasing values of b, the first non-zero value is obtained when b/Δ^2 is approximately equal to μ. The precision of the result is later improved, by using more precise "truncated" means.

This algorithm is extended (Theorem B.1) to cover the common situation where one knows a non-increasing function f such that $f(\mu) \geq 1+\sigma/\mu$, instead of having explicitly $\Delta \geq 1+\sigma/\mu$. For this purpose, we exhibit another property (Corollary 2.4 and Remark 2.6) of the amplitude estimation algorithm, namely that it always outputs a number smaller than the estimated value (up to a constant factor) with high probability. This shall be seen as a quantum equivalent of the Markov inequality. Combined with the previous algorithm, it allows us to find a value $f(\widetilde{\mu}) \geq 1+\sigma/\mu$, with a second logarithmic search on $\widetilde{\mu}$.

Next, we study the quantum analogue of the following standard fact: s classical samples, each taking average time T_{av} to be computed, can be obtained in total average time $s \cdot T_{av}$. The notion of average time is adapted to the quantum setting, using the framework of variable-time algorithms introduced by Ambainis. We develop a variable-time amplitude estimation algorithm (Theorem C.2) that approximates the target value efficiently when some branches of the computation stop earlier than the others. It can be used in place of the standard amplitude estimation in all our results (Theorem C.3).

Applications We describe two applications that illustrate the use of the above results. We first study the problem of approximating the frequency moments F_k of order $k \geq 3$ in the multi-pass streaming model with updates. Classically, the best P-pass algorithms with memory M satisfy $PM = \Theta\left(n^{1-2/k}\right)$ [51, 63]. We give a quantum algorithm for which $P^2M = \widetilde{\mathcal{O}}\left(n^{1-2/k}\right)$ (Theorem 5.3). This problem was studied before in [53], where the author obtained quantum speed-ups for F_0, F_2 and F_∞, but no significant improvement for $k \geq 3$. Similar tradeoff results are known for DISJOINTNESS ($P^2M = \widetilde{\Theta}(n)$ in the quantum streaming model [46] vs. $PM = \Theta(n)$ classically), and DYCK(2) ($P^3M = \Omega(\sqrt{n})$ [54] vs. $PM = \widetilde{\Theta}(\sqrt{n})$ [50, 17, 38]).

Our construction starts with a classical one-pass *linear sketch* streaming algorithm [51, 6] with memory polylog$n$, that samples (approximately) from a distribution with mean F_k and variance $\mathcal{O}\left(n^{1-2/k}F_k^2\right)$. We implement it with a quantum sampler, that needs two passes for one quantum sample. The crucial observation (Appendix D) is that the reverse computation of a linear sketch algorithm can be done efficiently in one pass (whereas usually that would require processing the same stream but in the reverse direction).

As a second application, we study the approximation of graph parameters using neighbor, vertex-pair and degree queries. We show that the numbers m of edges and t of triangles, in an n-vertex graph, can be estimated with $\widetilde{\Theta}\left(n^{1/2}/m^{1/4}\right)$ (Theorem 5.4) and $\widetilde{\Theta}\left(\sqrt{n}/t^{1/6} + m^{3/4}/\sqrt{t}\right)$ (Theorem 5.6) quantum queries respectively. This is a quadratic speed-up over the best classical algorithms [33, 26]. The lower bounds (Theorems 5.5 and 5.7) are obtained with a property testing to communication complexity reduction method.

The number of edges is approximated by translating a classical estimator [58] into a quantum sampler. The triangle counting algorithm is more involved. We need a classical estimator [26] approximating the number t_v of adjacent triangles to any vertex v. Its average running time being small, we obtain a quadratic speed-up for estimating t_v (Proposition E.6) using our mean estimation algorithm for variable-time samplers. We then diverge from the classical triangle counting algorithm of [26], that requires to set up a data structure for sampling edges uniformly in the graph. This technique seems to be an obstacle for a quadratic speed-up. We circumvent this problem by adapting instead a bucketing approach from [25] that partitions the graph's vertices according to the value of t_v. The size of each bucket is estimated using a second quantum sampler.

2 Preliminaries

2.1 Computational model

In this paper we consider probability distributions d on some finite sample spaces $\Omega \subset \mathbb{R}^+$. We denote by $d(x)$ the probability to sample $x \in \Omega$ in the distribution d. We also make the assumption, which is satisfied for most of applications, that Ω is equipped with an efficient encoding of its elements $x \in \Omega$. In particular, we can perform quantum computations on the Hilbert space $\mathcal{H}_\Omega$ defined by the basis $\{|x\rangle\}_{x\in\Omega}$. Moreover, given any two values $0 \leq a < b$, we assume the existence of a unitary $R_{a,b}$ that can perform the *Bernoulli sampling* (see below) in time polylogarithmic in b. In the rest of the paper we will neglect this complexity, including the required precision for implementing any of those unitary operators.

Definition 2.1. *Given a finite space $\Omega \subset \mathbb{R}^+$ and two reals $0 \leq a < b$, an* (a,b)-Bernoulli sampler over Ω *is a unitary $R_{a,b}$ acting on $\mathcal{H}_\Omega \otimes \mathbb{C}^2$ and satisfying for all $x \in \Omega$:*

$$R_{a,b}(|x\rangle|0\rangle) = \begin{cases} |x\rangle\left(\sqrt{1-\frac{x}{b}}|0\rangle + \sqrt{\frac{x}{b}}|1\rangle\right) & \text{when } a \leq x < b, \\ |x\rangle|0\rangle & \text{otherwise.} \end{cases}$$

We say that Ω is Bernoulli samplable *if any (a,b)-Bernoulli sampler can be implemented in polylogarithmic time in b, when a, b have polylog-size encodings in b.*

The $R_{a,b}$ operation can be implemented with a controlled rotation, and is reminiscent of related works on mean estimation (e.g. [62, 12, 52]). In what follows, we always use $a = 0$ or $a = b/2$.

We can now define what a *quantum sample* is.

Definition 2.2. *Given a finite Bernoulli samplable space $\Omega \subset \mathbb{R}^+$ and a distribution d on Ω, a* (quantum) sampler $\mathcal{S}$ *for d is a unitary operator acting on $\mathcal{H}_g \otimes \mathcal{H}_\Omega$, for some Hilbert space $\mathcal{H}_g$, such that*

$$\mathcal{S}(|0\rangle|0\rangle) = \sum_{x\in\Omega} \sqrt{d(x)}|\psi_x\rangle|x\rangle$$

where $|\psi_x\rangle$ are arbitrary unit vectors. A quantum sample *is one execution of $\mathcal{S}$ or $\mathcal{S}^{-1}$ (including their controlled versions). The* output *of $\mathcal{S}$ is the random variable $v(\mathcal{S})$ obtained by measuring the x-register of $\mathcal{S}(|0\rangle|0\rangle)$. Its* mean *is denoted by $\mu_\mathcal{S}$, its* variance *by $\sigma^2_\mathcal{S}$, and its* second moment *by $\phi^2_\mathcal{S} = \mathbb{E}\left[v(\mathcal{S})^2\right]$.*

Given a non-negative random variable X and two numbers $0 \leq a \leq b$, we define the random variable $X_{a,b} = \mathrm{id}_{a,b}(X)$ where $\mathrm{id}_{a,b}(x) = x$ when $a \leq x < b$ and $\mathrm{id}_{a,b}(x) = 0$ otherwise. If $a = 0$, we let $X_{<b} = X_{0,b}$. Similarly, $X_{\geq b} = \mathrm{id}_{\geq b}(X)$ where $\mathrm{id}_{\geq b}(x) = x$ when $x \geq b$ and $\mathrm{id}_{\geq b}(x) = 0$ otherwise.

We motivate the use of a Bernoulli sampler $R_{a,b}$ by the following observation: for any sampler $\mathcal{S}$ and values $0 \leq a < b$, the modified sampler $\hat{\mathcal{S}} = (I_{\mathcal{H}_g} \otimes R_{a,b})(\mathcal{S} \otimes I_{\mathbb{C}^2})$ acting on $\mathcal{H}_{\hat{g}} \otimes \mathcal{H}_{\hat{\Omega}}$, where $\mathcal{H}_{\hat{g}} = \mathcal{H}_g \otimes \mathcal{H}_\Omega$ and $\hat{\Omega} = \{0,1\}$, generates the Bernoulli distribution $d(0) = 1-p$, $d(1) = p$ of mean $p = \mathbb{E}\left[v(\hat{\mathcal{S}})\right] = b^{-1}\mathbb{E}\left[v(\mathcal{S})_{a,b}\right]$ (see the proof of Corollary 2.4). This central result will be used all along this paper.

Other quantum sampling models Instead of having access to the unitary $\mathcal{S}$, one could only have copies of the state $\sum_{x\in\Omega}\sqrt{d(x)}|\psi_x\rangle|x\rangle$ (as in [7] for instance). However, as we show in Theorem 4.3, the speed-up presented in this paper is impossible to achieve in this model. On another note, Aharonov and Ta-Shma [2] studied the *Qsampling* problem, which is the ability to prepare $\sum_{x\in\Omega}\sqrt{d(x)}|x\rangle$ given the decription of a classical circuit with output distribution d. This problem becomes straightforward if a garbage register ψ_x can be added (using standard reversible-computation techniques). Bravyi, Harrow and Hassidim [14] considered an oracle-based model, that is provably weaker than Qsampling, where a distribution $d = (d(1), \ldots, d(N))$ on $\Omega = [N]$ is represented by an oracle $O_d : [S] \to [N]$ (for some S), such that $d(x)$ equals the proportion of inputs $s \in [S]$ with $O_d(s) = x$. It is extended to the quantum query

framework with a unitary $\mathcal{O}_d$ such that $\mathcal{O}_d|s\rangle|0\rangle = |s\rangle|O_d(s)\rangle$. It is not difficult to see that applying $\mathcal{O}_d$ on a uniform superposition gives $\sum_{x\in[N]} \sqrt{d(x)} \left(\frac{1}{\sqrt{d(x)S}} \sum_{s\in[S]:O_d(s)=x} |s\rangle \right) |x\rangle$, as required by Definition 2.2 (where $|\psi_x\rangle = \frac{1}{\sqrt{d(x)S}} \sum_{s\in[S]:O_d(s)=x} |s\rangle$). Finally, Montanaro [52] presented a model that is similar to ours, where he replaced the x-register of $\mathcal{S}(|0\rangle|0\rangle)$ with a k-qubit register (for some k) combined with a mapping $\phi : \{0,1\}^k \to \Omega$ where $x = \phi(s)$ is the sample associated to each $s \in \{0,1\}^k$.

2.2 Amplitude estimation

The essential building block of this paper is the amplitude estimation algorithm [13], combined with ideas from [62, 12, 52], to estimate the modified mean $b^{-1}\mathbb{E}[v(\mathcal{S})_{a,b}]$ of a quantum sampler $\mathcal{S}$ to which a Bernoulli sampler $R_{a,b}$ has been applied. We will need the following result about amplitude estimation.

Theorem 2.3. *There is a quantum algorithm* AmplEst*, called* Amplitude Estimation*, that takes as input a unitary operator U, an orthogonal projector Π, and an integer $t > 2$. The algorithm outputs an estimate $\widetilde{p} = \mathsf{AmplEst}(U,\Pi,t)$ of $p = \langle\psi|\Pi|\psi\rangle$, where $|\psi\rangle = U|0\rangle$, such that*

$$\begin{cases} |\widetilde{p} - p| \le 2\pi\frac{\sqrt{p}}{t} + \frac{\pi^2}{t^2}, & \text{with probability } 8/\pi^2; \\ \widetilde{p} = 0, & \text{with probability } \frac{\sin^2(t\theta)}{t^2\sin^2(\theta)}. \end{cases}$$

and $0 \le \theta \le \pi/2$ satisfies $\sin(\theta) = \sqrt{p}$. It uses $\mathcal{O}\left(\log^2(t)\right)$ 2-qubit quantum gates (independent of U and Π) and makes $2t+1$ calls to (the controlled versions of) U and U^{-1}, and t calls to the reflection $I - 2\Pi$.

We now present an adaptation of the algorithms from [62, 12, 52] for estimating $b^{-1}\mathbb{E}[v(\mathcal{S})_{a,b}]$.

Input: a sampler $\mathcal{S}$ acting on $\mathcal{H}_g \otimes \mathcal{H}_\Omega$, two values (a,b), an integer t, a failure parameter $0 < \delta < 1$.
Output: an estimate $\widetilde{p} = \mathsf{BasicEst}(\mathcal{S},(a,b),t,\delta)$ of $p = b^{-1}\mathbb{E}\left[v(\mathcal{S})_{a,b}\right]$

1. Let $U = (I_{\mathcal{H}_g} \otimes R_{a,b})(\mathcal{S} \otimes I_{\mathbb{C}^2})$ and $\Pi = I_{\mathcal{H}_g} \otimes I_{\mathcal{H}_\Omega} \otimes |1\rangle\langle 1|$.
2. For $i = 1,\ldots,\Theta(\log(1/\delta))$: compute $\widetilde{p}_i = \mathsf{AmplEst}(U,\Pi,t)$.
3. Output $\widetilde{p} = \text{median}\{\widetilde{p}_1,\ldots,\widetilde{p}_{\Theta(\log(1/\delta))}\}$.

Algorithm 1: the Basic Estimation algorithm BasicEst.

Corollary 2.4. *Consider a quantum sampler $\mathcal{S}$ and two values $0 \le a < b$. Denote $p = b^{-1}\mathbb{E}[v(\mathcal{S})_{a,b}]$. Given an integer $t > 2$ and a real $0 < \delta < 1$, $\mathsf{BasicEst}(\mathcal{S},(a,b),t,\delta)$ (see Algorithm 1) uses $\mathcal{O}(t\log(1/\delta))$ quantum samples and outputs $\widetilde{p}$ satisfying all of the following inequalities with probability $1-\delta$:*

(1) $|\widetilde{p} - p| \le 2\pi\frac{\sqrt{p}}{t} + \frac{\pi^2}{t^2}$, *for any t;* (2) $\widetilde{p} \le (1+2\pi)^2 \cdot p$, *for any t;*

(3) $\widetilde{p} = 0$, *when $t < \frac{1}{2\sqrt{p}}$;* (4) $|\widetilde{p} - p| \le \varepsilon \cdot p$, *when $t \ge \frac{8}{\varepsilon\sqrt{p}}$ and $0 < \varepsilon < 1$.*

Proof. We show that each $\widetilde{p}_i$ satisfies the inequalities stated in the corollary, with probability $8/\pi^2$. Since $\widetilde{p}$ is the median of $\Theta(\log 1/\delta)$ such values, the probability is increased to $1-\delta$ using the Chernoff bound.

For each $x \in \Omega$, denote $\nu_x = \frac{x}{b}$ if $a \le x < b$, and $\nu_x = 0$ otherwise. Since $p = \sum_{x\in\Omega} \nu_x d(x)$, observe that

$$U(|0\rangle|0\rangle|0\rangle) = \sum_{x\in\Omega} \sqrt{d(x)}|\psi_x\rangle|x\rangle \left(\sqrt{1-\nu_x}|0\rangle + \sqrt{\nu_x}|1\rangle\right) = \sqrt{1-p}|\psi_0'\rangle|0\rangle + \sqrt{p}|\psi_1'\rangle|1\rangle$$

where $|\psi_0'\rangle = \frac{1}{\sqrt{1-p}} \sum_{x\in\Omega} \sqrt{d(x)}\sqrt{1-\nu_x}|\psi_x\rangle|x\rangle$ and $|\psi_1'\rangle = \frac{1}{\sqrt{p}} \sum_{x\in\Omega} \sqrt{d(x)}\sqrt{\nu_x}|\psi_x\rangle|x\rangle$ are unit vectors. Thus, the output $\widetilde{p}_i$ of the AmplEst algorithm applied on U and Π is an estimate of p satisfying the output conditions of Theorem 2.3. Therefore $|\widetilde{p}_i - p| \le 2\pi\frac{\sqrt{p}}{t} + \frac{\pi^2}{t^2}$ with probability $8/\pi^2$, for any t. By plugging $t \ge \frac{8}{\varepsilon\sqrt{p}}$ into this inequality we have $|\widetilde{p}_i - p| \le \varepsilon \cdot p$. By plugging $t \ge \frac{1}{2\sqrt{p}}$ we also have

$|\widetilde{p}_i - p| \leq (4\pi + 4\pi^2)p$, and thus $\widetilde{p}_i \leq (1+2\pi)^2 \cdot p$. Finally, if $t < \frac{1}{2\sqrt{p}}$, denote $0 \leq \theta \leq \pi/2$ such that $\sin(\theta) = \sqrt{p}$ and observe that $\theta \leq \frac{\pi}{2}\sqrt{p} \leq \frac{\pi}{4t}$ (since $\frac{2}{\pi}x \leq \sin(x) \leq x$, for $x \in [0, \pi/2]$). The probability to obtain $\widetilde{p}_i = 0$ is $\frac{\sin^2(t\theta)}{t^2\sin^2(\theta)} \geq \frac{\sin^2(t\pi/(4t))}{t^2\sin^2(\pi/(4t))} \geq \frac{\sin^2(\pi/4)}{t^2(\pi/(4t))^2} = 8/\pi^2$, since $x \mapsto \sin^2(tx)/(t^2\sin^2(x))$ is decreasing for $0 < x \leq \pi/t$. Moreover, when $t < \frac{1}{2\sqrt{p}}$, the first two inequalities are obviously satisfied if $\widetilde{p}_i = 0$. □

The four results on p in Corollary 2.4 lie at the heart of this paper. We make a few comments on them.

Remark 2.5. *Consider a sampler $\mathcal{S}$ over $\Omega = \{0,1\}$ for the Bernoulli distribution of parameter p. Using the Chebyshev inequality, we get that $\mathcal{O}\left((1-p)/(\varepsilon^2 p)\right)$ classical samples are enough for estimating p with relative error ε. The inequality (4) of Corollary 2.4 shows that $t = \mathcal{O}\left(1/(\varepsilon\sqrt{p})\right)$ quantum samples are sufficient. Our main result (Section 3) generalizes this quadratic speed-up to the non-Bernoulli case.*

Remark 2.6. *The inequality (2) shall be seen as an equivalent of the Markov inequality[3], namely that $\widetilde{p}$ does not exceed p by a large factor with large probability. This property will be used in Appendix B.*

Remark 2.7. *If $p \neq 0$, inequalities (3) and (4) imply that, with large probability, $t < 8/\sqrt{p}$ when $\widetilde{p} = 0$, and $t \geq 1/(2\sqrt{p})$ when $\widetilde{p} \neq 0$. This phenomenon, at $t = \Theta(1/\sqrt{p})$, is crucially used in the next section.*

3 Quantum Chebyshev's inequality

We describe our main algorithm for estimating the mean $\mu_{\mathcal{S}}$ of any quantum sampler $\mathcal{S}$, given an upper bound $\Delta_{\mathcal{S}} \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$ (we recall that $\phi_{\mathcal{S}}^2 = \mathbb{E}\left[v(\mathcal{S})^2\right]$ and $\sigma_{\mathcal{S}}/\mu_{\mathcal{S}} \leq \phi_{\mathcal{S}}/\mu_{\mathcal{S}} \leq 1 + \sigma_{\mathcal{S}}/\mu_{\mathcal{S}}$). The two main tools used in this section are the BasicEst algorithm of Corollary 2.4, and the following lemma on "truncated" means. We recall that $X_{<b}$ (resp. $X_{\geq b}$) is defined from a non-negative random variable X by substituting the outcomes greater or equal to b (resp. less than b) with 0. Note that $X = X_{<b} + X_{\geq b}$ for all $b > 0$.

Fact 3.1. *For any random variable X and numbers $0 < a \leq b$, we have $\mathbb{E}[X_{a,b}] \leq \frac{\mathbb{E}\left[X_{a,b}^2\right]}{a}$ and $\mathbb{E}[X_{\geq b}] \leq \frac{\mathbb{E}\left[X_{\geq b}^2\right]}{b}$.*

Lemma 3.2. *Let X be a non-negative random variable and $\Delta \geq \sqrt{\mathbb{E}[X^2]}/\mathbb{E}[X]$. Then, for all $c_1, c_2, M > 0$ such that $c_1 \cdot \mathbb{E}[X] \leq M \leq c_2 \cdot \mathbb{E}[X]$, we have*

$$\left(1 - \frac{1}{c_1}\right) \cdot \mathbb{E}[X] \leq \mathbb{E}[X_{<M\Delta^2}] \leq \mathbb{E}[X] \qquad \text{and} \qquad \sqrt{c_1} \cdot \Delta \leq \frac{1}{\sqrt{\mathbb{E}[X_{<M\Delta^2}]/(M\Delta^2)}} \leq \sqrt{c_2\left(1 - \frac{1}{c_1}\right)} \cdot \Delta$$

Proof. The left hand side term is a consequence of $\mathbb{E}[X_{<M\Delta^2}] = \mathbb{E}[X] - \mathbb{E}\left[X_{\geq M\Delta^2}\right]$ and $0 \leq \mathbb{E}\left[X_{\geq M\Delta^2}\right] \leq \mathbb{E}\left[X_{\geq M\Delta^2}^2\right]/(M\Delta^2) \leq \mathbb{E}\left[X^2\right]/(M\Delta^2) \leq (1/c_1) \cdot \mathbb{E}[X]$ (using Fact 3.1). The right hand side term is a direct consequence of the left one, and of the hypothesis $c_1 \cdot \mathbb{E}[X] \leq M \leq c_2 \cdot \mathbb{E}[X]$. □

Our mean estimation algorithm works in two stages. We first compute a rough estimate $M \in [2\mu_{\mathcal{S}}, 2500\mu_{\mathcal{S}}]$ with $\widetilde{\mathcal{O}}(\Delta_{\mathcal{S}} \cdot \log(H/L))$ quantum samples (where $0 < L < \mu_{\mathcal{S}} < H$ are known bounds on $\mu_{\mathcal{S}}$). Then, we improve the accuracy of the estimate to any value ε, at extra cost $\widetilde{\mathcal{O}}\left(\Delta_{\mathcal{S}}/\varepsilon^{3/2}\right)$.

[3] The Markov inequality for a non-negative random variable X states that $\mathbb{P}(X \geq k\mathbb{E}[X]) \leq 1/k$ for any $k > 0$. Here, although we do not need this result, it is possible to prove that $\mathbb{P}(\widetilde{p} \geq kp) \leq C/\sqrt{k}$, for some absolute constant C.

Input: a sampler $\mathcal{S}$, an integer $\Delta_{\mathcal{S}}$, two values $0 < L < H$, two reals $0 < \varepsilon, \delta < 1/2$.
Output: an estimate $\widetilde{\mu}_{\mathcal{S}}$ of $\mu_{\mathcal{S}}$.

1. Set $M = 8H$ and $\widetilde{p} = 0$
2. While $\widetilde{p} = 0$ and $M \geq 2L$:
   (a) Set $M = M/2$.
   (b) Compute $\widetilde{p} = \mathsf{BasicEst}\left(\mathcal{S}, (0, M\Delta_{\mathcal{S}}^2), 25\Delta_{\mathcal{S}}, \delta'\right)$ where $\delta' = \frac{\delta}{2(3+\log(H/L))}$.
3. If $M < 2L$ then output $\widetilde{\mu}_{\mathcal{S}} = 0$.
4. Else, compute $\widetilde{q} = \mathsf{BasicEst}\left(\mathcal{S}, (0, \varepsilon^{-1}M\Delta_{\mathcal{S}}^2), 35^2\varepsilon^{-3/2}\Delta_{\mathcal{S}}, \delta/2\right)$ and output $\widetilde{\mu}_{\mathcal{S}} = (\varepsilon^{-1}M\Delta_{\mathcal{S}}^2) \cdot \widetilde{q}$.

Algorithm 2: $\varepsilon-$approximation of the mean of a quantum sampler $\mathcal{S}$.

Theorem 3.3. *If $\Delta_{\mathcal{S}} \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$ and $L < \mu_{\mathcal{S}} < H$ then the output $\widetilde{\mu}_{\mathcal{S}}$ of Algorithm 2 satisfies $|\widetilde{\mu}_{\mathcal{S}} - \mu_{\mathcal{S}}| \leq \varepsilon\mu_{\mathcal{S}}$ with probability $1-\delta$. Moreover, for any $\Delta_{\mathcal{S}}, L, H$ it satisfies $\widetilde{\mu}_{\mathcal{S}} \leq (1+2\pi)^2\mu_{\mathcal{S}}$ with probability $1-\delta$. The number of quantum samples used by the algorithm is $\mathcal{O}\left(\Delta_{\mathcal{S}} \cdot \left(\log\left(\frac{H}{L}\right)\log\left(\frac{\log(H/L)}{\delta}\right) + \varepsilon^{-3/2}\log\left(\frac{1}{\delta}\right)\right)\right)$.*

Proof. Assume that $\Delta_{\mathcal{S}} \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$ and $L < \mu_{\mathcal{S}} < H$. We denote $p = (M\Delta_{\mathcal{S}}^2)^{-1} \cdot \mathbb{E}\left[v(\mathcal{S})_{<M\Delta_{\mathcal{S}}^2}\right]$. By Lemma 3.2, if $M \geq 2500\mu_{\mathcal{S}}$ then $25\Delta_{\mathcal{S}} \leq \frac{1}{2\sqrt{p}}$, and if $2\mu_{\mathcal{S}} \leq M \leq 4\mu_{\mathcal{S}}$ then $25\Delta_{\mathcal{S}} > \frac{8}{\sqrt{p}}$. Therefore, by Corollary 2.4, with probability $1-\delta'$, the value $\widetilde{p}$ computed at Step 2.(b) is equal to 0 when $M \geq 2500\mu_{\mathcal{S}}$, and is different from 0 when $2\mu_{\mathcal{S}} \leq M \leq 4\mu_{\mathcal{S}}$. Thus, the first time Step 2.(b) of Algorithm 2 computes $\widetilde{p} \neq 0$ happens for $M \in [2\mu_{\mathcal{S}}, 2500\mu_{\mathcal{S}}]$, with probability at least $(1-\delta')^{1+\log(4H/(2\mu_{\mathcal{S}}))} > 1-\delta/2$.

Consequently, we can assume that Step 4 is executed with $M \in [2\mu_{\mathcal{S}}, 2500\mu_{\mathcal{S}}]$, and we let $M' = M/\varepsilon$. According to Lemma 3.2 we have $(1-\varepsilon/2)\mu_{\mathcal{S}} \leq \mathbb{E}\left[v(\mathcal{S})_{<M'\Delta_{\mathcal{S}}^2}\right] \leq \mu_{\mathcal{S}}$ and $35^2\varepsilon^{-3/2}\Delta_{\mathcal{S}} \geq \frac{8}{(\varepsilon/2)\sqrt{q}}$, where $q = (M'\Delta_{\mathcal{S}}^2)^{-1} \cdot \mathbb{E}\left[v(\mathcal{S})_{<M'\Delta_{\mathcal{S}}^2}\right]$. Thus, according to Corollary 2.4, the value $\widetilde{q}$ satisfies $|\widetilde{q} - q| \leq (\varepsilon/2)q$ with probability $1-\delta/2$. Using the triangle inequality, it implies $|(\varepsilon^{-1}M\Delta_{\mathcal{S}}^2) \cdot \widetilde{q} - \mu_{\mathcal{S}}| \leq \varepsilon\mu_{\mathcal{S}}$.

If $L \geq \mu_{\mathcal{S}}$, this may only increase the probability to stop at Step 3 and output $\widetilde{\mu}_{\mathcal{S}} = 0$. If Step 4 is executed, we still have $\widetilde{\mu}_{\mathcal{S}} \leq (1+2\pi)^2\mu_{\mathcal{S}}$ with probability $1-\delta$, as a consequence of Corollary 2.4. □

Remark 3.4. *If $\Delta_{\mathcal{S}} \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$ and $H > \mu_{\mathcal{S}}$, observe that the output of Algorithm 2 satisfies $\widetilde{\mu}_{\mathcal{S}} = 0$ when $L \geq 1250\mu_{\mathcal{S}}$ and $\widetilde{\mu}_{\mathcal{S}} \neq 0$ when $L < \mu_{\mathcal{S}}$, with probability $1-\delta$.*

We show in Appendix A (Algorithm 5) how to modify the last step of Algorithm 2 so that it uses $\widetilde{\mathcal{O}}\left(\Delta_{\mathcal{S}} \cdot \varepsilon^{-1}\log(1/\delta)\right)$ quantum samples only (Theorem A.2). Using Remark 3.4, we also remove the input parameter L while keeping the number of quantum samples small in expectation (Algorithm 6). Altogether, it leads to the following result.

Theorem 3.5. *There is an algorithm that, given a sampler $\mathcal{S}$, an integer $\Delta_{\mathcal{S}}$, a value $H > 0$, and two reals $0 < \varepsilon, \delta < 1$, outputs an estimate $\widetilde{\mu}_{\mathcal{S}}$. If $\Delta_{\mathcal{S}} \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$ and $H > \mu_{\mathcal{S}}$, it satisfies $|\widetilde{\mu}_{\mathcal{S}} - \mu_{\mathcal{S}}| \leq \varepsilon\mu_{\mathcal{S}}$ with probability $1-\delta$, and the algorithm uses $\widetilde{\mathcal{O}}\left(\Delta_{\mathcal{S}} \cdot \varepsilon^{-1}\log^3(H/\mu_{\mathcal{S}})\log(1/\delta)\right)$ quantum samples in expectation.*

In Section 4, we describe an $\Omega((\Delta_{\mathcal{S}}-1)/\varepsilon)$ lower bound for this mean estimation problem. Before, we present three kinds of generalizations of the above algorithms.

- **Higher moments.** Given an upper-bound $\Delta_{\mathcal{S}}^2 \geq \left(\mathbb{E}\left[v(\mathcal{S})^k\right]/\mathbb{E}\left[v(\mathcal{S})\right]^k\right)^{1/(k-1)}$ on the relative moment of order $k \geq 2$, one can easily generalize Facts 3.1, Lemma 3.2 and Theorem A.2 to show that $\mu_{\mathcal{S}}$ can be estimated using $\widetilde{\mathcal{O}}\left(\Delta_{\mathcal{S}} \cdot \varepsilon^{-1/(2(k-1))}\log(H/L)\log(1/\delta)\right)$ quantum samples.
- **Implicit upper bound on $\phi_{\mathcal{S}}/\mu_{\mathcal{S}}$.** If instead of an explicit value $\Delta_{\mathcal{S}} \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$ we are given a non-increasing function f such that $f(\mu_{\mathcal{S}}) \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$, we can still estimate the mean $\mu_{\mathcal{S}}$ using $\widetilde{\mathcal{O}}\left(f(\mu_{\mathcal{S}}/c) \cdot \varepsilon^{-1}\log(H/L)\log(1/\delta)\right)$ quantum samples, where $c > 1$ is an absolute constant (Algorithm 7 in Appendix B). The proof crucially uses the Markov-like inequality "$\widetilde{\mu}_{\mathcal{S}} \leq (1+2\pi)^2\mu_{\mathcal{S}}$" of Corollary 2.4.

- **Time complexity and variable-time samplers.** The *time complexity* (number of quantum gates) of all above algorithms is essentially equal to the number of quantum samples multiplied by the time complexity $T_{max}(\mathcal{S})$ of the considered sampler. Often, this last quantity is much larger than the more desirable *ℓ_2-average running time* $T_{\ell_2}(\mathcal{S})$ defined by Ambainis [5] in the context of *variable-time amplitude amplification*. In Appendix C, we develop a new *variable-time amplitude estimation* algorithm (Theorem C.2), and we use it into our above algorithm to show that $\mu_{\mathcal{S}}$ can be estimated *in time* $\widetilde{\mathcal{O}}\left(\Delta_{\mathcal{S}} \cdot \varepsilon^{-2} T_{\ell_2}(\mathcal{S}) \cdot \log^4(T_{max}(\mathcal{S})) \log(H/L) \log(1/\delta)\right)$ (Theorem C.3).

The last two results are combined together in Section 5.2 and Appendix E.2 to describe an optimal quantum query algorithm that approximates the number of triangles in any graph.

4 Optimality and separation results

Using a result due to Nayak and Wu [55] on approximate counting, we can show a corresponding lower bound to Theorem 3.5 already in the simple case of Bernoulli variables. For this purpose, we define that an algorithm $\mathcal{A}$ solves the *Mean Estimation problem for parameters* ε, Δ if, for any sampler $\mathcal{S}$ satisfying $\phi_{\mathcal{S}}/\mu_{\mathcal{S}} \in [\Delta, 4\Delta]$ (the constant 4 is arbitrary), it outputs a value $\widetilde{\mu}_{\mathcal{S}}$ satisfying $|\widetilde{\mu}_{\mathcal{S}} - \mu_{\mathcal{S}}| \leq \varepsilon \mu_{\mathcal{S}}$ with probability $2/3$.

Theorem 4.1. *Any algorithm solving the Mean Estimation problem for parameters $0 < \varepsilon < 1/5$ and $\Delta > 1$ on the sample space $\Omega = \{0,1\}$ must use $\Omega((\Delta-1)/\varepsilon)$ quantum samples.*

Proof. Consider an algorithm $\mathcal{A}$ solving the Mean Estimation problem for parameters $0 < \varepsilon < 1/5$, $\Delta > 1$ using N quantum samples. Take two integers $0 < t < n$ large enough such that $\sqrt{2}\Delta \leq \sqrt{n/t} \leq 4\Delta$ and $\varepsilon t > 1$. For any oracle $\mathcal{O} : \{1, \ldots, n\} \to \{0,1\}$, define the quantum sampler $\mathcal{S}_{\mathcal{O}}(|0\rangle|0\rangle) = \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i\rangle|\mathcal{O}(i)\rangle$ and let $t_{\mathcal{O}} = |\{i \in [n] : \mathcal{O}(i) = 1\}|$. Observe that $\mu_{\mathcal{S}_{\mathcal{O}}} = \phi^2_{\mathcal{S}_{\mathcal{O}}} = t_{\mathcal{O}}/n$, and one quantum sample from $\mathcal{S}_{\mathcal{O}}$ can be implemented with one quantum query to $\mathcal{O}$.

According to [55, Corollary 1.2], any algorithm that can distinguish $t_{\mathcal{O}} = t$ from $t_{\mathcal{O}} = \lceil (1+4\varepsilon)t \rceil$ makes $\Omega\left(\sqrt{n/(\varepsilon t)} + \sqrt{t(n-t)}/(\varepsilon t)\right) = \Omega\left((\sqrt{n/t} - 1)/\varepsilon\right) = \Omega((\Delta-1)/\varepsilon)$ quantum queries to $\mathcal{O}$. However, given the promise that $t_{\mathcal{O}} = t$ or $t_{\mathcal{O}} = \lceil (1+4\varepsilon)t \rceil$ we can use $\mathcal{A}$ with input $\mathcal{S}_{\mathcal{O}}$, ε, Δ to distinguish between the two cases using N samples, that is N queries to $\mathcal{O}$. Indeed, $\phi_{\mathcal{S}_{\mathcal{O}}}/\mu_{\mathcal{S}_{\mathcal{O}}} = \sqrt{n/t_{\mathcal{O}}} \in [\Delta, 4\Delta]$ for such samplers (since $\lceil (1+4\varepsilon)t \rceil \leq (1+5\varepsilon)t \leq 2t$). Thus, $\mathcal{A}$ must use $N = \Omega((\Delta-1)/\varepsilon)$ quantum samples. □

One may wonder whether the quantum speed-up presented in this paper holds if we only have access to copies of a quantum state $\sum_{x \in \Omega} \sqrt{d(x)} |\psi_x\rangle|x\rangle$ (instead of access to a unitary $\mathcal{S}$ preparing it). Below we answer this question negatively. For this purpose, we define that an algorithm $\mathcal{A}$ solves the *state-based Mean Estimation problem for parameters* ε, Δ if, using access to some copies of an unknown state $|d\rangle = \sum_{x \in \Omega} \sqrt{d(x)}|x\rangle$ satisfying $\phi_d/\mu_d \in [\Delta, 4\Delta]$ (where $\mu_d = \sum_x d(x)x$ and $\phi_d^2 = \sum_x d(x)x^2$), it outputs a value $\widetilde{\mu}_d$ satisfying $|\widetilde{\mu}_d - \mu_d| \leq \varepsilon \mu_d$ with probability $2/3$.

Lemma 4.2. *Consider two distributions d, d' represented by the quantum states $|d\rangle = \sum_{x \in \Omega} \sqrt{d(x)}|x\rangle$ and $|d'\rangle = \sum_{x \in \Omega} \sqrt{d'(x)}|x\rangle$. The smallest integer T needed to be able to discriminate $|d\rangle^{\otimes T}$ and $|d'\rangle^{\otimes T}$ with success probability $2/3$ satisfies $T \geq \frac{\ln(9/8)}{D(d||d')}$, where $D(d||d')$ is the KL-divergence from d to d'.*

Proof. According to Helstrom's bound [37] the best success probability to discriminate two states $|\psi\rangle$ and $|\phi\rangle$ is $\frac{1}{2}(1 + \sqrt{1 - |\langle\psi|\phi\rangle|^2})$. Consequently, T must satisfy $\frac{1}{2}(1 + \sqrt{1 - \langle d|d'\rangle^{2T}}) \geq 2/3$, which implies

$$T \geq \frac{\ln(9/8)}{-\ln(\langle d|d'\rangle^2)} = \frac{\ln(9/8)}{-2\ln\left(\sum_x d(x)\sqrt{d'(x)/d(x)}\right)} \geq \frac{\ln(9/8)}{\sum_x d(x)\ln\left(d(x)/d'(x)\right)} = \frac{\ln(9/8)}{D(d||d')}$$

where we used the concavity of the $-\ln$ function. □

Theorem 4.3. *Any algorithm solving the state-based Mean Estimation problem for parameters $0 < \varepsilon < 1/100$ and $\Delta > 1$ on the sample space $\Omega = \{0,1\}$ must use $\Omega\left((\Delta^2-1)/\varepsilon^2\right)$ copies of the input state.*

Proof. Consider an algorithm $\mathcal{A}$ solving the state-based Mean Estimation problem for parameters $0 < \varepsilon < 1/100$, $\Delta > 1$ using N copies of the input state. Given any $|d\rangle = \sqrt{1-p}|0\rangle + \sqrt{p}|1\rangle$ with $\phi_d/\mu_d \in [\sqrt{6}\Delta, \sqrt{8}\Delta]$ (notice that $\mu_d = \phi_d^2 = p$ and $1 - p \geq 5/6 \geq 12\varepsilon$), we show how to construct a state $|d'\rangle = \sqrt{1-p'}|0\rangle + \sqrt{p'}|1\rangle$ such that

$$(1)\ (1+4\varepsilon)\mu_d < \mu_{d'} < (1+24\varepsilon)\mu_d\ ; \qquad (2)\ \phi_{d'}/\mu_{d'} \in [\Delta, 4\Delta]\ ; \qquad (3)\ D(d||d') \leq (12\varepsilon)^2/(\Delta^2-1).$$

It is clear that $\mathcal{A}$ can be used to discriminate two such states. On the other hand, according to Lemma 4.2, any such algorithm muse use $N = \Omega\left(1/D(d||d')\right) = \Omega\left((\Delta^2-1)/\varepsilon^2\right)$ copies of the input state.

The construction of d' is adapted from [22, Section 7]. We set $p' = pe^{\alpha(1-p)}/\psi$ where $\alpha = 12\varepsilon/(1-p) < 1$ and $\psi = (1-p)e^{-\alpha p} + pe^{\alpha(1-p)}$ (so that $1 - p' = (1-p)e^{-\alpha p}/\psi$). We let $\dot{\psi}$ (resp. $\ddot{\psi}$) denote the first (resp. second) derivative of ψ with respect to α. A simple calculation shows that $\mu_{d'} - \mu_d = \dot{\psi}/\psi$ and $D(d||d') = \ln\psi$. Moreover, $\sigma_{d'}^2 = \mathbb{E}_{x\sim d'}\left[(x-\mu_{d'})^2\right] = \mathbb{E}_{x\sim d'}\left[(x-\mu_d)^2\right] + 2(\mu_d - \mu_{d'})\mathbb{E}_{x\sim d'}\left[x-\mu_d\right] + (\mu_d - \mu_{d'})^2 = \mathbb{E}_{x\sim d}\left[(x-p)^2 e^{\alpha(x-p)-\ln\psi}\right] - (\mu_d - \mu_{d'})^2 = \ddot{\psi}/\psi - (\dot{\psi}/\psi)^2$.

Since $\psi = \mathbb{E}_{x\sim d}\left[e^{\alpha(x-p)}\right]$, it can be deduced from the standard inequality $1 + u + u^2/3 \leq e^u \leq 1 + u + u^2$ (when $|u| \leq 1$) that $1 \leq 1 + \frac{p(1-p)}{3}\cdot\alpha^2 \leq \psi \leq 1 + p(1-p)\cdot\alpha^2 \leq 2$. Consequently, $\frac{2p(1-p)}{3}\cdot\alpha \leq \dot{\psi} \leq 2p(1-p)\cdot\alpha$ and $\frac{2p(1-p)}{3} \leq \ddot{\psi} \leq 2p(1-p)$. It implies that $4\varepsilon p \leq \mu_{d'} - \mu_d \leq 24\varepsilon p$ and $p(1-p)/3 - (24\varepsilon p)^2 \leq \sigma_{d'}^2 \leq 2p(1-p)$. Thus, $(1+4\varepsilon)\mu_d \leq \mu_{d'} \leq (1+24\varepsilon)\mu_d \leq \sqrt{2}\mu_d$ and $\frac{1}{6}\sigma_d^2/\mu_d^2 - (24\varepsilon/\sqrt{2})^2 \leq \sigma_{d'}^2/\mu_{d'}^2 \leq 2\sigma_d^2/\mu_d^2$. Since $\sigma_{d'}^2/\mu_{d'}^2 = \phi_{d'}^2/\mu_{d'}^2 - 1$ and $\phi_d/\mu_d \in [\sqrt{6}\Delta, \sqrt{8}\Delta]$, we obtain that $\Delta \leq \frac{1}{\sqrt{6}}\phi_d/\mu_d \leq \phi_{d'}/\mu_{d'} \leq \sqrt{2}\phi_d/\mu_d \leq 4\Delta$. Finally, $D(d||d') = \ln\psi \leq p(1-p)\cdot\alpha^2 = (12\varepsilon)^2 p/(1-p) \leq (12\varepsilon)^2/(\Delta^2-1)$. □

Remark 4.4. *An intermediate version of Theorem 4.1 can be deduced from Theorem 4.3, when $\mathcal{S}$ is accessed via the* reflection oracle $\mathcal{O}_{\mathcal{S}} = I - 2\mathcal{S}(|0\rangle|0\rangle)(\langle 0|\langle 0|)\mathcal{S}^{-1}$ *only (observe that this is the case for our algorithms). Indeed, according to [42, Theorem 4], for any algorithm performing q queries to a reflection oracle $\mathcal{O} = I - 2|\phi\rangle\langle\phi|$, it is possible to remove the queries to $\mathcal{O}$ by using $\sim q^2$ copies of $|\phi\rangle$ instead.*

5 Applications

We describe two applications of the Quantum Chebyshev Inequality. The first one (Section 5.1) concerns the computation of the frequency moments F_k of order $k \geq 3$ in the streaming model. We design a P-pass algorithm with quantum memory M satisfying a tradeoff of $P^2M = \widetilde{\mathcal{O}}\left(n^{1-2/k}\right)$, whereas the best algorithm with classical memory requires $PM = \Theta(n^{1-2/k})$. We then study (Section 5.2) the edge and triangle counting problems in the general graph model with quantum query access. We describe nearly optimal algorithms that approximate these parameters quadratically faster than in the classical query model.

5.1 Frequency moments in the multi-pass streaming model

In the streaming model with update (*turnstile model*), the input is a vector $x \in \mathbb{R}^n$ obtained through a stream $\vec{u} = u_1, u_2, \ldots$ of updates. Initially, $x(0) = (0,\ldots,0)$, and each $u_j = (i,\lambda) \in [n]\times\mathbb{R}$ modifies the i-th coordinate of $x(j)$ by adding λ to it. The goal of a *streaming algorithm* $\mathcal{T}$ is to output, at the end of the stream, some function of the final vector x while minimizing the number $M \ll n$ of memory cells. In the *multi-pass* model, the same stream is repeated for a certain number P of passes, before the algorithm outputs its result.

The *frequency moment of order k* is defined, for the final vector $x = (x_1,\ldots,x_n)$, as $F_k(x) = \sum_{i\in[n]}|x_i|^k$. The problem of approximating F_k when $k \geq 3$ has been addressed first with the AMS algorithm [4], that uses $\mathcal{O}\left(n^{1-1/k}\right)$ classical memory cells in the insertion-only model (where $u_j \in [n]\times\mathbb{R}^+$). A series of works in the turnstile model culminated in optimal one-pass algorithms with memory $\Theta\left(n^{1-2/k}\right)$ [49, 31], and nearly optimal P-pass algorithms with memory $\widetilde{\Theta}\left(n^{1-2/k}/P\right)$ [51, 6, 63]. In the quantum setting, Montanaro [53] obtained a small improvement in terms of the approximation parameter ε only.

Our algorithm relies on a classical procedure for ℓ_2 *sampling*. Given $x \in \mathbb{R}^n$, we let $D_{q,x}$ denotes the ℓ_q *distribution* that returns $i \in [n]$ with probability $\frac{|x_i|^q}{F_q(x)}$. One can observe that the (suboptimal) AMS algorithm [4] essentially samples $i \sim D_{1,x}$ and computes $F_1 \cdot |x_i|^{k-1}$. This is an unbiased estimator for $F_k(x)$ with variance $\mathcal{O}\left(n^{1-1/k}F_k(x)^2\right)$ (thus requiring to compute $\mathcal{O}\left(n^{1-1/k}\right)$ samples in one pass). Instead, we base our algorithm on the estimator $F_2(x) \cdot |x_i|^{k-2}$ where $i \sim D_{2,x}$. It reduces the variance to $\mathcal{O}\left(n^{1-2/k}F_k(x)^2\right)$ [51], but it requires a procedure for ℓ_2 sampling. To this end, we use the following algorithm from [6] to sample from an (ε,δ)-*approximator* to $D_{2,x}$ (meaning that each $i \in [n]$ is sampled with a probability p_i satisfying $(1-\varepsilon)\frac{|x_i|^2}{F_2(x)} - \delta \leq p_i \leq (1+\varepsilon)\frac{|x_i|^2}{F_2(x)} + \delta$).

Theorem 5.1 ([6])**.** *There is a randomized streaming algorithm that, given a stream $\vec{u}$ with final vector x, a real $0 < \varepsilon < 1/3$ and a value $\widetilde{F}_2$ such that $|\widetilde{F}_2 - F_2(x)| \leq (1/2) \cdot F_2(x)$, outputs a value $i \in [n]$ that is distributed according to an (ε, n^{-2})-approximator to $D_{2,x}$. The algorithm uses $M = \mathcal{O}\left(\varepsilon^{-2}\log^3 n\right)$ classical memory cells. Moreover, each element of the stream is processed in time $T_{upd} = \mathcal{O}\left(\varepsilon^{-1}\log n\right)$, and the output is computed in time $T_{rec} = \mathcal{O}\left(\varepsilon^{-1} n \log n\right)$ after the last element is received.*

Input: a stream $\vec{u}$, an integer $k \geq 3$, a real $\widetilde{F}_2$, an approximation parameter $0 < \varepsilon < 1$.
Output: an estimate $\widetilde{F}_k$ of the frequency moment of order k of $\vec{u}$.

1. Compute $i \in [n]$ using the streaming algorithm of Theorem 5.1 with input $\vec{u}$, $\varepsilon/4$, $\widetilde{F}_2$.
2. Compute x_i using a second pass over $\vec{u}$.
3. Output $\widetilde{F}_2 \cdot |x_i|^{k-2}$.

Estimator 3: frequency moment F_k of a stream.

Proposition 5.2 ([51, 6])**.** *If we let X denote the output random variable of Estimator 3, then $\mathbb{E}[X] = (1 \pm \varepsilon/2)F_k$ and $\mathrm{Var}[X] \leq \mathcal{O}\left(n^{1-2/k}F_k^2\right)$, when $|\widetilde{F}_2 - F_2| \leq (\varepsilon/4) \cdot F_2$.*

Using standard techniques, the algorithm of Estimator 3 can be made reversible and therefore implemented by a quantum sampler $\mathcal{S}$. We need to be careful that the reverse computation $\mathcal{S}^{-1}$ can also be done efficiently. Usually, that would require processing the same stream but in the *reverse* direction. However, the construction given in [6] has the particularity to be a *linear sketch* algorithm (the memory content is a linear function $L(x)$ of the input x, see Definition D.1). In Appendix D (Proposition D.2), we show that the reverse computation of such algorithms can be done efficiently with one pass in the *direct* direction. We combine the quantum sampler that is obtained from this result with the Quantum Chebyshev Inequality (Theorem 3.5) to obtain the following tradeoff.

Theorem 5.3. *There is a quantum streaming algorithm that, given a stream $\vec{u}$, two integers $P \geq 1$, $k \geq 3$ and an approximation parameter $0 < \varepsilon < 1$, outputs an estimate $\widetilde{F}_k$ such that $|\widetilde{F}_k - F_k| \leq \varepsilon F_k$ with probability $2/3$. The algorithm uses $\widetilde{\mathcal{O}}\left(n^{1-2/k}/(\varepsilon P)^2\right)$ quantum memory cells, and it makes $\widetilde{\mathcal{O}}\left(P \cdot (k\log n + \varepsilon^{-1})\right)$ passes over the stream $\vec{u}$.*

Proof. We first compute, in one pass, a value $\widetilde{F}_2$ such that $|\widetilde{F}_2 - F_2| \leq (\varepsilon/2)F_2$ with high probability, using [4, 53] for instance. The complexity is absorbed by the final result. Then, using Estimator 3 together with Proposition D.2, we can design a quantum sampler $\mathcal{S}$ using memory $M = \widetilde{\mathcal{O}}\left(\varepsilon^{-2}\log^3 n\right)$ such that $\mathcal{S}(|0\rangle|0\rangle) = \sum_{r\in\{0,1\}^M} |r\rangle|\psi_r\rangle|f_r\rangle$ where each $|r\rangle$ corresponds to a different random seed for the linear sketch algorithm of Theorem 5.1, $|f_r\rangle$ is the output of Estimator 3, and $|\psi_r\rangle$ is some garbage state obtained when making Estimator 3 reversible. According to Proposition 5.2, we have $\mu_{\mathcal{S}} = (1 \pm \varepsilon/2)F_k$ and $\sigma_{\mathcal{S}} \leq \mathcal{O}\left(\sqrt{n^{1-2/k}}F_k\right)$. Moreover one quantum sample can be implemented with two passes over the stream.

We concatenate $Q = n^{1-2/k}/P^2$ such samplers, and compute the mean $\bar{f} = Q^{-1} \cdot (f_{r_1} + \cdots + f_{r_Q})$ of their results, i.e. $\bar{\mathcal{S}}(|0\rangle|0\rangle) = \sum_{r_1,\ldots,r_Q\in\{0,1\}^M} |r_1,\ldots,r_Q\rangle|\psi_1,\ldots,\psi_Q\rangle|f_{r_1},\ldots,f_{r_Q}\rangle|\bar{f}\rangle$. This sampler satisfies

$\sigma_{\bar{\mathcal{S}}} \leq \mathcal{O}(PF_k)$, and it requires two passes and memory $\bar{M} = \widetilde{\mathcal{O}}\left(Q \cdot \varepsilon^{-2} \log^3 n\right)$ to be implemented. Finally, we approximate F_k by applying Theorem 3.5 on $\bar{\mathcal{S}}$, which uses $\widetilde{\mathcal{O}}\left(P \cdot (k \log n + \varepsilon^{-1})\right)$ quantum samples. □

5.2 Approximating graph parameters in the query model

In this section, we consider the *general graph model* [44, 32] that provides query access to a graph $G = (V, E)$ through the following operations: (1) *degree query* (given $v \in V$, returns the degree d_v of v), (2) *neighbor query* (given $v \in V$ and i, returns the i-th neighbor of v if $i \leq d_v$, and $\perp$ otherwise), and (3) *vertex-pair query* (given $u, v \in V$, indicates if $(u, v) \in E$). This is a combination of the dense graph model (pair queries) and the bounded-degree model (neighbor and degree queries). We refer the reader to [32, Chapter 10] for a more detailed discussion about it. It can be extended to the standard quantum query framework. A quantum degree query is represented as a unitary $\mathcal{O}_{deg}$ such that $\mathcal{O}_{deg}|v\rangle|b\rangle = |v\rangle|y \oplus d_v\rangle$ where $v \in V$ and $b \in \{0,1\}^{\lceil \log n \rceil}$. The quantum neighbor $\mathcal{O}_{neigh}$ and vertex-pair $\mathcal{O}_{pair}$ queries are defined similarly. The *query complexity* of an algorithm in the *quantum general graph model* is the number of times it uses $\mathcal{O}_{deg}$, $\mathcal{O}_{nei}$ or $\mathcal{O}_{pair}$.

In the following, we let n denote the number of vertices, m the number of edges and t the number of triangles in G. We consider the problems of estimating m and t, for which we provide nearly optimal quantum algorithms. The description and analysis of these algorithms is deferred to Appendix E.

Edge counting In the classical setting, with degree queries only, Feige [30] showed that $\Theta(n/(\varepsilon\sqrt{m}))$ queries are sufficient to compute a factor $(2+\varepsilon)$ approximation of m, but no factor $(2-\varepsilon)$ approximation can be obtained in sublinear time. Using both degree and neighbor queries, it is possible to compute a factor $(1+\varepsilon)$ approximation with $\Theta\left(n/(\sqrt{\varepsilon m})\right)$ classical queries [33, 58, 27]. These results were generalized to k-star counting in [35, 27]. In the quantum setting, we prove the following results in Appendix E.1.

Theorem 5.4. *There is an algorithm that, given query access to any n-vertex graph G with m edges, and an approximation parameter $\varepsilon < 1$, outputs an estimate $\widetilde{m}$ of m such that $|\widetilde{m} - m| \leq \varepsilon m$ with probability $2/3$. This algorithm performs $\widetilde{\mathcal{O}}\left(\frac{n^{1/2}}{\varepsilon m^{1/4}}\right)$ quantum degree and neighbor queries in expectation. Moreover, it does not use vertex-pair queries.*

Theorem 5.5. *Any algorithm that computes an ε-approximation of the number m of edges in any n-vertex graph, given query access to it, must use $\Omega\left(\frac{n^{1/2}}{(\varepsilon m)^{1/4}} \cdot \log^{-1}(n)\right)$ quantum queries in expectation.*

Triangle counting In the classical general graph model, the triangle counting problem requires $\widetilde{\Theta}(n/t^{1/3} + \min(m, m^{3/2}/t))$ queries in expectation [25, 26]. This result was generalized to k-clique counting in [28]. In the quantum setting, we prove the following results in Appendix E.2.

Theorem 5.6. *There is an algorithm that, given query access to any n-vertex graph G with m edges and t triangles, and an approximation parameter $\varepsilon < 1$, outputs an estimate $\widetilde{t}$ of t such that $|\widetilde{t} - t| \leq \varepsilon t$ with probability $2/3$. This algorithm performs $\widetilde{\mathcal{O}}\left(\left(\frac{\sqrt{n}}{t^{1/6}} + \frac{m^{3/4}}{\sqrt{t}}\right) \cdot poly(1/\varepsilon)\right)$ quantum queries in expectation.*

Theorem 5.7. *Any algorithm that computes an ε-approximation to the number t of triangles in any n-vertex graph with m vertices, given query access to it, must use $\Omega\left(\left(\frac{\sqrt{n}}{t^{1/6}} + \frac{m^{3/4}}{\sqrt{t}}\right) \cdot \log^{-1}(n)\right)$ quantum queries in expectation.*

6 Open questions

Is it possible to improve the complexity of our main result (Theorem 3.5) to $\mathcal{O}(\Delta_{\mathcal{S}}/\varepsilon)$ exactly? Can we generalize it to sample spaces with negative values? What are other possible applications? Two promising problems are minimum spanning tree weight [20] and arbitrary subgraph counting [28, 8].

Acknowledgements

The authors want to thank the anonymous referees for their valuable comments and suggestions which helped to improve this paper.

References

[1] S. Aaronson and A. Ambainis. Quantum search of spatial regions. *Theory of Computing*, 1(4):47–79, 2005.

[2] D. Aharonov and A. Ta-Shma. Adiabatic quantum state generation. *SIAM Journal on Computing*, 37(1):47–82, 2007.

[3] Y. Ai, W. Hu, Y. Li, and D. P. Woodruff. New characterizations in turnstile streams with applications. In *Proceedings of the 31st Conference on Computational Complexity*, CCC '16, pages 20:1–20:22, 2016.

[4] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *J. Comput. Syst. Sci.*, 58(1):137–147, 1999.

[5] A. Ambainis. Variable time amplitude amplification and a faster quantum algorithm for solving systems of linear equations. Technical Report arxiv:1010.4458, arXiv.org, 2010.

[6] A. Andoni, R. Krauthgamer, and K. Onak. Streaming algorithms via precision sampling. In *Proceedings of the 52nd Symposium on Foundations of Computer Science*, FOCS '11, pages 363–372, 2011.

[7] S. Arunachalam and R. de Wolf. Optimal quantum sample complexity of learning algorithms. In *Proceedings of the 32nd Computational Complexity Conference*, CCC '17, pages 25:1–25:31, 2017.

[8] S. Assadi, M. Kapralov, and S. Khanna. A simple sublinear-time algorithm for counting arbitrary subgraphs via edge sampling. Technical Report arxiv:1811.07780, arXiv.org, 2018.

[9] C. Badescu, R. O'Donnell, and J. Wright. Quantum state certification. Technical Report arxiv:1708.06002, arXiv.org, 2017.

[10] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing closeness of discrete distributions. *J. ACM*, 60(1):4:1–4:25, 2013.

[11] C. Bennett. Time/space trade-offs for reversible computation. *SIAM Journal on Computing*, 18(4):766–776, 1989.

[12] G. Brassard, F. Dupuis, S. Gambs, and A. Tapp. An optimal quantum algorithm to approximate the mean and its application for approximating the median of a set of points over an arbitrary distance. Technical Report arxiv:1106.4267, arXiv.org, 2011.

[13] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. *Quantum Computation and Quantum Information: A Millennium Volume*, 1:53–74, 2002.

[14] S. Bravyi, A. W. Harrow, and A. Hassidim. Quantum algorithms for testing properties of distributions. *IEEE Transactions on Information Theory*, 57(6):3971–3981, 2011.

[15] H. Buhrman, R. Cleve, and A. Wigderson. Quantum vs. classical communication and computation. In *Proceedings of the 33th Symposium on Theory of Computing*, STOC '98, pages 63–68, 1998.

[16] C. L. Canonne, I. Diakonikolas, D. M. Kane, and A. Stewart. Testing conditional independence of discrete distributions. In *Proceedings of the 50th Symposium on Theory of Computing*, STOC '18, pages 735–748, 2018.

[17] A. Chakrabarti, G. Cormode, R. Kondapally, and A. McGregor. Information cost tradeoffs for augmented index and streaming language recognition. *SIAM Journal on Computing*, 42(1):61–83, 2013.

[18] S. Chakraborty, A. Gilyén, and S. Jeffery. The power of block-encoded matrix powers: improved regression techniques via faster Hamiltonian simulation. Technical Report arxiv:1804.01973, arXiv.org, 2018.

[19] S. Chan, I. Diakonikolas, P. Valiant, and G. Valiant. Optimal algorithms for testing closeness of discrete distributions. In *Proceedings of the 25th Symposium on Discrete Algorithms*, SODA '14, pages 1193–1203, 2014.

[20] B. Chazelle, R. Rubinfeld, and L. Trevisan. Approximating the minimum spanning tree weight in sublinear time. *SIAM Journal on Computing*, 34(6):1370–1379, 2005.

[21] A. N. Chowdhury and R. D. Somma. Quantum algorithms for Gibbs sampling and hitting-time estimation. *Quantum Info. Comput.*, 17(1-2):41–64, 2017.

[22] P. Dagum, R. Karp, M. Luby, and S. Ross. An optimal algorithm for Monte Carlo estimation. *SIAM Journal on Computing*, 29(5):1484–1496, 2000.

[23] N. Destainville, B. Georgeot, and O. Giraud. Quantum algorithm for exact Monte Carlo sampling. *Phys. Rev. Lett.*, 104:250502, 2010.

[24] M. Dyer, A. Frieze, and R. Kannan. A random polynomial-time algorithm for approximating the volume of convex bodies. *J. ACM*, 38(1):1–17, 1991.

[25] T. Eden, A. Levi, and D. Ron. Approximately counting triangles in sublinear time. Technical Report TR15-046, ECCC, 2015.

[26] T. Eden, A. Levi, D. Ron, and C. Seshadhri. Approximately counting triangles in sublinear time. *SIAM J. Comput.*, 46(5):1603–1646, 2017.

[27] T. Eden, D. Ron, and C. Seshadhri. Sublinear time estimation of degree distribution moments: The degeneracy connection. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming*, ICALP '17, pages 7:1–7:13, 2017.

[28] T. Eden, D. Ron, and C. Seshadhri. On approximating the number of k-cliques in sublinear time. In *Proceedings of the 50th Symposium on Theory of Computing*, STOC '18, pages 722–734, 2018.

[29] T. Eden and W. Rosenbaum. Lower bounds for approximating graph parameters via communication complexity. In *Proceedings of the Workshop on Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, APPROX/RANDOM '18, pages 11:1–11:18, 2018.

[30] U. Feige. On sums of independent random variables with unbounded variance and estimating the average degree in a graph. *SIAM Journal on Computing*, 35(4):964–984, 2006.

[31] S. Ganguly. Taylor polynomial estimator for estimating frequency moments. In *Proceedings of the 42nd International Colloquium on Automata, Languages and Programming*, ICALP '15, pages 542–553, 2015.

[32] O. Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.

[33] O. Goldreich and D. Ron. Approximating average parameters of graphs. *Random Struct. Algorithms*, 32(4):473–493, 2008.

[34] O. Goldreich and D. Ron. On testing expansion in bounded-degree graphs. In *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation*, pages 68–75. Springer-Verlag, 2011.

[35] M. Gonen, D. Ron, and Y. Shavitt. Counting stars and other small subgraphs in sublinear-time. *SIAM Journal on Discrete Mathematics*, 25(3):1365–1411, 2011.

[36] S. Heinrich. Quantum summation with an application to integration. *Journal of Complexity*, 18(1):1 – 50, 2002.

[37] C. W. Helstrom. Quantum detection and estimation theory. *Journal of Statistical Physics*, 1(2):231–252, Jun 1969.

[38] R. Jain and A. Nayak. The space complexity of recognizing well-parenthesized expressions in the streaming model: the index function revisited. *IEEE Transactions on Information Theory*, 60(10):6646–6668, 2014.

[39] M. Jerrum and A. Sinclair. The Markov chain Monte Carlo method: An approach to approximate counting and integration. In *Approximation Algorithms for NP-hard Problems*, chapter 12, pages 482–520. PWS Publishing, 1996.

[40] M. Jerrum, A. Sinclair, and E. Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *J. ACM*, 51(4):671–697, 2004.

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

[42] Z. Ji, Y.-K. Liu, and F. Song. Pseudorandom quantum states. In *Advances in Cryptology*, CRYPTO '18, pages 126–152, 2018.

[43] R. M. Karp and M. Luby. Monte-Carlo algorithms for enumeration and reliability problems. In *Proceedings of the 24th Symposium on Foundations of Computer Science*, FOCS '83, pages 56–64, 1983.

[44] T. Kaufman, M. Krivelevich, and D. Ron. Tight bounds for testing bipartiteness in general graphs. *SIAM Journal on Computing*, 33(6):1441–1483, 2004.

[45] E. Knill, G. Ortiz, and R. D. Somma. Optimal quantum measurements of expectation values of observables. *Phys. Rev. A*, 75:012328, 2007.

[46] F. Le Gall. Exponential separation of quantum and classical online space complexity. *Theor. Comp. Sys.*, 45(2):188–202, 2009.

[47] T. Li and X. Wu. Quantum query complexity of entropy estimation. Technical Report arxiv:1710.06025, arXiv.org, 2017.

[48] Y. Li, H. L. Nguyen, and D. P. Woodruff. Turnstile streaming algorithms might as well be linear sketches. In *Proceedings of the 46th Symposium on Theory of Computing*, STOC '14, pages 174–183, 2014.

[49] Y. Li and D. P. Woodruff. A tight lower bound for high frequency moment estimation with small error. In *Proceedings of the Workshop on Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques*, APPROX/RANDOM '13, pages 623–638, 2013.

[50] F. Magniez, C. Mathieu, and A. Nayak. Recognizing well-parenthesized expressions in the streaming model. *SIAM Journal on Computing*, 43(6):1880–1905, 2014.

[51] M. Monemizadeh and D. P. Woodruff. 1-pass relative-error Lp-sampling with applications. In *Proceedings of the 21st Symposium on Discrete Algorithms*, SODA '10, pages 1143–1160, 2010.

[52] A. Montanaro. Quantum speedup of Monte Carlo methods. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 471(2181), 2015.

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

[54] A. Nayak and D. Touchette. Augmented index and quantum streaming algorithms for DYCK(2). In *Proceedings of the 32nd Conference on Computational Complexity*, CCC '17, pages 23:1–23:21, 2017.

[55] A. Nayak and F. Wu. The quantum query complexity of approximating the median and related statistics. In *Proceedings of the 31st Symposium on Theory of Computing*, STOC '99, pages 384–393, 1999.

[56] D. Poulin and P. Wocjan. Sampling from the thermal quantum Gibbs state and evaluating partition functions with a quantum computer. *Phys. Rev. Lett.*, 103:220502, 2009.

[57] A. A. Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya: Mathematics*, 67(1):145–159, 2003.

[58] C. Seshadhri. A simpler sublinear algorithm for approximating the triangle count. Technical Report arxiv:1505.01927, arXiv.org, 2015.

[59] K. Temme, T. J. Osborne, K. Vollbrecht, D. Poulin, and F. Verstraete. Quantum metropolis sampling. *Nature*, 471:87, 2011.

[60] D. Štefankovič, S. Vempala, and E. Vigoda. Adaptive simulated annealing: A near-optimal connection between sampling and counting. *J. ACM*, 56(3):18:1–18:36, 2009.

[61] P. Wocjan and A. Abeyesinghe. Speedup via quantum sampling. *Phys. Rev. A*, 78:042336, 2008.

[62] P. Wocjan, C.-F. Chiang, D. Nagaj, and A. Abeyesinghe. Quantum algorithm for approximating partition functions. *Phys. Rev. A*, 80:022340, 2009.

[63] D. P. Woodruff and Q. Zhang. Tight bounds for distributed functional monitoring. In *Proceedings of the 44th Symposium on Theory of Computing*, STOC '12, pages 941–960, 2012.

A A faster algorithm for mean approximation

We show first how to improve the dependence on ε of Algorithm 2. To this end, we need a finer version of an algorithm from [36, 52], where we introduce a new parameter Γ (the result presented in [52] corresponds to $\Gamma = 1$).

Input: a sampler $\mathcal{S}$, a parameter $\Gamma > 0$, an integer $t > 2$, a failure parameter $0 < \delta < 1$.
Output: an estimate $\widetilde{\mu}_{\mathcal{S}}$ of $\mu_{\mathcal{S}}$.

1. Set $k = \lceil \log t \rceil - 1$, $t_0 = \lceil 3\pi^2 t\sqrt{\log t} \rceil$.
2. Compute $\widetilde{p}_0 = \mathsf{BasicEst}\,(\mathcal{S}, (0,\Gamma), t_0, \delta/(k+1))$.
3. For $\ell = 1, \ldots, k$:
   (a) Compute $\widetilde{p}_\ell = \mathsf{BasicEst}\left(\mathcal{S}, (2^{\ell-1}\Gamma, 2^\ell\Gamma), t_0, \delta/(k+1)\right)$.
4. Output $\widetilde{\mu}_{\mathcal{S}} = \sum_{\ell=0}^{k} 2^\ell \Gamma \cdot \widetilde{p}_\ell$.

Algorithm 4: subroutine for approximating the mean of a quantum sampler $\mathcal{S}$.

Proposition A.1. *The output $\widetilde{\mu}_{\mathcal{S}}$ of Algorithm 4 satisfies $|\widetilde{\mu}_{\mathcal{S}} - \mu_{\mathcal{S}}| \leq \frac{1}{t}\left(\sqrt{\Gamma} + \frac{\phi_{\mathcal{S}}}{\sqrt{\Gamma}}\right)^2$ and $\widetilde{\mu}_{\mathcal{S}} \leq (1+2\pi)^2 \mu_{\mathcal{S}}$ with probability $1-\delta$. The number of quantum samples used by the algorithm is $\mathcal{O}(t \log^{3/2}(t) \log(\log(t)/\delta))$.*

Proof. Observe that $\mu_{\mathcal{S}} = \sum_{\ell=0}^{k} 2^\ell \Gamma \cdot p_\ell + \mathbb{E}\left[v(\mathcal{S})_{\geq 2^k\Gamma}\right]$, where $p_0 = \frac{\mathbb{E}[v(\mathcal{S})_{0,\Gamma}]}{\Gamma}$ and $p_\ell = \frac{\mathbb{E}\left[v(\mathcal{S})_{2^{\ell-1}\Gamma, 2^\ell\Gamma}\right]}{2^\ell\Gamma}$. Using Corollary 2.4 and a union bound, we can assume $|\widetilde{p}_\ell - p_\ell| \leq \pi^2\left(\frac{\sqrt{p_\ell}}{t_0} + \frac{1}{t_0^2}\right)$ and $\widetilde{p}_\ell \leq (1+2\pi)^2 p_\ell$ for all ℓ, with probability $1-\delta$. It implies $\widetilde{\mu}_{\mathcal{S}} \leq (1+2\pi)^2 \widetilde{\mu}_{\mathcal{S}}$. On the other hand, using the triangle inequality,

$$
\begin{aligned}
|\widetilde{\mu}_{\mathcal{S}} - \mu_{\mathcal{S}}| &\leq \pi^2 \left(\frac{\Gamma}{t_0} + \frac{1}{t_0} \sum_{\ell=1}^{k} \sqrt{2^\ell \Gamma \cdot \mathbb{E}\left[v(\mathcal{S})_{2^{\ell-1}\Gamma, 2^\ell\Gamma}\right]} + \frac{\Gamma}{t_0^2} \sum_{\ell=0}^{k} 2^\ell \right) + \mathbb{E}\left[v(\mathcal{S})_{\geq 2^k\Gamma}\right] \\
&\leq \pi^2 \left(\frac{\Gamma}{t_0} + \frac{1}{t_0}\sqrt{k} \sqrt{\sum_{\ell=1}^{k} 2^\ell \Gamma \cdot \frac{\mathbb{E}\left[v(\mathcal{S})^2_{2^{\ell-1}\Gamma, 2^\ell\Gamma}\right]}{2^{\ell-1}\Gamma} + \frac{2^{k+1}}{t_0^2}\Gamma} \right) + \frac{\phi_{\mathcal{S}}^2}{2^k\Gamma} \\
&\leq \pi^2 \left(\frac{\Gamma}{t_0} + \frac{\sqrt{2k}}{t_0} \cdot \phi_{\mathcal{S}} + \frac{2^{k+1}}{t_0^2}\Gamma \right) + \frac{\phi_{\mathcal{S}}^2}{2^k\Gamma} \leq \frac{1}{t}\left(\sqrt{\Gamma} + \frac{\phi_{\mathcal{S}}}{\sqrt{\Gamma}}\right)^2
\end{aligned}
$$

where we used Fact 3.1 and the Cauchy-Schwarz inequality, at the second step. □

If we set $\Gamma = \phi_{\mathcal{S}}$ in the above inequality, we obtain $|\widetilde{\mu}_{\mathcal{S}} - \mu_{\mathcal{S}}| \leq 4\phi_{\mathcal{S}}/t$, and thus $|\widetilde{\mu}_{\mathcal{S}} - \mu_{\mathcal{S}}| \leq \varepsilon\mu_{\mathcal{S}}$ when $t = \Omega\left(\varepsilon^{-1}\Delta_{\mathcal{S}}\right)$. Since $\phi_{\mathcal{S}}$ is unknown, we approximate it by $\widetilde{\phi}_{\mathcal{S}} = M\Delta_{\mathcal{S}}$ instead, where $M \in [2\mu_{\mathcal{S}}, 2500\mu_{\mathcal{S}}]$ is obtained with the same method as in Algorithm 2.

Input: a sampler $\mathcal{S}$, an integer $\Delta_{\mathcal{S}}$, two values $0 < L < H$, two reals $0 < \varepsilon, \delta < 1/2$.
Output: an estimate $\widetilde{\mu}_{\mathcal{S}}$ of $\mu_{\mathcal{S}}$.

1. Set $M = 8H$ and $\widetilde{p} = 0$
2. While $\widetilde{p} = 0$ and $M \geq 2L$:
   (a) Set $M = M/2$.
   (b) Compute $\widetilde{p} = \mathsf{BasicEst}\left(\mathcal{S}, (0, M\Delta_{\mathcal{S}}^2), 25\Delta_{\mathcal{S}}, \delta'\right)$ where $\delta' = \frac{\delta}{2(3+\log(H/L))}$.
3. If $M < 2L$ then output $\widetilde{\mu}_{\mathcal{S}} = 0$.
4. Else, run Algorithm 4 on input $\mathcal{S}$, $\Gamma = M \cdot \Delta_{\mathcal{S}}$, $t = 51^2\varepsilon^{-1}\Delta_{\mathcal{S}}$, $\delta/2$ and output the result as $\widetilde{\mu}_{\mathcal{S}}$.

Algorithm 5: $\varepsilon-$approximation of the mean of a quantum sampler $\mathcal{S}$.

Theorem A.2. *If $\Delta_{\mathcal{S}} \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$ and $L < \mu_{\mathcal{S}} < H$ then the output $\widetilde{\mu}_{\mathcal{S}}$ of Algorithm 5 satisfies $|\widetilde{\mu}_{\mathcal{S}} - \mu_{\mathcal{S}}| \leq \varepsilon\mu_{\mathcal{S}}$ with probability $1-\delta$. Moreover, for any $\Delta_{\mathcal{S}}, L, H$ it satisfies $\widetilde{\mu}_{\mathcal{S}} \leq (1+2\pi)^2\mu_{\mathcal{S}}$ with probability $1-\delta$. The number of quantum samples used by the algorithm is*

$$\mathcal{O}\left(\Delta_{\mathcal{S}} \cdot \left(\log\left(\frac{H}{L}\right)\log\left(\frac{\log(H/L)}{\delta}\right) + \varepsilon^{-1}\log^{3/2}(\Delta_{\mathcal{S}})\log\left(\frac{\log\Delta_{\mathcal{S}}}{\delta}\right)\right)\right).$$

Proof. Steps 1 to 3 are identical to the beginning of Algorithm 2. Consequently, by the same arguments as in the proof of Theorem 3.3, when $\Delta_{\mathcal{S}} \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$ and $L < \mu_{\mathcal{S}} < H$ we can assume (with probability $1-\delta/2$) that Step 4 is executed with $M \in [2\mu_{\mathcal{S}}, 2500\mu_{\mathcal{S}}]$. In this case, according to Proposition A.1, the output $\widetilde{\mu}_{\mathcal{S}}$ satisfies $|\widetilde{\mu}_{\mathcal{S}} - \mu_{\mathcal{S}}| \leq \frac{1}{51^2\varepsilon^{-1}\Delta_{\mathcal{S}}}\left(\sqrt{2500\mu_{\mathcal{S}}\Delta_{\mathcal{S}}} + \frac{\phi_{\mathcal{S}}}{\sqrt{2\mu_{\mathcal{S}}\Delta_{\mathcal{S}}}}\right)^2 \leq \frac{(\sqrt{2500}+1/\sqrt{2})^2}{51^2}\varepsilon\mu_{\mathcal{S}} \leq \varepsilon\mu_{\mathcal{S}}$ with probability $1-\delta/2$. □

The next algorithm details how to replace the input parameter L with a logarithmic search on decreasing values of L. This causes the factor $\log(H/L)$ in the complexity bounds to become $\log^3(H/\mu_{\mathcal{S}})$. A similar result can be obtained for all the other algorithms of Section 3.

Input: a sampler $\mathcal{S}$, an integer $\Delta_{\mathcal{S}}$, a value $H > 0$, two reals $0 < \varepsilon, \delta < 1/2$.
Output: an estimate $\widetilde{\mu}_{\mathcal{S}}$ of $\mu_{\mathcal{S}}$.

1. Set $i = 1$.
2. Run Algorithm 5 on input $\mathcal{S}$, $\Delta_{\mathcal{S}}$, $L = H/2^i$, H, $\delta/2^i$.
   (a) If the result is non-zero, run Algorithm 5 on input $\mathcal{S}$, $\Delta_{\mathcal{S}}$, $L/1250$, H, ε, $\delta/2^{i+1}$ and output its result as $\widetilde{\mu}_{\mathcal{S}}$.
   (b) Else, set $i = i+1$ and go to Step 2.

Algorithm 6: $\varepsilon-$approximation of the mean of a quantum sampler $\mathcal{S}$.

Proof of Theorem 3.5. We show that Algorithm 6 satisfies the properties specified in Theorem 3.5.

Suppose that $\Delta_{\mathcal{S}} \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$ and $H > \mu_{\mathcal{S}}$. Since Remark 3.4 also applies to Algorithm 5, the probability that Algorithm 6 stops with $L \geq 1250\mu_{\mathcal{S}}$ is at most $\sum_{i=1}^{\lfloor\log(H/(1250\mu_{\mathcal{S}}))\rfloor}\delta/2^i$. On the other hand, if $L < 1250\mu_{\mathcal{S}}$ at Step 2.(a) then, according to Theorem A.2, the output $\widetilde{\mu}_{\mathcal{S}}$ satisfies $|\widetilde{\mu}_{\mathcal{S}} - \mu_{\mathcal{S}}| \leq \varepsilon\mu_{\mathcal{S}}$ with probability $1-\delta/2^{i+1}$. Consequently, the output is correct with probability at least $1-\sum_{i=1}^{\infty}\delta/2^i \geq \delta$.

According to Remark 3.4, when $L < \mu_{\mathcal{S}}$ the probability that Step 2 computes a non-zero value is at least $1-\delta/2^i$. Thus, Algorithm 6 uses

$$\widetilde{\mathcal{O}}\left(\Delta_{\mathcal{S}} \cdot \varepsilon^{-1}\log(1/\delta)\left(\sum_{i=1}^{\lfloor\log(H/\mu_{\mathcal{S}})\rfloor} i^2 + \sum_{i=\lfloor\log(H/\mu_{\mathcal{S}})\rfloor}^{\infty} i^2 \cdot \delta/2^i\right)\right) = \widetilde{\mathcal{O}}\left(\Delta_{\mathcal{S}} \cdot \varepsilon^{-1}\log^3(H/\mu_{\mathcal{S}})\log(1/\delta)\right)$$

quantum samples in expectation. □

B Approximating the mean when $\Delta_{\mathcal{S}}$ is implicit

We show how to approximate the mean $\mu_{\mathcal{S}}$ of a quantum sampler $\mathcal{S}$ given a non-increasing function f such that $f(\mu_{\mathcal{S}}) \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$. Our result combines Algorithm 5 (or Algorithm 2) with a new stopping rule that is based on the Markov-like inequality "$\widetilde{\mu}_{\mathcal{S}} \leq (1+2\pi)^2\mu_{\mathcal{S}}$" of Theorem A.2.

Input: a sampler $\mathcal{S}$, a non-increasing function f such that $f(\mu_{\mathcal{S}}) \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$, two values $0 < L < H$, two reals $0 < \varepsilon, \delta < 1/2$.
Output: an estimate $\widetilde{\mu}_{\mathcal{S}}$ of $\mu_{\mathcal{S}}$.

1. Set $M = 2H$, $\Delta_{\mathcal{S}} = f(M)$ and $\widetilde{\mu} = 0$.
2. While $\widetilde{\mu} < M/6$ and $M \geq L/2$:
   (a) Set $M = M/2$ and $\Delta_{\mathcal{S}} = f(M)$.
   (b) Run Algorithm 5 on input $\mathcal{S}$, $\Delta_{\mathcal{S}}$, L, H, $\varepsilon' = 5/6$, $\delta' = \frac{\delta}{2\left(2+\log\left(\frac{H}{L}\right)\right)}$. Denote the result by $\widetilde{\mu}$.
3. If $M < L/2$ then output $\widetilde{\mu}_{\mathcal{S}} = 0$.
4. Else, run Algorithm 5 on input $\mathcal{S}$, $\Delta_{\mathcal{S}} = f\left(M/(6(1+2\pi)^2)\right)$, L, H, ε, $\delta/2$ and output its result as $\widetilde{\mu}_{\mathcal{S}}$.

Algorithm 7: $\varepsilon-$approximation of the mean of a quantum sampler $\mathcal{S}$ for implicit $\Delta_{\mathcal{S}}$.

Theorem B.1. *If $L \leq \mu_{\mathcal{S}} < H$ then the output $\widetilde{\mu}_{\mathcal{S}}$ of Algorithm 2 satisfies $|\widetilde{\mu}_{\mathcal{S}} - \mu_{\mathcal{S}}| \leq \varepsilon\mu_{\mathcal{S}}$ with probability $1-\delta$. Moreover, for any L it satisfies $\widetilde{\mu}_{\mathcal{S}} \leq (1+2\pi)^2\mu_{\mathcal{S}}$ with probability $1-\delta$. The number of quantum samples used by the algorithm is*

$$\widetilde{\mathcal{O}}\left(f\left(\frac{\max(L/4, 2^{-T}\mu_{\mathcal{S}})}{6(1+2\pi)^2}\right)\cdot\varepsilon^{-1}\log\left(\frac{H}{L}\right)\log\left(\frac{1}{\delta}\right)\right)$$

for some integer random variable T such that $\mathbb{P}(T=1) \geq 1-\delta$ and $\mathbb{P}(T=\ell) \leq \delta^{\ell}$ for all $\ell > 1$.

Proof. Assume first that $L \leq \mu_{\mathcal{S}}$. According to Theorem A.2, the estimate $\widetilde{\mu}$ computed at Step 2.(b) of Algorithm 7 satisfies $\widetilde{\mu} \leq (1+2\pi)^2\mu_{\mathcal{S}}$ with probability $1-\delta'$. Consequently, when $M > 6(1+2\pi)^2\mu_{\mathcal{S}}$, we have $\widetilde{\mu} < M/6$ with probability $1-\delta'$. On the other hand, when $M \leq \mu_{\mathcal{S}}$, since $\Delta_{\mathcal{S}} = f(M) \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$ the value $\widetilde{\mu}$ satisfies $|\widetilde{\mu} - \mu_{\mathcal{S}}| \leq (5/6)\cdot\mu_{\mathcal{S}}$ with probability $1-\delta'$ (by Theorem A.2). In particular, it implies $\widetilde{\mu} \geq \mu_{\mathcal{S}}/6 \geq M/6$ with probability $1-\delta'$. Using these two points, we conclude that the first time Step 2.(b) of Algorithm 7 obtains $\widetilde{\mu} \geq M/6$ happens for $M \in [\mu_{\mathcal{S}}/2, 6(1+2\pi)^2\mu_{\mathcal{S}}]$, with probability at least $(1-\delta')^{1+\log(H/(\mu_{\mathcal{S}}/2))} > 1-\delta/2$. In this case, $\Delta_{\mathcal{S}} \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$ at Step 4 of the algorithm, and the output $\widetilde{\mu}_{\mathcal{S}}$ satisfies $|\widetilde{\mu} - \mu_{\mathcal{S}}| \leq \varepsilon\mu_{\mathcal{S}}$ with probability $1-\delta/2$ (by Theorem A.2). The total success probability is $(1-\delta/2)^2 \geq 1-\delta$.

If $L > \mu_{\mathcal{S}}$, this may only increase the probability to stop at Step 3 and output $\widetilde{\mu}_{\mathcal{S}} = 0$. If Step 4 is executed, we still have $\widetilde{\mu}_{\mathcal{S}} \leq (1+2\pi)^2\mu_{\mathcal{S}}$ with probability $1-\delta$, as a consequence of Theorem A.2.

We analyse the number of quantum samples used in the algorithm. The value taken by M at Step 4 satisfies $M \geq \mu_{\mathcal{S}}/2$ with probability at least $1-\delta$, and $2^{-\ell}\mu_{\mathcal{S}} > M \geq 2^{-(\ell+1)}\mu_{\mathcal{S}}$ with probability at most δ^{ℓ} (for any $\ell \geq 1$). Moreover, the total number of quantum samples used in Algorithm 7 is dominated (up to a polylogarithmic factor in H/L) by the number of quantum samples used at Step 4, that is $\widetilde{\mathcal{O}}\left(f\left(\frac{2^{-\ell}\mu_{\mathcal{S}}}{12(1+2\pi)^2}\right)\cdot\varepsilon^{-1}\log\left(\frac{H}{L}\right)\log\left(\frac{1}{\delta}\right)\right)$ when $M \geq 2^{-\ell}\mu_{\mathcal{S}}$. The smallest possible value for M at Step 4 is $L/4$. Thus, the total number of quantum samples is $\widetilde{\mathcal{O}}\left(f\left(\frac{\max(L/4, 2^{-T}\mu_{\mathcal{S}})}{12(1+2\pi)^2}\right)\cdot\varepsilon^{-1}\log\left(\frac{H}{L}\right)\log\left(\frac{1}{\delta}\right)\right)$, where $T=1$ with probability at least $1-\delta$ and $T=\ell$ with probability at most δ^{ℓ}, for all $\ell \geq 2$. □

We simplify the above statement when the function f is of the form $f : x \mapsto A/x^{\alpha}$ for some $A, \alpha > 0$ (this result is sufficient for our applications in Section 5.2).

Corollary B.2. *If $L \leq \mu_{\mathcal{S}} < H$ and $f : x \mapsto A/x^{\alpha}$ for some reals $A, \alpha > 0$ with $\delta < 2^{-2\alpha}$, then the output $\widetilde{\mu}_{\mathcal{S}}$ of Algorithm 2 satisfies $|\widetilde{\mu}_{\mathcal{S}} - \mu_{\mathcal{S}}| \leq \varepsilon\mu_{\mathcal{S}}$ with probability $1-\delta$. Moreover, for any L it satisfies $\widetilde{\mu}_{\mathcal{S}} \leq (1+2\pi)^2\mu_{\mathcal{S}}$ with probability $1-\delta$. The algorithm uses*

$$\widetilde{\mathcal{O}}\left(f(\max(L, \mu_{\mathcal{S}}))\cdot\varepsilon^{-1}\log\left(\frac{H}{L}\right)\log\left(\frac{1}{\delta}\right)\right)$$

quantum samples in expectation (both for the ℓ_1 and ℓ_2 average).

Proof. The average (for the ℓ_1 norm) number of quantum samples used in Algorithm 7 is

$$\widetilde{\mathcal{O}}\left(\sum_{\ell=0}^{\infty}\delta^{\ell}\cdot f\left(\frac{\max(L/4,2^{-(\ell+1)}\mu_{\mathcal{S}})}{6(1+2\pi)^2}\right)\cdot\varepsilon^{-1}\log\left(\frac{H}{L}\right)\log\left(\frac{1}{\delta}\right)\right)$$

Since $f:x\mapsto A/x^{\alpha}$ for some $A,\alpha>0$, it becomes $\widetilde{\mathcal{O}}\left(\frac{A}{\max(L,\mu_{\mathcal{S}})^{\alpha}}\cdot\varepsilon^{-1}\log\left(\frac{H}{L}\right)\log\left(\frac{1}{\delta}\right)\right)\right)$ when $\delta<2^{-\alpha}$. Similarly, for the ℓ_2 norm, the average number of quantum samples used in Algorithm 7 is

$$\widetilde{\mathcal{O}}\left(\left(\sum_{\ell=0}^{\infty}\delta^{\ell}\cdot\left(f\left(\frac{\max(L/4,2^{-(\ell+1)}\mu_{\mathcal{S}})}{6(1+2\pi)^2}\right)\cdot\varepsilon^{-1}\log\left(\frac{H}{L}\right)\log\left(\frac{1}{\delta}\right)\right)^2\right)^{1/2}\right)$$

which becomes $\widetilde{\mathcal{O}}\left(\frac{A}{\max(L,\mu_{\mathcal{S}})^{\alpha}}\cdot\varepsilon^{-1}\log\left(\frac{H}{L}\right)\log\left(\frac{1}{\delta}\right)\right)\right)$ when $\delta<2^{-2\alpha}$. □

C Approximating the mean of variable-time samplers

Definition C.1 (Variable-time algorithm [5, 18])**.** *Consider two Hilbert spaces $\mathcal{H}_F=\otimes_{i=1}^{m}\mathcal{H}_{F_i}$ (for some integer m) and $\mathcal{H}_C$, where each $\mathcal{H}_{F_i}$ is equipped with a standard basis $\{|stop\rangle,|cont\rangle\}$. We say that a unitary U acting on $\mathcal{H}_F\otimes\mathcal{H}_C$ is a* variable-time algorithm with stopping times $t_1<\cdots<t_m$ *if it can be decomposed as a product of unitary operators $U=U_m\cdots U_1$, such that each U_i has time complexity $T_{max}(U_i)=t_i-t_{i-1}$ (where $t_0=0$) and acts on $\mathcal{H}_{F_i}\otimes\mathcal{H}_C$ controlled on the first $(i-1)^{th}$ registers being $|cont\rangle^{\otimes i-1}\in\otimes_{j=1}^{i-1}\mathcal{H}_{F_j}$. The* probability to stop at step *i is defined as*

$$p_{stop,i}=\|\Pi_{stop}(U_i\cdots U_1|init\rangle)\|^2-\|\Pi_{stop}(U_{i-1}\cdots U_1|init\rangle)\|^2$$

where $|init\rangle=|cont\rangle^{\otimes m}|0\rangle\in\mathcal{H}_F\otimes\mathcal{H}_C$ and Π_{stop} is the projector on $\mathrm{Span}\left(|cont\rangle^{\otimes m}\right)^{\perp}\otimes\mathcal{H}_C$ *(i.e. on the states containing $|stop\rangle$). The* ℓ_2-average running time *of U is defined as $T_{\ell_2}(U)=(\sum_{i=1}^{m}p_{stop,i}\cdot t_i^2)^{1/2}$.*

The previous definition expresses the fact that some branches of computation may stop earlier than the others. When a branch is completed at time t_i, the corresponding register in $\mathcal{H}_{F_i}$ is set to $|stop\rangle$, and this part of the state cannot be changed afterward. Ambainis [5] studied the question of quantum search and amplitude amplification for variable-time unitaries $U=U_m\cdots U_1$. We extend this work by developing the following *variable-time amplitude estimation* algorithm.[4]

Theorem C.2. *Consider two Hilbert spaces $\mathcal{H}_F=\otimes_{i=1}^{m}\mathcal{H}_{F_i}$ (for some integer m) and $\mathcal{H}_C$, where each $\mathcal{H}_{F_i}$ is equipped with a standard basis $\{|stop\rangle,|cont\rangle\}$. There is a quantum algorithm that takes as input a variable-time algorithm $U=U_m\cdots U_1$ on $\mathcal{H}_F\otimes\mathcal{H}_C$, an orthogonal projector Π_C on $\mathcal{H}_C$, two reals $t,T_{\ell_2}>1$, and two reals $0<\varepsilon,\delta<1$. If $T_{\ell_2}\geq T_{\ell_2}(U)$, then the algorithm outputs an estimate $\widetilde{p}$ of $p=\langle\psi|\Pi|\psi\rangle$, where $\Pi=(I_{\mathcal{H}_F}-(|cont\rangle\langle cont|)^{\otimes m})\otimes\Pi_C$ and $|\psi\rangle=U(|cont\rangle^{\otimes m}|0\rangle)$, such that*

$$(1)\ \ \widetilde{p}\leq 2\cdot p,\ \text{for any } t; \qquad (2)\ \ |\widetilde{p}-p|\leq\varepsilon\cdot p,\ \text{when } t\geq\tfrac{2}{\sqrt{p}}; \qquad (3)\ \ \widetilde{p}=0,\ \text{when } t<\tfrac{1}{\sqrt{2p}}.$$

with probability $1-\delta$. The time complexity of this algorithm is

$$\mathcal{O}\left(\left(\min\left(T_{max}(U),t\cdot T_{\ell_2}\varepsilon^{-1/2}\right)+t\cdot T_{\ell_2}\right)\varepsilon^{-1}\cdot\log^4(T_{max}(U))\log\left(\frac{\log(T_{max}(U))}{\delta}\right)\right).$$

[4] We have been aware, during the redaction of this paper, of a similar result recently obtained in [18] with time complexity $\mathcal{O}\left((T_{max}(U)+t\cdot T_{\ell_2})\varepsilon^{-1}\cdot\log^3(T_{max}(U))\log\left(\log(T_{max}(U))/\delta\right)\right)$ that is too large for our applications.

Using this new result in place of the standard amplitude estimation in Algorithm 5, we obtain directly the following result.

Theorem C.3. *There is an algorithm that, given a variable-time sampler $\mathcal{S}$, an integer $\Delta_{\mathcal{S}}$, two values $0 < L < H$, a real $T_{\ell_2} \geq 1$, and two reals $0 < \varepsilon, \delta < 1$, outputs an estimate $\widetilde{\mu}_{\mathcal{S}}$ of $\mu_{\mathcal{S}}$. If $\Delta_{\mathcal{S}} \geq \phi_{\mathcal{S}}/\mu_{\mathcal{S}}$, $T_{\ell_2} \geq T_{\ell_2}(\mathcal{S})$ and $L < \mu_{\mathcal{S}} < H$, then it satisfies $|\widetilde{\mu}_{\mathcal{S}} - \mu_{\mathcal{S}}| \leq \varepsilon \mu_{\mathcal{S}}$ with probability $1-\delta$. Moreover, for any $\Delta_{\mathcal{S}}, L, H, T_{\ell_2}$ it satisfies $\widetilde{\mu}_{\mathcal{S}} \leq 2 \cdot \mu_{\mathcal{S}}$ with probability $1-\delta$. The* time complexity *of this algorithm is*

$$\widetilde{\mathcal{O}}\left(\Delta_{\mathcal{S}}\left(\varepsilon^{-2} + \log\left(\frac{H}{L}\right)\right) \cdot T_{\ell_2} \cdot \log^4(T_{max}(\mathcal{S})) \log\left(\frac{1}{\delta}\right)\right).$$

The rest of this section is dedicated to the proof of Theorem C.2. Our approach (Algorithms 10 and 11) consists in estimating at each intermediate time step t_i of U a multiplicative portion $\widetilde{p}_i$ of p (the final estimate $\widetilde{p}$ of p being the product of the $\widetilde{p}_i$'s). To this end, we apply the amplitude estimation algorithm on two particular state generation algorithms $(\mathcal{B}_i)_i$ and $(\mathcal{A}_i)_i$ (Algorithms 8 and 9) originating from the work of Ambainis [5].

C.1 Preliminaries

We need a modified version of the amplitude estimation algorithm that does not need input time parameter.

Proposition C.4 ([13, Theorem 15])**.** *There is a quantum algorithm, denoted* $\mathsf{AmplEst}^\star$*, that takes as input a unitary operator U, an orthogonal projector Π, and two reals $0 < \varepsilon, \delta < 1$. With probability $1-\delta$, this algorithm outputs an estimate $\widetilde{p} = \mathsf{AmplEst}^\star(U, \Pi, \varepsilon, \delta)$ satisfying $|\widetilde{p} - p| \leq \varepsilon p$ and runs in time*

$$\mathcal{O}\left(\frac{T_{max}(U)}{\varepsilon\sqrt{p}} \cdot \log\left(\frac{1}{\delta}\right)\right)$$

where $p = \langle\psi|\Pi|\psi\rangle$ and $|\psi\rangle = U|0\rangle$.

We also use the following careful analysis of the amplitude amplification algorithm.

Proposition C.5 ([1, Lemma 5.2])**.** *Let $\mathcal{H}$ be some Hilbert space. Let U be a unitary operator and Π an orthogonal projector on $\mathcal{H}$. Denote $p = \langle\psi|\Pi|\psi\rangle$ where $|\psi\rangle = \sqrt{p}|\psi_\Pi\rangle + \sqrt{1-p}|\psi_{\Pi^\perp}\rangle = U|0\rangle$ and $|\psi_\Pi\rangle$, $|\psi_{\Pi^\perp}\rangle$ are two unit vectors invariant by Π and $\Pi^\perp$ respectively. Given an integer t such that*

$$t \leq \frac{\pi}{4\arcsin\sqrt{p}} - \frac{1}{2}$$

the Amplitude Amplification algorithm [13, Section 2] on input (U, Π, t) outputs in time $\mathcal{O}(t)$ the description of a quantum circuit $\mathsf{Amplify}(U, \Pi, t)$ acting on $\mathcal{H}$ such that

$$\mathsf{Amplify}(U, \Pi, t)|0\rangle = \sqrt{p'}|\psi_\Pi\rangle + \sqrt{1-p'}|\psi_{\Pi^\perp}\rangle$$

where

$$p' \geq \left(1 - \frac{(2t+1)^2}{3}p\right)(2t+1)^2 p$$

Moreover, $\mathsf{Amplify}(U, \Pi, t)$ runs in time $\mathcal{O}(t \cdot T_{max}(U))$.

C.2 Notations

For clarity, and without loss of generality, we assume that each intermediate state $|\psi_i\rangle = U_i \cdots U_1 |init\rangle$ of the variable-time algorithm $U = U_m \cdots U_1$ can be written as

$$\begin{cases} |\psi_i\rangle = \sqrt{p_{rej,\leq i}}|stop_i^0\rangle|\psi_i^0\rangle|0\rangle + \sqrt{p_{acc,\leq i}}|stop_i^1\rangle|\psi_i^1\rangle|1\rangle + \sqrt{p_{stop,>i}}|cont_i\rangle|\psi_i^2\rangle|2\rangle, & \text{for } i < m; \\ |\psi_m\rangle = |\psi\rangle = \sqrt{p_{rej}}|stop_m^0\rangle|\psi^0\rangle|0\rangle + \sqrt{p_{acc}}|stop_m^1\rangle|\psi^1\rangle|1\rangle, & \text{where } p = p_{acc}. \end{cases}$$

for some unit vectors $|stop_i^1\rangle, |stop_i^0\rangle \in \text{Span}\left(|cont\rangle^{\otimes i}\right)^{\perp} \otimes_{j=i+1}^{m} \mathcal{H}_{F_j} \otimes \mathcal{H}_C$, $|cont_i\rangle \in \text{Span}\left(|cont\rangle^{\otimes i}\right) \otimes_{j=i+1}^{m} \mathcal{H}_{F_j} \otimes \mathcal{H}_C$, $|\psi_i^0\rangle, |\psi_i^1\rangle, |\psi_i^2\rangle, |\psi^0\rangle, |\psi^1\rangle \in \mathcal{H}_{C'}$ where $\mathcal{H}_C = \mathcal{H}_{C'} \otimes \mathbb{C}^3$, and some probabilities $p_{acc,\leq i}$, $p_{rej,\leq i}$, $p_{stop,>i}$, p_{acc}, p_{rej}. The last register indicates if the computation is not finished (value 2), if it is finished and corresponds to the *accepting part* whose amplitude has to be estimated (value 1), or if it is finished and corresponds to the *rejecting part* (value 0). The proportion $1 - p_{stop,>i}$ of computation that is finished at step i is decomposed as $p_{acc,\leq i}$ for the accepting part and $p_{rej,\leq i}$ for the rejecting part. We assume that all the computations are finished at step m (i.e. $p_{stop,>m} = 0$, $p_{acc,\leq m} = p_{acc} = p$ and $p_{rej,>m} = p_{rej}$). We also denote $p_{rej,\leq 0} = p_{acc,\leq 0} = 0$, $p_{stop,>0} = 1$. Finally, we define the following two projectors on $\mathcal{H}_F \otimes \mathcal{H}_C$:

$$\begin{cases} \Pi_1 = I_{\mathcal{H}_F \otimes \mathcal{H}_{C'}} \otimes |1\rangle\langle 1|; \\ \Pi_{1,2} = I_{\mathcal{H}_F \otimes \mathcal{H}_{C'}} \otimes (|1\rangle\langle 1| + |2\rangle\langle 2|). \end{cases}$$

C.3 State generation algorithms

We recall the definition of the state generation algorithms $(\mathcal{B}_i)_i$ and $(\mathcal{A}_i)_i$ from [5].

Input: a variable-time algorithm $U = U_m \cdots U_1$ with stopping times $t_1 < \cdots < t_m$, a step $i \in \{1, \ldots, m\}$, a sequence of estimates $(\widetilde{b}_k)_{1 \leq k \leq i-1}$.
Output: a state generation algorithm $\mathcal{B}_i = \mathsf{Gen}_{\mathcal{B}}\left(U, i, (\widetilde{b}_k)_{1 \leq k \leq i-1}\right)$.

1. If $i = 1$, output $\mathcal{B}_1 = U_1$.
2. If $i > 1$, output $\mathcal{B}_i = U_i \mathcal{A}_{i-1}$ where $\mathcal{A}_{i-1} = \mathsf{Gen}_{\mathcal{A}}\left(U, i-1, (\widetilde{b}_k)_{1 \leq k \leq i-1}\right)$.

Algorithm 8: state generation algorithm $\mathsf{Gen}_{\mathcal{B}}$.

Input: a variable-time algorithm $U = U_m \cdots U_1$ with stopping times $t_1 < \cdots < t_m$, a step $i \in \{1, \ldots, m\}$, a sequence of estimates $(\widetilde{b}_k)_{1 \leq k \leq i}$.
Output: a state generation algorithm $\mathcal{A}_i = \mathsf{Gen}_{\mathcal{A}}\left(U, i, (\widetilde{b}_k)_{1 \leq k \leq i}\right)$.

1. Set $\mathcal{B}_i = \mathsf{Gen}_{\mathcal{B}}\left(U, i, (\widetilde{b}_k)_{1 \leq k \leq i-1}\right)$.
2. If $\widetilde{b}_i > \frac{1}{9m}$, output $\mathcal{A}_i = \mathcal{B}_i$.
3. If $\widetilde{b}_i \leq \frac{1}{9m}$, output $\mathcal{A}_i = \mathsf{Amplify}\left(\mathcal{B}_i, \Pi_{1,2}, k\right)$ for the smallest k satisfying $1/(9m) \leq (2k+1)^2 \widetilde{b}_i \leq 1/m$.

Algorithm 9: state generation algorithm $\mathsf{Gen}_{\mathcal{A}}$.

We let $|\psi_{\mathcal{B}_i}\rangle = \mathcal{B}_i|init\rangle$ and $|\psi_{\mathcal{A}_i}\rangle = \mathcal{A}_i|init\rangle$ denote the states generated by the $(\mathcal{B}_i)_i$ and $(\mathcal{A}_i)_i$ algorithms respectively. The goal of the $(\mathcal{A}_i)_i$ algorithms is to amplify at each intermediate step i the amplitude of the potentially accepting part $\sqrt{p_{acc,\leq i}}|stop_i^1\rangle|\psi_{\leq i}^1\rangle|1\rangle + \sqrt{p_{stop,>i}}|cont_i\rangle|\psi_{>i}^2\rangle|2\rangle$ into $|\psi_{\mathcal{B}_i}\rangle$ from $b_i = \|\Pi_{1,2}|\psi_{\mathcal{B}_i}\rangle\|^2$ to $a_i = \|\Pi_{1,2}|\psi_{\mathcal{A}_i}\rangle\|^2 \geq \max(b_i, \Omega(1/m))$. The goal of the $(\mathcal{B}_i)_i$ algorithms is to continue the execution of U: $|\psi_{\mathcal{B}_{i+1}}\rangle = U_{i+1}|\psi_{\mathcal{A}_i}\rangle$. Below we summarize the main results from [5] we need about these algorithms.

Proposition C.6 ([5])**.** *Consider a variable-time algorithm $U = U_m \cdots U_1$ with stopping times $t_1 < \cdots < t_m$, a step $i \in \{1,\ldots,m\}$ and a sequence of estimates $(\widetilde{b}_k)_{1\le k\le i}$. For each $1 \le j \le i$, denote $\mathcal{B}_j = \mathsf{Gen}_{\mathcal{B}}\left(U, j, (\widetilde{b}_k)_{1\le k\le j-1}\right)$, $\mathcal{A}_j = \mathsf{Gen}_{\mathcal{A}}\left(U, j, (\widetilde{b}_k)_{1\le k\le j}\right)$, and let $b_j = \|\Pi_{1,2}(\mathcal{B}_j|init\rangle)\|^2$, $a_j = \|\Pi_{1,2}(\mathcal{A}_j|init\rangle)\|^2$. We have that*

$$b_i = a_{i-1}\frac{1-p_{rej,\le i}}{1-p_{rej,\le i-1}} \tag{1}$$

where $a_0 = 0$. Moreover, if $|\widetilde{b}_j - b_j| \le b_j/(3m)$ for all $1 \le j \le i$, then the running time $T_{max}(\mathcal{A}_i)$ of $\mathcal{A}_i$ is

$$T_{max}(\mathcal{A}_i) \le C\sqrt{m}\left(t_i + i\frac{T_{\ell_2}(U)}{\sqrt{1-p_{rej,\le i}}}\right)$$

for some constant C, and

$$a_i \ge \left(1-\frac{1}{3m}\right)\frac{1}{9m}.$$

C.4 Variable-time amplitude estimation algorithm

We describe the two algorithms that constitute our variable-time amplitude estimation algorithm. First, we show how to approximate $p_{acc,\le i}$ for any step i (Algorithm 10). Then, we describe the algorithm proving Theorem C.2 (Algorithm 11). Our results rely on the following consequence of Equation 1.

Lemma C.7. *Using the notations of Proposition C.6, we have that*

$$p_{acc,\le i} = b_1 \cdot \prod_{j=2}^{i-1}\frac{b_j}{a_{j-1}} \cdot \frac{b_{i,1}}{a_{i-1}}$$

where $b_{i,1} = \|\Pi_1(\mathcal{B}_i|init\rangle)\|^2 = a_{i-1}\frac{p_{acc,\le i}}{1-p_{rej,\le i-1}}$.

Input: a variable-time algorithm $U = U_m \cdots U_1$, a step $i \in \{1,\ldots,m\}$, two reals $0 < \varepsilon, \delta < 1$.
Output: an estimate $\widetilde{p}_{acc,\le i}$ of $p_{acc,\le i}$.

1. For $j = 1,\ldots,i-1$:
   (a) Set $\mathcal{B}_j = \mathsf{Gen}_{\mathcal{B}}\left(U, j, (\widetilde{b}_k)_{1\le k\le j-1}\right)$ and compute $\widetilde{b}_j = \mathsf{AmplEst}^{\star}\left(\mathcal{B}_j, \Pi_{1,2}, \frac{\varepsilon}{4m}, \frac{\delta}{2m}\right)$.
   (b) Set $\mathcal{A}_j = \mathsf{Gen}_{\mathcal{A}}\left(U, j, (\widetilde{b}_k)_{1\le k\le j}\right)$ and compute $\widetilde{a}_j = \mathsf{AmplEst}^{\star}\left(\mathcal{A}_j, \Pi_{1,2}, \frac{\varepsilon}{8m}, \frac{\delta}{2m}\right)$.
2. Set $\mathcal{B}_i = \mathsf{Gen}_{\mathcal{B}}\left(U, i, (\widetilde{b}_k)_{1\le k\le i-1}\right)$ and compute $\widetilde{b}_{i,1} = \mathsf{AmplEst}^{\star}\left(\mathcal{B}_i, \Pi_1, \frac{\varepsilon}{4m}, \frac{\delta}{2m}\right)$.
3. Output $\widetilde{p}_{acc,\le i} = \widetilde{b}_1 \cdot \prod_{j=2}^{i-1}\frac{\widetilde{b}_j}{\widetilde{a}_{j-1}} \cdot \frac{\widetilde{b}_{i,1}}{\widetilde{a}_{i-1}}$.

Algorithm 10: estimation of $p_{acc,\le i}$.

Proposition C.8. *With probability $1-\delta$, Algorithm 10 outputs an estimate $\widetilde{p}_{acc,\le i}$ satisfying $|\widetilde{p}_{acc,\le i} - p_{acc,\le i}| \le \varepsilon p_{acc,\le i}$ and runs in time $\mathcal{O}\left(\frac{m^3}{\varepsilon}\sqrt{\frac{1-p_{rej,\le i}}{p_{acc,\le i}}}\left(t_i + i\frac{T_{\ell_2}(U)}{\sqrt{1-p_{rej,\le i}}}\right)\log\left(\frac{m}{\delta}\right)\right)$.*

Proof. Using Proposition C.4, together with a union bound over all the calls to $\mathsf{AmplEst}^{\star}$ in Algorithm 10, we can assume with probability $1-\delta$ that (for all j) $\widetilde{b}_j$ and $\widetilde{b}_{i,1}$ are $\frac{\varepsilon}{4m}$-approximations of b_j and $b_{i,1}$ respectively, and $\widetilde{a}_j$ is an $\frac{\varepsilon}{8m}$-approximation of a_j (which implies $\left|\frac{1}{\widetilde{a}_j} - \frac{1}{a_j}\right| \le \frac{\varepsilon}{4m}\cdot\frac{1}{a_j}$). Consequently,

$$\widetilde{b}_1 \cdot \prod_{j=2}^{i-1}\frac{\widetilde{b}_j}{\widetilde{a}_{j-1}} \cdot \frac{\widetilde{b}_{i,1}}{\widetilde{a}_{i-1}} \le \left(1+\frac{\varepsilon}{4m}\right)^{2i} b_1 \cdot \prod_{j=2}^{i-1}\frac{b_j}{a_{j-1}} \cdot \frac{b_{i,1}}{a_{i-1}} \le \left(1+\frac{4i}{4m}\varepsilon\right)\cdot p_{acc,\le i} \le (1+\varepsilon)p_{acc,\le i}$$

where we used Lemma C.7 and the inequalities $1+x \le e^x$ and $e^y - 1 \le 2y$ (for $y \in [0,1]$). On the other hand,

$$\widetilde{b}_1 \cdot \prod_{j=2}^{i-1} \frac{\widetilde{b}_j}{\widetilde{a}_{j-1}} \cdot \frac{\widetilde{b}_{i,1}}{\widetilde{a}_{i-1}} \ge \left(1 - \frac{\varepsilon}{4m}\right)^{2i} b_1 \cdot \prod_{j=2}^{i-1} \frac{b_j}{a_{j-1}} \cdot \frac{b_{i,1}}{a_{i-1}} \ge \left(1 - \frac{2i}{4m}\varepsilon\right) \cdot p_{acc,\le i} \ge (1-\varepsilon) p_{acc,\le i}$$

where we used Lemma C.7 and Bernoulli's inequality. Thus, $|\widetilde{p}_{acc,\le i} - p_{acc,\le i}| \le \varepsilon p_{acc,\le i}$.

We analyse the time complexity of the algorithm. Using the same union bound as above we can assume with probability $1-\delta$ that (for all j) Step 1.(a) runs in time $\mathcal{O}\left(\frac{m}{\varepsilon\sqrt{b_j}} T_{max}(\mathcal{B}_j)\log(m/\delta)\right)$, Step 1.(b) runs in time $\mathcal{O}\left(\frac{m}{\varepsilon\sqrt{a_j}} T_{max}(\mathcal{A}_j)\log(m/\delta)\right)$ and Step 2 runs in time $\mathcal{O}\left(\frac{m}{\varepsilon\sqrt{b_{i,1}}} T_{max}(\mathcal{B}_i)\log(m/\delta)\right)$. Moreover, observe that if $\widetilde{b}_j > \frac{1}{9m}$ then $a_j = b_j$ and $T_{max}(\mathcal{A}_j) \ge T_{max}(\mathcal{B}_j)$, and if $\widetilde{b}_j \le \frac{1}{9m}$ then $T_{max}(\mathcal{A}_j) = \Omega\left(\sqrt{\frac{a_j}{b_j}} T_{max}(\mathcal{B}_j)\right)$, by definitions of $(\mathcal{B}_j)_j$ and $(\mathcal{A}_j)_j$. In both cases we obtain $\frac{T_{max}(\mathcal{B}_j)}{\sqrt{b_j}} = \mathcal{O}\left(\frac{T_{max}(\mathcal{A}_j)}{\sqrt{a_j}}\right)$. Similarly, $\frac{T_{max}(\mathcal{B}_i)}{\sqrt{b_{i,1}}} = \mathcal{O}\left(\sqrt{\frac{b_i}{a_i b_{i,1}}} T_{max}(\mathcal{A}_i)\right)$. Consequently, using Proposition C.6, the total time complexity is $\mathcal{O}\left(\left(\sum_{j=1}^{i-1} \frac{m}{\varepsilon\sqrt{a_j}} T_{max}(\mathcal{A}_j) + \sqrt{\frac{b_i}{a_i b_{i,1}}} T_{max}(\mathcal{A}_i)\right)\log\left(\frac{m}{\delta}\right)\right) = \mathcal{O}\left(\frac{m^3}{\varepsilon}\sqrt{\frac{1-p_{rej,\le i}}{p_{acc,\le i}}}\left(t_i + i\frac{T_{\ell_2}(U)}{\sqrt{1-p_{rej,\le i}}}\right)\log\left(\frac{m}{\delta}\right)\right)$. □

In the following, we make the basic assumption (also used in [5, 18]) that $U = U_m \cdots U_1$ has stopping times $t_j = 2^j$, for $j = 1, \ldots, m$ and $m = \log(T_{max}(U))$.

Input: a variable-time algorithm $U = U_m \cdots U_1$ with stopping times $t_j = 2^j$ ($1 \le j \le m$), an integer t, a value $T_{\ell_2} \ge T_{\ell_2}(U)$, two reals $0 < \varepsilon, \delta < 1$.
Output: an estimate $\widetilde{p}_{acc}$ of p_{acc}.

1. Set $i = \min\left(m, \lceil \log(t\varepsilon^{-1/2} \cdot T_{\ell_2})\rceil\right)$ and $t' = 2D\frac{m^3}{\varepsilon}(t_i + i \cdot t \cdot T_{\ell_2})\log\left(\frac{m}{\delta}\right)$, where D is the constant hidden in the $\mathcal{O}(.)$ notation of Proposition C.8.
2. Run Algorithm 10 with input U, i, $\varepsilon/2$, δ for at most t' computation steps.
   (a) If the computation has not ended after t' steps, stop it and output $\widetilde{p}_{acc} = 0$.
   (b) Else, let $\widetilde{p}_{acc,\le i}$ denote the result of Algorithm 10. If $\widetilde{p}_{acc,\le i} = 0$ or $t < 1/\sqrt{\widetilde{p}_{acc,\le i}}$ then output $\widetilde{p}_{acc} = 0$, else output $\widetilde{p}_{acc} = \widetilde{p}_{acc,\le i}$.

Algorithm 11: estimation of p_{acc}.

Proof of Theorem C.2. We show that Algorithm 11 satisfies the statements of Theorem C.2.

Assume first that $t \ge \frac{2}{\sqrt{p_{acc}}}$. Since $T_{\ell_2} \ge T_{\ell_2}(U) \ge \sqrt{p_{stop,>i} \cdot t_i^2} = \sqrt{p_{stop,>i} \cdot 2^{2i}}$ for all i, by choosing $i = \min\left(m, \lceil \log(t\varepsilon^{-1/2} \cdot T_{\ell_2})\rceil\right)$ we obtain $p_{stop,>i} \le T_{\ell_2}^2/t_i \le (\varepsilon/4) \cdot p_{acc}$. Thus $p_{acc,\le i}$ satisfies

$$p_{acc} \ge p_{acc,\le i} \ge p_{acc} - p_{stop,>i} \ge (1 - \varepsilon/4) \cdot p_{acc}$$

and $1 - p_{rej,\le i} \le p_{acc} + p_{stop,>i} \le 2p_{acc}$. It implies $D\frac{m^3}{\varepsilon}\sqrt{\frac{1-p_{rej,\le i}}{p_{acc,\le i}}}\left(t_i + i\frac{T_{\ell_2}(U)}{\sqrt{1-p_{rej,\le i}}}\right)\log\left(\frac{m}{\delta}\right) < t'$. Consequently, according to Proposition C.8, with probability $1-\delta$ the computation does not stop at Step 2 and $\widetilde{p}_{acc,\le i}$ satisfies $|\widetilde{p}_{acc,\le i} - p_{acc,\le i}| \le (\varepsilon/2) \cdot p_{acc,\le i}$. In this case, using the triangle inequality, we have $|\widetilde{p}_{acc,\le i} - p_{acc}| \le \varepsilon \cdot p_{acc}$ and $1/\sqrt{\widetilde{p}_{acc,\le i}} \le \sqrt{2/p_{acc}} \le t$.

Assume now that $t < \frac{2}{\sqrt{p_{acc}}}$. According to Proposition C.8, the output $\widetilde{p}_{acc,\le i}$ of Algorithm 10 satisfies $\widetilde{p}_{acc,\le i} \le (1+\varepsilon/2)p_{acc,\le i} \le 2p_{acc}$ with probability $1-\delta$. Since the output $\widetilde{p}_{acc}$ of Algorithm 11 is either 0 or $\widetilde{p}_{acc,\le i}$, it also satisfies $\widetilde{p}_{acc} \le 2p_{acc}$ with probability $1-\delta$. Finally, if $t < \frac{1}{\sqrt{2p_{acc}}}$ and $0 \ne \widetilde{p}_{acc,\le i} \le 2p_{acc}$ then $t < \frac{1}{\sqrt{\widetilde{p}_{acc,\le i}}}$ and $\widetilde{p}_{acc} = 0$. □

D Making streaming algorithms reversible

Reversibility is an intrinsic property of quantum computing that we often used in this paper. It is known that any deterministic computation can be made reversible, and therefore implemented by a unitary map with a limited overhead on the time and space complexities [11]. Nonetheless, implementing the reverse computation of a streaming algorithm would require processing the same stream but in the *reverse* direction, which may not be always possible. This motivates our specific notion of *reversible streaming algorithms*. We say that a streaming algorithm $\mathcal{T}$ with memory size M is *reversible* if there exists a streaming algorithm $\mathcal{T}^{-1}$ with memory size M such that each computational steps of $\mathcal{T}$ and $\mathcal{T}^{-1}$ are reversible, and in addition each pass of $\mathcal{T}$ can be undone by one pass of $\mathcal{T}^{-1}$ in the *same* direction.

Even if it is not clear how to make any streaming algorithm reversible, it is sufficient for our purpose to show how to achieve it when the streaming algorithm is a *linear sketch*.

Definition D.1. *We say that a (one-pass) streaming algorithm $\mathcal{T}$ is a* linear sketch algorithm *with memory M, update time T_{upd} and reconstruction time T_{rec} if there exists a family $\{L_r\}_{r\in\{0,1\}^M}$ of linear functions $L_r : \mathbb{R}^n \to \mathbb{R}^M$, and two deterministic algorithms $\mathcal{A}_{upd}$ and $\mathcal{A}_{rec}$ running in time T_{upd} and T_{rec} (respectively) and space $\mathcal{O}(M)$, such that $\mathcal{T}$ behaves as follows:*

1. *Draw $r \in \{0,1\}^M$ uniformly at random and store it in memory. Initialize $\mathtt{L} = 0$.*
2. *Given $u_j = (i,\lambda)$, apply $\mathcal{A}_{upd}$ on input r, u_j to compute $L_r(\lambda e_i)$ and update $\mathtt{L} \leftarrow \mathtt{L} + L_r(\lambda e_i)$*
3. *At the end of the stream, apply $\mathcal{A}_{rec}$ on input r, $\mathtt{L}$ to compute the output of the algorithm*

Observe that, by linearity of L_r, the value of $\mathtt{L}$ in Definition D.1 after the j-th item has been processed is $\mathtt{L} = L_r(x(j))$. Linear sketch algorithms play an important role in the turnstile model, since they can implement essentially *all* streaming algorithms [48, 3]. Moreover, they are highly parallelizable, which facilitates their adaptation to the multi-pass model. In addition they can be made reversible as proved below. This property stems from the fact that the content of the memory, at any step of the computation, is unchanged under any permutation of the order of arrival of the updates received so far (because of the linearity of L_r).

Proposition D.2. *For any linear sketch algorithm $\mathcal{T}$ with parameters (M, T_{upd}, T_{rec}), there exists a* reversible *streaming algorithm $\mathcal{R}(\mathcal{T})$ with memory size $\mathcal{O}\left(M \cdot \log\left(T_{upd} \cdot T_{rec}\right)\right)$ that computes the same output as $\mathcal{T}$.*

Proof. First we observe from [11] that any (non-streaming) classical algorithm $\mathcal{A}$ can be turned into a reversible one $\mathcal{R}(\mathcal{A})$ that computes the same output as $\mathcal{A}$, performs T^2 computation steps and uses $\mathcal{O}(M \log T)$ memory cells.

We assume that the random seed $r \in \{0,1\}^M$ is pre-loaded in memory. Algorithm $\mathcal{R}(\mathcal{T})$ is implemented as follows. For each update $u(j) = (i,\lambda)$, use algorithm $\mathcal{R}(\mathcal{A}_{upd})$ to compute reversibly $L_r(\lambda e_i)$, copy the result to $\mathtt{L} \leftarrow \mathtt{L} + L_r(\lambda e_i)$, and undo the computation of $L_r(\lambda e_i)$ with $\mathcal{R}(\mathcal{A}_{upd})^{-1}$. The reconstruction part is done at the end of the stream using $\mathcal{R}(\mathcal{A}_{rec})$.

The reverse algorithm $\mathcal{R}(\mathcal{T})^{-1}$ first uncomputes the reconstruction part using $\mathcal{R}(\mathcal{A}_{rec})^{-1}$. Then, for each update $u(j) = (i,\lambda)$, it computes $L_r(\lambda e_i)$ with $\mathcal{R}(\mathcal{A}_{upd})$, updates $\mathtt{L} \leftarrow \mathtt{L} - L_r(\lambda e_i)$, and uncomputes $L_r(\lambda e_i)$ using $\mathcal{R}(\mathcal{A}_{upd})^{-1}$. □

E Approximating graph parameters in the query model

We fix a few notations that are used in the next two sections.

Notations E.1. *Let $G = (V,E)$ be a graph, where $V = [n]$ for some integer n. For each vertex $v \in V$, we let N_v equal the set of neighbor vertices to v, E_v the set of edges adjacent to v, and $d_v = |N_v| = |E_v|$ the degree of v. Similarly, T_v is the set of triangles adjacent to v, and $t_v = |T_v|$ its cardinality. We define the total order $\prec$ on $V = [n]$ where $u \prec v$ if $d_u < d_v$, or $d_u = d_v$ and $u < v$ (where $<$ is the natural order on $[n]$). We let d_v^+ equal the number of neighbors w of v such that $d_v \prec d_w$.*

Fact E.2. *For all vertex $v \in V$, we have $d_v^+ \leq \sqrt{2m}$.*

We will also use the following combination of Theorems B.1 and C.3.

Theorem E.3. *There is an algorithm that takes as input a variable-time sampler $\mathcal{S}$, a function $f : x \mapsto A/x^\alpha$ for some reals $A, \alpha > 0$, two values $0 < L < H$, a real $T_{\ell_2} \geq 1$, and two reals $0 < \varepsilon, \delta < 1$ with $\delta < 2^{-2\alpha}$. If $f(\mu_\mathcal{S}) \geq \phi_\mathcal{S}/\mu_\mathcal{S}$, $T_{\ell_2} \geq T_{\ell_2}(\mathcal{S})$ and $L \leq \mu_\mathcal{S} < H$, this algorithm outputs an estimate $\widetilde{\mu}_\mathcal{S}$ that satisfies $|\widetilde{\mu}_\mathcal{S} - \mu_\mathcal{S}| \leq \varepsilon\mu_\mathcal{S}$ with probability $1-\delta$, and it uses*

$$\widetilde{\mathcal{O}}\left(f(\max(L,\mu_\mathcal{S})) \cdot T_{\ell_2} \cdot \varepsilon^{-2} \log^4(T_{max}(\mathcal{S})) \log\left(\frac{H}{L}\right) \log\left(\frac{1}{\delta}\right)\right)$$

quantum samples in expectation (both for the ℓ_1 and ℓ_2 average).

E.1 Approximating the number of edges

We show how to approximate the number m of edges with $\widetilde{\mathcal{O}}\left(n^{1/2}/(\varepsilon m^{1/4})\right)$ quantum queries in expectation. We need the following estimator from Seshadhri [58].

Input: query access to a graph $G = (V,E)$.
Output: an estimate of $m = |E|$.
1. Sample $v \in V$ uniformly at random. Sample $w \in N_v$ uniformly at random.
2. If $v \prec w$, output nd_v, else output 0.

Estimator 12: number m of edges in a graph $G = (V,E)$ (from [58]).

Proposition E.4. *If we let X denote the output random variable of Estimator 12, then $\mathbb{E}[X] = m$ and $\mathbb{E}\left[X^2\right] \leq 2\sqrt{2}nm^{3/2}$.*

Proof. On the one hand, $\mathbb{E}[X] = n^{-1}\sum_v (d_v^+/d_v) \cdot nd_v = \sum_v d_v^+ = m$. On the other hand, $\mathbb{E}\left[X^2\right] = n\sum_v d_v^+ \cdot d_v \leq 2\sqrt{2}nm^{3/2}$, where we used Fact E.2. □

We can now prove Theorem 5.4.

Proof of Theorem 5.4. We can implement Estimator 12 with a sampler $\mathcal{S}$ that computes, in constant time,

$$\mathcal{S}(|0\rangle|0\rangle) = \sum_{v\in V} \sum_{w\in N_v} |v\rangle|w\rangle|\lambda(v,w)\rangle$$

where $\lambda(v,w) = nd_v$ if $v \prec w$, and $\lambda(v,w) = 0$ otherwise. According to Proposition E.4, we have $\mu_\mathcal{S} = m$ and $\phi_\mathcal{S}/\mu_\mathcal{S} \leq 8^{1/4}n^{1/2}/m^{1/4}$. Consequently, using Corollary B.2 with $f : x \mapsto 8^{1/4}n^{1/2}/x^{1/4}$, $L = 1$, $H = n^2$ and $\delta = 1/3$, we can estimate $\widetilde{m}$ with accuracy ε and success probability $2/3$ using $\widetilde{\mathcal{O}}\left(\frac{n^{1/2}}{\varepsilon m^{1/4}}\right)$ quantum samples in expectation. □

Lower bound We obtain a nearly matching lower bound by using a reduction from the two-player communication problem DISJOINTNESS. The proof is based on a construction from [29].

Proof of Theorem 5.5. Fix n, m, $\varepsilon < 1/4$. Given an instance $(x,y) \in \{0,1\}^N \times \{0,1\}^N$ of size $N = n/(2\sqrt{4\varepsilon m})$ for DISJOINTNESS, we construct a graph $G_{x,y}$ on n vertices such that

$$\begin{cases} \text{DISJOINTNESS}(x,y) = 1 \iff G_{x,y} \text{ has exactly } m \text{ edges} \\ \text{DISJOINTNESS}(x,y) = 0 \iff G_{x,y} \text{ has at least } (1+4\varepsilon)m \text{ edges.} \end{cases}$$

The construction is as follows (see [29, Section 4.1]): fix any graph H with $n/2$ vertices and m edges, use half of the n vertices in $G_{x,y}$ to construct a subgraph isomorphic to H, and partition the remaining $n/2$ vertices into N sets $K_1,\dots,K_N$ of size $\sqrt{4\varepsilon m}$. If $x_j = y_j = 1$ then K_j is a clique, otherwise it is a set of isolated vertices. It is clear that at least one K_j is a clique if and only if $\textsc{Disjointness}(x,y) = 0$.

Consider now an algorithm that approximates with relative error ε the number of edges in any graph G with n vertices and m edges using at most Q quantum queries. Using the reduction above, it can be used on input $G_{x,y}$ to deduce the value of $\textsc{Disjointness}(x,y)$. We show how to implement it into a communication protocol of cost $\mathcal{O}(Q\log n)$ on input (x,y), using a standard technique from [15]. Alice runs the Q-query algorithm for $G_{x,y}$. When there is a vertex-pair query, her state is in a superposition $\sum_{v,w,b}\alpha_{v,w}|v,w\rangle|b\rangle|\phi_{v,w}\rangle$ over all pair of vertices (v,w) in $G_{x,y}$. She has to compute $\sum_{v,w,b}\alpha_{v,w}|v,w\rangle|b\oplus e_{v,w}\rangle|\phi_{v,w}\rangle$ where $e_{v,w}=1$ if and only if there is an edge between v and w. If (v,w) is an edge from the subgraph isomorphic to H, she can map directly $|v,w\rangle|b\rangle \mapsto |v,w\rangle|b\oplus 1\rangle$. If v and w belong to a same K_j, she appends $|0\rangle$ to $|v,w\rangle|b\rangle$, computes $|v,w\rangle|b\rangle|0\rangle \mapsto |v,w\rangle|b\rangle|x_j\rangle$, and sends the three registers to Bob. Then, Bob computes $|v,w\rangle|b\rangle|x_j\rangle \mapsto |v,w\rangle|b\oplus(x_j\cdot y_j)\rangle|x_j\rangle = |v,w\rangle|b\oplus e_{v,w)}\rangle|x_j\rangle$ and sends the result back to Alice who maps $|v,w\rangle|b\oplus e_{v,w}\rangle|x_j\rangle \mapsto |v,w\rangle|b\oplus e_{v,w)}\rangle|0\rangle$ to obtain the desired result. The degree and neighbor queries are implemented similarly. Each query requires $\mathcal{O}(\log n)$ qubits of communication, hence the total communication cost is $\mathcal{O}(Q\log n)$. Since the quantum communication complexity of any protocol computing $\textsc{Disjointness}$ must be $\Omega\left(\sqrt{N}\right)$ [57], we obtain that $Q = \Omega(\sqrt{N}/\log n) = \Omega\left(\frac{n^{1/2}}{(\varepsilon m)^{1/4}}\cdot\log^{-1}(n)\right)$. □

E.2 Approximating the number of triangles

We show how to approximate the number t of triangles with $\widetilde{\mathcal{O}}\left(\frac{\sqrt{n}}{t^{1/6}}+\frac{m^{3/4}}{\sqrt{t}}\right)$ quantum queries in expectation. In order to keep this section concise, we describe an algorithm that computes a $(4/5+\varepsilon)$-approximation of t, though it is possible to obtain an ε-approximation with similar ideas.

We begin with a simple estimator from [26] for approximating the number t_v of triangles adjacent to a given vertex $v\in V$.

Input: query access to a graph $G=(V,E)$, a vertex $v\in V$.
Output: an estimate of t_v/d_v.

1. Sample $e\in E_v$ uniformly at random. Let w be the endpoint of e that is not v. Let u be the smaller endpoint of e according to $\prec$.
2. If $d_u\le\sqrt{2m}$, set $r=1$ with probability $d_u/\sqrt{2m}$, output 0 otherwise. If $d_u>\sqrt{2m}$, set $r=\lceil d_u/\sqrt{2m}\rceil$.
3. For $i=1,\dots,r$:
   (a) Pick a neighbor x of u uniformly at random.
   (b) If e and x form a triangle and $w\prec x$, set $X_i=\max(d_u,\sqrt{2m})$. Else, set $X_i=0$.
4. Output $\frac{1}{r}\sum_{i=1}^{r}X_i$.

Estimator 13: ratio of the number of adjacent triangles t_v to the degree d_v of a vertex v (from [26]).

Proposition E.5. *If we let X denote the output random variable of Estimator 13, then $\mathbb{E}[X]=t_v/d_v$ and $\mathrm{Var}[X]\le 2\sqrt{2m}t_v/d_v$. Moreover, the ℓ_2-average running time of Estimator 13 is $\mathcal{O}(1)$.*

Proof. For each edge $e=(v,w)$, we let $t_{e,v}$ be the number of triangles (v,w,x) such that $w\prec x$. It is clear that $t_v=\sum_{e\in E_v}t_{e,v}$. Moreover, $t_{e,v}\le\sqrt{2m}$. Indeed, either $d_w\le\sqrt{2m}$ (and thus $t_{e,v}\le d_w\le\sqrt{2m}$), or $d_w>\sqrt{2m}$ and in this case w cannot have more than $\sqrt{2m}$ neighbors of degree at least $\sqrt{2m}$.

We first compute the mean of X conditionned on the edge e chosen at Step 1 and the value taken by d_u. We have $\mathbb{E}\left[X|e,d_u\le\sqrt{2m}\right]=(d_u/\sqrt{2m})\cdot(t_{e,v}/d_u)\cdot\sqrt{2m}=t_{e,v}$ and $\mathbb{E}\left[X|e,d_u>\sqrt{2m}\right]=(t_{e,v}/d_u)\cdot d_u=t_{e,v}$. Consequently, $\mathbb{E}[X]=\frac{1}{d_v}\sum_{e\in E_v}\mathbb{E}[X|e]=t_v/d_v$. Similarly, $\mathrm{Var}\left[X^2|e,d_u\le\sqrt{2m}\right]\le\mathbb{E}\left[X^2|e,d_u\le\sqrt{2m}\right]=\sqrt{2m}t_{e,v}$ and $\mathrm{Var}\left[X^2|e,d_u>\sqrt{2m}\right]\le(\sqrt{2m}/d_u)\cdot\mathbb{E}\left[X_i^2|e,d_u>\sqrt{2m}\right]\le(\sqrt{2m}/d_u)\cdot(t_{e,v}/d_u)\cdot d_u^2=\sqrt{2m}t_{e,v}$.

Thus, using the low of total variance, $\mathrm{Var}[X] \leq \frac{1}{d_v}\sum_{e\in E_v}(\sqrt{2m}t_{e,v} + t_{e,v}^2)$. Since $t_{e,v} \leq \sqrt{2m}$, it implies $\mathrm{Var}[X] \leq 2\sqrt{2m}t_v/d_v$. Finally, the ℓ_2-average running time of Step 3 is $\frac{1}{d_v}\sum_{w\in N_v}\left(\frac{\min(d_v,d_w)}{\sqrt{2m}}\right)^2 \leq \frac{1}{2md_v}\sum_{w\in N_v} d_v d_w \leq \mathcal{O}(1)$. The other steps of the estimator run in constant time. □

Proposition E.6. *There is a quantum algorithm that, given query access to any n-vertex graph G with m edges, a vertex $v \in V$, an integer L, an approximation parameter $\varepsilon < 1$ and a failure parameter $\delta < 2^{-1}$, outputs an estimate $\widetilde{t_v}$ of the number t_v of triangles adjacent to v. If $L \leq t_v$, this estimate satisfies $|\widetilde{t_v} - t_v| \leq \varepsilon t_v$ with probability $1-\delta$. Moreover, for any L, it satisfies $\widetilde{t_v} \leq 2t_v$ with probability $1-\delta$. The ℓ_2-average running time of this algorithm, including its number of queries, is $\widetilde{\mathcal{O}}\left(\left(1+\frac{m^{1/4}\sqrt{d_v}}{\varepsilon^2\sqrt{L}}\right)\cdot\log(1/\delta)\right)$.*

Proof. It is straightforward to implement Estimator 13 with a quantum sampler $\mathcal{S}$, in a similar way as we did in the proof of Theorem 5.4. This sampler satisfies $\mu_{\mathcal{S}} = t_v/d_v$ and $\phi_{\mathcal{S}}/\mu_{\mathcal{S}} \leq 1 + (8m)^{1/4}\sqrt{d_v/t_v}$ according to Proposition E.5. Moreover, its ℓ_2-*average* running time is $T_{\ell_2}(\mathcal{S}) = \mathcal{O}(1)$. We estimate t_v by applying Theorem E.3 on $\mathcal{S}$ with $f : x \mapsto 1 + (cm)^{1/4}\sqrt{d_v/x}$ (for a small enough constant c), $L' = L/d_v$ and $H = n^2$. The ℓ_2-average running time of this algorithm is $\widetilde{\mathcal{O}}\left(\left(1+\frac{m^{1/4}\sqrt{d_v}}{\varepsilon^2\sqrt{L}}\right)\cdot\log(1/\delta)\right)$. □

The remaining part of our algorithm diverges from the approach taken in [26], that requires to set up a data structure for sampling edges uniformly in G. This technique seems to be an obstacle for improving the term $\mathcal{O}\left(m^{3/2}/t\right)$ in the complexity. We circumvent this problem by combining [26] with a bucketing approach from [25], that partitions the graph's vertices into $k+1 = \mathcal{O}(\log n)$ buckets $B_0, \ldots, B_k$, where

$$B_i = \{v \in V : t_v \in [(1+c)^{i-1}, (1+c)^i]\}$$

for a small value $0 < c < 1$ to be chosen later. If we estimate the size $b_i = |B_i|$ of each bucket, then we would obtain an approximation of $\frac{1}{3}\sum_i |B_i|\cdot(1+c)^i \in [t, (1+c)t]$. We first show that the smallest sizes $|B_i|$ can be discarded, at the cost of a certain factor in the approximation.

Lemma E.7. *If $I^+ \subseteq \{0,\ldots,k\}$ denotes the set of indices i such that $|B_i| \geq \frac{(ct)^{1/3}}{k+1}$ and $|B_i| \geq \frac{ct}{(k+1)(1+c)^i}$, then*

$$\frac{(1-2c)}{3}t \leq \frac{1}{3}\sum_{i\in I^+}|B_i|\cdot(1+c)^i \leq (1+c)t$$

Proof. Define $B(v)$ to be the bucket that $v \in V$ belongs to, and let $V_{bad,1} = \left\{v \in V : |B(v)| < \frac{(ct)^{1/3}}{k+1}\right\}$ and $V_{bad,2} = \left\{v \in V : |B(v)| < \frac{ct}{(k+1)(1+c)^i}\right\}$. There are at most $(ct)^{1/3}$ vertices in $V_{bad,1}$. Consequently, at most ct triangles have their three endpoints in V_{bad}. It implies $\sum_{v\in V_{bad,1}} t_v < 3ct + 2(1-c)t$. On the other hand, we have $\sum_{v\in V_{bad,2}} t_v \leq \sum_{i:|B_i|<\frac{ct}{(k+1)(1+c)^i}} |B_i|\cdot(1+c)^i < ct$. Consequently, $\frac{1}{3}\sum_{i\in I^+}|B_i|\cdot(1+c)^i \geq t - \frac{1}{3}\sum_{v\in V_{bad,1}\cup V_{bad,2}} t_v > \frac{1}{3}(1-2c)t$. □

We are now ready to state the main result of this section.

Theorem E.8. *There is a quantum algorithm that, given query access to an n-vertex graph G with m edges and an approximation parameter $\varepsilon < 1$, outputs an estimate $\widetilde{t}$ of the number t of triangles of G such that $|\widetilde{t} - t| \leq (4/5+\varepsilon)t$ with probability 2/3. This algorithm performs $\widetilde{\mathcal{O}}\left(\left(\frac{\sqrt{n}}{t^{1/6}} + \frac{m^{3/4}}{\sqrt{t}}\right)\cdot poly(1/\varepsilon)\right)$ queries in expectation.*

Sketch of the proof. In the following, we assume that the threshold values $\frac{(ct)^{1/3}}{k+1}$ and $\frac{ct}{(k+1)(1+c)^i}$ used to define I^+ are known, although t is part of their definitions. In fact, it is easy to see that if t is replaced with

any value $\bar{t}$ in these expressions then the output of the algorithm described below will likely be smaller than $\bar{t}$ when $\bar{t} > 20t$, and it will likely be larger than $\bar{t}$ when $\bar{t} < t/20$. Thus, it suffices to perform a logarithmic search on $\bar{t}$ (starting with $\bar{t} = n^3$) to approximate the right threshold values.

The general appropach of the algorithm is to compute separately an estimate $\widetilde{b}_i$ of the size of each B_i for $i \in I^+$, and then to recombine them into $\sum_{i\in I^+} \widetilde{b}_i \cdot (1+c)^i$. If we had access to an oracle that returns t_v for each $v \in V$, then it would suffice to perform order of $\sqrt{n/|B_i|}$ quantum queries for estimating $|B_i|$. Instead, we use the algorithm of Proposition E.6 with threshold $L = (1+c)^{i-1}$ to decide if $v \in B_i$. Since we cannot distinguish efficiently $v \in B_i$ from $v \in B_{i+1}$ when t_v is close to $(1+c)^i$, we are estimating a value between $|B_i|$ and $|B_{i-1}| + |B_i| + |B_{i+1}|$ instead. This adds a factor of $(1+c)^{-1} + 1 + (1+c) \leq 3 + c$ to the final approximation.

In more details, we assign $v \in V$ to bucket B_i if the output $\widetilde{t_v}$ of the algorithm of Proposition E.6 with input v, $L = (1+c)^{i-1}$, $\varepsilon' = c/2$, $\delta = \varepsilon/poly(n)$ satisfies $\widetilde{t_v} \in [(1+c)^{i-1}, (1+c)^i]$. We apply this algorithm on a superposition over all vertices $v \in V$ to obtain a quantum sampler $\mathcal{S}_i(|0\rangle|0\rangle) = n^{-1}\sum_{v\in V} |v\rangle|\psi_v\rangle|e_v\rangle$ over $\Omega = \{0,1\}$, where $|\psi_v\rangle$ is some garbage state, and $|e_v\rangle$ is a one-qubit state that equals $|1\rangle$ to indicate $v \in B_i$, and $|0\rangle$ otherwise. This sampler implements a Bernoulli distribution of mean $\mu_{\mathcal{S}} \in [(1-\varepsilon/8)|B_i|, (1+\varepsilon/8)(3+c)|B_i|]$ (the $\varepsilon/8$ error comes from the fact that the algorithm of Proposition E.6 has probability $\delta = \varepsilon/poly(n)$ to fail).

According to Proposition E.6, the ℓ_2-average running time to compute each $|\psi_v\rangle|e_v\rangle$ is of the order of $\widetilde{\mathcal{O}}\left(\left(1 + \frac{m^{1/4}\sqrt{d_v}}{\varepsilon^2\sqrt{(1+c)^{i-1}}}\right)\log\left(\frac{n}{\varepsilon}\right)\right)$. Thus, the ℓ_2-average running time of $\mathcal{S}_i$ is

$$\widetilde{\mathcal{O}}\left(\left(1 + \sqrt{\frac{1}{n}\sum_{v\in V}\left(\frac{m^{1/4}\sqrt{d_v}}{\varepsilon^2\sqrt{(1+c)^{i-1}}}\right)^2}\right)\log\left(\frac{n}{\varepsilon}\right)\right) = \widetilde{\mathcal{O}}\left(\left(1 + \frac{m^{3/4}}{\varepsilon^2\sqrt{n(1+c)^{i-1}}}\right)\log\left(\frac{n}{\varepsilon}\right)\right)$$

We apply the algorithm of Theorem C.3 on input $\mathcal{S}_i$, $\Delta_{\mathcal{S}_i} = \sqrt{n/\max\left(\frac{(ct)^{1/3}}{k+1}, \frac{ct}{(k+1)(1+c)^i}\right)}$, $H = n$, $L = 1$, $T_{\ell_2} = \widetilde{\mathcal{O}}\left(\left(1 + \frac{m^{3/4}}{\varepsilon^2\sqrt{n(1+c)^{i-1}}}\right)\log\left(\frac{n}{\varepsilon}\right)\right)$, $\varepsilon' = \varepsilon/8$ and $\delta = \mathcal{O}(1/\log(n))$ to obtain an estimate $\widetilde{b}_i \in \left[(1-\varepsilon/8)^2|B_i|, (1+\varepsilon/8)^2(3+c)|B_i|\right]$, in time

$$\widetilde{\mathcal{O}}\left(\sqrt{\frac{n}{\max\left(\frac{(ct)^{1/3}}{k+1}, \frac{ct}{(k+1)(1+c)^i}\right)}}\left(1 + \frac{m^{3/4}}{\sqrt{n(1+c)^i}}\right)\cdot poly(1/\varepsilon)\right) = \widetilde{\mathcal{O}}\left(\left(\frac{\sqrt{n}}{t^{1/6}} + \frac{m^{3/4}}{\sqrt{t}}\right)\cdot poly(1/\varepsilon)\right)$$

Finally, we choose $c = \varepsilon/4$ to define the buckets' width, which implies $\frac{1}{3}\sum_{i\in I^+}|B_i|\cdot(1+c)^i \in [\frac{1}{3}(1-\varepsilon/2)t, (1+\varepsilon/4)t]$ according to Lemma E.7, and $\widetilde{b}_i \in [(1-\varepsilon/4)|B_i|, 3(1+\varepsilon/4)|B_i|]$ with large probability. Thus, $\frac{1}{3}\sum_{i\in I^+}\widetilde{b}_i\cdot(1+c)^i \in [\frac{1}{3}(1-\varepsilon)t, 3(1+\varepsilon)t]$. Consequently, for $\widetilde{t} = \frac{1}{5}\sum_{i\in I^+}\widetilde{b}_i\cdot(1+c)^i$, we have $|\widetilde{t}-t| \leq (4/5+\varepsilon)t$ with large probability. □

The approximation factor can be improved from $(4/5+\varepsilon)$ to ε, by using a refined algorithm that combines techniques from [25] and [26]. The first main idea is to randomly perturbate the buckets' boundaries (see [25, Section 3.3.1]) to ensure that few vertices are close to them (this removes the previous factor $3(1+c)$ in the approximation). The second main idea is to modified the estimator used in Proposition E.6 to compensate the loss introduced by discarding the buckets outside of I^+. This leads to Theorem 5.6.

Lower bound A nearly matching lower bound can be obtained with the same method as in Theorem 5.5, using the constructions given in Sections 4.1 and 4.3 of [29] for the reduction to DISJOINTNESS. This leads to Theorem 5.7.