TheoretiCS

Multidimensional Quantum Walks, with Application to k-Distinctness

Received Sep 27, 2023
Revised Aug 28, 2024
Accepted Oct 28, 2024
Published Mar 4, 2025

Key words and phrases
quantum walk, random walk, k-distinctness, element distinctness, welded trees, quantum algorithm

Stacey Jeffery [a]
Sebastian Zur [a]

a CWI & QuSoft, the Netherlands

ABSTRACT. While the quantum query complexity of k-distinctness is known to be $O(n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^k-1}})$ for any constant $k \geq 4$ [Belovs, FOCS 2012], the best previous upper bound on the time complexity was $\widetilde{O}(n^{1-1/k})$. We give a new upper bound of $\widetilde{O}(n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^k-1}})$ on the time complexity, matching the query complexity up to polylogarithmic factors. In order to achieve this upper bound, we give a new technique for designing quantum walk search algorithms, which is an extension of the electric network framework. We also show how to solve the welded trees problem in $O(n)$ queries and $O(n^2)$ time using this new technique, showing that the new quantum walk framework can achieve exponential speedups.

1. Introduction

In the problem of *element distinctness*, the input is a list of n integers, and the output is a bit indicating whether the integers are all distinct, or there exists a pair of integers that are the same, called a *collision*. This problem has been studied as a fundamental problem in query complexity, but also for its relationship to other more practical problems, such as sorting, or *collision finding*, which is similar, but one generally assumes there are many collisions and one wants to find one. In the worst case, element distinctness requires $\Theta(n)$ classical queries [2].

The first quantum algorithm to improve on this was a $O(n^{3/4})$ query algorithm [16], which is a variation of an optimal quantum algorithm for collision finding [15], whose main technique is amplitude amplification [14]. The algorithm of [16] could also be implemented time efficiently,

This work is supported by ERC STG grant 101040624-ASC-Q, NWO Klein project number OCENW.Klein.061, and ARO contract no W911NF2010327. SJ is a CIFAR Fellow in the Quantum Information Science Program. A preliminary version of this article appeared at STOC 2023 [**27**].

in $\widetilde{O}(n^{3/4})$ steps, with a log factor overhead from storing large subsets of the input in a sorted data structure. This was later improved to $O(n^{2/3})$ queries, and $\widetilde{O}(n^{2/3})$ time by Ambainis [4], which is optimal [1]. Ambainis' algorithm has been modified to solve other problems in various domains, from k-sum [20], to path finding in isogeny graphs [35, 24]. Moreover, this algorithm was a critical step in our understanding of quantum query complexity, and quantum algorithms in general, as the algorithm used a new technique that was later generalised by Szegedy into a generic speedup for random walk search algorithms of a particular form [34].

For any constant integer $k \geq 2$, the problem *k-distinctness* is to decide if an input list of integers contains k copies of the same integer. When $k = 2$, this is exactly element distinctness. Ambainis [4] actually gave a quantum algorithm for k-distinctness for any $k \geq 2$, with query complexity $O(n^{1-1/(k+1)})$, and time complexity $\widetilde{O}(n^{1-1/(k+1)})$. For $k \geq 3$, Belovs gave an improved quantum query upper bound of $O(n^{3/4-\frac{1}{4}\frac{1}{2^k-1}})$ [9], however, this upper bound was not constructive. Belovs proved this upper bound by exhibiting a dual adversary solution, which can be turned into a quantum algorithm that relies on controlled calls to a particular unitary. This unitary can be implemented in one query, but actually implementing this algorithm requires giving an efficient circuit for the unitary, which is not possible in general. This is analogous to being given a classical table of values, but no efficient circuit description. While it seems reasonable to guess that the time complexity of k-distinctness should not be significantly higher than the query complexity – what could one possibly do aside from querying and sorting well-chosen sets of inputs? – the problem of finding a matching time upper bound was open for ten years.

In the meantime, lower bounds of $\Omega(n^{\frac{3}{4}-\frac{1}{2k}})$ for $k \geq 3$ [18] and $\Omega(n^{\frac{3}{4}-\frac{1}{4k}})$ for $k \geq 4$ [33] were exhibited. Progress was also made for the $k = 3$ case. Two simultaneous works, [10] and [22] (published together as [12]), gave a $\widetilde{O}(n^{5/7})$ time upper bound for 3-distinctness. Ref. [10] achieved this bound using a generalization of Szegedy's quantum walk framework, called the *electric network framework*. Ref. [22] used the MNRS quantum walk framework [32], and could also be generalised to give a slight improvement on the time upper bound to $\widetilde{O}(n^{1-1/k})$ for any $k > 3$ [26].

In this work, we give an upper bound of $\widetilde{O}(n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^k-1}})$ on the time complexity of k-distinctness, matching the best known query upper bound up to polylogarithmic factors. We do this using ideas from Belovs' query upper bound in a new framework for quantum walk algorithms, the *multidimensional quantum walk framework*, which is an extension of the electric network framework – the most general of the quantum walk frameworks [7]. We give a high-level overview of this extension in Section 1.1.

Quantum walk search frameworks, discussed more in Section 1.1, are important because they allow one to design a quantum algorithm by first designing a classical random walk algorithm of a particular form, which can be compiled into an often faster quantum algorithm. While quantum walk frameworks make it extremely easy to design quantum algorithms, even without an in-depth knowledge of quantum computing, as evidenced by their wide application

across domains, the major drawback is that they can achieve at most a quadratic speedup over the best classical algorithm. This is because a quantum walk search algorithm essentially takes a classical random walk algorithm, and produces a quantum algorithm that is up to quadratically better.

This drawback does not hold for the multidimensional quantum walk framework. We give a quantum algorithm in our framework that solves the *welded trees* problem in $O(n)$ queries and $O(n^2)$ time, which is an exponential speedup over the classical lower bound of $2^{\Omega(n)}$ [21]. While a poly(n) quantum algorithm based on continuous-time quantum walks was already known, this proof-of-concept application shows that our framework is capable of exponential speedups. We emphasise that unlike the quantum walk search frameworks mentioned here that give generic speedups over classical random walk algorithms, continuous-time quantum walks are not easily designed and analysed, and their applications have been limited (with some exceptions based on converting quantum walk search algorithms into continuous-time quantum walks, such as [6]). Our multidimensional quantum walk framework, as a generalization of the electric network framework, is in principle similarly easy to apply, but with the potential for significantly more dramatic speedups.

1.1 Quantum Walks

We give a brief overview of previous work on quantum walk search algorithms, with sufficient detail to understand, at a high level, the improvements we make, before describing these improvements at the end of this section.

The first quantum walk search framework is due to Szegedy [34], and is a generalization of the technique used by Ambainis in his element distinctness algorithm [4]. The framework can be described in analogy to a classical random walk algorithm that first samples an initial vertex according to the stationary distribution π of some random walk (equivalently, reversible Markov process) P, and repeatedly takes a step of the random walk by sampling a neighbour of the current vertex, checking each time if the current vertex belongs to some *marked set* M. Let $HT(P, M)$ be the hitting time, or the expected number of steps needed by a walker starting from π to reach a vertex in M. If S is the cost of sampling from π, U is the cost of sampling a neighbour of any vertex, C is the cost of checking if a vertex is marked, and H is an upper bound on $HT(P, M)$ assuming $M \neq \emptyset$, then this classical algorithm finds a marked vertex with bounded error in complexity:

$$O(\mathsf{S} + H(\mathsf{U} + \mathsf{C})).$$

Szegedy showed that given such a P and M, if S is the cost of coherently[1] sampling from π, i.e. generating $\sum_u \sqrt{\pi(u)}|u\rangle$, and U is the cost of generating, for any u, the superposition over its neighbours $\sum_v \sqrt{P_{u,v}}|v\rangle$, then there is a quantum algorithm that detects if $M \neq \emptyset$ with bounded

1 Technically the classical S and U might be different from the quantum ones, but in practice they are often similar.

error in complexity:

$$O(\mathsf{S} + \sqrt{H}(\mathsf{U} + \mathsf{C})).$$

This result was extended to the case of *finding* a marked vertex, rather than just *detecting* a marked vertex in [5]. This framework and subsequent related frameworks have been widely applied, because this is a very simple way to design a quantum algorithm.

Belovs generalised this framework to the *electric network framework*, by allowing the initial state to be $|\sigma\rangle = \sum_u \sqrt{\sigma(u)}|u\rangle$ for *any* distribution σ, analogous to starting a random walk in some arbitrary initial distribution. Then if S_σ is the cost to generate $|\sigma\rangle$, there is a quantum algorithm that detects a marked vertex with bounded error in complexity:

$$O(\mathsf{S}_\sigma + \sqrt{C}(\mathsf{U} + \mathsf{C})),$$

where C is a quantity that may be the same, or much larger than the hitting time of the classical random walk starting at σ. For example, if $\sigma = \pi$, then $C = H$ as above, but when σ is supported on a single vertex s, and $M = \{t\}$, C is the *commute time* from s to t [19], which is the expected number of steps needed to get from s to t, and then back to s. If the hitting time from s to t is the same as the hitting time from t to s, this is just twice that hitting time. However, in some cases the hitting time from t to s may be significantly larger than the hitting time from s to t.

A second incomparable quantum walk search framework that is similarly easy to apply is the MNRS framework [32]. Loosely speaking, this is the quantum analogue of a classical random walk that does not check if the current vertex is marked at every step, but rather, only after sufficiently many steps have been taken so that the current vertex is independent of the previously checked vertex. Ref. [7] extended the electric network framework to be able to *find* a marked vertex, and also showed that the MNRS framework can be seen as a special case of the resulting framework. Thus, the finding version of the electric network framework captures all quantum walk search frameworks in one unified framework.

We now discuss, at a high level, how a quantum walk search algorithm works – particularly in the electric network framework (but others are similar)[2]. We will suppose for simplicity that σ is supported on a single vertex s, and either $M = \emptyset$ or $M = \{t\}$. Fix a graph G, possibly with weighted edges, such that $s, t \in V(G)$. It is simplest if we imagine that G is bipartite, so let $V(G) = V_{\mathcal{A}} \cup V_{\mathcal{B}}$ be a bipartition, with $s \in V_{\mathcal{A}}$. Let G' be the graph G with a single extra vertex v_0, connected to s, and connected to t if and only if $t \in M$. For $u \in V_{\mathcal{A}}$, define *star states*:

$$|\psi^{G'}_\star(u)\rangle = \sum_{v \in V_{\mathcal{B}} \cup \{v_0\} : \{u,v\} \in E(G')} \sqrt{\mathsf{w}_{u,v}}|u, v\rangle,$$

2 We discuss the classic construction of such algorithms, without modifications that were more recently made in [5] and [7] to not only detect, but find.

where $w_{u,v}$ is the weight of the edge $\{u,v\}$. If we normalise this state, we get $\sum_v \sqrt{P_{u,v}}|u,v\rangle$, where P is the transition matrix of the random walk on G'. For $v \in V_{\mathcal{B}}$, define:

$$|\psi_\star^{G'}(v)\rangle = \sum_{u \in V_{\mathcal{A}} \cup \{v_0\} : \{u,v\} \in E(G')} \sqrt{w_{u,v}}|u,v\rangle.$$

Let

$$\mathcal{A} := \operatorname{span}\{|\psi_\star^{G'}(u)\rangle : u \in V_{\mathcal{A}}\} \text{ and } \mathcal{B} := \operatorname{span}\{|\psi_\star^{G'}(v)\rangle : v \in V_{\mathcal{B}}\}.$$

Then a quantum walk algorithm works by performing phase estimation of the unitary

$$U_{\mathcal{AB}} := (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$$

on initial state $|s, v_0\rangle$ to some sufficiently high precision – this precision determines the complexity of the algorithm. Let us consider why this algorithm can distinguish $M = \emptyset$ from $M = \{t\}$.

First suppose $M = \{t\}$. Assume there is a path from s to t in G (otherwise a random walk from s will never find t), which means there is a cycle in G' containing the edge (v_0, s), obtained by adding (t, v_0) and (v_0, s) to the st-path in G. We can define a cycle state for a cycle $u_1, \ldots, u_d = u_1$ as:

$$\sum_{i=1}^{d-1} \frac{|e_{u_i,u_{i+1}}\rangle}{\sqrt{w_{u_i,u_{i+1}}}} \text{ where } |e_{u,v}\rangle := \begin{cases} |u,v\rangle & \text{if } (u,v) \in V_{\mathcal{A}} \times V_{\mathcal{B}} \text{ or } v = v_0 \\ -|v,u\rangle & \text{if } (u,v) \in V_{\mathcal{B}} \times V_{\mathcal{A}} \text{ or } u = v_0. \end{cases}$$

A cycle state is orthogonal to all star states: if the cycle goes through a vertex u, it is supported on 2 of the edges adjacent to u: one contributing -1 because it goes into u, and the other $+1$ because it comes out of u. Thus, a cycle state is in the $(+1)$-eigenspace of $U_{\mathcal{AB}}$. If there is a cycle that uses the edge (v_0, s), then it has non-zero overlap with the initial state $|s, v_0\rangle$, and so the initial state has non-zero overlap with the $(+1)$-eigenspace of $U_{\mathcal{AB}}$, and so the phase estimation algorithm will have a non-zero probability of outputting a phase estimate of 0. The shorter the cycle (i.e. the shorter the st-path) the greater this overlap is relative to the size of the cycle state. We can make a similar argument if we take not just a single st-path in G, but a superposition of paths called an st-flow. Then the *energy* of this flow (see Definition 2.2) controls the probability of getting Ta phase estimate of 0. The minimum energy of a unit flow from s to t is called the *effective resistance* between s and t, denoted $\mathcal{R}_{s,t}(G)$.

On the other hand, suppose $M = \emptyset$. Then we claim that

$$|s, v_0\rangle = \sum_{u \in V_{\mathcal{A}}} |\psi_\star^{G'}(u)\rangle - \sum_{v \in V_{\mathcal{B}}} |\psi_\star^{G'}(v)\rangle \in \mathcal{A} + \mathcal{B} = (\mathcal{A}^\perp \cap \mathcal{B}^\perp)^\perp.$$

Since $|s, v_0\rangle$ also only overlaps with the star state of $s \in V_{\mathcal{A}}$, it is orthogonal to $\mathcal{B}$, and Thus, to $\mathcal{A} \cap \mathcal{B}$. Combined, this means that our initial state has no overlap with the $(+1)$-eigenspace of $U_{\mathcal{AB}}$, which is exactly $(\mathcal{A} \cap \mathcal{B}) \oplus (\mathcal{A}^\perp \cap \mathcal{B}^\perp)$, so if we could do phase estimation with infinite

precision, the probability we would measure a phase estimate of 0 would be 0. Our precision is not infinite, but using a linear algebraic tool called the *effective spectral gap lemma*, we can show that precision proportional to

$$\left\| \sum_{u \in V_{\mathcal{A}}} |\psi_{\star}^{G'}(u)\rangle \right\|^2 = \sum_{e \in G'} \mathsf{w}_e =: \mathcal{W}(G)$$

is sufficient.

Combining these two analyses for the $M = \{t\}$ and $M = \emptyset$ case yield (in a non-obvious way) that approximately $\sqrt{\mathcal{R}\mathcal{W}}$ steps of the quantum walk is sufficient, if $\mathcal{R}$ is an upper bound on $\mathcal{R}_{s,t}(G)$ whenever $M = \{t\}$, and $\mathcal{W}$ is an upper bound on $\mathcal{W}(G)$ whenever $M = \emptyset$. A nice way to interpret this is that the quantity $\mathcal{R}_{s,t}(G)\mathcal{W}(G)$ is equal to the *commute time* from s to t – the expected number of steps a random walker starting from s needs to reach t, and then return to s. For a discussion of how to interpret this quantity in the case of more general σ and M, see [7].

The Multidimensional Quantum Walk Framework: We extend this algorithm in two ways:

Edge Composition To implement the unitary $U_{\mathcal{A}\mathcal{B}}$, we perform a mapping that acts, for any $u \in V_{\mathcal{A}}$, as $|u, 0\rangle \mapsto |\psi_{\star}^{G'}(u)\rangle$ (up to normalization), and a similar mapping for $v \in V_{\mathcal{B}}$. Loosely speaking, what this usually means is that we have a labelling of the edges coming out of u, and some way of computing (u, v) from (u, i), where v is the i-th neighbour of u. If this computation costs $\mathsf{T}_{u,i}$ steps, then it takes $O(\max_{u,i} \mathsf{T}_{u,i})$ steps to implement $U_{\mathcal{A}\mathcal{B}}$. However, in case this cost varies significantly over different u, i, we can do much better. We show how we can obtain a unitary with polylogarithmic cost, and essentially consider, in the analysis of the resulting algorithm, a quantum walk on a modified graph in which an edge $\{u, v\}$, where v is the i-th neighbour of u, is replaced by a path of length $\mathsf{T}_{u,i}$. A similar thing was already known for *learning graphs*, when a transition could be implemented with $\mathsf{T}_{u,i}$ *queries* [11]. This is an extremely useful, if not particularly surprising, feature of the framework, which we use in our application to k-distinctness.

Alternative Neighbourhoods The more interesting way we augment the electric network framework is to allow the use of *alternative neighbourhoods*. In order to generate the star state of a vertex u, which is a superposition of the edges coming out of u, one must, in some sense, know the neighbours of u, as well as their relative weights. In certain settings, the algorithm will know that the star state for u is one of a small set of easily preparable states $\Psi_{\star}(u) = \{|\psi_{\star}^{1}(u)\rangle, |\psi_{\star}^{2}(u)\rangle, \dots\}$, but computing precisely which one of these is the correct state would be computationally expensive. In that case, we include all of $\Psi_{\star}(u)$ when constructing the spaces $\mathcal{A}$ and $\mathcal{B}$. In the case when $M = \emptyset$, the analysis is the same – by increasing $\mathcal{A} + \mathcal{B}$, we have only made the analysis easier. However, in the case $M \neq \emptyset$, the analysis has become more constrained. For the analysis of this case, we used a circulation,

because it is orthogonal to all star states. However, now there are some extra states in $\mathcal{A}+\mathcal{B}$, and we need to take extra care to find a circulation that is also orthogonal to these.

The alternative neighbourhoods technique is best understood through examples, of which we shortly describe two. We first remark on the unifying idea from which both these techniques follow.

If we let $\{|\psi_\star(u)\rangle\}_{u\in V}$ be *any* set of states, we can make a graph G on V by letting u and v be adjacent if and only if $\langle\psi_\star(u)|\psi_\star(v)\rangle \neq 0$. Then, if this graph is bipartite, and we can reflect around the span of each state individually, we can reflect around $\operatorname{span}\{|\psi_\star(u)\rangle : u \in V\}$. Quantum walk search algorithms can be seen as a special case of this, where we additionally exploit the structure of the graph to analyse the complexity of this procedure. One way of viewing alternative neighbourhoods is that we extend this reasoning to the case where we have *spaces* $\{\operatorname{span}\{\Psi_\star(u)\}\}_{u\in V}$, each of which we can efficiently reflect around, and G is now a bipartite graph encoding the overlap of the *spaces*, hence the qualifier *multidimensional*.

Edge composition also exploits this picture. We can define a sequence of subspaces $\{\Psi_t^{u,v}\}_{t=1}^{\mathsf{T}_{u,i}}$ that only overlap for adjacent t, and such that the subroutine computing $|v,j\rangle$ from $|u,i\rangle$ can be seen as moving through these spaces. Now the overlap graph of all these spaces will look like G, except with each edge (u,v) replaced by a path of length $\mathsf{T}_{u,i}$. See Figure 5 and Figure 7 for examples of such overlap graphs.

Before moving on to our examples, we comment that unlike the finding version of the electric network framework [7], our extension does not allow one to find a marked vertex, but only to detect if there is one or not. We leave extending our framework to finding as future work.

1.2 Welded Trees

We motivate the alternative neighbourhoods modification by an application to the welded trees problem [21]. In the welded trees problem, the input is an oracle O_G for a graph G with $s,t \in V(G) \subset \{0,1\}^{2n}$. Each of s and t is the root of a full binary tree with 2^n leaves, and we connect these leaves with a pair of random matchings. This results in a graph in which all vertices except s and t have degree 3, and s and t each have degree 2. Given a string $u \in \{0,1\}^{2n}$, the oracle O_G returns $\perp$ if $u \notin V(G)$, which is true for all but at most a 2^{-n+2} fraction of strings, and otherwise it returns a list of the 2 or 3 neighbours of u. We assume $s = 0^{2n}$, so we can use s as our starting point, and the goal is to find t, which we can recognise since it is the only other vertex with only 2 neighbours. The classical query complexity of this problem is $2^{\Omega(n)}$ [21]. Intuitively that is because this problem is set up so that a classical algorithm has no option but to do a random walk, starting from s, until it hits t. However, this takes $2^{\Omega(n)}$ steps, because wherever a walker is in the graph, the probability of moving towards the centre, where the leaves of the two trees are connected, is twice the probability of moving away from the centre,

towards s or t. So a walker quickly moves from s to the centre, but then it takes exponential time to escape to t.

While we know there is a quantum algorithm that solves this problem in $\mathrm{poly}(n)$ queries[3] to O_G [21], if we try to reproduce this result in the electric network framework, we will get an exponential-time algorithm, essentially because the total weight of the graph is exponential.

Suppose we could add weights to the edges of G, so that at any vertex u, the probability of moving towards the centre or away from the centre were the same: that is, if w is the weight on the edge from u to its *parent*, then the other two edges should have weight w/2. This would already be very helpful for a classical random walk, however, a bit of thought shows that this is not possible to implement. By querying u, we learn the labels of its three neighbours, v_1, v_2, v_3, which are random $2n$-bit strings, but we get no indication which is the parent. However, we know that the correct star state in the weighted graph that we would like to be able to walk on is proportional to one of the following:

$$|u, v_1\rangle + \frac{1}{2}|u, v_2\rangle + \frac{1}{2}|u, v_3\rangle, \quad |u, v_2\rangle + \frac{1}{2}|u, v_1\rangle + \frac{1}{2}|u, v_3\rangle, \quad \text{or} \quad |u, v_3\rangle + \frac{1}{2}|u, v_1\rangle + \frac{1}{2}|u, v_2\rangle.$$

Thus, we add all three states (up to some minor modifications) to $\Psi_\star(u)$, which yields an algorithm that can learn any bit of information about t in $O(n)$ queries. By composing this with the Bernstein-Vazirani algorithm we can find t. For details, see Section 4.

We emphasise that our application to the welded trees problem does not use the edge composition technique. It would be trivial to embed any known exponential speedup in our framework by simply embedding the exponentially faster quantum algorithm in one of the edges of the graph, but we are able to solve the welded trees problem using only the alternative neighbourhoods idea.

1.3 3-Distinctness

We describe an attempt at a quantum walk algorithm for 3-distinctness, how it fails, and how the Multidimensional Quantum Walk Framework comes to the rescue. While our result for $k = 3$ is not new, our generalization to $k > 3$ is, and the case of $k = 3$ is already sufficient to illustrate our techniques. Formally, the problem of 3-distinctness is: given a string $x \in [q]^n$, output a 1 if and only if there exist distinct $a_1, a_2, a_3 \in [n]$ such that $x_{a_1} = x_{a_2} = x_{a_3}$. We make the standard simplifying assumptions (without loss of generality) that if such a 3-collision exists, it is unique, and moreover, there is an equipartition $[n] = A_1 \cup A_2 \cup A_3$ such that $a_1 \in A_1$, $a_2 \in A_2$ and $a_3 \in A_3$.

3 The best previous query complexity was $O(n^{1.5})$ [8], although it is likely that continuous time quantum walks could also be used to solve this problem in $O(n)$ queries.

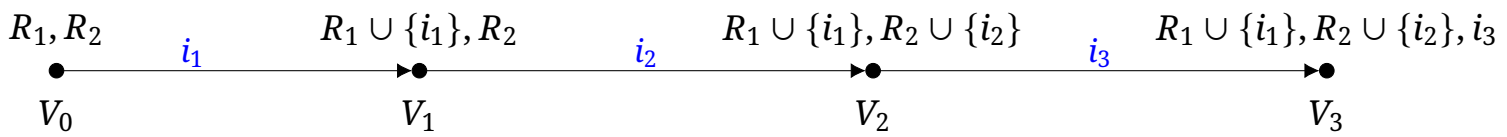


Figure 1. A sample path from V_0 to V_3 in our first attempt at a quantum walk for 3-distinctness. The indices shown in blue can be seen to label the edges.

We now describe a graph that will be the basis for a quantum walk attempt. A vertex v_{R_1,R_2} is described by a pair of sets $R_1 \subset A_1$ and $R_2 \subset A_2$. v_{R_1,R_2} stores these sets, as well as input-dependent *data* consisting of the following:

— Queried values for all of R_1: $D_1(R) := \{(i, x_i) : i \in R_1\}$.

— Queried values for those elements of R_2 that have a match in R_1:

$$D_2(R) := \{(i_1, i_2, x_{i_1}) : i_1 \in R_1, i_2 \in R_2, x_{i_1} = x_{i_2}\}.$$

By only keeping track of the values in R_2 that have a match in R_1, we save the cost of initially querying the full set R_2. The vertices will be in 4 different classes, for some parameters r_1 and r_2 with $r_1 \ll r_2$:

$$\begin{aligned}
V_0 &= \{v_{R_1,R_2} : |R_1| = r_1, |R_2| = r_2\} \\
V_1 &= \{v_{R_1,R_2} : |R_1| = r_1 + 1, |R_2| = r_2\} \\
V_2 &= \{v_{R_1,R_2} : |R_1| = r_1 + 1, |R_2| = r_2 + 1\} \\
V_3 &= \{v_{R_1,R_2,i_3} : |R_1| = r_1 + 1, |R_2| = r_2 + 1, i_3 \in A_3\}.
\end{aligned}$$

The vertices $v_{R_1,R_2,i_3} \in V_3$ are just like the vertices in V_2, except there is an additional index $i_3 \in A_3$ stored. We connect vertices in V_ℓ and $V_{\ell+1}$ in the obvious way: $v_{R_1,R_2} \in V_\ell$ is adjacent to $v_{R_1',R_2'} \in V_{\ell+1}$ if and only if $R_1 \subseteq R_1'$ and $R_2 \subseteq R_2'$ (exactly one of these inclusions is proper); and $v_{R_1,R_2} \in V_2$ is adjacent to $v_{R_1,R_2,i_3} \in V_3$ for any $i_3 \in A_3$ (see Figure 1).

We say a vertex $v_{R_1,R_2,i_3} \in V_3$ is marked if $a_1 \in R_1$, $a_2 \in R_2$, and $a_3 = i_3$, where (a_1, a_2, a_3) is the unique 3-collision. Thus, a quantum walk that decides if there is a marked vertex or not decides 3-distinctness.

We imagine a quantum walk that starts in a uniform superposition over V_0. To construct this initial state, we first take a uniform superposition over all sets R_1 of r_1 indices, and query them. Next we take a uniform superposition over all sets R_2 of size r_2, but rather than query everything in R_2, we search for all indices in R_2 that have a match in R_1. This saves us the cost of querying all r_2 elements of R_2, which is important because we will set r_2 to be larger than the total complexity we aim for (in this case, $r_2 \gg n^{5/7}$), so we could not afford to spend so much time. However, we do not only care about query complexity, but also the total time spent on

non-query operations, so we also do not want to spend time writing down the set R_2, even if we do not query it, which is the first problem with this approach:

Problem 1: Writing down R_2 would take too long.

The fix for Problem 1 is rather simple: we will not let R_2 be a uniform random set of size r_2. Instead, we will assume that A_2 is partitioned into m_2 blocks, each of size $n/(3m_2)$, and R_2 will be made up of $t_2 := 3m_2r_2/n$ of these blocks. This also means that when we move from V_1 to V_2, we will add an entire block, rather than just a single index. The main implication of this is that when we move from V_1 to V_2, we will have to search the new block of indices that we are adding to R_2 for any index that collides with R_1. This means that transitions from V_1 to V_2 have a non-trivial cost, n^ε for some small constant ε, unlike all other transitions, which have polylogarithmic cost. Naively we would incur a multiplicative factor of n^ε on the whole algorithm, but we avoid this because the edge composition technique essentially allows us to only incur the cost n^ε on the edges that actually incur this cost, and not on every edge in the graph. Otherwise, our solution to Problem 1 is technical, but not deep, and so we gloss over Problem 1 and its solution for the remainder of this high-level synopsis. This is the only place we use the edge composition part of the framework in our applications, but we suspect it can be used in much more interesting ways.

Moving on, in order to take a step from a vertex $v_{R_1,R_2} \in V_0$ to a vertex $v_{R_1\cup\{i_1\},R_2} \in V_1$, we need to select a uniform new index i_1 to add to R_1, and then also update the data we store with each vertex. That means we have to query i_1 and add (i_1, x_{i_1}) to $D_1(R)$, which is simple, and can be done in $O(\log n)$ basic operations as long as we use a reasonable data structure to store $D_1(R)$; and we also have to update $D_2(R)$ by finding anything in R_2 that collides with i_1. Since R_2 has not been queried, this latter update would require an expensive search, which we do not have time for, so we want to avoid this. However, if we do not search R_2 for any i_2 such that $x_{i_2} = x_{i_1}$, then whenever we add some i_1 that has a match in R_2, the data becomes incorrect, and we have introduced what is referred to in [9] as a *fault*. This is a serious issue, because if i_1 is the unique index in R_1 such that there exists $i_2 \in R_2$ with $x_{i_1} = x_{i_2}$, but this is not recorded in $D_2(R)$, then i_1 is "remembered" as having been added after i_2. That is, the resulting vertex does not only depend on $R_1 \cup \{i_1\}, R_2$, but on i_1 as well. For quantum interference to happen, it is crucial that when we are at a vertex v, the state does not remember anything about how we got there.

Problem 2: When we add i_1 to R_1 without searching for a match in R_2, we may introduce a *fault*.

Our handling of this is inspired by the solution to an analogous problem in the query upper bound of [9]. We partition R_1 into three sets: $R_1(\{1\})$, $R_1(\{2\})$, and $R_1(\{1,2\})$; and R_2 into two sets $R_1(1)$ and $R_1(2)$. Then $D_2(R)$ will only store collisions (i_1, i_2, x_{i_1}) such that $x_{i_1} = x_{i_2}$ if $i_1 \in R_1(S)$ and $i_2 \in R_2(s)$ for some $s \in S$. This is shown in Figure 2.

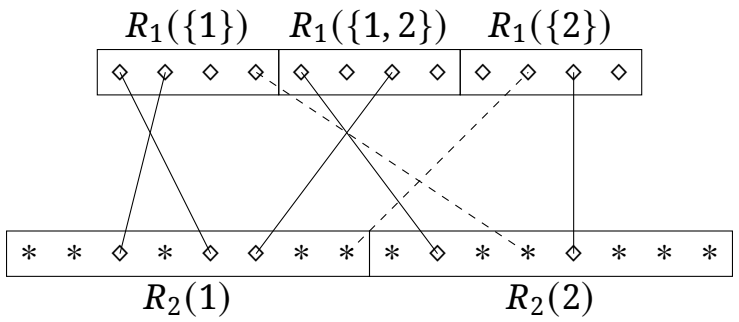


Figure 2. The data we keep track of for a vertex v_{R_1,R_2}. $\diamond$ represents a queried index. $*$ represents an index whose query value is not stored. We only store the query value of an index in $R_2(s)$ if it collides with something in $R_1(\{s\}) \cup R_1(\{1,2\})$, shown here by a solid line. If $i_2 \in R_2(1)$ collides with some value in $R_1(\{2\})$, shown here by a dashed line, we do not record that, and do not store x_{i_2}.

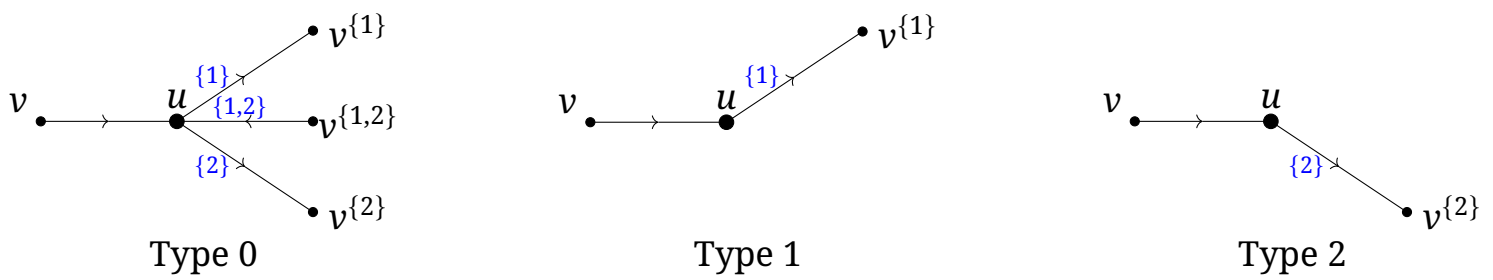


Figure 3. The possible neighbourhoods of $u = v_{R_1,R_2,i_1} \in V_0^+$, depending on the type of vertex. $v^S \in V_1$ is obtained from v by adding i_1 to $R_1(S)$. The backwards neighbour $v = v_{R_1,R_2} \in V_0$ is always the same.

Now when we add i_1 to R_1, we have three choices: we can add it to $R_1(\{1\})$, $R_1(\{2\})$, or $R_1(\{1,2\})$. Importantly, at least one of these choices does not introduce a fault. To see this, suppose there is some $i_2 \in R_2$ such that $x_{i_1} = x_{i_2}$. We claim there can be at most one such index, because otherwise there would be a 3-collision in $A_1 \cup A_2$, and we are assuming the unique 3-collision has one part in A_3. This leads to three possibilities:

Type 1: $i_2 \in R_2(2)$, in which case, adding i_1 to $R_1(\{1\})$ does not introduce a fault.
Type 2: $i_2 \in R_2(1)$, in which case, adding i_1 to $R_1(\{2\})$ does not introduce a fault.
Type 0: There is no such i_2, in which case, adding i_1 to $R_1(\{1\})$ or $R_1(\{2\})$ or $R_1(\{1,2\})$ does not introduce a fault.

We modify the graph so that we first move from $v_{R_1,R_2} \in V_0$ to $v_{R_1,R_2,i_1} \in V_0^+$ by selecting a new $i_1 \in A_1 \setminus R_1$, and then move from v_{R_1,R_2,i_1} to $v_{R_1 \cup \{i_1\},R_2} \in V_1$ – here there are three possibilities for $R_1 \cup \{i_1\}$, depending on to which of the three parts of R_1 we add i_1. However, we will only add i_1 to a part of R_1 that does not introduce a fault. Thus, a vertex v_{R_1,R_2,i_1} in V_0^+ has one edge leading back to V_0, and either one or three edges leading forward to V_1, as shown in Figure 3.

On its own, this is not a solution, because for a given v_{R_1,R_2,i_1}, in order to determine its type, we would have to search for an $i_2 \in R_2$ such that $x_{i_1} = x_{i_2}$, which is precisely what we want to

avoid. However, this is exactly the situation where the alternative neighbourhood technique is useful. For all $u \in V_0^+$, we will let $\Psi_\star(u)$ contain all three possibilities shown in Figure 3, of which exactly one is the correct state. We are then able to carefully construct a flow that is orthogonal to all three states, in our analysis. The idea is that all incoming flow from v must leave along the edge $(u, v^{\{1\}})$ so that the result is a valid flow in case of Type 1. However, in order to be a valid flow in case of Type 2, all incoming flow from v must leave along the edge $(u, v^{\{2\}})$. But now to ensure that we also have a valid flow in case of Type 0, we must have negative flow on the edge $(u, v^{\{1,2\}})$, or equivalently, flow from $v^{\{1,2\}}$ to u. This is indicated by the arrows on the edges in Figure 3. For details, see Section 5.2.

Model of Computation: Our k-distinctness algorithm works in the same model as previous k-distinctness algorithms, which we try to make more explicit than has been done in previous work. In addition to arbitrary 1- and 2-qubit gates, we assume *quantum random access* to a large *quantum* memory (QRAM). This version of QRAM is fully quantum, whereas some previous works have used "QRAM" to refer to classical memory that can be read in superposition by a quantum machine. We describe precisely what we mean by QRAM in Section 2.2.

1.4 Organization

The remainder of this article is organised as follows. In Section 2, we give preliminaries on graph theory, quantum subroutines, quantum data structures, and probability theory, including several non-standard definitions, which we encourage the experienced reader not to skip. In Section 3, we present the Multidimensional Quantum Walk Framework, which is stated as Theorem 3.10. In Section 4, we present our first application to the welded trees problem. This section is mostly self-contained, explicitly constructing and analysing an algorithm rather than referring to our new framework, which we feel gives an intuitive demonstration of the framework. In Section 5, we present our new application to k-distinctness.

2. Preliminaries

2.1 Graph Theory

In this section, we define graph theoretic concepts and notation.

DEFINITION 2.1 (Network). A network is a weighted graph G with an (undirected) edge set $E(G)$, vertex set $V(G)$, and some weight function $\mathsf{w} : E(G) \to \mathbb{R}_{>0}$. Since edges are undirected, we can equivalently describe the edges by some set $\overrightarrow{E}(G)$ such that for all $\{u, v\} \in E(G)$, exactly one of (u, v) or (v, u) is in $\overrightarrow{E}(G)$. The choice of edge directions is arbitrary. Then we can view the weights as a function $\mathsf{w} : \overrightarrow{E}(G) \to \mathbb{R}_{>0}$, and for all $(u, v) \in \overrightarrow{E}$, define $\mathsf{w}_{v,u} = \mathsf{w}_{u,v}$. For convenience, we will define $\mathsf{w}_{u,v} = 0$ for every pair of vertices such that $\{u, v\} \notin E(G)$. The *total*

weight of G is

$$\mathcal{W}(G) := \sum_{e \in \overrightarrow{E}(G)} \mathsf{w}_e.$$

For an implicit network G, and $u \in V(G)$, we will let $\Gamma(u)$ denote the *neighbourhood* of u:

$$\Gamma(u) := \{v \in V(G) : \{u, v\} \in E(G)\}.$$

We use the following notation for *the out- and in-neighbourhoods* of $u \in V(G)$:

$$\begin{aligned} \Gamma^+(u) &:= \{v \in \Gamma(u) : (u, v) \in \overrightarrow{E}(G)\} \\ \Gamma^-(u) &:= \{v \in \Gamma(u) : (v, u) \in \overrightarrow{E}(G)\}, \end{aligned} \tag{1}$$

DEFINITION 2.2 (Flow, Circulation). A *flow* on a network G is a real-valued function $\theta : \overrightarrow{E}(G) \to \mathbb{R}$, extended to edges in both directions by $\theta(u, v) = -\theta(v, u)$ for all $(u, v) \in \overrightarrow{E}(G)$. For any flow θ on G, and vertex $u \in V(G)$, we define $\theta(u) = \sum_{v \in \Gamma(u)} \theta(u, v)$ as the flow coming out of u. If $\theta(u) = 0$, we say flow is conserved at u. If flow is conserved at every vertex, we call θ a *circulation*. If $\theta(u) > 0$, we call u a *source*, and if $\theta(u) < 0$ we call u a *sink*. A flow with unique source s and unique sink t is called an *st-flow*. The *energy* of θ is

$$\mathcal{E}(\theta) := \sum_{(u,v) \in \overrightarrow{E}(G)} \frac{\theta(u, v)^2}{\mathsf{w}_{u,v}}.$$

Accessing G: In computations involving a (classical) random walk on a graph G, it is usually assumed that for any $u \in V(G)$, it is possible to sample a neighbour $v \in \Gamma(u)$ according to the distribution

$$\Pr[v] = \frac{\mathsf{w}_{u,v}}{\mathsf{w}_u} \text{ where } \mathsf{w}_u := \sum_{v' \in \Gamma(u)} \mathsf{w}_{u,v'}.$$

It is standard to assume this is broken into two steps: (1) sampling some $i \in [d_u]$, where $d_u := |\Gamma(u)|$ is the degree of u, and (2) computing the i-th neighbour of u. That is, we assume that for each $u \in V(G)$, there is an efficiently computable function $f_u : [d_u] \to V(G)$ such that $\mathrm{im}(f_u) = \Gamma(u)$, and we call $f_u(i)$ the *i-th neighbour of u*. In the quantum case (see Definition 2.3 below), we assume that the sample (1) can be done coherently, and we use a reversible version of the map $(u, i) \mapsto f_u(i)$. We will also find it convenient to suppose the indices i of the neighbours of u come from some more general set $L(u)$, which may equal $[d_u]$, or some other convenient set, which we call the *edge labels of u*. It is possible to have $|L(u)| > |\Gamma(u)| = d_u$, meaning that some elements of $L(u)$ do not label an edge adjacent to u (these labels should be sampled with probability 0). We assume we have a partition of $L(u)$ into disjoint $L^+(u)$ and $L^-(u)$ such that:

$$\begin{aligned} L^+(u) &\supseteq \{i \in L(u) : (u, f_u(i)) \in \overrightarrow{E}(G)\} = \{i \in L(u) : f_u(i) \in \Gamma^+(u)\} \\ L^-(u) &\supseteq \{i \in L(u) : (f_u(i), u) \in \overrightarrow{E}(G)\} = \{i \in L(u) : f_u(i) \in \Gamma^-(u)\}. \end{aligned}$$

Note that for any $(u, v) \in \overrightarrow{E}(G)$, with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, any of (u, v), (v, u), (u, i), or (v, j) fully specify the edge. Thus, it will be convenient to denote the weight of the edge using

any of the alternatives:

$$\mathsf{w}_{u,v} = \mathsf{w}_{v,u} = \mathsf{w}_{u,i} = \mathsf{w}_{v,j}.$$

For any $i \in L(u)$, we set $\mathsf{w}_{u,i} = 0$ if and only if $\{u, f_u(i)\} \notin E(G)$.

DEFINITION 2.3 (Quantum Walk access to G). For each $u \in V(G)$, let $L(u) = L^+(u) \cup L^-(u)$ be some finite set of *edge labels*, and $f_u : L(u) \to V(G)$ a function such that $\Gamma(u) \subseteq \mathrm{im}(f_u)$. A quantum algorithm has *quantum walk access* to G if it has access to the following subroutines:

- A subroutine that "samples" from $L(u)$ by implementing a map $U_\star$ in cost $\mathsf{A}_\star$ that acts as:

$$U_\star|u,0\rangle \propto \sum_{i\in L^+(u)} \sqrt{\mathsf{w}_{u,i}}|u,i\rangle - \sum_{i\in L^-(u)} \sqrt{\mathsf{w}_{u,i}}|u,i\rangle =: |\psi^G_\star(u)\rangle.$$

- A subroutine that implements the *transition map* $|u,i\rangle \mapsto |v,j\rangle$ (possibly with some error) where $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, with costs $\{\mathsf{T}_{u,i} = \mathsf{T}_{u,v}\}_{(u,v)\in\overrightarrow{E}(G)}$.
- Query access to the total vertex weights $\mathsf{w}_u = \sum_{v\in\Gamma(u)} \mathsf{w}_{u,v}$.

We call $\{\mathsf{T}_e\}_{e\in\overrightarrow{E}(G)}$ the set of *transition costs* and $\mathsf{A}_\star$ the *cost of generating the star states*.

DEFINITION 2.4 (Networks with lengths). If G is a network, and $\ell : \overrightarrow{E}(G) \to \mathbb{Z}_{\geq 1}$ a positive-integer-valued function on the edges of G, we define G^ℓ to be the graph obtained from replacing each edge $(u,v) \in \overrightarrow{E}(G)$ of G with a path from u to v of length $\ell_{u,v}$, and giving each edge in the path the weight $\mathsf{w}_{u,v}$. We define:

$$\mathcal{W}^\ell(G) := \mathcal{W}(G^\ell) = \sum\nolimits_{e\in\overrightarrow{E}(G)} \mathsf{w}_e \ell_e,$$

and for any flow θ on G, we let θ^ℓ be the flow on G^ℓ obtained by assigning flow $\theta(u,v)$ to any edge in the path from u to v, and define:

$$\mathcal{E}^\ell(\theta) := \mathcal{E}(\theta^\ell) = \sum\nolimits_{e\in\overrightarrow{E}(G)} \frac{\theta(e)^2}{\mathsf{w}_e}\ell_e.$$

2.2 Model of Computation and Quantum Subroutines

We will work in the (fully quantum) QRAM model, which we now describe. By QRAM, we mean *quantum* memory, storing an arbitrary quantum state, to which we can apply random access gates. By this, we mean we can implement, for $i \in [n]$, $b \in \{0,1\}$, $x \in \{0,1\}^n$, a random access read:

$$\text{READ} : |i\rangle|b\rangle|x\rangle \mapsto |i\rangle|b \oplus x_i\rangle|x\rangle,$$

or a random access write:

$$\text{WRITE} : |i\rangle|b\rangle|x\rangle \mapsto |i\rangle|b\rangle|x_1,\ldots,x_{i-1},x_i \oplus b, x_{i+1},\ldots,x_n\rangle,$$

on any superposition. By applying READ · WRITE · READ, we can implement a controlled swap:

$$\sum_{i\in[n]} |i\rangle\langle i| \otimes \mathrm{SWAP}_{0,i}(|i\rangle|b\rangle|x\rangle) = |i\rangle|x_i\rangle|x_1,\dots,x_{i-1},b,x_{i+1},\dots,x_n\rangle.$$

Aside from these operations, we count the number of elementary gates, by which we mean arbitrary unitaries that act on $O(1)$ qubits.

We will be interested in running different iterations of a subroutine on different branches of a superposition, for which we use the concept of a quantum subroutine. We note that Definition 2.5 is *not* the most general definition, but it is sufficient for our purposes.

DEFINITION 2.5 (Quantum Subroutine). A *quantum subroutine* is a sequence of unitaries $U_0,\dots,U_{\mathsf{T}_{\max}-1}$ on $H_{\mathcal{Z}} = \mathrm{span}\{|z\rangle : z \in \mathcal{Z}\}$ for some finite set $\mathcal{Z}$. For $X, Y \subseteq \mathcal{Z}$, we say the subroutine computes an injective function $f : X \to Y$ in times $\{\mathsf{T}_x \le \mathsf{T}_{\max}\}_{x\in X}$ with errors $\{\epsilon_x\}_{x\in X}$ if:

1. The map $\sum_{t=0}^{\mathsf{T}_{\max}-1} |t\rangle\langle t| \otimes U_t$ can be implemented in $\mathrm{polylog}(\mathsf{T}_{\max})$ complexity.
2. For all $x \in X$, $\left\||f(x)\rangle - U_{\mathsf{T}_x-1}\dots U_0(|x\rangle)\right\|^2 \le \epsilon_x$.
3. The maps $x \mapsto \mathsf{T}_x$ and $y \mapsto \mathsf{T}_{f^{-1}(y)}$ can both be implemented in $\mathrm{polylog}(\mathsf{T}_{\max})$ complexity.
4. There exists a decomposition $\mathcal{Z} = \bigcup_{x\in X} \mathcal{Z}_x$ such that $x, f(x) \in \mathcal{Z}_x$, and for all $t \in \{0,\dots,\mathsf{T}_{\max}-1\}, U_t\dots U_0|x\rangle \in \mathrm{span}\{|z\rangle : z \in \mathcal{Z}_x\}$.

While not all of our assumptions are general, they are reasonable in our setting. Item 1 is standard in subroutines that will be run in superposition (see e.g. [3]), and is reasonable, for example, in settings where the algorithm is sufficiently structured to compute U_t from a standard gate set on the fly, which we formalise in Lemma 2.6 below (see also the discussion in [23, Section 2.2]).

Item 3 is not always necessary, but it is often true, and simplifies things considerably. It means, in particular, that one can decide, based on the input, how many steps of the algorithm should be applied, and then, based on the output, uncompute this information.

Item 4 is not a standard assumption, but it is also not unreasonable. For example, if $X = X' \times \{0\}$ and $f(x,0) = (x, g(x))$ for some function g, the algorithm may simply use x as a control, and so its state always encodes x, and therefore remains orthogonal for different x.

LEMMA 2.6. *Call unitaries $U_0,\dots,U_{\mathsf{T}_{\max}-1}$ on H a* uniform quantum algorithm *if there exists $\ell = \mathrm{polylog}(\mathsf{T}_{\max})$, unitaries $W_1,\dots,W_\ell$, and maps $g : \{0,\dots,\mathsf{T}_{\max}-1\} \to [\ell]$ and $g' : \{0,\dots,\mathsf{T}_{\max}-1\} \to 2^{[\log\dim H]}$ such that:*

1. *For each $j \in [\ell]$, W_j can be implemented by $\mathrm{polylog}(\mathsf{T}_{\max})$ gates from some implicit gate set (and therefore acts on $m = \mathrm{polylog}(\mathsf{T}_{\max})$ qubits).*
2. *g and g' can be computed in $\mathrm{polylog}(\mathsf{T}_{\max})$ complexity.*
3. *For all $t \in \{0,\dots,\mathsf{T}_{\max}-1\}$, $U_t = W_{g(t)}(g'(t))$, where $W_\ell(S)$ denotes W_ℓ applied to the qubits specified by S.*

Then $\sum_{t=0}^{\mathsf{T}_{\max}-1} |t\rangle\langle t| \otimes U_t$ *can be implemented in* $\mathrm{polylog}(\mathsf{T}_{\max})$ *gates.*

PROOF. We describe how to implement $\sum_{t=0}^{\mathsf{T}_{\max}-1} |t\rangle\langle t| \otimes U_t$ on $|t\rangle|z\rangle$ for $|z\rangle \in H$. Append registers $|0\rangle_A|0\rangle_{A'} \in \mathrm{span}\{|j,S\rangle : j \in \{0,\dots,\ell\}, S \in \mathcal{S}\}$, where $\mathcal{S}$ is the set of subsets of $[\log \dim H]$ of size at most m. Compute $g(t)$ and $g'(t)$ to get: $|t\rangle|z\rangle|0\rangle_A|0\rangle_{A'} \mapsto |t\rangle|z\rangle|g(t)\rangle_A|g'(t)\rangle_{A'}$. Controlled on $g'(t)$, we can swap the qubits acted on by U_t into the first m positions. Then we can implement $\sum_{j=1}^{\ell} |j\rangle\langle j| \otimes W_j + |0\rangle\langle 0| \otimes I$ by decomposing it into a sequence of ℓ controlled operations:

$$\prod_{j=1}^{\ell} \left(|j\rangle\langle j| \otimes W_j + (I - |j\rangle\langle j|) \otimes I\right).$$

The result follows from noticing that each of these $\ell = \mathrm{polylog}(\mathsf{T}_{\max})$ operations can be implemented with $\mathrm{polylog}(\mathsf{T}_{\max})$ controlled gates. ■

LEMMA 2.7. *Fix a constant integer* c*, and for* $j \in [c]$*, let* $\mathcal{S}_j$ *be a quantum subroutine on* $H_j = \mathrm{span}\{|j\rangle\} \otimes H$ *for some space* H *that takes time* $\{\mathsf{T}_x = \mathsf{T}_j\}_{x \in X_j}$ *with errors* $\{\epsilon_x = \epsilon_j\}_{x \in X_j}$*. Then there is a quantum algorithm that implements* $\sum_{j=1}^{c} |j\rangle\langle j| \otimes \mathcal{S}_j$ *in variable times* $\mathsf{T}_{j,x} = O(\mathsf{T}_j)$ *and errors* $\epsilon_{j,x} = \epsilon_j$ *for all* $x \in X_j$*.*

PROOF. Pad each algorithm with identities so that they all have the same number, $\mathsf{T}_{\max} = \max_{j \in [c]} \mathsf{T}_{\max}^{(c)}$ of unitaries. Then for $t = \{0,\dots,c\mathsf{T}_{\max} - 1\}$, with $t = qc + r$ for $r \in \{0,\dots,c\}$, let $U_t = |r\rangle\langle r| \otimes U_q^{(r)} + (I - |r\rangle\langle r|) \otimes I$. ■

2.3 Quantum Data Structures

We will assume we have access to a data structure that can store a set of keyed items, $S \subset \mathcal{I} \times \mathcal{K}$, for finite sets $\mathcal{K}$ and $\mathcal{I}$. For such a stored set S, we assume the following can be implemented in $\mathrm{polylog}(|\mathcal{I} \times \mathcal{K}|)$ complexity:

1. For $(i,k) \in \mathcal{I} \times \mathcal{K}$, insert (i,k) into S.
2. For $(i,k) \in S$, remove (i,k) from S.
3. For $k \in \mathcal{K}$, query the number of $i \in \mathcal{I}$ such that $(i,k) \in S$.
4. For $k \in \mathcal{K}$, return the smallest i such that $(i,k) \in S$.
5. Generate a uniform superposition over all $(i,k) \in S$.

In addition, for quantum interference to take place, we assume the data structure is coherent, meaning it depends only on S, and not on, for example, the order in which elements were added. See [17, Section 3.1] for an example of such a data structure.

2.4 Probability Theory

Hypergeometric Distribution: In the hypergeometric distribution with parameters (N,K,d), we draw d objects uniformly without replacement from a set of N objects, K of which are marked, and consider the number of marked objects that are drawn. We will use the following:

LEMMA 2.8 (Hypergeometric Tail Bounds [25]). *Let Z be a hypergeometric random variable with parameters (N, K, d), and $\mu = \frac{Kd}{N}$. Then for every $B \geq 7\mu$, $\Pr[Z \geq B] \leq e^{-B}$. Furthermore, for every $\epsilon > 0$,*

$$\Pr[|Z - \mu| \geq \epsilon\mu] \leq 2\exp\{-((1+\epsilon)\log(1+\epsilon) - \epsilon)\mu\}.$$

We will make use of the second bound from Lemma 2.8 in the following form, when $\mu = o(1)$:

COROLLARY 2.9. *Let Z be a hypergeometric random variable with parameters (N, K, d), and $\mu = \frac{Kd}{N}$. Then $\Pr[Z \geq c] \leq 2e^c(c/\mu)^{-c}$.*

***d*-wise Independence** It will be convenient to divide the input into blocks, which we will argue are random. In order to avoid the $\widetilde{\Theta}(n)$ cost of sampling a uniform random permutation to define these blocks, we use a d-wise independent family of permutations.

DEFINITION 2.10. Let $\{\tau_s : [n] \to [n]\}_{s\in\mathcal{S}}$ for some finite seed set $\mathcal{S}$. For $d \in \mathbb{N}$ and $\delta \in (0,1)$, we say that τ_s is a d-wise δ-independent permutation (family) if for s chosen uniformly at random from $\mathcal{S}$, for any distinct $i_1, \ldots, i_d \in [n]$ and distinct $i'_1, \ldots, i'_d \in [n]$,

$$\left|\Pr[\tau_s(i_1) = i'_1, \ldots, \tau_s(i_d) = i'_d] - \frac{1}{n(n-1)\ldots(n-d+1)}\right| \leq \delta.$$

For $d \in \text{polylog}(n)$, and any $\delta \in (0,1)$, there exist families of d-wise δ-independent permutations with the following properties (see, for example, [28]):

- s can be sampled in $O(d \log n \log \frac{1}{\delta})$ time and space.
- For any $s \in \mathcal{S}, i \in [n]$, $\tau_s(i)$ can be computed in time $\text{poly}(d \log n \log \frac{1}{\delta})$.
- For any $s \in \mathcal{S}, i' \in [n]$, $\tau_s^{-1}(i')$ can be computed in time $\text{poly}(d \log n \log \frac{1}{\delta})$.[4]

We will design our algorithms assuming such a construction for $\delta = 0$, although this is not known to exist. By taking δ to be a sufficiently small inverse polynomial, our algorithm will not notice the difference.

3. Framework

In this section, we present the Multidimensional Quantum Walk Framework, which defines a quantum algorithm from a network, and certain subroutines. In Section 3.1, we describe the type of algorithm that will be used to prove our main theorem. In Section 3.2, we state and prove our main theorem, Theorem 3.10.

4 For example, in d-wise independent permutation families based on Feistel networks applied to d-wise independent functions h, inverting τ_h is as easy as computing h.

3.1 Phase Estimation Algorithms

In this section, we formally define a particular kind of quantum algorithm that uses phase estimation [30], and describe ingredients sufficient to analyse such an algorithm. All algorithms in this paper are of this specific form.

DEFINITION 3.1 (Parameters of a Phase Estimation Algorithm). For an implicit input $x \in \{0,1\}^*$, fix a finite-dimensional complex inner product space H, a unit vector $|\psi_0\rangle \in H$, and sets of vectors $\Psi^{\mathcal{A}}, \Psi^{\mathcal{B}} \subset H$. We further assume that $|\psi_0\rangle$ is orthogonal to every vector in $\Psi^{\mathcal{B}}$. Let $\Pi_{\mathcal{A}}$ be the orthogonal projector onto $\mathcal{A} = \text{span}\{\Psi^{\mathcal{A}}\}$, and similarly for $\Pi_{\mathcal{B}}$.

Then $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ defines a quantum algorithm as follows. Let

$$U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I). \tag{2}$$

Do phase estimation[5] of $U_{\mathcal{AB}}$ on initial state $|\psi_0\rangle$ to a certain precision, measure the phase register, and output 1 if the measured phase is 0, and output 0 otherwise. Theorem 3.8 at the end of this section describes what precision is sufficient, and when we can expect the output to be 1 and when 0.

In practice, unitaries like $U_{\mathcal{AB}}$ that are the product of two reflections are nice to work with because if each of $\Psi^{\mathcal{A}}$ and $\Psi^{\mathcal{B}}$ is a pairwise orthogonal set, implementing $U_{\mathcal{AB}}$ can be reduced to generating the states in $\Psi^{\mathcal{A}}$ and $\Psi^{\mathcal{B}}$ respectively, and a product of reflections has sufficient structure to analyse the relevant eigenspaces, as will become clear throughout this section.

Negative Analysis: The first of the two cases we want to distinguish with a phase estimation algorithm is the *negative case*, in which there exists a negative witness, defined as follows.

DEFINITION 3.2 (Negative Witness). A *δ-negative witness* for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ is a pair of vectors $|w_{\mathcal{A}}\rangle, |w_{\mathcal{B}}\rangle \in H$ such that $|\psi_0\rangle = |w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle$; and $|w_{\mathcal{A}}\rangle$ is mostly in the space $\mathcal{A}$, and $|w_{\mathcal{B}}\rangle$ is mostly in the space $\mathcal{B}$, in the sense that $\|(I - \Pi_{\mathcal{A}})|w_{\mathcal{A}}\rangle\|^2 \leq \delta$ and $\|(I - \Pi_{\mathcal{B}})|w_{\mathcal{B}}\rangle\|^2 \leq \delta$.

For intuition, it is useful to think of the case when $\delta = 0$. There exists a 0-negative witness precisely when $|\psi_0\rangle \in \mathcal{A} + \mathcal{B} = (\mathcal{A}^\perp \cap \mathcal{B}^\perp)^\perp$. For the rest of this section, we write Λ_Θ for the orthogonal projector onto the span of the $e^{i\theta}$-eigenspaces of $U_{\mathcal{AB}}$ with $|\theta| \leq \Theta$. The negative analysis relies on the effective spectral gap lemma:

LEMMA 3.3 (Effective Spectral Gap Lemma [31]). *Fix $\Theta \in (0, \pi)$. If $|\psi_{\mathcal{A}}\rangle \in \mathcal{A}$, then*

$$\|\Lambda_\Theta(I - \Pi_{\mathcal{B}})|\psi_{\mathcal{A}}\rangle\| \leq \frac{\Theta}{2} \||\psi_{\mathcal{A}}\rangle\| .$$

5 For what is meant precisely by “phase estimation”, refer to the proof of Theorem 3.8.

LEMMA 3.4 (Negative Analysis). *Fix $\delta \geq 0$ and $\Theta \in (0, \pi)$. Suppose there exists a δ-negative witness, $|w_{\mathcal{A}}\rangle$, $|w_{\mathcal{B}}\rangle$, for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$. Then we have:*

$$\|\Lambda_\Theta|\psi_0\rangle\| \leq \frac{\Theta}{2}\||w_{\mathcal{A}}\rangle\| + 2\sqrt{\delta}.$$

PROOF. We can apply the effective spectral gap lemma to $\Pi_{\mathcal{A}}|w_{\mathcal{A}}\rangle \in \mathcal{A}$, to get:

$$\begin{aligned}
&\frac{\Theta}{2}\|\Pi_{\mathcal{A}}|w_{\mathcal{A}}\rangle\| \geq \|\Lambda_\Theta(I-\Pi_{\mathcal{B}})\Pi_{\mathcal{A}}|w_{\mathcal{A}}\rangle\| \\
&\frac{\Theta}{2}\||w_{\mathcal{A}}\rangle\| \geq \|\Lambda_\Theta\left(I-\Pi_{\mathcal{B}}-(I-\Pi_{\mathcal{B}})(I-\Pi_{\mathcal{A}})\right)|w_{\mathcal{A}}\rangle\| \\
&\geq \|\Lambda_\Theta(I-\Pi_{\mathcal{B}})|w_{\mathcal{A}}\rangle\| - \|\Lambda_\Theta(I-\Pi_{\mathcal{B}})(I-\Pi_{\mathcal{A}})|w_{\mathcal{A}}\rangle\| && \text{by the triangle ineq.} \\
&\geq \|\Lambda_\Theta(I-\Pi_{\mathcal{B}})(|\psi_0\rangle - |w_{\mathcal{B}}\rangle)\| - \|(I-\Pi_{\mathcal{A}})|w_{\mathcal{A}}\rangle\| && \text{since } |\psi_0\rangle = |w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle. \\
&\geq \|\Lambda_\Theta(I-\Pi_{\mathcal{B}})|\psi_0\rangle\| - \|\Lambda_\Theta(I-\Pi_{\mathcal{B}})|w_{\mathcal{B}}\rangle\| - \|(I-\Pi_{\mathcal{A}})|w_{\mathcal{A}}\rangle\|. && \text{by the triangle ineq.}
\end{aligned}$$

Since $|\psi_0\rangle$ is orthogonal to $\mathcal{B}$, and $\|(I-\Pi_{\mathcal{A}})|w_{\mathcal{A}}\rangle\| \leq \sqrt{\delta}$ and similarly for $\mathcal{B}$, the result follows. ■

Positive Analysis: We want to distinguish the case where there exists a negative witness (the negative case) from the *positive case*, which is the case where there exists a positive witness, defined as follows.

DEFINITION 3.5 (Positive Witness). A δ-positive witness for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ is a vector $|w\rangle \in H$ such that $\langle\psi_0|w\rangle \neq 0$ and $|w\rangle$ is almost orthogonal to all $|\psi\rangle \in \Psi^{\mathcal{A}} \cup \Psi^{\mathcal{B}}$, in the sense that $\|\Pi_{\mathcal{A}}|w\rangle\|^2 \leq \delta\||w\rangle\|^2$ and $\|\Pi_{\mathcal{B}}|w\rangle\|^2 \leq \delta\||w\rangle\|^2$.[6]

Again, for intuition, we consider the case where $\delta = 0$. A 0-positive witness is exactly a component of $|\psi_0\rangle$ in $(\mathcal{A}+\mathcal{B})^\perp$, which exists precisely when $|\psi_0\rangle \notin \mathcal{A}+\mathcal{B}$. Thus, the case where there exists a 0-positive witness is the complement of the case where there exists a 0-negative witness, so it is theoretically possible to distinguish these two cases. When $\delta > 0$, the two cases may or may not be distinct, depending on δ, and the overlap between $\mathcal{A}$ and $\mathcal{B}$.

When $|w\rangle$ is a 0-positive witness, it is straightforward to see that

$$\|\Lambda_0|\psi_0\rangle\| \geq \frac{|\langle w|\psi_0\rangle|}{\||w\rangle\|},$$

where Λ_0 is the orthogonal projector onto the $(+1)$-eigenspace of $U_{\mathcal{AB}}$. For the case of $\delta > 0$, we need the following lemma, analogous to the effective spectral gap lemma.

6 We note that for technical reasons, positive witness error is defined multiplicatively (relative error), whereas negative witness error was defined additively. We could also have defined negative witness error as relative error, but it would have been relative to $\||w_{\mathcal{A}}\rangle\|^2$ for both $\|(I-\Pi_{\mathcal{A}})|w_{\mathcal{A}}\rangle\|^2$ (makes perfect sense) and $\|(I-\Pi_{\mathcal{B}})|w_{\mathcal{B}}\rangle\|^2$ (confusing).

LEMMA 3.6 (Effectively Zero Lemma). *Fix $\delta \geq 0$ and $\Theta \in (0, \pi)$. For $|\psi\rangle \in H$ such that $\|\Pi_{\mathcal{A}}|\psi\rangle\|^2 \leq \delta \||\psi\rangle\|^2$ and $\|\Pi_{\mathcal{B}}|\psi\rangle\|^2 \leq \delta \||\psi\rangle\|^2$,*

$$\|(I - \Lambda_\Theta)|\psi\rangle\|^2 \leq \frac{4\pi^2 \delta \||\psi\rangle\|^2}{\Theta^2}.$$

PROOF. Let $\{\theta_j\}_{j\in J} \subset (-\pi, \pi]$ be the set of eigenphases of $U_{\mathcal{AB}}$, and let Π_j be the orthogonal projector onto the $e^{i\theta_j}$-eigenspace of $U_{\mathcal{AB}}$, so we can write:

$$U_{\mathcal{AB}} = \sum\nolimits_{j\in J} e^{i\theta_j} \Pi_j. \tag{3}$$

We have (see (2))

$$U_{\mathcal{AB}}|\psi\rangle = |\psi\rangle + 4\Pi_{\mathcal{A}}\Pi_{\mathcal{B}}|\psi\rangle - 2\Pi_{\mathcal{A}}|\psi\rangle - 2\Pi_{\mathcal{B}}|\psi\rangle, \tag{4}$$

and using the triangle inequality, $\|\Pi_{\mathcal{A}}|\psi\rangle\|^2 \leq \delta \||\psi\rangle\|^2$ and $\|\Pi_{\mathcal{B}}|\psi\rangle\|^2 \leq \delta \||\psi\rangle\|^2$, we can compute

$$\begin{aligned} \|4\Pi_{\mathcal{A}}\Pi_{\mathcal{B}}|\psi\rangle - 2\Pi_{\mathcal{A}}|\psi\rangle - 2\Pi_{\mathcal{B}}|\psi\rangle\|^2 &= \|2(2\Pi_{\mathcal{A}} - I)\Pi_{\mathcal{B}}|\psi\rangle - 2\Pi_{\mathcal{A}}|\psi\rangle\|^2 \\ &\leq \left(\|2(2\Pi_{\mathcal{A}} - I)\Pi_{\mathcal{B}}|\psi\rangle\| + \|2\Pi_{\mathcal{A}}|\psi\rangle\|\right)^2 \leq 16\delta \||\psi\rangle\|^2 . \end{aligned} \tag{5}$$

Thus, by (4) and (5):

$$\begin{aligned} \|U_{\mathcal{AB}}|\psi\rangle - |\psi\rangle\|^2 &\leq 16\delta \||\psi\rangle\|^2 \\ \sum_{j\in J} |e^{i\theta_j} - 1|^2 \left\|\Pi_j|\psi\rangle\right\|^2 &\leq 16\delta \||\psi\rangle\|^2 && \text{by (3)} \\ \sin^2\frac{\Theta}{2} \sum_{j\in J:|\theta_j|>\Theta} \left\|\Pi_j|\psi\rangle\right\|^2 &\leq 4\delta \||\psi\rangle\|^2 && \text{since } |e^{i\theta_j} - 1|^2 = 4\sin^2\frac{\theta_j}{2} \\ \|(I - \Lambda_\Theta)|\psi\rangle\|^2 &\leq \frac{4\delta \||\psi\rangle\|^2}{\sin^2\frac{\Theta}{2}}. \end{aligned}$$

Then since $\sin^2\frac{\Theta}{2} \geq \frac{4}{\pi^2}\frac{\Theta^2}{4}$ whenever $\Theta \in (-\pi, \pi)$, the result follows. ∎

LEMMA 3.7 (Positive Analysis). *Fix $\delta \geq 0$ and $\Theta \in (0, \pi)$. Suppose there exists a δ-positive witness $|w\rangle$ for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$. Then*

$$\|\Lambda_\Theta|\psi_0\rangle\| \geq \frac{|\langle\psi_0|w\rangle|}{\||w\rangle\|} - \frac{2\sqrt{\delta}\pi}{\Theta}.$$

PROOF. We compute:

$$\begin{aligned} |\langle\psi_0|\Lambda_\Theta|w\rangle| &\geq |\langle\psi_0|w\rangle| - |\langle\psi_0|(I - \Lambda_\Theta)|w\rangle| && \text{by the triangle ineq.} \\ &\geq |\langle\psi_0|w\rangle| - \||\psi_0\rangle\| \cdot \|(I - \Lambda_\Theta)|w\rangle\| && \text{by Cauchy-Schwarz} \\ &\geq |\langle\psi_0|w\rangle| - \frac{2\sqrt{\delta}\pi}{\Theta} \||w\rangle\|, \end{aligned}$$

where we used $\||\psi_0\rangle\| = 1$ and Lemma 3.6. Then:

$$\|\Lambda_\Theta|\psi_0\rangle\| \geq \left\| \frac{\Lambda_\Theta|w\rangle\langle w|\Lambda_\Theta}{\|\Lambda_\Theta|w\rangle\|^2}|\psi_0\rangle \right\| = \frac{|\langle w|\Lambda_\Theta|\psi_0\rangle|}{\|\Lambda_\Theta|w\rangle\|} \geq \frac{|\langle\psi_0|w\rangle| - \frac{2\sqrt{\delta}\pi}{\Theta}\||w\rangle\|}{\||w\rangle\|} = \frac{|\langle\psi_0|w\rangle|}{\||w\rangle\|} - \frac{2\sqrt{\delta}\pi}{\Theta}. \blacksquare$$

Phase Estimation Algorithm: By Lemma 3.7, if there exists a δ-positive witness, which happens precisely when there is some component of $|\psi_0\rangle$ that is nearly orthogonal to $\mathcal{A}+\mathcal{B}$, then $|\psi_0\rangle$ overlaps the $e^{i\theta}$-eigenspaces of $U_{\mathcal{AB}}$ for small θ, say with $|\theta| \leq \Theta_0$ for some small-ish choice of Θ_0. The precise overlap depends on the size of this component, and allows us to lower bound the probability that phase estimation of $U_{\mathcal{AB}}$ on $|\psi_0\rangle$ will result in a 0 in the phase register. On the other hand, if $|\psi_0\rangle$ is actually in $\mathcal{A}+\mathcal{B}$, then Lemma 3.4 upper bounds the overlap of $|\psi_0\rangle$ with small phase spaces, where "small" is determined by the parameter $\Theta > \Theta_0$. This allows us to upper bound the probability that phase estimation of $U_{\mathcal{AB}}$ on $|\psi_0\rangle$, to precision Θ, will result in a 0 in the phase register. The key is then to choose the parameter Θ small enough so that there is a constant gap between the lower bound on the probability of a 0 phase in the positive case, and the upper bound on the probability of a 0 phase in the negative case. This leads to the following theorem.

THEOREM 3.8. *Fix $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ as in Definition 3.1. Suppose we can generate the state $|\psi_0\rangle$ in cost* S*, and implement $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$ in cost* A*.*

Let $c_+ \in [1, 50]$ be some constant, and let $C_- \geq 1$ be a positive real number that may scale with $|x|$. Let δ and δ' be positive real parameters such that

$$\delta \leq \frac{1}{(8c_+)^3\pi^8 C_-} \quad \text{and} \quad \delta' \leq \frac{3}{4}\frac{1}{\pi^4 c_+}.$$

Suppose we are guaranteed that exactly one of the following holds:

Positive Condition: *There is a δ-positive witness $|w\rangle$ (see Definition 3.5), s.t. $\frac{|\langle w|\psi_0\rangle|^2}{\||w\rangle\|^2} \geq \frac{1}{c_+}$.*
Negative Condition: *There is a δ'-negative witness $|w_{\mathcal{A}}\rangle, |w_{\mathcal{B}}\rangle$ (Definition 3.2), s.t. $\||w_{\mathcal{A}}\rangle\|^2 \leq C_-$.*

Suppose we perform $T = \sqrt{8}\pi^4 c_+ \sqrt{C_-}$ steps of phase estimation of $U_{\mathcal{AB}}$ on initial state $|\psi_0\rangle$, and output 1 if and only if the measured phase is 0, otherwise we output 0. Then

Positive Case: *If the positive condition holds, the algorithm outputs 1 with probability $\geq \frac{2.25}{\pi^2 c_+} \geq \frac{2.25}{50\pi^2}$.*
Negative Case: *If the negative condition holds, the algorithm outputs 1 with probability $\leq \frac{2}{\pi^2 c_+}$.*

Thus, the algorithm distinguishes between these two cases with bounded error, in cost

$$O\left(\mathsf{S} + \sqrt{C_-}\mathsf{A}\right).$$

PROOF. Let $\{\theta_j\}_{j\in J} \subset (-\pi, \pi]$ be the set of phases of $U_{\mathcal{AB}}$, and let Π_j be the orthogonal projector onto the $e^{i\theta_j}$-eigenspace of $U_{\mathcal{AB}}$, so we can write:

$$U_{\mathcal{AB}} = \sum\nolimits_{j\in J} e^{i\theta_j}\Pi_j.$$

After making a superposition over t from 0 to $T-1$ in the phase register, and applying $U^t_{\mathcal{AB}}$ to the input register conditioned on the phase register, as one does in phase estimation [30], we have the state:

$$\sum_{t=0}^{T-1} \frac{1}{\sqrt{T}}|t\rangle U^t_{\mathcal{AB}}|\psi_0\rangle = \sum_{j\in J}\sum_{t=0}^{T-1} \frac{1}{\sqrt{T}}|t\rangle e^{it\theta_j}\Pi_j|\psi_0\rangle.$$

The phase estimation algorithm then proceeds by applying an inverse Fourier transform, $F_T^\dagger$, to the first register, and then measuring the result, to obtain some $t \in \{0, \ldots, T-1\}$. We choose the output bit based on whether $t = 0$ or not. The probability of measuring 0 is:

$$\begin{aligned} p_0 &:= \left\| \langle 0|F_T^\dagger \otimes I \left(\sum_{j\in J}\sum_{t=0}^{T-1} \frac{1}{\sqrt{T}}|t\rangle e^{it\theta_j}\Pi_j|\psi_0\rangle \right) \right\|^2 = \left\| \sum_{t=0}^{T-1} \frac{1}{\sqrt{T}}\langle t| \otimes I \left(\sum_{j\in J}\sum_{t=0}^{T-1} \frac{1}{\sqrt{T}}|t\rangle e^{it\theta_j}\Pi_j|\psi_0\rangle \right) \right\|^2 \\ &= \frac{1}{T^2}\left\| \sum_{j\in J}\sum_{t=0}^{T-1} e^{it\theta_j}\Pi_j|\psi_0\rangle \right\|^2 = \frac{1}{T^2}\sum_{j\in J:\theta_j\neq 0} \left| \frac{1-e^{i\theta_j T}}{1-e^{i\theta_j}} \right|^2 \left\|\Pi_j|\psi_0\rangle\right\|^2 + \|\Lambda_0|\psi_0\rangle\|^2 \\ &= \frac{1}{T^2}\sum_{j\in J:\theta_j\neq 0} \frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)} \left\|\Pi_j|\psi_0\rangle\right\|^2 + \|\Lambda_0|\psi_0\rangle\|^2, \end{aligned} \tag{6}$$

since $\left|\sum_{t=0}^{T-1} e^{it\theta}\right| = \left|\frac{1-e^{i\theta T}}{1-e^{i\theta}}\right|$, and $|1-e^{i\theta}|^2 = 4\sin^2\frac{\theta}{2}$ for any $\theta \in \mathbb{R}$. We will analyse the positive and negative cases one-by-one.

Positive Case: Assume the positive condition, which allows us to apply Lemma 3.7. In the following, we will use the identities $\sin^2\theta \leq \theta^2$ for all θ, and $\sin^2\theta \geq 4\theta^2/\pi^2$ whenever $|\theta| \leq \pi/2$. Let $\Theta_0 = \pi/T$. Continuing from (6), we can lower bound the probability of measuring a 0 in the phase register by:

$$\begin{aligned} p_0 &\geq \frac{1}{T^2}\sum_{j\in J:0<|\theta_j|\leq\Theta_0} \frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)} \left\|\Pi_j|\psi_0\rangle\right\|^2 + \|\Lambda_0|\psi_0\rangle\|^2 \\ &\geq \frac{1}{T^2}\sum_{j\in J:0<|\theta_j|\leq\Theta_0} \frac{4(T\theta_j/2)^2/\pi^2}{(\theta_j/2)^2} \left\|\Pi_j|\psi_0\rangle\right\|^2 + \|\Lambda_0|\psi_0\rangle\|^2 && \text{since } |T\theta_j/2| \leq T\Theta_0/2 = \pi/2 \\ &\geq \frac{4}{\pi^2}\left(\sum_{j\in J:0<|\theta_j|\leq\Theta_0} \left\|\Pi_j|\psi_0\rangle\right\|^2 + \|\Lambda_0|\psi_0\rangle\|^2 \right) \\ &\geq \frac{4}{\pi^2}\left\|\Lambda_{\Theta_0}|\psi_0\rangle\right\|^2 \geq \frac{4}{\pi^2}\left(\frac{|\langle\psi_0|w\rangle|}{\||w\rangle\|} - \frac{2\sqrt{\delta}\pi}{\Theta_0} \right)^2 && \text{by Lemma 3.7} \end{aligned}$$

$$\geq \frac{4}{\pi^2}\left(\frac{1}{\sqrt{c_+}} - \frac{2\pi T}{\pi}\frac{1}{(8c_+)^{3/2}\pi^4\sqrt{C_-}}\right)^2 \qquad \sqrt{\delta} \leq \frac{1}{(8c_+)^{3/2}\pi^4\sqrt{C_-}}$$
$$= \frac{4}{\pi^2}\left(\frac{1}{\sqrt{c_+}} - \frac{2\sqrt{8}\pi^4 c_+\sqrt{C_-}}{(8c_+)^{3/2}\pi^4\sqrt{C_-}}\right)^2 = \frac{4}{\pi^2}\left(\frac{3}{4}\right)^2\frac{1}{c_+} = \frac{2.25}{\pi^2}\frac{1}{c_+} \geq \frac{2.25}{50\pi^2}.$$

Negative Case: Assume the negative condition, which allows us to apply Lemma 3.4. In the following, we will use the identities $\sin^2\theta \leq \min\{1,\theta^2\}$ for all θ, and $\sin^2(\theta/2) \geq \theta^2/\pi^2$ whenever $|\theta| \leq \pi$. Let $\Theta = \pi^{-2}(c_+C_-)^{-1/2}$. Continuing from (6), we can *upper* bound the probability of measuring a 0 in the phase register by:

$$\begin{aligned}
p_0 &= \frac{1}{T^2}\sum_{j\in J:0<|\theta_j|\leq\Theta}\frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)}\left\|\Pi_j|\psi_0\rangle\right\|^2 + \frac{1}{T^2}\sum_{j\in J:|\theta_j|>\Theta}\frac{\sin^2(T\theta_j/2)}{\sin^2(\theta_j/2)}\left\|\Pi_j|\psi_0\rangle\right\|^2 + \||\Lambda_0|\psi_0\rangle\|^2 \\
&\leq \frac{1}{T^2}\sum_{j\in J:0<|\theta_j|\leq\Theta}\frac{(T\theta_j/2)^2}{(\theta_j/\pi)^2}\left\|\Pi_j|\psi_0\rangle\right\|^2 + \frac{1}{T^2}\sum_{j\in J:|\theta_j|>\Theta}\frac{1}{(\theta_j/\pi)^2}\left\|\Pi_j|\psi_0\rangle\right\|^2 + \||\Lambda_0|\psi_0\rangle\|^2 \\
&\leq \frac{\pi^2}{4}\||\Lambda_\Theta|\psi_0\rangle\|^2 + \frac{1}{T^2}\frac{\pi^2}{\Theta^2} \\
&\leq \frac{\pi^2}{4}\left(\frac{\Theta}{2}\||w_{\mathcal{A}}\rangle\| + 2\sqrt{\delta'}\right)^2 + \frac{1}{8\pi^8 c_+^2 C_-}\frac{\pi^2}{\Theta^2} && \text{by Lemma 3.4} \\
&\leq \frac{\pi^2}{4}\left(\frac{1}{\pi^2\sqrt{c_+C_-}}\sqrt{C_-} + 2\frac{\sqrt{3}}{2\pi^2\sqrt{c_+}}\right)^2 + \frac{\pi^2}{8\pi^8 c_+^2 C_-}\pi^4 c_+ C_- && \sqrt{\delta'} \leq \frac{\sqrt{3}}{2}\frac{1}{\pi^2\sqrt{c_+}} \text{ and } \||w_{\mathcal{A}}\rangle\|^2 \leq C_- \\
&\leq \frac{1}{4\pi^2 c_+}\left(1+\sqrt{3}\right)^2 + \frac{1}{8\pi^2 c_+} \leq \frac{2}{\pi^2 c_+}.
\end{aligned}$$

To complete the proof, it is easily verified that the described algorithm has the claimed cost. ■

3.2 Multidimensional Quantum Walks

Our new framework extends the electric network framework, so for intuition, we first describe this framework and sketch how it works, before stating our main theorem extending the electric network framework in Section 3.2.2.

3.2.1 Sketch of the Electric Network Framework

We begin by sketching the electric network framework of Belovs [10], on which our new framework is based. This explanation is for intuition, and the expert reader may skip it. Fix a network G that can depend on an input $x \in \{0,1\}^n$, as in Definition 2.1. Let $V_0, V_{\mathsf{M}} \subset V(G)$ be disjoint sets, σ an *initial distribution* on V_0, and $M \subseteq V_{\mathsf{M}}$ a *marked set*. For simplicity, let us assume that G is bipartite, which is always possible to ensure, for example, by replacing each edge with a path

of length two.[7] Let $(V_{\mathcal{A}}, V_{\mathcal{B}})$ be the bipartition, and assume for convenience that $V_0 \subseteq V_{\mathcal{A}}$, and $V_M \subseteq V_{\mathcal{B}}$.

Then the electric network framework is a way of designing a quantum algorithm that detects if $M \neq \emptyset$ with bounded error. To explain how this algorithm works conceptually, we will modify G by adding a vertex v_0, which is connected to each vertex $u \in V_0$ by an edge of weight $w_0\sigma(u)$ for some parameter w_0, and to each vertex in M by an edge of weight w_M. Call this new graph G'. We assume the new edges are pointing into v_0, so

$$V(G') = V(G) \cup \{v_0\} \text{ and } \overrightarrow{E}(G') = \overrightarrow{E}(G) \cup \{(u, v_0) : u \in V_0 \cup M\}.$$

The Algorithm: We describe a phase estimation algorithm, of the form given in Section 3.1. Let

$$H = \text{span}\{|u, v\rangle : (u, v) \in \overrightarrow{E}(G')\}.$$

For any $(u, v) \in \overrightarrow{E}(G')$, H does not contain a vector $|v, u\rangle$, so we define:

$$|v, u\rangle := -|u, v\rangle.$$

It is conceptually convenient to think of negation as reversing the direction of an edge, but note that this means that (u, v) and (v, u) are *not* labelling orthogonal states. For each $u \in V(G) \setminus (V_0 \cup M)$, we define a *star state*[8]:

$$|\psi_\star^{G'}(u)\rangle := \sum_{v \in \Gamma(u)} \sqrt{w_{u,v}}|u, v\rangle = \sum_{v \in \Gamma^+(u)} \sqrt{w_{u,v}}|u, v\rangle - \sum_{v \in \Gamma^-(u)} \sqrt{w_{u,v}}|v, u\rangle,$$

where the last expression shows how to express $|\psi_\star^{G'}(u)\rangle$ in the standard basis of H, which only includes $(u, v) \in \overrightarrow{E}(G')$. Recall that $\Gamma^+(u)$ and $\Gamma^-(u)$ are the out- and in-neighbourhoods of u, defined in (1). Similarly, for $u \in V_0$, we define

$$|\psi_\star^{G'}(u)\rangle := \sum_{v \in \Gamma(u)} \sqrt{w_{u,v}}|u, v\rangle + \sqrt{w_0\sigma(u)}|u, v_0\rangle,$$

and for $u \in M$,

$$|\psi_\star^{G'}(u)\rangle := \sum_{v \in \Gamma(u)} \sqrt{w_{u,v}}|u, v\rangle + \sqrt{w_M}|u, v_0\rangle,$$

which simply includes the new edges we add when going from G to G'. The star states are not normalised, but if we normalise $|\psi_\star^{G'}(u)\rangle$, we get a *quantum walk state*: a state that, if measured, would allow one to sample from the neighbours of u as in a random walk on G'.

We use an initial state based on the initial distribution σ:

$$|\psi_0\rangle = \sum_{u \in V_0} \sqrt{\sigma(u)}|u, v_0\rangle = |\sigma\rangle|v_0\rangle.$$

---

7 In fact, we essentially do this in the proof of our new framework, since we replace each edge with a sort of "algorithm gadget", which is analogous to a path, and always has even length.

8 To simplify things here, these are slightly different from the star states we use in Theorem 3.10 and Definition 3.9. They are the same, up to sign, if $\overrightarrow{E}(G) \subset V_{\mathcal{A}} \times V_{\mathcal{B}}$.

Finally, let

$$\Psi^{\mathcal{A}} = \{|\psi_\star^{G'}(u)\rangle : u \in V_{\mathcal{A}}\} \text{ and } \Psi^{\mathcal{B}} = \{|\psi_\star^{G'}(u)\rangle : u \in V_{\mathcal{B}}\}.$$

Since $V_{\mathcal{A}}$ and $V_{\mathcal{B}}$ are each independent sets, each set is a pairwise orthogonal set of states. Thus, being able to generate these states[9] is sufficient to be able to reflect around $\mathcal{A} = \text{span}\{\Psi^{\mathcal{A}}\}$ and $\mathcal{B} = \text{span}\{\Psi^{\mathcal{B}}\}$, in order to implement:

$$U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I).$$

It can then be verified that $(\mathcal{A} + \mathcal{B})^\perp$ is the span of all *circulation* states on G':

$$|C\rangle = \sum_{(u,v)\in\vec{E}(G')} \frac{C(u,v)}{\sqrt{\mathsf{w}_{u,v}}}|u,v\rangle,$$

where C is a circulation (see Definition 2.2). Thus, a 0-positive witness for a phase estimation algorithm with these parameters (see Definition 3.5) is always a circulation.

Positive Case: Suppose that whenever $M \neq \emptyset$, there exists a flow θ on G (see Definition 2.2) with sources in V_0 and sinks in M, and suppose $\theta(u) = \sigma(u)$ for all $u \in V_0$. Then we can extend θ to a circulation, C_θ, on G' by sending all excess flow from M into v_0, and then sending the flow out from v_0 to V_0, distributed according to σ. The state corresponding to C_θ will include a term $\sum_{u\in V_0} \frac{\sigma(u)}{\sqrt{\sigma(u)\mathsf{w}_0}}|u,v_0\rangle = \frac{1}{\sqrt{\mathsf{w}_0}}|\psi_0\rangle$ – the part that distributes flow from v_0 to V_0 according to σ. Thus, $|C_\theta\rangle$ is a positive witness: a state in $(\mathcal{A} + \mathcal{B})^\perp$ that has non-zero overlap with $|\psi_0\rangle$. In particular, $\langle\psi_0|C_\theta\rangle = \frac{1}{\sqrt{\mathsf{w}_0}}$, and one can check that $\||C_\theta\rangle\|^2 \approx \frac{1}{\mathsf{w}_0} + \mathcal{E}(\theta)$, where $\mathcal{E}(\theta)$ is the energy of θ (see Definition 2.2). So in particular,

$$\frac{\||C_\theta\rangle\|^2}{|\langle\psi_0|C_\theta\rangle|^2} \approx 1 + \mathsf{w}_0\mathcal{E}(\theta).$$

Negative Case: On the other hand, whenever $M = \emptyset$, if we add up *all* star states $|\psi_\star^{G'}(u)\rangle$ for $u \in V(G) = V_{\mathcal{A}} \cup V_{\mathcal{B}}$ (this does not include v_0), for every edge $(u,v) \in \vec{E}(G)$, we will get a contribution of $\sqrt{\mathsf{w}_{u,v}}|u,v\rangle$ from $|\psi_\star^{G'}(u)\rangle$, and a contribution of $\sqrt{\mathsf{w}_{u,v}}|v,u\rangle = -\sqrt{\mathsf{w}_{u,v}}|u,v\rangle$ from $|\psi_\star^{G'}(v)\rangle$, which will add up to 0. However, the edges in $\vec{E}(G') \setminus \vec{E}(G)$, which are precisely the edges from $u \in V_0$ to v_0 (as $M = \emptyset$) will only appear in $|\psi_\star^{G'}(u)\rangle$, which contributes $\sqrt{\mathsf{w}_0\sigma(u)}|u,v_0\rangle$. Thus, adding up all star states results in the vector $\sqrt{\mathsf{w}_0}|\psi_0\rangle$, so if we let

$$|w_{\mathcal{A}}\rangle = \frac{1}{\sqrt{\mathsf{w}_0}}\sum_{u\in V_{\mathcal{A}}}|\psi_\star^{G'}(u)\rangle \text{ and } |w_{\mathcal{B}}\rangle = \frac{1}{\sqrt{\mathsf{w}_0}}\sum_{u\in V_{\mathcal{B}}}|\psi_\star^{G'}(u)\rangle$$

then these form a 0-negative witness (see Definition 3.2), with

$$\||w_{\mathcal{A}}\rangle\|^2 = \frac{1}{\mathsf{w}_0}\sum_{e\in\vec{E}(G')}\mathsf{w}_e = \frac{1}{\mathsf{w}_0}\mathcal{W}(G') \approx \frac{1}{\mathsf{w}_0}\mathcal{W}(G).$$

9 Note that $v_0 \notin V_{\mathcal{A}} \cup V_{\mathcal{B}}$. This is important, because generating the star state for v_0 would require knowing precisely which vertices of $V_{\mathcal{B}}$ are in M.

Electric Network Framework: By applying Theorem 3.8, we can get the following. Let $\mathcal{R}$ be an upper bound on $\min_\theta \mathcal{E}(\theta)$ where θ runs over all flows from σ to M, whenever $M \neq \emptyset$. Let $\mathcal{W}$ be an upper bound on $\mathcal{W}(G)$. Define:

$$c_+ = 1 + \mathsf{w}_0\mathcal{R} \text{ and } C_- = \frac{1}{\mathsf{w}_0}\mathcal{W},$$

and let $\mathsf{w}_0 = \mathcal{R}^{-1}$ so that $c_+ = O(1)$. If S is the complexity of generating the state $|\sigma\rangle$, and A is the cost of generating the star states (which requires checking if a vertex is marked), then there is a quantum algorithm that decides if $M = \emptyset$ with bounded error in complexity

$$O(\mathsf{S} + \sqrt{C_-}\mathsf{A}) = O(\mathsf{S} + \sqrt{\mathcal{R}\mathcal{W}}\mathsf{A}).$$

3.2.2 The Multidimensional Quantum Walk Framework

We now state our extension of the electric network framework. In the electric network framework, we assume we can generate all star states in some cost A, which implicitly assumes that for any vertex u, we can compute the neighbours of u in time at most A. However, in some cases, the actual *transition cost* from u to v, $\mathsf{T}_{u,v}$ may vary significantly for different u, and different neighbours v of u. Our modified framework takes this variation into account, avoiding incurring a factor of the maximum $\mathsf{T}_{u,v}$. Instead, the complexity will scale as if we replaced each edge $\{u, v\}$ in G by a path of length $\mathsf{T}_{u,v}$.

The second modification we make to the electric network framework is to allow for *alternative neighbourhoods*, which we now formally define.

DEFINITION 3.9 (Alternative Neighbourhoods). For a network G, as in Definition 2.1 and Definition 2.3, a set of *alternative neighbourhoods* is a collection of states:

$$\Psi_\star = \{\Psi_\star(u) \subset \text{span}\{|u, i\rangle : i \in L(u)\} : u \in V(G)\}$$

such that for all $u \in V(G)$,

$$|\psi^G_\star(u)\rangle := \sum_{i \in L^+(u)} \sqrt{\mathsf{w}_{u,i}}|u, i\rangle - \sum_{i \in L^-(u)} \sqrt{\mathsf{w}_{u,i}}|u, i\rangle \in \Psi_\star(u).$$

We view the states of $\Psi_\star(u)$ as different possibilities for $|\psi^G_\star(u)\rangle$, only one of which is "correct." Let $d_{\max} = \max\{|L(u)| : u \in V(G)\}$. We say we can *generate* $\Psi_\star$ *in complexity* $\mathsf{A}_\star$, for some $\mathsf{A}_\star = \Omega(\log d_{\max})$, if there is a map $U_\star$ such that:

- for each $u \in V(G)$, there is an orthonormal basis $\overline{\Psi}(u) = \{|\overline{\psi}_{u,0}\rangle, \ldots, |\overline{\psi}_{u,a_u-1}\rangle\}$ for the span of $\Psi_\star(u)$, such that for all $k \in [a_u]$, $U_\star|u, k\rangle = |\overline{\psi}_{u,k}\rangle$, and
- $U_\star$ can be implemented with complexity $\mathsf{A}_\star$.

It may be possible to implement a set of alternative neighbourhoods $\Psi_\star$ for G faster than it would be possible to generate the star states of G. This happens when, given u, it is expensive to

determine the correct form of $|\psi_\star^G(u)\rangle$, but we do know that it is one of a set of easily generated states, say $|\psi_\star^1(u)\rangle$ or $|\psi_\star^2(u)\rangle$ (see the discussion in Section 1.1).

We now state the main result of this paper, from which the applications in Section 4 and Section 5 follow.

THEOREM 3.10 (Multidimensional Quantum Walk Framework). *Fix a family of networks G that may depend on some implicit input x, with disjoint sets $V_0, V_M \subset V(G)$ such that for any vertex, checking if $v \in V_0$ (resp. if $v \in V_M$) can be done in at most $\mathsf{A}_\star$ complexity. Let $M \subseteq V_M$ be the* marked set, *and σ an* initial distribution *on V_0. Let $\Psi_\star = \{\Psi_\star(u) : u \in V(G)\}$ be a set of alternative neighbourhoods for G (see Definition 3.9). For all $u \in V_0 \cup V_M$, assume that $\Psi_\star(u) = \{|\psi_\star^G(u)\rangle\}$. Fix some positive real-valued $\mathcal{W}^\mathsf{T}$ and $\mathcal{R}^\mathsf{T}$, that may scale with $|x|$. Suppose the following conditions hold.*

Setup Subroutine: *The state $|\sigma\rangle = \sum_{u\in V_0} \sqrt{\sigma(u)}|u\rangle$ can be generated in cost S, and furthermore, for any $u \in V_0$, $\sigma(u)$ can be computed in $O(1)$ complexity.*
Star State Generation Subroutine: *We can generate $\Psi_\star$ in complexity $\mathsf{A}_\star$.*
Transition Subroutine: *There is a quantum subroutine (see Definition 2.5) that implements the transition map of G (see Definition 2.3) with errors $\{\epsilon_{u,v}\}_{(u,v)\in\overrightarrow{E}(G)}$ and costs $\{\mathsf{T}_{u,v}\}_{(u,v)\in\overrightarrow{E}(G)}$. We make the following assumptions on the errors $\epsilon_{u,v}$, where $\tilde{E} \subset \overrightarrow{E}(G)$ is some (possibly unknown) set of edges on which we allow the subroutine to fail:*

TS1 *For all $(u,v) \in \overrightarrow{E}(G) \setminus \tilde{E}$, $\epsilon_{u,v} \le \epsilon$, where $\epsilon = o\left(\frac{1}{\mathcal{W}^\mathsf{T}\mathcal{R}^\mathsf{T}}\right)$.*

TS2 *For all $(u,v) \in \tilde{E}$, there is no non-trivial upper bound on $\epsilon_{u,v}$, but $\widetilde{\mathcal{W}} := \sum_{e\in\tilde{E}} \mathsf{w}_e = o\left(\frac{1}{\mathcal{R}^\mathsf{T}}\right)$.*

Checking Subroutine: *There is an algorithm that checks, for any $u \in V_M$, if $u \in M$, in cost $\mathsf{A}_\star$.*[10]
Positive Condition: *Interpreting $\mathsf{T}_{u,v}$ as a length function on $\overrightarrow{E}(G)$, G^T is the graph obtained by replacing each edge (u,v) of G with a path of length $\mathsf{T}_{u,v}$ (see Definition 2.4). If $M \neq \emptyset$, then there exists a flow θ on G (see Definition 2.2) such that*

P1 *For all $(u,v) \in \tilde{E}$, $\theta(u,v) = 0$.*
P2 *For all $u \in V(G) \setminus (V_0 \cup M)$ and $|\psi_\star(u)\rangle \in \Psi_\star(u)$,*

$$\sum_{i\in L^+(u)} \frac{\theta(u, f_u(i))\langle\psi_\star(u)|u,i\rangle}{\sqrt{\mathsf{w}_{u,i}}} - \sum_{i\in L^-(u)} \frac{\theta(u, f_u(i))\langle\psi_\star(u)|u,i\rangle}{\sqrt{\mathsf{w}_{u,i}}} = 0.$$[11]

---

10 This is without loss of generality. Suppose the checking cost is some higher value $\mathsf{C} > \mathsf{A}_\star$. Then we can simply put an outgoing edge on each vertex $u \in V_M$ that ends at a new vertex (u,b) that encodes whether $u \in M$ in the bit b. Such an edge can be implemented with transition cost C.

11 Whenever $|\psi_\star(u)\rangle = |\psi_\star^G(u)\rangle$, this condition is simply saying that $\theta(u) = 0$ (i.e. flow is conserved at u), but we require that the condition also hold for all other states in $\Psi_\star(u)$ as well.

P3 $\sum_{u\in V_0}\theta(u)=1$.[12]
P4 $\sum_{u\in V_0}\frac{|\theta(u)-\sigma(u)|^2}{\sigma(u)}\leq 1$.[13]
P5 $\mathcal{E}^{\mathsf{T}}(\theta)\leq\mathcal{R}^{\mathsf{T}}$.

Negative Condition: *If* $M=\emptyset$*, then* $\mathcal{W}(G^{\mathsf{T}})\leq\mathcal{W}^{\mathsf{T}}$.

Then there is a quantum algorithm that decides if $M=\emptyset$ *or not with bounded error in complexity:*

$$O\left(\mathsf{S}+\sqrt{\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}}}\,(\mathsf{A}_\star+\mathsf{polylog}(\mathsf{T}_{\max}))\right).$$

In the remainder of this section, we prove Theorem 3.10 by describing (parameters of) a phase estimation algorithm and analysing it using Theorem 3.8.

REMARK 3.11. For an edge $(u,v)\in\tilde{E}$, we may without loss of generality assume that $v\notin V(G)$. Suppose $i=f_u^{-1}(v)$. Then since we don't actually implement the transition $|u,i\rangle\rightarrow|v,j\rangle$ correctly anyway, we can assume that $v=(u,i)$, which is distinct from all vertices in $V(G)$, and so we can consider it an almost isolated vertex with the single backwards neighbour u. We can equivalently think of these as dangling edges, without an endpoint.

3.2.3 The Transition Subroutine

Recall from Definition 2.5 that a quantum subroutine is given by a sequence $U_0,\ldots,U_{\mathsf{T}_{\max}-1}$ of unitaries on $H=\mathrm{span}\{|z\rangle:z\in\mathcal{Z}\}$, such that we can implement $\sum_{t=0}^{\mathsf{T}_{\max}-1}|t\rangle\langle t|\otimes U_t$ in cost $\mathsf{polylog}(\mathsf{T}_{\max})$. In our case, the subroutine computes the *transition map* (see Definition 2.3), $|u,i\rangle\mapsto|v,j\rangle$, so we assume

$$\{(u,i):u\in V(G),i\in L(u)\}\subseteq\mathcal{Z}.$$

Then by conditions **TS1** and **TS2** from Theorem 3.10, for any $(u,v)\in\overrightarrow{E}(G)$, with $i=f_u^{-1}(v)$ and $j=f_v^{-1}(u)$, we have

$$\left\||v,j\rangle-U_{\mathsf{T}_{u,v}-1}\ldots U_0|u,i\rangle\right\|^2=\epsilon_{u,v},\tag{7}$$

where $\epsilon_{u,v}\leq\epsilon$ whenever $(u,v)\in\overrightarrow{E}(G)\setminus\tilde{E}$. Otherwise, we only have the trivial upper bound $\epsilon_{u,v}\leq 4$.

We will assume that in $O(1)$ time, we can check, for any $z\in\mathcal{Z}$, if $z=(u,i)$ for some $u\in V(G)$ and $i\in L(u)$, and further, whether $i\in L^+(u)$ or $i\in L^-(u)$. This is without loss of generality, by the following construction. Assume that for all $u\in V(G)$, every label in $L^+(u)$ ends with the symbol $\rightarrow$, and every label in $L^-(u)$ ends with the symbol $\leftarrow$. Further assume that no

12 Intuitively, we want to think of the flow as coming in at V_0, and exiting at M. While we do not make it a strict requirement that all sources are in V_0 and all sinks in M, this condition implies that we do not simply have all the flow coming in at vertices in V_0 and then leaving again through other vertices in V_0.

13 Intuitively, θ should be a σ-M flow, meaning that for all $u\in V_0$, $\theta(u)=\sigma(u)$. We don't make this a strict requirement, but this condition means it should hold in some approximate sense.

other $z \in \mathcal{Z}$ ends with these symbols. Then it is sufficient to check a single constant-dimensional register.

We will assume that $\mathsf{T}_{u,v}$ is always even. This assumption incurs at most a small constant slowdown. We will also assume that after exactly $\mathsf{T}_{u,v}$ steps, the algorithm sets an *internal flag register* to 1, and we will let this 1-flag be part of the final state (v, j) by letting each $i \in L(u)$ contain an extra bit set to 1. This also ensures that the state of the algorithm is never $|v, j\rangle$ before $\mathsf{T}_{u,v}$ steps have passed. This assumption is without loss of generality, because we can simply let the algorithm use an internal timer in order to decide to set a flag after exactly $\mathsf{T}_{u,v}$ steps, and uncompute this timer using our ability to compute $\mathsf{T}_{u,v}$ from the final correct state $|v, j\rangle$.

Recall from Definition 2.5 that for any $u \in V(G)$, $i \in L(u)$ and $t \in \{0, \ldots, \mathsf{T}_{\max} - 1\}$,

$$U_t \ldots U_0 |u, i\rangle \in \operatorname{span}\{|z\rangle : z \in \mathcal{Z}_{u,i}\}.$$

For convenience, we will let $\mathcal{Z}_{u,v} = \mathcal{Z}_{u,i}$, where $v = f_u(i)$. For $b \in \{0, 1\}$, let $\mathcal{Z}^b_{u,v} \subset \mathcal{Z}_{u,v}$ be the subset of states in which the algorithm's internal flag register is set to b. So by the above discussion, we have $(v, j) \in \mathcal{Z}^1_{u,v}$,

$$\forall t < \mathsf{T}_{u,v},\ U_t \ldots U_0 |u, i\rangle \in \mathcal{Z}^0_{u,v}, \text{ and } \forall t \geq \mathsf{T}_{u,v},\ U_t \ldots U_0 |u, i\rangle \in \mathcal{Z}^1_{u,v}.$$

3.2.4 Parameters of the Phase Estimation Algorithm

Our phase estimation algorithm will work on the space:

$$\begin{aligned} H' = {} & \operatorname{span}\{|u, i\rangle|0\rangle : u \in V(G), i \in L^+(u) \cup \{0\}\} \oplus \operatorname{span}\{|v, j\rangle|0\rangle : v \in V(G), j \in L^-(v)\} \\ & \oplus \bigoplus_{(u,v) \in \overrightarrow{E}(G)} \operatorname{span}\{|z\rangle|t\rangle : z \in \mathcal{Z}^0_{u,v}, t \in [\mathsf{T}_{u,v} - 1]\} \cup \{|z\rangle|\mathsf{T}_{u,v}\rangle : z \in \mathcal{Z}^1_{u,v}\}. \end{aligned} \tag{8}$$

We now define sets of states $\Psi^{\mathcal{A}}$ and $\Psi^{\mathcal{B}}$ in H'.

Star States: We slightly modify the star states to get states in H'. To all star states $|\psi_\star(u)\rangle \in \Psi_\star(u)$ (see Definition 3.9), we append a register $|0\rangle$, but for $u \in V_0 \cup M$, we make a further modification. Conceptually, we modify the graph G, by adding a new vertex, v_0, to get a new graph G' (see Figure 4). The new vertex is connected to every $u \in V_0$ by an edge of weight $\mathsf{w}_0\sigma(u)$, for some w_0 to be assigned later; and it is connected to every $u \in M$ by an edge of weight w_M, for some w_M to be assigned later. So for $u \in V_0 \cup M$, we modify the star state $\Psi_\star(u) = \{|\psi^G_\star(u)\rangle\}$ by adding the extra register $|0\rangle$, but we also account for the additional edge to v_0. We assume that for all $u \in V_0 \cup M$, the edge to v_0 is labelled by $0 \notin L(u)$. With this intuition, we define, for

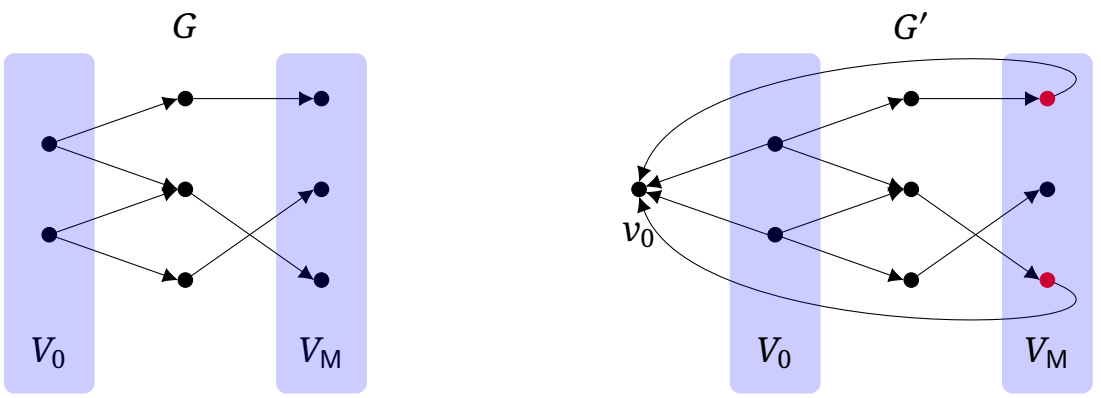


Figure 4. Example of a graph G with $V_0, V_M \subseteq V(G)$ and the induced graph G' that is obtained from G by adding a new vertex v_0. This new vertex is connected to all vertices in V_0 and only connected to those vertices in V_M which are marked (visualised by the red vertices).

$u \in V_0$:

$$|\psi_\star^{G'}(u)\rangle|0\rangle := \underbrace{\sum_{i\in L^+(u)} \sqrt{w_{u,i}}|u,i\rangle|0\rangle - \sum_{i\in L^-(u)} \sqrt{w_{u,i}}|u,i\rangle|0\rangle}_{=|\psi_\star^{G}(u)\rangle|0\rangle} + \sqrt{w_0\sigma(u)}|u,0\rangle|0\rangle \tag{9}$$

and for $u \in M$:

$$|\psi_\star^{G'}(u)\rangle|0\rangle := \underbrace{\sum_{i\in L^+(u)} \sqrt{w_{u,i}}|u,i\rangle|0\rangle - \sum_{i\in L^-(u)} \sqrt{w_{u,i}}|u,i\rangle|0\rangle}_{=|\psi_\star^{G}(u)\rangle|0\rangle} + \sqrt{w_M}|u,0\rangle|0\rangle. \tag{10}$$

For $u \in V(G) \setminus (V_0 \cup M)$, the neighbours and weights in G' are the same as G, so we let $|\psi_\star^{G'}(u)\rangle = |\psi_\star^{G}(u)\rangle$, which we know is in $\Psi_\star(u)$ (possibly among other states). We let:

$$\Psi'_\star := \bigcup_{u\in V(G)\setminus(V_0\cup M)} \underbrace{\{|\psi_\star(u)\rangle|0\rangle : |\psi_\star(u)\rangle \in \Psi_\star(u)\}}_{=:\Psi'_\star(u)} \cup \bigcup_{u\in V_0\cup M} \underbrace{\left\{|\psi_\star^{G'}(u)\rangle|0\rangle\right\}}_{=:\Psi'_\star(u)}. \tag{11}$$

Algorithm States: For each $u \in V(G)$ and $i \in L^+(u)$, define a state

$$|\psi_{\rightarrow}^{u,i}\rangle := |u,i\rangle|0\rangle - U_0|u,i\rangle|1\rangle. \tag{12}$$

These represent a transition from an outgoing edge to the first step of the algorithm implementing that edge transition. For each $(u,v) \in \overrightarrow{E}(G)$, and $t \in [\mathsf{T}_{u,v} - 1]$, define states:

$$\Psi_t^{u,v} := \left\{|\psi_t^z\rangle := |z\rangle|t\rangle - U_t|z\rangle|t+1\rangle : z \in \mathcal{Z}_{u,v}^0\right\}. \tag{13}$$

These represent steps of the edge transition subroutine. For each $v \in V(G)$ and $j \in L^-(v)$, with $u = f_v(j)$, define a state:

$$|\psi_{\leftarrow}^{v,j}\rangle := |v,j\rangle|\mathsf{T}_{u,v}\rangle - |v,j\rangle|0\rangle. \tag{14}$$

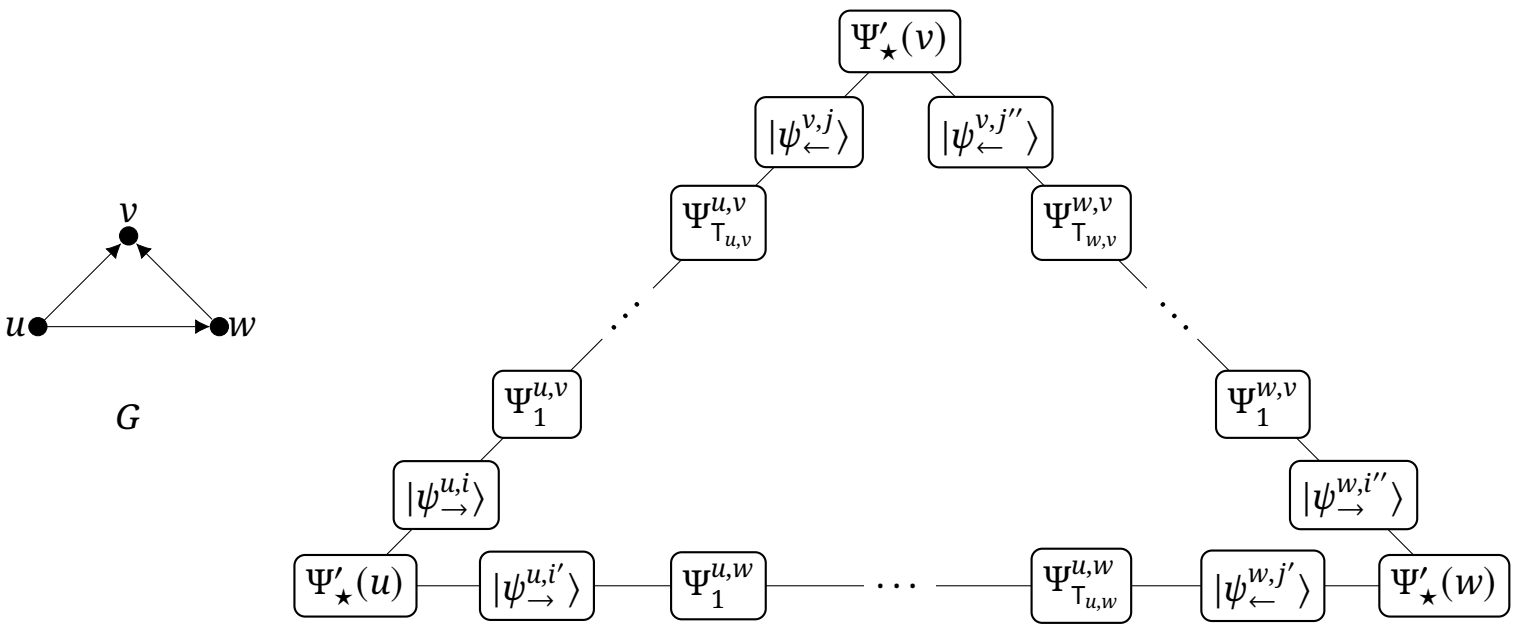


Figure 5. A graph showing the overlap of various sets of states, for an example graph G. With the exception of the spaces $\Psi'_\star(u)$ (which we will replace with orthonormal bases in Section 3.2.5), each node represents an orthonormal set. There is an edge between two nodes if and only if the sets contain overlapping vectors.

These represent exiting the algorithm to an edge going into vertex v. Letting $\Psi'_\star$ be as in (11), define

$$\Psi^{\mathcal{A}} = \Psi'_\star \cup \bigcup_{(u,v)\in\overrightarrow{E}(G)} \bigcup_{\substack{t=1:\\ t \text{ odd}}}^{\mathsf{T}_{u,v}-1} \Psi^{u,v}_t$$
$$\Psi^{\mathcal{B}} = \{|\psi^{u,i}_{\rightarrow}\rangle : u \in V(G), i \in L^+(u)\} \cup \{|\psi^{v,j}_{\leftarrow}\rangle : v \in V(G), j \in L^-(v)\} \cup \bigcup_{(u,v)\in\overrightarrow{E}(G)} \bigcup_{\substack{t=1:\\ t \text{ even}}}^{\mathsf{T}_{u,v}-1} \Psi^{u,v}_t. \tag{15}$$

The reason we have divided the states in this way between $\Psi^{\mathcal{A}}$ and $\Psi^{\mathcal{B}}$ is so that if we replace each $\Psi_\star(u)$ with an orthonormal basis, all states in $\Psi^{\mathcal{A}}$ (or $\Psi^{\mathcal{B}}$) are pairwise orthogonal. We leave it up to the reader to verify that this is the case (it is implicitly proven in Section 3.2.5), but we note that this fact relies on the assumption that $\mathsf{T}_{u,v}$ is always even. This ensures that for even t, $\langle t+1|\mathsf{T}_{u,v}\rangle = 0$, so $\langle\psi^z_t|\psi^{v,j}_{\leftarrow}\rangle = 0$. Figure 5 shows a graph of the overlap between various sets of states, and we can observe that the sets in $\Psi^{\mathcal{A}}$ and the sets in $\Psi^{\mathcal{B}}$ form a bipartition of this overlap graph into independent sets.

Finally, we define the initial state of the algorithm:

$$|\psi_0\rangle := |\sigma\rangle|0\rangle|0\rangle = \sum_{u\in V_0} \sqrt{\sigma(u)}|u,0\rangle|0\rangle. \tag{16}$$

3.2.5 Implementing the Unitary

Let $\mathcal{A} = \text{span}\{\Psi^{\mathcal{A}}\}$ and $\mathcal{B} = \text{span}\{\Psi^{\mathcal{B}}\}$ (see (15)), and let $\Pi_{\mathcal{A}}$ and $\Pi_{\mathcal{B}}$ be the orthogonal projectors onto $\mathcal{A}$ and $\mathcal{B}$. In this section we will prove:

LEMMA 3.12. *The unitary $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$ on H' can be implemented in complexity $O(\mathsf{A}_\star + \text{polylog}(\mathsf{T}_{\max}))$.*

This essentially follows[14] from the fact that we can efficiently generate orthonormal bases for each of $\Psi^{\mathcal{A}}$ and $\Psi^{\mathcal{B}}$, since:

— By the **Star State Generation Subroutine** condition of Theorem 3.10, we can generate an orthonormal basis for $\bigcup_{u\in V(G)} \Psi_\star(u)$. Since we can also efficiently check if a vertex is in V_0 or M, we can generate orthonormal bases for $\Psi'_\star = \bigcup_{u\in V(G)} \Psi'_\star(u)$ (see Claim 3.13).

— Generating the states $|\psi^z_t\rangle = |z\rangle|t\rangle - U_t|z\rangle|t+1\rangle$ for odd t can be done using $\sum_t |t\rangle\langle t| \otimes U_t$ (see Claim 3.14). The same is true for even t (Claim 3.15), also including the states $|\psi^{u,i}_{\rightarrow}\rangle = |u,i\rangle|0\rangle - U_0|u,i\rangle|1\rangle$.

— Generating the states $|\psi^{v,j}_{\leftarrow}\rangle = |v,j\rangle(|\mathsf{T}_{u,v}\rangle - |0\rangle)$ can be done efficiently because we can compute $\mathsf{T}_{u,v}$ from (v,j) (see Claim 3.16).

There is nothing conceptually new in this proof, and the reader may skip ahead to Section 3.2.6 with no loss of understanding.

CLAIM 3.13. *Let $R_\star = 2\Pi_\star - I$, where $\Pi_\star$ is the orthogonal projector onto $\text{span}\{\Psi'_\star\}$. Then $R_\star$ can be implemented in complexity $O(\mathsf{A}_\star + \log \mathsf{T}_{\max})$.*

PROOF. By the **Star State Generation Subroutine** condition of Theorem 3.10, we can generate $\Psi_\star$ in cost $\mathsf{A}_\star$, which means (see Definition 3.9) that for each $u \in V(G)$, there is an orthonormal basis $\overline{\Psi}(u) = \{|\overline{\psi}_{u,1}\rangle, \ldots, |\overline{\psi}_{u,a_u}\rangle\}$ for $\text{span}\{\Psi_\star(u)\}$, and a unitary $U_\star$ with complexity $\mathsf{A}_\star$, such that for all $u \in V(G)$ and $k \in \{0,\ldots,a_u - 1\}$, $U_\star|u,k\rangle = |\overline{\psi}_{u,k}\rangle$. Then for all $u \in V(G) \setminus (V_0 \cup V_{\mathsf{M}})$, $\overline{\Psi}'(u) := \{|\overline{\psi}_{u,1}\rangle|0\rangle, \ldots, |\overline{\psi}_{u,a_u}\rangle|0\rangle\}$ is an orthonormal basis for $\Psi'(u)$ (see (11)). For $u \in V_0 \cup V_{\mathsf{M}}$, we have $\Psi'_\star(u) = \{|\psi^{G'}_\star(u)\rangle|0\rangle\}$, so $\overline{\Psi}'(u)$ is just the single normalization of this state.

We will first define a unitary $U'_\star$ that acts, for $u \in V(G)$, $k \in \{0,\ldots,a_u - 1\}$, as $U'_\star|u,k\rangle|0\rangle = |\overline{\psi}'_{u,k}\rangle$. We define $U'_\star$ by its implementation. To begin we will append a qutrit register, $|0\rangle_A$ (this will be uncomputed, so that the action described is indeed unitary), and set it to $|1\rangle_A$ if $u \in V_0$, and $|2\rangle_A$ if $u \in M$. We are assuming we can check if $u \in V_0$ or $u \in M$ in at most $\mathsf{A}_\star$ complexity. We proceed in three cases, controlled on the value of the ancilla.
First, controlled on $|0\rangle_A$, we apply $U_\star$, in cost $\mathsf{A}_\star$, to get:

$$|u,k\rangle|0\rangle|0\rangle_A \mapsto |\overline{\psi}_{u,k}\rangle|0\rangle|0\rangle_A = |\overline{\psi}'_{u,k}\rangle|0\rangle_A.$$

14 For a simple example of how reflecting around a set of states reduces to generating the set, see Claim 3.16.

Next, controlled on $|1\rangle_A$, we implement, on the last register, a single qubit rotation that acts as

$$|0\rangle \mapsto \sqrt{\frac{\mathsf{w}_u}{\mathsf{w}_u + \mathsf{w}_0\sigma(u)}}|1\rangle + \sqrt{\frac{\mathsf{w}_0\sigma(u)}{\mathsf{w}_u + \mathsf{w}_0\sigma(u)}}|0\rangle.$$

This requires that we can query w_u and $\sigma(u)$ (w_0 is a parameter of the algorithm). Controlled on $|1\rangle$ in the last register (and also still $|1\rangle_A$ in the third), we apply $U_\star$ to get (when $u \in V_0$ we only care about the behaviour for $k = 0$):

$$|u,0\rangle|0\rangle|1\rangle_A \mapsto \left(\sqrt{\frac{\mathsf{w}_u}{\mathsf{w}_u + \mathsf{w}_0\sigma(u)}}\frac{|\psi_\star^G(u)\rangle}{\sqrt{\mathsf{w}_u}}|1\rangle + \sqrt{\frac{\mathsf{w}_0\sigma(u)}{\mathsf{w}_u + \mathsf{w}_0\sigma(u)}}|u,0\rangle|0\rangle\right)|1\rangle_A.$$

Above we have used the fact that when $u \in V_0$,

$$|\overline{\psi}_{u,0}\rangle = \frac{|\psi_\star^G(u)\rangle}{\||\psi_\star^G(u)\rangle\|} = \frac{|\psi_\star^G(u)\rangle}{\sqrt{\mathsf{w}_u}}.$$

To complete the map for the case $u \in V_0$, note that $|\psi_\star^G(u)\rangle$ is supported on $|u,i\rangle$ for $i \neq 0$, so we can uncompute the second register to get:

$$\frac{|\psi_\star^G(u)\rangle|0\rangle + \sqrt{\mathsf{w}_0\sigma(u)}|u,0\rangle|0\rangle}{\sqrt{\mathsf{w}_u + \mathsf{w}_0\sigma(u)}}|1\rangle_A = \frac{|\psi_\star^{G'}(u)\rangle|0\rangle}{\||\psi_\star^{G'}(u)\rangle\|}|1\rangle_A.$$

Finally, controlled on $|2\rangle_A$, we do something very similar, but now the single qubit map we use is

$$|0\rangle \mapsto \sqrt{\frac{\mathsf{w}_u}{\mathsf{w}_u + \mathsf{w}_\mathsf{M}}}|1\rangle + \sqrt{\frac{\mathsf{w}_\mathsf{M}}{\mathsf{w}_u + \mathsf{w}_\mathsf{M}}}|0\rangle.$$

This is possible given query access to w_u (w_M is a parameter of the algorithm).

Since all states still have $|u\rangle$ in the first register, controlled on u, we can uncompute the ancilla. Thus, we can implement $U'_\star$ in complexity $O(\mathsf{A}_\star)$, and $U'_\star$ maps the subspace

$$\mathcal{L}_\star := \mathrm{span}\{|u,k\rangle|0\rangle : u \in V(G), k \in \{0,\dots,a_u - 1\}\}$$

of H' to the $\mathrm{span}\{\Psi'_\star\}$. Thus $(2\Pi_\star - I) = U'_\star(2\Pi_{\mathcal{L}_\star} - I){U'_\star}^\dagger$, so we complete the proof by describing how to implement $2\Pi_{\mathcal{L}_\star} - I$. Initialise two ancillary flag qubits, $|0\rangle_{F_1}|0\rangle_{F_2}$. For a computational basis state $|z\rangle|t\rangle$, if $t \neq 0$, flip F_1 to $|1\rangle_{F_1}$. This check costs $\log \mathsf{T}_{\max}$. If $t = 0$, we can assume that z has the form (u,k), and interpret k as an integer. If $k < a_u$, which can be checked in $O(\log d_{\max}) = O(\mathsf{A}_\star)$, flip F_2 to $|1\rangle_{F_2}$. Reflect conditioned on either of the flags being set to 1, and then uncompute both flags. ∎

CLAIM 3.14. *Let $R_{\mathrm{odd}} = 2\Pi_{\mathrm{odd}} - I$, where Π_{odd} is the orthogonal projector onto $\mathrm{span}\{\Psi_t^{u,v} : (u,v) \in \overrightarrow{E}(G), t \in \{1,\dots,\mathsf{T}_{u,v}-1\}$ odd$\}$. Then R_{odd} can be implemented in complexity $\mathrm{polylog}(\mathsf{T}_{\max})$.*

PROOF. We first describe the implementation of a unitary U_{odd} such that:

$$\forall (u,v) \in \overrightarrow{E}(G), z \in \mathcal{Z}^0_{u,v}, t \in [\mathsf{T}_{u,v} - 1] \text{ odd}, \; U_{\mathrm{odd}}|z\rangle|t\rangle = \frac{1}{\sqrt{2}}|z\rangle|t\rangle - \frac{1}{\sqrt{2}}U_t|z\rangle|t+1\rangle = \frac{1}{\sqrt{2}}|\psi_z^t\rangle.$$

We begin by decrementing the $|t\rangle$ register, which costs $\log \mathsf{T}_{\max}$. Next we apply an X gate, followed by a Hadamard gate, to the last qubit of $|t-1\rangle$. If t is odd, $t-1$ is even and the last qubit is $|0\rangle \overset{\mathsf{HX}}{\mapsto} (|0\rangle - |1\rangle)/\sqrt{2}$, so we get:

$$|z\rangle|t\rangle \mapsto |z\rangle|t-1\rangle \mapsto (|z\rangle|t-1\rangle - |z\rangle|t\rangle)/\sqrt{2}.$$

Then controlled on the last qubit of $|t\rangle$ being $|1\rangle$ (i.e. on odd parity of t) apply $\sum_{t=0}^{\mathsf{T}_{\max}-1} |t\rangle\langle t| \otimes U_t$, which can be done in cost $\mathrm{polylog}(\mathsf{T}_{\max})$ by assumption, to get: $(|z\rangle|t-1\rangle - U_t|z\rangle|t\rangle)/\sqrt{2}$. Complete the operation by incrementing the $|t\rangle$ register. Thus, U_{odd} maps the subspace:

$$\mathcal{L}_{\mathsf{odd}} := \bigoplus_{(u,v)\in\overrightarrow{E}(G)} \mathrm{span}\{|z\rangle|t\rangle : z \in \mathcal{Z}^0_{u,v}, t \in \{1,\dots,\mathsf{T}_{u,v}-1\},\ \mathrm{odd}\}$$

of H' to the support of Π_{odd}, and so $R_{\mathsf{odd}} = U_{\mathsf{odd}}(2\Pi_{\mathcal{L}_{\mathsf{odd}}} - I)U^\dagger_{\mathsf{odd}}$. We complete the proof by describing how to implement $2\Pi_{\mathcal{L}_{\mathsf{odd}}} - I$. For $|z\rangle|t\rangle$, we can check if t is odd in $O(1)$, and if not, set an ancillary flag F_1. Next, we will ensure that $z \in \mathcal{Z}^0_{u,v}$ for some $(u,v) \in \overrightarrow{E}(G)$, which also ensures that $t < \mathsf{T}_{u,v}$, by the structure of H', and if not, set a flag F_2. Reflect if either F_1 or F_2 is set, and then uncompute both of them. ■

CLAIM 3.15. *Let $R_{\mathsf{even}} = 2\Pi_{\mathsf{even}} - I$, where Π_{even} is the orthogonal projector onto the span of*

$$\bigcup_{(u,v)\in\overrightarrow{E}(G)} \Psi^{u,v}_t \cup \{|\psi^{u,i}_{\rightarrow}\rangle : u \in V(G), i \in L^+(u)\}.$$
$$t\in\{1,\dots,\mathsf{T}_{u,v}-1\}: t \text{ even}$$

Then R_{even} can be implemented in complexity $\mathrm{polylog}(\mathsf{T}_{\max})$.

PROOF. We describe the implementation of a unitary U_{even} such that for all $(u,v) \in \overrightarrow{E}(G)$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$:

$$U_{\mathsf{even}}|u,i\rangle|0\rangle = \frac{1}{\sqrt{2}}(|u,i\rangle|0\rangle - U_0|u,i\rangle|1\rangle) = \frac{1}{\sqrt{2}}|\psi^{u,i}_{\rightarrow}\rangle$$
$$\forall z \in \mathcal{Z}^0_{u,v}, t \in [\mathsf{T}_{u,v}-1] \text{ even},\ U_{\mathsf{even}}|z\rangle|t\rangle = \frac{1}{\sqrt{2}}(|z\rangle|t\rangle - U_t|z\rangle|t+1\rangle) = \frac{1}{\sqrt{2}}|\psi^z_t\rangle.$$

We can implement such a mapping nearly identically to the proof of Claim 3.14, except the decrementing of t happens after the Hadamard is applied. Thus, U_{even} maps the subspace:

$$\mathcal{L}_{\mathsf{even}} := \bigoplus_{u\in V(G), i\in L^+(u)} \mathrm{span}\{|u,i\rangle|0\rangle\} \oplus \bigoplus_{(u,v)\in\overrightarrow{E}(G)} \mathrm{span}\{|z\rangle|t\rangle : z \in \mathcal{Z}^0_{u,v}, t \in \{1,\dots,\mathsf{T}_{u,v}-1\},\ \mathrm{even}\}$$

of H' to the support of Π_{even}, and so $R_{\mathsf{even}} = U_{\mathsf{even}}(2\Pi_{\mathcal{L}_{\mathsf{even}}} - I)U^\dagger_{\mathsf{even}}$. We complete the proof by describing how to implement $2\Pi_{\mathcal{L}_{\mathsf{even}}} - I$. For $|z\rangle|t\rangle$, we can check if t is even in $O(1)$ steps, and if not, set an ancillary flag F_1. Next, we check if $z \in \mathcal{Z}^0_{u,v}$ by checking the subroutine's internal flag, which also ensures that $t < \mathsf{T}_{u,v}$, and if not, set an ancillary flag F_2. Note that if $t = 0$, z has the form (u,i) for some $i \in L(u)$, by the structure of H', and by the discussion in Section 3.2.3,

we can check if $i \in L^+(u)$ in $O(1)$ time, and otherwise, set a flag F_3. Reflect if either F_1, F_2 or F_3 is set, and then uncompute all three flags. ■

CLAIM 3.16. *Let $R_\leftarrow = 2\Pi_\leftarrow - I$, where $\Pi_\leftarrow$ is the orthogonal projector onto the span of $\{|\psi_\leftarrow^{v,j}\rangle : v \in V(G), j \in L^-(v)\}$. Then $R_\leftarrow$ can be implemented in complexity* $\mathsf{polylog}(\mathsf{T}_{\max})$.

PROOF. We describe the implementation of a unitary $U_\leftarrow$ that acts, for all $v \in V(G)$ and $j \in L^-(v)$, with $u = f_v(j)$, as:

$$U_\leftarrow|v,j\rangle|0\rangle = \frac{1}{\sqrt{2}}\left(|v,j\rangle|0\rangle - |v,j\rangle|\mathsf{T}_{u,v}\rangle\right) = -\frac{1}{\sqrt{2}}|\psi_\leftarrow^{v,j}\rangle.$$

First, append an ancilla $|-\rangle_A$. Controlled on this ancilla, compute $\mathsf{T}_{u,v}$ from (v,j), which we can do in $\mathsf{polylog}(\mathsf{T}_{\max})$ basic operations, by the assumptions of Definition 2.5, to get:

$$|v,j\rangle|0\rangle|-\rangle_A \mapsto |v,j\rangle\left(|0\rangle|0\rangle_A - |\mathsf{T}_{u,v}\rangle|1\rangle_A\right)/\sqrt{2}.$$

Uncompute the ancilla by adding 1 into register A conditioned on the time register having a value greater than 0. Thus, $U_\leftarrow$ maps the subspace

$$\mathcal{L}_\leftarrow := \mathrm{span}\{|v,j\rangle|0\rangle : v \in V(G), j \in L^-(v)\}$$

of H' to the support of $\Pi_\leftarrow$, and so $R_\leftarrow = U_\leftarrow(2\Pi_{\mathcal{L}_\leftarrow} - I)U_\leftarrow^\dagger$. We complete the proof by describing how to implement $2\Pi_{\mathcal{L}_\leftarrow} - I$. Append two ancillary qubits, $|0\rangle_{F_1}$ and $|0\rangle_{F_2}$. For a computational basis state $|z\rangle|t\rangle$, if $t \neq 0$, which can be checked in $O(\log \mathsf{T}_{\max})$ time, flip F_1 to get $|1\rangle_{F_1}$. By the discussion in Section 3.2.3, we can check if z has the form (v,j) for some $v \in V(G)$ and $j \in L^-(v)$ in $O(1)$ time, and if not, flip F_2 to get $|1\rangle_{F_2}$. Reflect the state if either flag is set to 1, and then uncompute both flags. ■

PROOF OF LEMMA 3.12. We can see that $\Pi_\star\Pi_{\mathsf{odd}} = 0$, since $\Pi_\star$ is supported on states with 0 in the last register, and Π_{odd} is the span of states with an odd $t \in \{1,\ldots,\mathsf{T}_{u,v}-1\}$ in the first term, and an even $t \in \{2,\ldots,\mathsf{T}_{u,v}\}$ in the second term. Thus,

$$(2\Pi_{\mathsf{even}} - I)(2\Pi_\star - I) = -(2(\Pi_{\mathsf{even}} + \Pi_\star) - I) = -(2\Pi_{\mathcal{A}} - I),$$

where the last equality is because the support of $\Pi_{\mathcal{A}}$ is the direct sum of the supports of $\Pi_\star$ and Π_{even}, by their definitions. By a similar argument, $(2\Pi_{\mathsf{odd}} - I)(2\Pi_\leftarrow - I) = (2\Pi_{\mathcal{B}} - I)$, and Thus, the result follows from Claim 3.13, Claim 3.15, Claim 3.14 and Claim 3.16. ■

3.2.6 Positive Analysis

Suppose there is a flow θ on G satisfying conditions **P1-P5** of Theorem 3.10. We will use it to make a positive witness as follows. For each $(u,v) \in \overrightarrow{E}(G)$, with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$,

define:

$$\begin{aligned} |w^0_{u,v}\rangle &:= |u,i\rangle \\ \forall t \in \{1,\dots,\mathsf{T}_{u,v}\},\ |w^t_{u,v}\rangle &:= U_{t-1}|w^{t-1}_{u,v}\rangle \\ |w_{u,v}\rangle &:= \sum_{t=0}^{\mathsf{T}_{u,v}} |w^t_{u,v}\rangle|t\rangle + |v,j\rangle|0\rangle. \end{aligned} \tag{17}$$

Then $|w_{u,v}\rangle$ is a kind of *history state* [29] for the algorithm on input (u,i). We first show it is almost orthogonal to all algorithm states, defined in (12), (13) and (14).

CLAIM 3.17. *For all $(u,v) \in \overrightarrow{E}(G)$, letting $j = f_v^{-1}(u)$:*

1. *For all $u' \in V(G)$ and $i' \in L^+(u)$, $\langle\psi_{\rightarrow}^{u',i'}|w_{u,v}\rangle = 0$.*
2. *For all $(u',v') \in \overrightarrow{E}(G)$, $z \in \mathcal{Z}^0_{u',v'}$ and $t \in \{1,\dots,\mathsf{T}_{u,v}-1\}$, $\langle\psi_t^z|w_{u,v}\rangle = 0$.*
3. *For all $v' \in V(G)$ and $j' \in L^-(u)$, $|\langle\psi_{\leftarrow}^{v',j'}|w_{u,v}\rangle|^2 \le \delta_{(v,j),(v',j')}\epsilon_{u,v}$, where δ_{xy} denotes the Kronecker delta function.*

PROOF. **Item 1:** Recalling that $|\psi_{\rightarrow}^{u',i'}\rangle = |u',i'\rangle|0\rangle - U_0|u',i'\rangle|1\rangle$, we have

$$\langle\psi_{\rightarrow}^{u',i'}|w_{u,v}\rangle = \langle u',i'|w^0_{u,v}\rangle - \langle u',i'|U_0^\dagger|w^1_{u,v}\rangle = \langle u',i'|u,i\rangle - \langle u',i'|U_0^\dagger U_0|u,i\rangle = 0.$$

Item 2: Recall that $|\psi_t^z\rangle = |z\rangle|t\rangle - U_t|z\rangle|t+1\rangle$. This is always orthogonal to the last term of $|w_{u,v}\rangle$, since $t > 0$. We also note that if $z \in \mathcal{Z}_{u',v'}$ for $(u',v') \neq (u,v)$, we have $\langle\psi_t^z|w_{u,v}\rangle = 0$, since $|w_{u,v}\rangle$ is only supported on $z \in \mathcal{Z}_{u,v}$. Thus, we can assume $(u,v) = (u',v')$, and so $t < \mathsf{T}_{u,v}$. Thus:

$$\langle\psi_t^z|w_{u,v}\rangle = \langle z|w^t_{u,v}\rangle - \langle z|U_t^\dagger|w^{t+1}_{u,v}\rangle = \langle z|w^t_{u,v}\rangle - \langle z|U_t^\dagger U_t|w^t_{u,v}\rangle = 0.$$

Item 3: Recall that $|\psi_{\leftarrow}^{v',j'}\rangle = |v',j'\rangle|\mathsf{T}_{u',v'}\rangle - |v',j'\rangle|0\rangle$. Again, if $(v',j') \neq (v,j)$, $(v',j') \notin \mathcal{Z}_{u,v}$, so $\langle\psi_{\leftarrow}^{v',j'}|w_{u,v}\rangle = 0$. So supposing $(v',j') = (v,j)$, we have:

$$\langle\psi_{\leftarrow}^{v',j'}|w_{u,v}\rangle = \langle v,j|w^{\mathsf{T}_{u,v}}_{u,v}\rangle - \langle v,j|w^0_{u,v}\rangle - \langle v,j|v,j\rangle = \langle v,j|w^{\mathsf{T}_{u,v}}_{u,v}\rangle - 1,$$

since $|w^0_{u,v}\rangle = |u,i\rangle$, so the middle term is 0. Then, using

$$\left|1 - \langle v,j|w^{\mathsf{T}_{u,v}}_{u,v}\rangle\right|^2 \le \left\||v,j\rangle - |w^{\mathsf{T}_{u,v}}_{u,v}\rangle\right\|^2 = \epsilon_{u,v},$$

by (7), we have $|\langle\psi_{\leftarrow}^{v',j'}|w_{u,v}\rangle|^2 \le \epsilon_{u,v}$. ∎

Next let θ be a flow satisfying conditions **P1**-**P5** of Theorem 3.10, which can only exist if $M \neq \emptyset$. Then we define:

$$|w\rangle = \sum_{(u,v)\in\overrightarrow{E}(G)} \frac{\theta(u,v)}{\sqrt{\mathsf{w}_{u,v}}}|w_{u,v}\rangle - \sum_{u\in V_0}\frac{\theta(u)}{\sqrt{\mathsf{w}_0\sigma(u)}}|u,0\rangle|0\rangle - \sum_{u\in M}\frac{\theta(u)}{\sqrt{\mathsf{w}_M}}|u,0\rangle|0\rangle, \tag{18}$$

which we show is a positive witness, in the sense of Definition 3.5.

LEMMA 3.18. *Let $\mathsf{w}_M = |V(G)|$, $\mathsf{w}_0 = 1/\mathcal{R}^\mathsf{T}$, and $c_+ = 7$. Then $\dfrac{\||w\rangle\|^2}{|\langle w|\sigma\rangle|^2} \le 7 = c_+$.*

PROOF. To analyse the positive witness, we compute (referring to (16)):

$$\langle\psi_0|w\rangle = \sum_{u\in V_0}\sqrt{\sigma(u)}\langle u,0,0|w\rangle = -\sum_{u\in V_0}\sqrt{\sigma(u)}\frac{\theta(u)}{\sqrt{\mathsf{w}_0\sigma(u)}} = -\frac{1}{\sqrt{\mathsf{w}_0}} = -\sqrt{\mathcal{R}^{\mathsf{T}}}, \tag{19}$$

by condition **P3** of Theorem 3.10. Since this is non-zero, $|w\rangle$ is a positive witness, though it may have some error. To continue, we compute:

$$\||w\rangle\|^2 = \sum_{(u,v)\in\overrightarrow{E}(G)}\frac{\theta(u,v)^2}{\mathsf{w}_{u,v}}\left\||w_{u,v}\rangle\right\|^2 + \sum_{u\in V_0}\frac{\theta(u)^2}{\mathsf{w}_0\sigma(u)} + \sum_{u\in M}\frac{\theta(u)^2}{\mathsf{w}_{\mathsf{M}}}. \tag{20}$$

To upper bound the first term of (20), we have $\left\||w_{u,v}\rangle\right\|^2 = \mathsf{T}_{u,v}+2$, so we have

$$\sum_{(u,v)\in\overrightarrow{E}(G)}\frac{\theta(u,v)^2}{\mathsf{w}_{u,v}}\left\||w_{u,v}\rangle\right\|^2 = \sum_{(u,v)\in\overrightarrow{E}(G)}\frac{\theta(u,v)^2}{\mathsf{w}_{u,v}}(\mathsf{T}_{u,v}+2) = \mathcal{E}^{\mathsf{T}}(\theta)+2\mathcal{E}(\theta) \le 2\mathcal{R}^{\mathsf{T}}, \tag{21}$$

by condition **P5** of Theorem 3.10, and using $2\mathcal{E}(\theta)\le\mathcal{E}^{\mathsf{T}}(\theta)$, since each $\mathsf{T}_{u,v}\ge 2$.
To upper bound the second term of (20), we have

$$\sum_{u\in V_0}\frac{\theta(u)^2}{\mathsf{w}_0\sigma(u)} \le \frac{1}{\mathsf{w}_0}2\sum_{u\in V_0}\frac{\sigma(u)^2+(\theta(u)-\sigma(u))^2}{\sigma(u)} = 2\mathcal{R}^{\mathsf{T}}\left(1+\sum_{u\in V_0}\frac{(\theta(u)-\sigma(u))^2}{\sigma(u)}\right) \le 4\mathcal{R}^{\mathsf{T}} \tag{22}$$

by condition **P4** of Theorem 3.10. Finally, we can upper bound the last term of (20) as:

$$\sum_{u\in M}\frac{\theta(u)^2}{\mathsf{w}_{\mathsf{M}}} = \frac{1}{|V(G)|}\sum_{u\in M}\left(\sum_{v\in\Gamma(u)}\theta(u,v)\right)^2 \le \frac{1}{|V(G)|}\sum_{u\in M}d_u\sum_{v\in\Gamma(u)}\theta(u,v)^2 \le \mathcal{E}(\theta)\le\mathcal{E}^{\mathsf{T}}(\theta)\le\mathcal{R}^{\mathsf{T}}. \tag{23}$$

Plug (21), (22) and (23) into (20) to get $\||w\rangle\|^2\le 7\mathcal{R}^{\mathsf{T}}$, which, with (19), completes the proof. ■

Next, we analyse the error of $|w\rangle$ as a positive witness, by upper bounding its overlap with the various states in $\Psi^{\mathcal{A}}\cup\Psi^{\mathcal{B}}$. First, we have the following corollary to Claim 3.17.

COROLLARY 3.19. *1. For all $u\in V(G)$, $i\in L^+(u)$, $\langle\psi^{u,i}_{\rightarrow}|w\rangle = 0$.*
2. For all $(u,v)\in\overrightarrow{E}(G)$, $z\in\mathcal{Z}^0_{u,v}$ and $t\in\{1,\dots,\mathsf{T}_{u,v}-1\}$, $\langle\psi^z_t|w\rangle = 0$.
3. For all $v\in V(G)$ and $j\in L^-(v)$, letting $u=f_v(j)$, we have: $|\langle\psi^{v,j}_{\leftarrow}|w\rangle| \le \frac{\theta(u,v)}{\sqrt{\mathsf{w}_{u,v}}}\sqrt{\epsilon_{u,v}}$.

Next we show that the states in $\Psi'_\star$ are orthogonal to $|w\rangle$.

CLAIM 3.20. *For all $u'\in V(G)$, and any $|\psi_\star(u')\rangle|0\rangle\in\Psi'_\star(u')$, $\langle\psi_\star(u'),0|w\rangle = 0$.*

PROOF. If $u'\notin V_0\cup M$, we have:

$$\begin{aligned}\langle\psi_\star(u'),0|w\rangle &= \sum_{(u,v)\in\overrightarrow{E}(G)}\frac{\theta(u,v)}{\sqrt{\mathsf{w}_{u,v}}}\langle\psi_\star(u'),0|(|u,f_u^{-1}(v)\rangle|0\rangle + |v,f_v^{-1}(u)\rangle|0\rangle)\\ &= \sum_{v\in\Gamma^+(u')}\frac{\theta(u',v)}{\sqrt{\mathsf{w}_{u',v}}}\langle\psi_\star(u')|u',f_{u'}^{-1}(v)\rangle + \sum_{u\in\Gamma^-(u')}\frac{\theta(u,u')}{\sqrt{\mathsf{w}_{u,u'}}}\langle\psi_\star(u')|u',f_{u'}^{-1}(u)\rangle = 0,\end{aligned}$$

by condition **P2** of Theorem 3.10, using $\theta(u,u') = -\theta(u',u)$, and the fact that $|\psi_\star(u')\rangle$ is supported on $|u',i\rangle$ such that $i \in L(u')$.
If $u' \in M$, the only state in $\Psi'_\star(u')$ is $|\psi_\star^{G'}(u')\rangle|0\rangle$ (see (10)), so we have:

$$\begin{aligned}\langle\psi_\star^{G'}(u'),0|w\rangle &= \sum_{(u,v)\in\overrightarrow{E}(G)} \frac{\theta(u,v)}{\sqrt{\mathsf{w}_{u,v}}}\langle\psi_\star^{G'}(u'),0|w_{u,v}\rangle - \sum_{u\in M}\frac{\theta(u)}{\sqrt{\mathsf{w}_{\mathsf{M}}}}\langle\psi_\star^{G'}(u'),0|u,0,0\rangle \\ &= \sum_{v\in\Gamma^+(u')}\frac{\theta(u',v)}{\sqrt{\mathsf{w}_{u',v}}}\sqrt{\mathsf{w}_{u',v}} - \sum_{u\in\Gamma^-(u')}\frac{\theta(u,u')}{\sqrt{\mathsf{w}_{u,u'}}}\sqrt{\mathsf{w}_{u,u'}} - \frac{\theta(u')}{\sqrt{\mathsf{w}_{\mathsf{M}}}}\sqrt{\mathsf{w}_{\mathsf{M}}} \\ &= \sum_{u\in\Gamma(u')}\theta(u,u') - \theta(u') = 0.\end{aligned}$$

Similarly, if $u' \in V_0$,

$$\langle\psi_\star^{G'}(u'),0|w\rangle = \sum_{v\in\Gamma^+(u')}\theta(u',v) - \sum_{u\in\Gamma^-(u')}\theta(u,u') - \frac{\theta(u')}{\sqrt{\mathsf{w}_0\sigma(u')}}\sqrt{\mathsf{w}_0\sigma(u')} = \sum_{u\in\Gamma(u')}\theta(u,u') - \theta(u') = 0.\blacksquare$$

We can combine these results in the following lemma:

LEMMA 3.21. *When $M \neq \emptyset$, $|w\rangle$ as defined in (18) is an $\epsilon/2$-positive witness (see Definition 3.5).*

PROOF. Note that $|w\rangle$ is only defined when $M \neq \emptyset$, as it is constructed with a flow from V_0 to M. For $|w\rangle$ to be a positive witness, we require that $\langle w|\psi_0\rangle \neq 0$, which follows from (19). All that remains is to show that $\|\Pi_{\mathcal{A}}|w\rangle\|^2$ and $\|\Pi_{\mathcal{B}}|w\rangle\|^2$ are both at most $\frac{\epsilon}{2}\||w\rangle\|^2$. By Claim 3.17 and Claim 3.20, we have

$$\|\Pi_{\mathcal{A}}|w\rangle\|^2 = 0 \text{ and } \|\Pi_{\mathcal{B}}|w\rangle\|^2 = \sum_{v\in V(G),j\in L^-(v)}\frac{|\langle\psi_{\leftarrow}^{v,j}|w\rangle|^2}{\left\||\psi_{\leftarrow}^{v,j}\rangle\right\|^2} \leq \sum_{v\in V(G),j\in L^-(v)}\frac{\theta(v,f_v(j))^2\epsilon_{f_v(j),v}/\mathsf{w}_{v,f_v(j)}}{2}.$$

Since $\theta(e) = 0$ for all $e \in \tilde{E}$ (condition **P1** of Theorem 3.10) and for all $e \in \overrightarrow{E}(G)\setminus\tilde{E}$, $\epsilon_{u,v} \leq \epsilon$ (condition **TS1** of Theorem 3.10), we can continue:

$$\|\Pi_{\mathcal{B}}|w\rangle\|^2 \leq \frac{1}{2}\sum_{(u,v)\in\overrightarrow{E}(G)\setminus\tilde{E}}\frac{\theta(u,v)^2\epsilon_{u,v}}{\mathsf{w}_{u,v}} \leq \frac{\epsilon}{2}\mathcal{E}(\theta) < \frac{\epsilon}{2}\mathcal{E}^{\mathsf{T}}(\theta). \tag{24}$$

We have used the fact that the energy of the flow in G (see Definition 2.2), $\mathcal{E}(\theta)$, is at most the energy of that flow extended to the graph G^{T} in which we replace the edges by paths of positive lengths determined by T (see Definition 2.4). For the final step of the proof, we know due to (21) that

$$\||w\rangle\|^2 \geq \sum_{(u,v)\in\overrightarrow{E}(G)}\frac{\theta(u,v)^2}{\mathsf{w}_{u,v}}\left\||w_{u,v}\rangle\right\|^2 \geq \mathcal{E}^{\mathsf{T}}(\theta),$$

and therefore, it follows from (24) that $\|\Pi_{\mathcal{B}}|w\rangle\|^2 \leq \frac{\epsilon}{2}\||w\rangle\|^2$, so $|w\rangle$ is an $\epsilon/2$-positive witness. $\blacksquare$

3.2.7 Negative Analysis

Let $\mathcal{A} = \text{span}\{\Psi^{\mathcal{A}}\}$ and $\mathcal{B} = \text{span}\{\Psi^{\mathcal{B}}\}$ (see (15)). In this section, we will define a negative witness, which is some $|w_{\mathcal{A}}\rangle, |w_{\mathcal{B}}\rangle \in H'$, such that $|\psi_0\rangle = |w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle$ and $|w_{\mathcal{A}}\rangle$ (resp. $|w_{\mathcal{B}}\rangle$) is almost in $\mathcal{A}$ (resp. $\mathcal{B}$) (see Definition 3.2). We first define, for all $(u,v) \in \overrightarrow{E}(G)$ with $i = f_u^{-1}(v)$,

$$\begin{aligned} |w_{u,v}^{\mathcal{A}}\rangle &= \sum_{t\in[\mathsf{T}_{u,v}-1]:t \text{ odd}} \sum_{z\in\mathcal{Z}_{u,v}^0} \langle z|w_{u,v}^t\rangle|\psi_t^z\rangle \in \mathcal{A} \\ |w_{u,v}^{\mathcal{B}}\rangle &= |\psi_{\rightarrow}^{u,i}\rangle + \sum_{t\in[\mathsf{T}_{u,v}-1]:t \text{ even}} \sum_{z\in\mathcal{Z}_{u,v}^0} \langle z|w_{u,v}^t\rangle|\psi_t^z\rangle \in \mathcal{B}, \end{aligned} \tag{25}$$

where $|w_{u,v}^t\rangle$ is defined in (17).

LEMMA 3.22. *For all $(u,v) \in \overrightarrow{E}(G)$ with $i = f_u^{-1}(v)$, $|w_{u,v}^{\mathcal{A}}\rangle + |w_{u,v}^{\mathcal{B}}\rangle = |u,i\rangle|0\rangle - |w_{u,v}^{\mathsf{T}_{u,v}}\rangle|\mathsf{T}_{u,v}\rangle$.*

PROOF. Below we use the fact that for $t < \mathsf{T}_{u,v}$, $|w_{u,v}^t\rangle \in \text{span}\{|z\rangle : z \in \mathcal{Z}_{u,v}^0\}$ (see Section 3.2.3), and that $|w_{u,v}^0\rangle = |u,i\rangle$ (see (17)).

$$\begin{aligned} |w_{u,v}^{\mathcal{A}}\rangle + |w_{u,v}^{\mathcal{B}}\rangle &= |\psi_{\rightarrow}^{u,i}\rangle + \sum_{t\in[\mathsf{T}_{u,v}-1]} \sum_{z\in\mathcal{Z}_{u,v}^0} \langle z|w_{u,v}^t\rangle|\psi_t^z\rangle \\ &= |\psi_{\rightarrow}^{u,i}\rangle + \sum_{t=1}^{\mathsf{T}_{u,v}-1} \sum_{z\in\mathcal{Z}_{u,v}^0} \langle z|w_{u,v}^t\rangle|z,t\rangle - \sum_{t=1}^{\mathsf{T}_{u,v}-1} \sum_{z\in\mathcal{Z}_{u,v}^0} \langle z|w_{u,v}^t\rangle U_t|z\rangle|t+1\rangle \quad \text{see (13)} \\ &= |\psi_{\rightarrow}^{u,i}\rangle + \sum_{t=1}^{\mathsf{T}_{u,v}-1} |w_{u,v}^t\rangle|t\rangle - \sum_{t=1}^{\mathsf{T}_{u,v}-1} U_t|w_{u,v}^t\rangle|t+1\rangle \\ &= |\psi_{\rightarrow}^{u,i}\rangle + \sum_{t=1}^{\mathsf{T}_{u,v}-1} |w_{u,v}^t\rangle|t\rangle - \sum_{t=1}^{\mathsf{T}_{u,v}-1} |w_{u,v}^{t+1}\rangle|t+1\rangle \quad \text{see (17)} \\ &= |u,i\rangle|0\rangle - U_0|u,i\rangle|1\rangle + |w_{u,v}^1\rangle|1\rangle - |w_{u,v}^{\mathsf{T}_{u,v}}\rangle|\mathsf{T}_{u,v}\rangle \quad \text{see (12)} \\ &= |u,i\rangle|0\rangle - |w_{u,v}^{\mathsf{T}_{u,v}}\rangle|\mathsf{T}_{u,v}\rangle, \end{aligned}$$

since $|w_{u,v}^1\rangle = U_0|u,i\rangle$ (see (17)). ∎

For $v \in V(G)$ and $j \in L^-(v)$, with $u = f_v(j)$, define

$$|\tilde{\psi}_{\leftarrow}^{v,j}\rangle := |w_{u,v}^{\mathsf{T}_{u,v}}\rangle|\mathsf{T}_{u,v}\rangle - |v,j\rangle|0\rangle. \tag{26}$$

We now define our negative witness:

$$\begin{aligned} |w_{\mathcal{A}}\rangle &= \frac{1}{\sqrt{\mathsf{W}_0}} \sum_{u\in V(G)} |\psi_\star^{G'}(u)\rangle|0\rangle - \frac{1}{\sqrt{\mathsf{W}_0}} \sum_{(u,v)\in\overrightarrow{E}(G)} \sqrt{\mathsf{W}_{u,v}}|w_{u,v}^{\mathcal{A}}\rangle \in \mathcal{A} \\ |w_{\mathcal{B}}\rangle &= -\frac{1}{\sqrt{\mathsf{W}_0}} \sum_{(u,v)\in\overrightarrow{E}(G)} \sqrt{\mathsf{W}_{u,v}} \left(|w_{u,v}^{\mathcal{B}}\rangle + |\tilde{\psi}_{\leftarrow}^{v,f_v^{-1}(u)}\rangle\right). \end{aligned}$$

We first show that this is indeed a negative witness in Lemma 3.23 and then analyse its error and complexity in Lemma 3.24.

LEMMA 3.23. *Let $|\psi_0\rangle$ be as in (16). Then if $M = \emptyset$, $|w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle = |\psi_0\rangle$.*

PROOF. We have:

$$\sqrt{\mathsf{w}_0}\,(|w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle) = \sum_{u\in V(G)} |\psi_\star^{G'}(u)\rangle|0\rangle - \sum_{(u,v)\in\overrightarrow{E}(G)} \sqrt{\mathsf{w}_{u,v}}(|w_{u,v}^{\mathcal{A}}\rangle + |w_{u,v}^{\mathcal{B}}\rangle + |\tilde{\psi}_{\leftarrow}^{v,f_v^{-1}(u)}\rangle). \tag{27}$$

Letting $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, we have:

$$\begin{aligned} |w_{u,v}^{\mathcal{A}}\rangle + |w_{u,v}^{\mathcal{B}}\rangle + |\tilde{\psi}_{\leftarrow}^{v,j}\rangle &= |u,i\rangle|0\rangle - |w_{u,v}^{\mathsf{T}_{u,v}}\rangle|\mathsf{T}_{u,v}\rangle + |\tilde{\psi}_{\leftarrow}^{v,j}\rangle \quad \text{by Lemma 3.22} \\ &= |u,i\rangle|0\rangle - |v,j\rangle|0\rangle \qquad\qquad\qquad\qquad \text{by (26).} \end{aligned} \tag{28}$$

Next we recall from (9) and (10) that for $u \in V(G) \setminus M = V(G)$ (since $M = \emptyset$), we have, letting $\delta_{u,V_0} = 1$ iff $u \in V_0$:

$$\begin{aligned} |\psi_\star^{G'}(u)\rangle|0\rangle &= \sum_{i\in L^+(u)} \sqrt{\mathsf{w}_{u,i}}|u,i\rangle|0\rangle - \sum_{j\in L^-(u)} \sqrt{\mathsf{w}_{u,j}}|u,j\rangle|0\rangle + \delta_{u,V_0}\sqrt{\mathsf{w}_0\sigma(u)}|u,0\rangle|0\rangle \\ \sum_{u\in V(G)} |\psi_\star^{G'}(u)\rangle|0\rangle &= \sum_{\substack{(u,v)\\ \in\overrightarrow{E}(G)}} \left(\sqrt{\mathsf{w}_{u,v}}|u,f_u^{-1}(v)\rangle - \sqrt{\mathsf{w}_{u,v}}|v,f_v^{-1}(u)\rangle\right)|0\rangle + \sum_{u\in V_0}\sqrt{\mathsf{w}_0\sigma(u)}|u,0\rangle|0\rangle. \end{aligned} \tag{29}$$

Plugging (28) and (29) back into (27), we get:

$$\sqrt{\mathsf{w}_0}\,(|w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle) = \sum_{u\in V_0}\sqrt{\mathsf{w}_0\sigma(u)}|u,0\rangle|0\rangle = \sqrt{\mathsf{w}_0}|\psi_0\rangle. \qquad \blacksquare$$

LEMMA 3.24. *Let $\mathsf{w}_0 = 1/\mathcal{R}^{\mathsf{T}}$, and $\delta' = \epsilon\mathcal{R}^{\mathsf{T}}\mathcal{W} + 4\mathcal{R}^{\mathsf{T}}\widetilde{\mathcal{W}}$. Then $|w_{\mathcal{A}}\rangle, |w_{\mathcal{B}}\rangle$ is a δ'-negative witness (see Definition 3.2), and*

$$\||w_{\mathcal{A}}\rangle\|^2 \le 2\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}} + 1.$$

PROOF. By construction, $|w_{\mathcal{A}}\rangle \in \mathcal{A}$, and the only part of $|w_{\mathcal{B}}\rangle$ that is not made up of states in $\Psi^{\mathcal{B}}$ (which are in $\mathcal{B} = \mathrm{span}\{\Psi^{\mathcal{B}}\}$) are the $|\tilde{\psi}_{\leftarrow}^{v,j}\rangle$ parts. Since $(I - \Pi_{\mathcal{B}})|\psi_{\leftarrow}^{v,j}\rangle = 0$ for all (v,j), we have:

$$\begin{aligned} (I-\Pi_{\mathcal{B}})|w_{\mathcal{B}}\rangle &= -\frac{1}{\sqrt{\mathsf{w}_0}} \sum_{v\in V(G), j\in L^-(v)} \sqrt{\mathsf{w}_{v,j}}(I-\Pi_{\mathcal{B}})|\tilde{\psi}_{\leftarrow}^{v,j}\rangle \\ &= -\frac{1}{\sqrt{\mathsf{w}_0}} \sum_{v\in V(G), j\in L^-(v)} \sqrt{\mathsf{w}_{v,j}}(I-\Pi_{\mathcal{B}})(|\tilde{\psi}_{\leftarrow}^{v,j}\rangle - |\psi_{\leftarrow}^{vj}\rangle) \\ &= -\frac{1}{\sqrt{\mathsf{w}_0}} \sum_{(u,v)\in\overrightarrow{E}(G)} \sqrt{\mathsf{w}_{u,v}}(I-\Pi_{\mathcal{B}})(|w_{u,v}^{\mathsf{T}_{u,v}}\rangle|\mathsf{T}_{u,v}\rangle - |v,f_v^{-1}(u)\rangle|\mathsf{T}_{u,v}\rangle) \quad \text{by (26) and (14).} \end{aligned}$$

Then by (17) and (7) we have:

$$\left\||w_{u,v}^{\mathsf{T}_{u,v}}\rangle - |v,f_v^{-1}(u)\rangle\right\|^2 = \epsilon_{u,v}.$$

Furthermore, the terms $|w_{u,v}^{\mathsf{T}_{u,v}}\rangle - |v,f_v^{-1}(u)\rangle \in \mathrm{span}\{|z\rangle : z \in \mathcal{Z}_{u,v}\}$ for different $(u,v) \in \overrightarrow{E}(G)$ are pairwise orthogonal. Let ϵ be the upper bound on $\epsilon_{u,v}$ for all $(u,v) \in \overrightarrow{E}(G) \setminus \tilde{E}$, from

Theorem 3.10. Then:

$$\begin{aligned}\||(I-\Pi_{\mathcal{B}})|w_{\mathcal{B}}\rangle\|^2 &\le \frac{1}{\mathsf{w}_0}\sum_{(u,v)\in\overrightarrow{E}(G)}\mathsf{w}_{u,v}\epsilon_{u,v}\\ &\le \mathcal{R}^{\mathsf{T}}\left(\sum_{(u,v)\in\overrightarrow{E}(G)\setminus\tilde{E}}\mathsf{w}_{u,v}\epsilon+4\sum_{(u,v)\in\tilde{E}}\mathsf{w}_{u,v}\right)\le \mathcal{R}^{\mathsf{T}}\left(\epsilon\mathcal{W}+4\widetilde{\mathcal{W}}\right),\end{aligned}$$

where we used $\mathsf{w}_0 = 1/\mathcal{R}^{\mathsf{T}}$, and the trivial upper bound $\epsilon_{u,v} \le 4$ when $(u,v) \in \tilde{E}$. To complete the proof, we give an upper bound on $\||w_{\mathcal{A}}\rangle\|^2$. We first note:

$$\||w_{\mathcal{A}}\rangle\|^2 = \frac{1}{\mathsf{w}_0}\sum_{u\in V(G)}\left\||\psi_\star^{G'}(u)\rangle\right\|^2+\frac{1}{\mathsf{w}_0}\sum_{(u,v)\in\overrightarrow{E}(G)}\mathsf{w}_{u,v}\left\||w_{u,v}^{\mathcal{A}}\rangle\right\|^2.$$

Then we can compute, for any $u \in V(G)$, letting $\delta_{u,V_0} = 1$ iff $u \in V_0$,

$$\left\||\psi_\star^{G'}(u)\rangle\right\|^2 = \sum\nolimits_{v\in\Gamma(u)}\mathsf{w}_{u,v}+\delta_{u,V_0}\mathsf{w}_0\sigma(u)$$

and for any $(u,v) \in \overrightarrow{E}(G)$, since for all $t < \mathsf{T}_{u,v}$, $\sum_{z\in\mathcal{Z}_{u,v}^0}|\langle z|w_{u,v}^t\rangle|^2 = \left\||w_{u,v}^t\rangle\right\|^2 = 1$,

$$\left\||w_{u,v}^{\mathcal{A}}\rangle\right\|^2 = \sum_{t\in\{0,\dots,\mathsf{T}_{u,v}-1\}:t\text{ odd}}\sum_{z\in\mathcal{Z}_{u,v}^0}|\langle z|w_{u,v}^t\rangle|^2\left\||\psi_t^z\rangle\right\|^2 = \left\lfloor\frac{\mathsf{T}_{u,v}}{2}\right\rfloor\cdot 2\le\mathsf{T}_{u,v}.$$

Thus

$$\begin{aligned}\||w_{\mathcal{A}}\rangle\|^2 &\le \frac{1}{\mathsf{w}_0}\sum_{u\in V(G)}\sum_{v\in\Gamma(u)}\mathsf{w}_{u,v}+\frac{1}{\mathsf{w}_0}\sum_{u\in V_0}\mathsf{w}_0\sigma(u)+\frac{1}{\mathsf{w}_0}\sum_{(u,v)\in\overrightarrow{E}(G)}\mathsf{w}_{u,v}\mathsf{T}_{u,v}\\ &= \frac{1}{\mathsf{w}_0}\mathcal{W}(G)+1+\frac{1}{\mathsf{w}_0}\mathcal{W}^{\mathsf{T}}(G).\end{aligned}$$

Note that $\mathcal{W}(G)$ is always less than $\mathcal{W}^{\mathsf{T}}(G)$ (see Definition 2.4). We Thus, complete the proof by substituting $\mathsf{w}_0 = 1/\mathcal{R}^{\mathsf{T}}$ and using the **Negative Condition** of Theorem 3.10 that $\mathcal{W}^{\mathsf{T}}(G) \le \mathcal{W}^{\mathsf{T}}$. ∎

3.2.8 Conclusion of Proof of Theorem 3.10

We now give the proof of Theorem 3.10, by appealing to Theorem 3.8, using $|\psi_0\rangle$ as defined in (16), and $\Psi^{\mathcal{A}}, \Psi^{\mathcal{B}}$ as defined in (15). By the **Setup Subroutine** condition of Theorem 3.10, we can generate $|\sigma\rangle$ in cost S. It follows that we can generate $|\psi_0\rangle = |\sigma\rangle|0\rangle|0\rangle$ in cost $\mathsf{S}' = \mathsf{S} + \log\mathsf{T}_{\max}$, since the last register is $\log\mathsf{T}_{\max}$ qubits. By Lemma 3.12, we can implement $U_{\mathcal{AB}}$ in cost $\mathsf{A}_\star + \mathrm{polylog}(\mathsf{T}_{\max})$.

We use $c_+ = 7$, so $1 \le c_+ \le 50$, as desired. We use

$$C_- = 2\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}}+1,\quad \delta=\frac{\epsilon}{2}\quad\text{and}\quad \delta'=\epsilon\mathcal{R}^{\mathsf{T}}\mathcal{W}+4\mathcal{R}^{\mathsf{T}}\widetilde{\mathcal{W}}.$$

To apply Theorem 3.8, we require that $\delta \leq \frac{1}{(8c_+)^3\pi^8 C_-}$, which follows, for sufficiently large $|x|$, from condition **TS1** of Theorem 3.10:

$$\delta = \frac{\epsilon}{2} = o\left(\frac{1}{\mathcal{R}^\mathsf{T}\mathcal{W}^\mathsf{T}}\right).$$

We also require that $\delta' \leq \frac{3}{4}\frac{1}{\pi^4 c_+} = \frac{3}{28\pi^4}$. The bound on ϵ implies that $\epsilon\mathcal{R}^\mathsf{T}\mathcal{W} = o(1)$, since $\mathcal{W} \leq \mathcal{W}^\mathsf{T}$. The bound $\widetilde{\mathcal{W}} = o(1/\mathcal{R}^\mathsf{T})$ from **TS2** of Theorem 3.10 implies that $4\mathcal{R}^\mathsf{T}\widetilde{\mathcal{W}} = o(1)$. Together these ensure that $\delta' = o(1)$. We verify the remaining conditions of Theorem 3.8 as follows.

Positive Condition: By Lemma 3.18 and Lemma 3.21, if $M \neq \emptyset$, there is a δ-positive witness $|w\rangle$ such that $\frac{|\langle w|\psi_0\rangle|^2}{\||w\rangle\|^2} \geq \frac{1}{c_+} = \frac{1}{7}$.
Negative Condition: By Lemma 3.24, if $M = \emptyset$, there is a δ'-negative witness with $\||w_{\mathcal{A}}\rangle\|^2 \leq C_-$.

Thus, the algorithm described in Theorem 3.8 distinguishes between the cases $M \neq \emptyset$ and $M = \emptyset$ with bounded error in complexity:

$$O\left(\mathsf{S} + \log \mathsf{T}_{\max} + \sqrt{C_-}\,(\mathsf{A}_\star + \mathrm{polylog}(\mathsf{T}_{\max}))\right) = O\left(\mathsf{S} + \sqrt{\mathcal{R}^\mathsf{T}\mathcal{W}^\mathsf{T}}\,(\mathsf{A}_\star + \mathrm{polylog}(\mathsf{T}_{\max}))\right)$$

which completes the proof of Theorem 3.10.

4. Welded Trees

A straightforward application of our technique is to the welded trees problem of [21], illustrating the power of the framework to achieve exponential speedups over classical algorithms. This application also serves as a pedagogic demonstration of the alternative neighbourhoods technique, as it does not make use of a non-trivial edge transition subroutine, and so the resulting algorithm is in that sense rather simple. Although it would be possible to apply our framework without looking "under the hood" at the underlying algorithm, to give intuition about the framework, we instead describe and analyse the full algorithm explicitly, proving our upper bound without appealing to Theorem 3.10.

The Welded Trees Problem: In the welded trees problem, the input is a graph G with $2^{n+2} - 2$ vertices from the set $\{0,1\}^{2n}$, consisting of two full binary trees of depth n (the 2^n leaves are at edge-distance n from the root), which we will refer to as the left and right trees, with additional edges connecting the leaves of one tree to another. Specifically, we assume there are two disjoint perfect matchings from the leaves of the left tree to the leaves of the right tree. Every vertex of this graph has degree 3 except for the roots of the two trees, which we denote by s and t. The graph's structure is shown in Figure 6.

We are promised that $s = 0^{2n}$ is the root of the left tree, but other than s, it is difficult to even find a vertex in the graph, since less than a 2^{-n+2} fraction of strings in $\{0,1\}^{2n}$ labels an actual vertex. We assume we have access to an oracle O_G that tells us the neighbours of any vertex. That is, for any string $\sigma \in \{0,1\}^{2n}$, we can query $O_G(\sigma)$ to learn either $\perp$, indicating it is not a vertex label, or a list of three neighbours (or in case of s and t, only two neighbours).

The welded trees problem is: given such an oracle O_G, output the label of t. We assume we can identify t when we see it, for example by querying it to see that it only has two neighbours. Classically, this problem requires $2^{\Omega(n)}$ queries [21], which is intuitively because the problem is set up to ensure that the only thing a classical algorithm can do is a random walk on G, starting from s. The hitting time from s to t is $2^{\Omega(n)}$ because a walker is always twice as likely to move towards the centre of the graph than away from it, and so a walker starting at s will quickly end up in the centre of the graph, but then will be stuck there for a long time. On the other hand a quantum algorithm can solve this problem in $\mathrm{poly}(n)$ time [21], with the best known upper bound being $O(n^{1.5} \log n)$ queries [8]. We show how to solve this problem in our new framework, with $O(n)$ queries and $O(n^2)$ time. Specifically, in the remainder of this section we show:

THEOREM 4.1. *Let $g : \{0,1\}^{2n} \to \{0,1\}$ be any function. Then there is a quantum algorithm that, given an oracle O_G for a welded trees graph G as above, decides if $g(t) = 1$ with bounded error in $O(n)$ queries to O_G. If g can be computed in $O(n)$ complexity, then the time complexity of this algorithm is $O(n^2)$.*

From this it immediately follows that we can solve the welded trees problem with $2n$ applications of this algorithm, letting $g(t) = t_i$ – the i-th bit of t – for $i = 1, \ldots, 2n$. However, we can also do slightly better by composing it with the Bernstein-Vazirani algorithm, which recovers a string t in a single quantum query to an oracle that computes $|z\rangle \mapsto (-1)^{z \cdot t}|z\rangle$ for any string $z \in \{0,1\}^{2n}$.

COROLLARY 4.2. *There is a quantum algorithm that can solve the welded trees problem in $O(n)$ queries and $O(n^2)$ time.*

PROOF. For any $z \in \{0,1\}^{2n}$, define $g_z(t) = z \cdot t = \sum_{i=1}^{2n} z_i t_i \mod 2$. Clearly g_z can be computed in complexity $O(n)$. To compute $g_z(t)$, we simply run the algorithm from Theorem 4.1. The Bernstein-Vazirani algorithm [13] outputs t using a single such query, and $O(n)$ additional gates. ■

Previous quantum algorithms for this problem are quantum walk algorithms in the sense that they construct a Hamiltonian based on the structure of the graph and simulate it, but this technique has not been replicated for many other problems, unlike quantum walk search

algorithms described in Section 1.1[15]. Our hope is that our new quantum walk search framework bridges the gap between a general and easily applied technique (quantum walk search algorithms) and exponential speedups.

G as a Weighted Network: Assume that n is even (this greatly simplifies notation, the proof is equivalent for the case where n is odd). We partition $V(G)$ into $V_0 \cup V_1 \cup \cdots \cup V_{2n+1}$, where V_k is the set of vertices at distance k from s, so $V_0 = \{s\}$, and $V_{2n+1} = \{t\}$. We first prove Theorem 4.1 under the assumption that it is possible to check, for any vertex, whether it is in $V_{\text{even}} := V_0 \cup V_2 \cup \cdots \cup V_{2n}$, or $V_{\text{odd}} := V_1 \cup V_3 \cup \cdots \cup V_{2n+1}$. At the end of this section, we will explain how to remove this assumption. Define $M = \{t\}$ if $g(t) = 1$ and otherwise $M = \emptyset$.

For $k \in \{1, \ldots, 2n+1\}$, define

$$E_k = \{(u,v) \in V_{k-1} \times V_k : \{u,v\} \in E(G)\}$$
$$\text{so } |E_k| = \begin{cases} 2^k & \text{if } k \in \{1, \ldots, n+1\} \\ 2^{2n+2-k} & \text{if } k \in \{n+1, \ldots, 2n+1\}. \end{cases} \tag{30}$$

We define the set of directed edges as follows (see Definition 2.1):

$$\overrightarrow{E}(G) = \bigcup_{\substack{k \in \{1,\ldots,2n+1\}: \\ k \bmod 4 \in \{0,1\}}} \{(u,v) : (u,v) \in E_k\} \cup \bigcup_{\substack{k \in \{1,\ldots,2n+1\}: \\ k \bmod 4 \in \{2,3\}}} \{(u,v) : (v,u) \in E_k\}. \tag{31}$$

Note that $E_k \subset \overrightarrow{E}(G)$ only holds when $k \mod 4 \in \{0,1\}$, so we do not always set the default directions left to right. At the moment it is not clear why we set the directions this way, but one thing this accomplishes is that the direction of edges switches at every layer of V_{odd}. Figure 6 illustrates how the directions of the edges change layer by layer. We now assign weights to all edges in G. We assign all edges in E_k the same weight, w_k, defined:

$$\mathsf{w}_k = \begin{cases} 2^{-2\lceil k/2 \rceil} & \text{if } k \in \{1, \ldots, n\} \\ 2^{-2(n+2-\lceil k/2 \rceil)} & \text{if } k \in \{n+1, \ldots, 2n+1\}. \end{cases} \tag{32}$$

It should be somewhat clear why this might be a useful weighting: we have increased the probability of moving away from the centre. Finally, we add a vertex v_0 connected to s by an edge of weight w_0, and connected to t by an edge of weight w_M if and only if t is marked, and call this resulting graph G'. We remark that we do not need to account for v_0 explicitly if we just want to appeal directly to Theorem 3.10, but we are going to explicitly construct an algorithm for the sake of exemplification.

Theorem 3.10 assumes we label the outgoing edges from a vertex u by indices from some set $L(u)$, and then implement a map $|u,i\rangle \mapsto |v,j\rangle$ for any $i \in L(u)$ (see also Definition 2.3). Here, since we assume we can simply query the set of the three neighbours of u, $\Gamma(u) = \{v_1, v_2, v_3\}$,

15 Where similar frameworks have been developed in the setting of continuous quantum walks [6], they are also limited to a quadratic speedup.

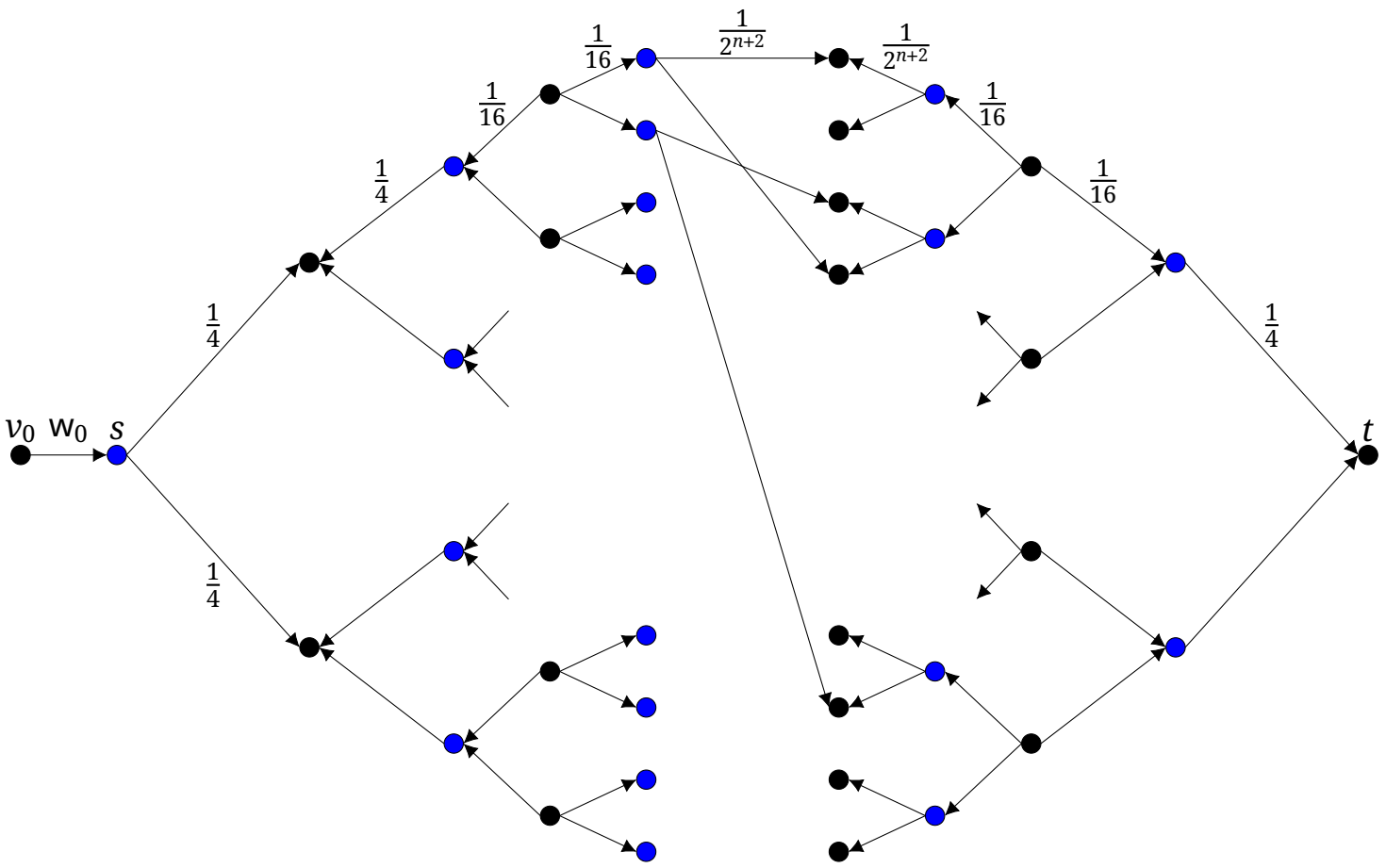


Figure 6. The weights of the graph G' (obtained from adding v_0 to G), and default edge directions.

in unit time, we can let $L(u) = \Gamma(u)$. In that case, the map $|u,i\rangle \mapsto |v,j\rangle$ is actually just $|u,v\rangle \mapsto |v,u\rangle$, which can be accomplished by swapping the two registers. The decomposition of $L(u)$ into $L^+(u) = \Gamma^+(u)$ and $L^-(u) = \Gamma^-(u)$ depends on the directions we assigned to the edges coming out of u in (31).

Our algorithm will be based on phase estimation of a unitary acting on:

$$H = \text{span}\{|u,v\rangle : u \in V(G), v \in \Gamma(u)\}. \tag{33}$$

Here we let $\Gamma(u) = \Gamma_{G'}(u)$ refer to the neighbours of u in G', meaning that $\Gamma(s)$ and $\Gamma(t)$ both include v_0. We emphasise that for all $(u,v) \in \overrightarrow{E}(G)$, we have included both $|u,v\rangle$ and $|v,u\rangle$ in H, as orthonormal vectors. This is different from the H defined in Section 3.2.1 (in which $|u,v\rangle = -|v,u\rangle$), and instead we should think of $|u,v\rangle$ as analogous to $|u,i\rangle|0\rangle$ in (8): v takes the place of the label i, and there is no $|0\rangle$ because there is no transition subroutine steps to count. Alternatively, we can think of $|u,v\rangle$ and $|v,u\rangle$ as labelling distinct edges on a path of length two connecting u and v: one adjacent to u, and the other adjacent to v (see also Figure 7).

For any $u \in V(G)$ and $v \in \Gamma(u)$, let $\Delta_{u,v} = 0$ if $v \in \Gamma^+(u)$ (i.e. $(u,v) \in \overrightarrow{E}(G') = \overrightarrow{E}(G) \cup \{(s,v_0),(t,v_0)\}$) and $\Delta_{u,v} = 1$ if $v \in \Gamma^-(u)$ (i.e. $(v,u) \in \overrightarrow{E}(G')$). We can then define star states in H as follows, for all $u \in V(G)$ with neighbours (in G') $\Gamma(u) = \{v_1,v_2,v_3\}$:

$$|\psi_\star^{G'}(u)\rangle := \sqrt{w_{u,v_1}}(-1)^{\Delta_{u,v_1}}|u,v_1\rangle + \sqrt{w_{u,v_2}}(-1)^{\Delta_{u,v_2}}|u,v_2\rangle + \sqrt{w_{u,v_3}}(-1)^{\Delta_{u,v_3}}|u,v_3\rangle. \tag{34}$$

We cannot efficiently generate these star states. For $\ell \in \{0, \ldots, n-1\}$ and $v \in V_{2\ell+1} \subset V_{\text{odd}}$ with neighbours $\Gamma(v) = \{u_1, u_2, u_3\}$, we have, referring to (31) and (32):

$$|\psi_\star^{G'}(v)\rangle = \begin{cases} (-1)^{\ell+1} \frac{1}{2^{\ell+1}} \left(|v, u_1\rangle + |v, u_2\rangle + |v, u_3\rangle\right) & \text{if } \ell \in \{0, \ldots, n/2-1\} \\ (-1)^{\ell+1} \frac{1}{2^{n-\ell+1}} \left(|v, u_1\rangle + |v, u_2\rangle + |v, u_3\rangle\right) & \text{if } \ell \in \{n/2, \ldots, n\}. \end{cases} \tag{35}$$

Even though we don't know which layer v is in, and therefore we do not know the precise scaling or direction of its edges, by querying the neighbours of v to learn the set $\{u_1, u_2, u_3\}$, we know that:

$$|\psi_\star^{G'}(v)\rangle \in \text{span}\{|v, u_1\rangle + |v, u_2\rangle + |v, u_3\rangle\}.$$

On the other hand, for $\ell \in \{1, \ldots, n\}$ and $u \in V_{2\ell} \subset V_{\text{even}}$, though we can compute $\Gamma(u) = \{v_1, v_2, v_3\}$, we do not know which neighbour is the parent – the unique neighbour of u that is further from the centre of the graph than u. Let $p(u) \in \{v_1, v_2, v_3\}$ be the parent of u, and $c_1(u)$, $c_2(u)$ the other two vertices in $\{v_1, v_2, v_3\}$. Then, referring to (31), (32) and (34), the star state of u has the form:

$$|\psi_\star^{G'}(u)\rangle = \begin{cases} (-1)^{\ell+1} \frac{1}{2^\ell} \left(|u, p(u)\rangle - \frac{1}{2}|u, c_1(u)\rangle - \frac{1}{2}|u, c_1(u)\rangle\right) & \text{if } \ell \in \left\{0, \ldots, \frac{n}{2}\right\} \\ (-1)^{\ell+1} \frac{1}{2^{n-\ell+1}} \left(|u, p(u)\rangle - \frac{1}{2}|u, c_1(u)\rangle - \frac{1}{2}|u, c_1(u)\rangle\right) & \text{if } \ell \in \left\{\frac{n}{2}+1, \ldots, n\right\}. \end{cases}$$

Generating this state would require knowing which of $\{v_1, v_2, v_3\}$ is the parent, $p(u)$, which is not something that can be learned from simply querying the neighbours of u. However, if we were to weight everything uniformly, our quantum walk would, like a classical random walk, suffer from the fact that the centre of the graph has exponential weight, and most time will be spent there. Thus, we employ the alternative neighbourhoods technique. For $u \in V_{\text{even}} \setminus \{s\}$ with neighbours $v_1 < v_2 < v_3$, define:

$$\forall j \in \{1, 2, 3\},\ |\psi_\star^j(u)\rangle = \sqrt{\frac{2}{3}} \left(|u, v_j\rangle - \frac{1}{2}|u, v_{j+1}\rangle - \frac{1}{2}|u, v_{j+2}\rangle\right), \tag{36}$$

where the indices add modulo 3. Then we know that

$$\frac{|\psi_\star^G(u)\rangle}{\left\||\psi_\star^G(u)\rangle\right\|} \in \{|\psi_\star^1(u)\rangle, |\psi_\star^2(u)\rangle, |\psi_\star^3(u)\rangle\} =: \Psi_\star(u),$$

though we do not know which one.

For s and t, suppose the neighbours are $\Gamma(s) = \{v_0, v_1, v_2\}$ and $\Gamma(t) = \{v_0, u_1, u_2\}$ – meaning that when we query s, we learn $\{v_1, v_2\}$, and when we query t we learn $\{u_1, u_2\}$, and then we add to each neighbourhood the additional special vertex v_0 (although if t is not marked, we set the weight $\mathsf{w}_{t,v_0} = 0$). Then the star states are, respectively:

$$\begin{aligned} |\psi_\star^{G'}(s)\rangle &= \sqrt{\mathsf{w}_0}|s, v_0\rangle + \frac{1}{2}|s, v_1\rangle + \frac{1}{2}|s, v_2\rangle \\ |\psi_\star^{G'}(t)\rangle &= \delta_{g(t),1}\sqrt{\mathsf{w}_\mathsf{M}}|t, v_0\rangle - \frac{1}{2}|t, u_1\rangle - \frac{1}{2}|t, u_2\rangle, \end{aligned} \tag{37}$$

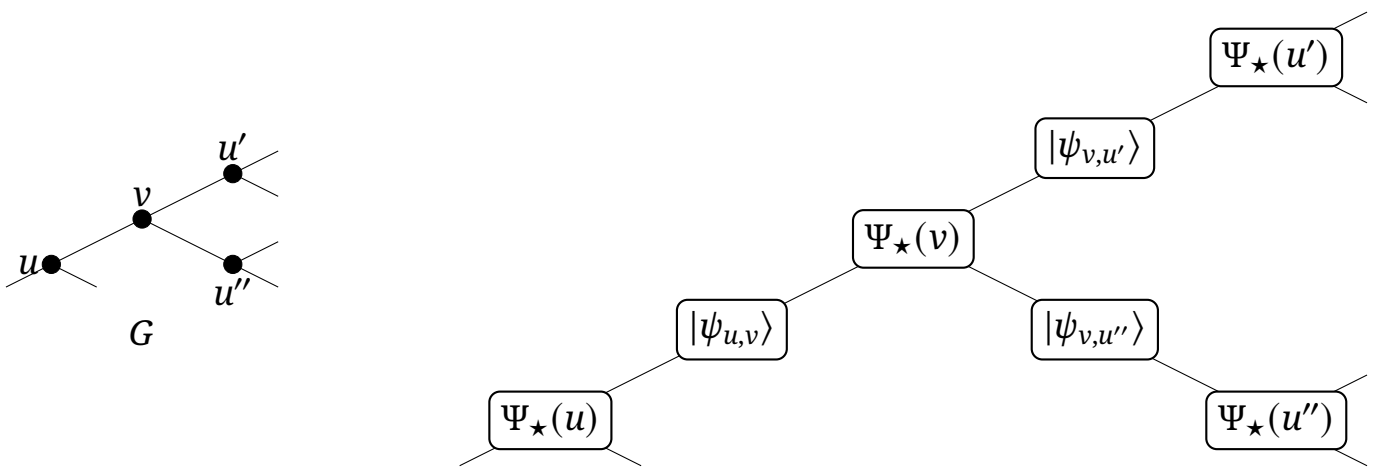


Figure 7. A piece of the graph G (left) and the corresponding piece of the overlap graph of the spaces $\mathrm{span}\{\Psi_\star(u)\}$ and $\mathrm{span}\{|\psi_{u,v}\rangle\}$. There is an edge between two nodes in the overlap graph if and only if the sets contain overlapping states. Compare with Figure 5.

where $\delta_{g(t),1} = 1$ if and only if $t \in M$. To see that this follows from (34), note that $(s, v_0), (t, v_0) \in \overrightarrow{E}(G')$ by definition, and for $i \in \{1, 2\}$, $(s, v_i) \in E_1$, so $(s, v_i) \in \overrightarrow{E}(G)$ since $1 = 1 \mod 4$, and $(u_i, t) \in E_{2n+1}$, so $(u_i, t) \in \overrightarrow{E}(G)$, since $2n + 1 = 1 \mod 4$ (we are assuming n is even), which is why we have minus signs in front of the $|t, u_i\rangle$ (see (31)). We can generate $|\psi_\star^{G'}(s)\rangle$ and $|\psi_\star^{G'}(t)\rangle$, because we can recognise s and t. Thus, for all $v \in V_{\mathrm{odd}} \cup \{s\}$ (the vertices with easy to generate star states), we let $\Psi_\star(v) := \{|\psi_\star^{G'}(v)\rangle\}$.

Finally, we define states corresponding to the transitions $|u, v\rangle \mapsto |v, u\rangle$:

$$\forall (u, v) \in \overrightarrow{E}(G),\ |\psi_{u,v}\rangle := |u, v\rangle - |v, u\rangle. \tag{38}$$

The star states and the transition states (38) will comprise all states of $\Psi^{\mathcal{A}} \cup \Psi^{\mathcal{B}}$ as follows:

$$\begin{aligned} \Psi^{\mathcal{A}} &:= \bigcup_{u \in V(G)} \Psi_\star(u) = \bigcup_{u \in V_{\mathrm{odd}} \cup \{s\}} \{|\psi_\star^{G'}(u)\rangle\} \cup \bigcup_{u \in V_{\mathrm{even}} \setminus \{s\}} \{|\psi_\star^1(u)\rangle, |\psi_\star^2(u)\rangle, |\psi_\star^3(u)\rangle\} \\ \Psi^{\mathcal{B}} &:= \{|\psi_{u,v}\rangle : (u, v) \in \overrightarrow{E}(G)\}. \end{aligned} \tag{39}$$

Then $\Psi^{\mathcal{B}}$ is a pairwise orthogonal set, and if we replace each $\Psi_\star(u)$ in $\Psi^{\mathcal{A}}$ with an orthonormal basis for $\mathrm{span}\{\Psi_\star(u)\}$ we get a pairwise orthogonal set. Figure 7 shows that the overlap graph for the sets $\Psi_\star(u)$ for $u \in V(G)$ and $\{|\psi_{u,v}\rangle\}$ for $(u, v) \in \overrightarrow{E}(G)$ is bipartite, and we have chosen $\Psi^{\mathcal{A}}$ and $\Psi^{\mathcal{B}}$ according to this bipartition.

Let $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$, where $\Pi_{\mathcal{A}}$ and $\Pi_{\mathcal{B}}$ are orthogonal projectors on $\mathcal{A} := \mathrm{span}\{\Psi^{\mathcal{A}}\}$ and $\mathcal{B} := \mathrm{span}\{\Psi^{\mathcal{B}}\}$ respectively. In the remainder of this section, we will show that we can solve the welded trees problem with bounded error by performing phase estimation of $U_{\mathcal{AB}}$ on initial state $|\psi_0\rangle = |s, v_0\rangle$, as described in Theorem 3.8.

Implementing the Unitary: In order to implement $U_{\mathcal{AB}}$, we need to be able to generate an orthonormal basis for each of $\mathcal{A}$ and $\mathcal{B}$, for which we use the following fact.

CLAIM 4.3. *Let $\omega_3 = e^{2\pi i/3}$ be a third root of unity. For a vertex $u \in V(G)$ with neighbours $v_1 < v_2 < v_3$, define for $j \in \{0,1,2\}$:*

$$|\widehat{\psi}^j(u)\rangle := \frac{1}{\sqrt{3}}\left(|u,v_1\rangle + \omega_3^j|u,v_2\rangle + \omega_3^{2j}|u,v_3\rangle\right).$$

Then these three vectors are an orthonormal set, and for $u \in V_{\text{even}} \setminus \{s\}$,

$$\text{span}\{|\psi_\star^1(u)\rangle, |\psi_\star^2(u)\rangle, |\psi_\star^3(u)\rangle\} = \text{span}\{|\widehat{\psi}^1(u)\rangle, |\widehat{\psi}^2(u)\rangle\}.$$

For $v \in V_{\text{odd}} \setminus \{t\}$,

$$|\psi_\star^{G'}(v)\rangle \in \text{span}\{|\widehat{\psi}^0(u)\rangle\}.$$

PROOF. Note that the states $|\widehat{\psi}^0(u)\rangle, |\widehat{\psi}^1(u)\rangle, |\widehat{\psi}^2(u)\rangle$ form an orthonormal basis for the space $\text{span}\{|u,v_1\rangle, |u,v_2\rangle, |u,v_3\rangle\}$ – it is the Fourier basis. Thus, the first part of the statement is simply proven by observing that for each $j \in \{1,2,3\}$, $|\psi_\star^j(u)\rangle \in \text{span}\{|u,v_1\rangle, |u,v_2\rangle, |u,v_3\rangle\}$ and $\langle\psi_\star^j(u)|\widehat{\psi}^0(u)\rangle = 0$; and that $\text{span}\{|\psi_\star^j(u)\rangle\}_{j=1}^3$ has dimension greater than 1. The second statement follows easily from (35). ∎

LEMMA 4.4. *The unitary $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$ can be implemented in $O(1)$ queries to O_G, and $O(n)$ elementary operations.*

PROOF. Let

$$H' = \text{span}\{|j\rangle|u,v\rangle : j \in \{0,1,2\}, u \in V(G), v \in \Gamma(u) \cup \{\perp\}\},$$

so in particular $|0\rangle \otimes H \subset H'$ (where H is as in (33)).

We first describe how to implement $2\Pi_{\mathcal{A}} - I$. We describe a unitary $U_\star$ on H', and in particular, its behaviour on states of the form $|j\rangle|u,\perp\rangle$, where $j = 0$ whenever $u \in V_{\text{odd}} \cup \{s\}$, and $j \in \{1,2\}$ whenever $u \in V_{\text{even}} \setminus \{s\}$. We begin by querying the neighbours of u in an ancillary register, Q, initialised to $|0\rangle$ using O_G:

$$|j\rangle|u,\perp\rangle|0\rangle_Q \mapsto |j\rangle|u,\perp\rangle|v_1,v_2,v_3\rangle_Q$$

where if $u \in \{s,t\}$, $v_1 < v_2$ and $v_3 = \perp$ (which we can interpret as v_0), and otherwise, since we assume $u \in V(G)$, $v_1 < v_2 < v_3$ are the neighbours of u. We initialise an ancilla qubit A, and compute a trit $|a\rangle_A$ for $a \in \{0,1,2\}$ as follows, to determine what happens next. If $v_3 \neq \perp$, then $a = 0$. Else if $u = 0^{2n} = s$, we let $a = 1$. Else if $v_3 = \perp$ but $u \neq 0^{2n}$, so $u = t$, we let $a = 2$.

Controlled on $|0\rangle_A$, apply a Fourier transfrom F_3 to $|j\rangle$ to get $|\hat{j}\rangle = (|1\rangle + \omega_3^j|2\rangle + \omega_3^{2j}|3\rangle)/\sqrt{3}$. Then, still conditioned on $|0\rangle_A$, swap the first and third registers, so now the first register contains $|\perp\rangle$, and then perform $|\perp\rangle \mapsto |0\rangle$ on the first register to get: $|0\rangle|u\rangle|\hat{j}\rangle|0\rangle|v_1,v_2,v_3\rangle_Q|0\rangle_A$. Then, conditioned on the value in the $|\hat{j}\rangle$ register, we can copy over the first, second or third value in the $|v_1,v_2,v_3\rangle$ register to get:

$$\frac{1}{\sqrt{3}}|0\rangle|u\rangle\left(|1\rangle|v_1\rangle + \omega_3^j|2\rangle|v_2\rangle + \omega_3^{2j}|3\rangle|v_3\rangle\right)|v_1,v_2,v_3\rangle_Q|0\rangle_A.$$

This requires $O(n)$ basic operations. We can uncompute the value $|i\rangle$ in $|i\rangle|v_i\rangle$ by referring to the last register to learn v_i's position, and then we are left with: $|0\rangle|\widehat{\psi^j}(u)\rangle|v_1, v_2, v_3\rangle_Q|0\rangle_A$.

Next, we control on $|1\rangle_A$, meaning $u = s$. In that case, we assume that $j = 0$. Using v_1 and v_2 in the last register, we can map $|\perp\rangle$ to a state proportional to $\sqrt{\mathsf{w}_0}|v_0\rangle + \frac{1}{2}|v_1\rangle + \frac{1}{2}|v_2\rangle$ to get

$$|0\rangle \frac{|\psi_\star^{G'}(s)\rangle}{\left\||\psi_\star^{G'}(s)\rangle\right\|^2}|v_1, v_2, v_3\rangle_Q|1\rangle_A.$$

Lastly, we control on $|2\rangle_A$, meaning $u = t$. We can compute $g(t)$ in a separate register, and using $g(t)$, v_1, and v_2, map $|\perp\rangle$ to a state proportional to: $\delta_{g(t),1}\sqrt{\mathsf{w}_\mathsf{M}}|v_0\rangle - \frac{1}{2}|v_1\rangle - \frac{1}{2}|v_2\rangle$ to get

$$|0\rangle \frac{|\psi_\star^{G'}(t)\rangle}{\left\||\psi_\star^{G'}(t)\rangle\right\|^2}|v_1, v_2, v_3\rangle_Q|2\rangle_A.$$

We can uncompute the ancilla A, since the registers containing u, and v_1, v_2, v_3 haven't changed. Since the register containing u has not changed, we can uncompute the register $|v_1, v_2, v_3\rangle_Q$ using another call to O_G. Then, removing ancillae, we have performed a map, $U_\star$ that acts, for $j = 0$ when $u \in V_{\text{odd}} \cup \{s\}$ and $j \in \{1, 2\}$ when $u \in V_{\text{even}} \setminus \{s\}$, as: $|j\rangle|u, \perp\rangle \mapsto |0\rangle|\widehat{\psi^j}(u)\rangle$, where, using Claim 4.3, for all $u \in V_{\text{odd}} \cup \{s\}$,

$$\text{span}\{|\widehat{\psi^0}(u)\rangle\} = \text{span}\{|\psi_\star^{G'}(u)\rangle\} = \text{span}\{\Psi_\star(u)\}$$

and for all $u \in V_{\text{even}} \setminus \{s\}$:

$$\text{span}\{|\widehat{\psi^1}(u)\rangle, |\widehat{\psi^2}(u)\rangle\} = \text{span}\{|\psi_\star^1(u)\rangle, |\psi_\star^2(u)\rangle, |\psi_\star^3(u)\rangle\} = \text{span}\{\Psi_\star(u)\}.$$

Thus, $U_\star$ maps the subspace

$$\mathcal{L} := \text{span}\{|0, u, \perp\rangle : u \in V_{\text{odd}} \cup \{s\}\} \cup \{|1, u, \perp\rangle, |2, u, \perp\rangle : u \in V_{\text{even}} \setminus \{s\}\}$$

of H' to $|0\rangle \otimes \text{span}\{\Psi^{\mathcal{A}}\} \cong \mathcal{A}$, and thus $2\Pi_{\mathcal{A}} - I = U_\star\,(2\Pi_{\mathcal{L}} - I)\,U_\star^\dagger$. We describe how to implement $2\Pi_{\mathcal{L}} - I$. Initialise ancillary flag qubits $|0\rangle_{F_1}|0\rangle_{F_2}|0\rangle_{F_3}$. For a computational basis state $|j\rangle|u, v\rangle$ of H', by assumption (which is removed at the end of this section) we can efficiently check whether u is in V_{odd} or V_{even}, and we can check whether $u = s = 0^{2n}$ in $O(n)$ cost. If $u \in V_{\text{odd}} \cup \{s\}$, we check if the first register is 0, and if not, flip F_1 to get $|1\rangle_{F_1}$. If $u \in V_{\text{even}} \setminus \{s\}$, we check if the first register is 1 or 2, and if not, flip F_2 to get $|1\rangle_{F_2}$. If the last register is not $\perp$, flip F_3 to get $|1\rangle_{F_3}$. Reflect if any flag is set, and then uncompute all flags. This can all be done in $O(n)$ basic operations.

Next, we describe how to implement $2\Pi_{\mathcal{B}} - I$. We describe a unitary U_S on H', and in particular, its behaviour on states of the form $|1\rangle|u, v\rangle$ for $\{u, v\} \in E(G)$ with $u < v$. First, apply a Hadamard gate to the first register, and then, controlled on its value, swap the second two registers to get:

$$(|0\rangle|u, v\rangle - |1\rangle|v, u\rangle)/\sqrt{2}.$$

We can uncompute the first register by adding in a bit indicating if the last two registers are in sorted order, to get:

$$|0\rangle\frac{1}{\sqrt{2}}(|u,v\rangle - |v,u\rangle) \in \begin{cases} \text{span}\{|0\rangle|\psi_{u,v}\rangle\} & \text{if } (u,v) \in \overrightarrow{E}(G) \\ \text{span}\{|0\rangle|\psi_{v,u}\rangle\} & \text{if } (v,u) \in \overrightarrow{E}(G). \end{cases}$$

Thus, $U_{\mathcal{S}}$ maps

$$\mathcal{L}' := \text{span}\{|1\rangle|u,v\rangle : \{u,v\} \in E(G), u < v\}$$

to $\text{span}\{|0\rangle|\psi_{u,v}\rangle : (u,v) \in \overrightarrow{E}(G)\} \cong \mathcal{B}$, and so $2\Pi_{\mathcal{B}} - I = U_{\mathcal{S}}(2\Pi_{\mathcal{L}'} - I)U_{\mathcal{S}}^{\dagger}$. To implement $(2\Pi_{\mathcal{L}'} - I)$, it is enough to check that the first register is 1, and u and v are in sorted order (we know $\{u,v\} \in E(G)$ by the structure of H'). This can be done in $O(n)$ basic operations. ■

Negative Analysis: For the negative analysis, it would be sufficient to upper bound the total weight of G and appeal to Theorem 3.10, but we will instead explicitly construct a negative witness (see Definition 3.2) in order to appeal to Theorem 3.8. That is, we show explicitly how to express $|\psi_0\rangle = |s, v_0\rangle$ as the sum of something in $\mathcal{A}$ and something in $\mathcal{B}$, when t is not marked. We let:

$$|w_{\mathcal{A}}\rangle := \frac{1}{\sqrt{\mathsf{w}_0}}\sum_{u\in V(G)}|\psi_{\star}^{G'}(u)\rangle \text{ and } |w_{\mathcal{B}}\rangle := -\frac{1}{\sqrt{\mathsf{w}_0}}\sum_{(u,v)\in\overrightarrow{E}(G)}\sqrt{\mathsf{w}_{u,v}}(-1)^{\Delta_{u,v}}|\psi_{u,v}\rangle. \tag{40}$$

Then we prove the following.

LEMMA 4.5. *Suppose $M = \emptyset$. Then $|w_{\mathcal{A}}\rangle, |w_{\mathcal{B}}\rangle$ form a 0-negative witness with $\||w_{\mathcal{A}}\rangle\|^2 = O(n/\mathsf{w}_0)$.*

PROOF. When $M = \emptyset$ (that is, $t \notin M$), the graph G' is simply G with an additional vertex v_0 connected to s by an edge from s to v_0 of weight w_0. Let $\Gamma_G(u)$ denote the neighbourhood of u in G, and $\Gamma_{G'}(u)$ the neighbourhood of u in G', so, assuming $M \neq \emptyset$, for all $u \in V(G) \setminus \{s\}$, $\Gamma_G(u) = \Gamma_{G'}(u)$, and $\Gamma_{G'}(s) = \Gamma_G(s) \cup \{v_0\}$. Thus, referring to (40) and (34), we have:

$$\begin{aligned}\sqrt{\mathsf{w}_0}|w_{\mathcal{A}}\rangle &= \sum_{u\in V(G)}\sum_{v\in\Gamma_{G'}(u)}\sqrt{\mathsf{w}_{u,v}}(-1)^{\Delta_{u,v}}|u,v\rangle \\ &= \sum_{u\in V(G)}\sum_{v\in\Gamma_{G}(u)}\sqrt{\mathsf{w}_{u,v}}(-1)^{\Delta_{u,v}}|u,v\rangle + \sqrt{\mathsf{w}_0}|s,v_0\rangle,\end{aligned}$$

and referring to (38), we have:

$$\begin{aligned}\sqrt{\mathsf{w}_0}|w_{\mathcal{B}}\rangle &= -\sum_{(u,v)\in\overrightarrow{E}(G)}\sqrt{\mathsf{w}_{u,v}}(-1)^{\Delta_{u,v}}(|u,v\rangle - |v,u\rangle) \\ &= -\sum_{u\in V(G)}\sum_{v\in\Gamma_G^+(u)}\sqrt{\mathsf{w}_{u,v}}(-1)^{\Delta_{u,v}}|u,v\rangle - \sum_{v\in V(G)}\sum_{u\in\Gamma_G^-(v)}\sqrt{\mathsf{w}_{v,u}}(-1)^{\Delta_{v,u}}|v,u\rangle \\ &= -\sum_{u\in V(G)}\sum_{v\in\Gamma_G(u)}\sqrt{\mathsf{w}_{u,v}}(-1)^{\Delta_{u,v}}|u,v\rangle,\end{aligned}$$

where we have used the fact that $\mathsf{w}_{u,v} = \mathsf{w}_{v,u}$ and $(-1)^{\Delta_{u,v}} = -(-1)^{\Delta_{v,u}}$. Thus, we see that:

$$\sqrt{\mathsf{w}_0}\left(|w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle\right) = \sqrt{\mathsf{w}_0}|s, v_0\rangle = \sqrt{\mathsf{w}_0}|\psi_0\rangle.$$

It is simple to check that $|w_{\mathcal{A}}\rangle \in \mathrm{span}\{\Psi^{\mathcal{A}}\}$ and $|w_{\mathcal{B}}\rangle \in \mathrm{span}\{\Psi^{\mathcal{B}}\}$ (see (39)), so we see that these states form a 0-negative witness.

We can analyse the complexity of this witness by computing an upper bound on $\||w_{\mathcal{A}}\rangle\|^2$:

$$\begin{aligned}\||w_{\mathcal{A}}\rangle\|^2 &= \frac{2}{\mathsf{w}_0}\sum_{e\in E(G)}\mathsf{w}_e = \frac{2}{\mathsf{w}_0}\sum_{k=0}^{2n+1}|E_k|\mathsf{w}_k \\ &= \frac{1}{\mathsf{w}_0}\sum_{k=0}^{n}2^k\frac{1}{2^{2\lceil k/2\rceil}} + \frac{2}{w_0}\sum_{k=n+1}^{2n+1}2^{2n+1-k+1}\frac{1}{2^{2n+4-2\lceil k/2\rceil}} = O(n/\mathsf{w}_0)\end{aligned}$$

using the fact that edges in E_k have weight w_k defined in (32), and $|E_k|$ in (30). ■

Positive Analysis: In the case when t is marked, so $M = \{t\} \neq \emptyset$, we exhibit a positive witness (see Definition 3.5) $|w\rangle$ that is orthogonal to all states in $\Psi^{\mathcal{A}} \cup \Psi^{\mathcal{B}}$, and that has non-zero overlap with $|\psi_0\rangle = |s, v_0\rangle$. If θ is any st-flow on G (see Definition 2.2), as long as $M = \{t\}$, so there is an edge from t to v_0, we can extend θ to a circulation on G' by sending the unit flow coming into t out to v_0, and then back into s. That is, define $\theta(t, v_0) = 1$, and $\theta(s, v_0) = -1$. Then if we define

$$|w\rangle = \frac{\theta(s,v_0)}{\sqrt{\mathsf{w}_0}}|s,v_0\rangle + \sum_{(u,v)\in\overrightarrow{E}(G)}\frac{\theta(u,v)}{\sqrt{\mathsf{w}_{u,v}}}(|u,v\rangle + |v,u\rangle) + \frac{\theta(t,v_0)}{\sqrt{\mathsf{w}_M}}|t,v_0\rangle \tag{41}$$

it turns out that this will always be orthogonal to all star states $|\psi_\star^{G'}(u)\rangle$, as well as all transition states $|\psi_{u,v}\rangle$. However, there are additional states $|\psi_\star^j(u)\rangle \in \Psi^{\mathcal{A}} \cup \Psi^{\mathcal{B}}$, and in order to be orthogonal to all of these, the flow must satisfy additional constraints. We will show that all these constraints are satisfied by the natural choice of flow that, for each vertex, comes in from the parent and then sends half to each child. That is, letting E_k for $k \in \{1, \ldots, 2n+1\}$ be as in (30), and $E_0 = \{(v_0, s)\}$, define:

$$\forall k \in \{1,\ldots,2n+1\}, (u,v) \in E_k, \quad \theta(u,v) := \frac{1}{|E_k|} = 2^{-k}. \tag{42}$$

Then we first prove the following:

CLAIM 4.6. *Let $u \in V_{\text{even}} \setminus \{s\}$, and let $|w_u\rangle = (|u\rangle\langle u| \otimes I)|w\rangle$. Then $|w_u\rangle \propto |\widehat{\psi^0}(u)\rangle$.*

PROOF. Since $u \notin \{s, t\}$, we have:

$$|w_u\rangle = \sum_{v\in\Gamma^+(u)}\frac{\theta(u,v)}{\sqrt{\mathsf{w}_{u,v}}}|u,v\rangle + \sum_{u'\in\Gamma^-(u)}\frac{\theta(u',u)}{\sqrt{\mathsf{w}_{u,u'}}}|u,u'\rangle = \sum_{v\in\Gamma(u)}(-1)^{\Delta_{u,v}}\frac{\theta(u,v)}{\sqrt{\mathsf{w}_{u,v}}}|u,v\rangle.$$

using $\theta(u,v) = -\theta(v,u)$, $\mathsf{w}_{u,v} = \mathsf{w}_{v,u}$, and $\Delta_{u,v} = 0$ if $v \in \Gamma^+(u)$, and 1 otherwise. Recall that u has three neighbours: a parent $p(u)$ and two children $c_1(u)$ and $c_2(u)$. Since $u \in V_{2\ell}$ for some ℓ, the edges adjacent to u are (up to direction) in $E_{2\ell}$ and $E_{2\ell+1}$. If ℓ is even, $2\ell = 0 \mod 4$

and $2\ell+1 = 1 \bmod 4$, so by (30), $\Delta_{p(u),u} = \Delta_{u,c_1(u)} = \Delta_{u,c_2(u)} = 0$ if $\ell \in \{1,\dots,n/2\}$ (i.e. u is in the left tree, so its parent is to its left) and $= 1$ otherwise. If ℓ is odd, $2\ell = 2 \bmod 4$ and $2\ell+1 = 3 \bmod 4$, so $\Delta_{p(u),u} = \Delta_{u,c_1(u)} = \Delta_{u,c_2(u)} = 1$ if $\ell \in \{1,\dots,n/2\}$ and $= 0$ otherwise. Thus, since $(-1)^{\Delta_{u,p(u)}} = -(-1)^{\Delta_{p(u),u}}$, we always have:

$$|w_u\rangle = \pm\left(-\frac{\theta(u,p(u))}{\sqrt{\mathsf{w}_{u,p(u)}}}|u,p(u)\rangle + \frac{\theta(u,c_1(u))}{\sqrt{\mathsf{w}_{u,c_1(u)}}}|u,c_1(u)\rangle + \frac{\theta(u,c_2(u))}{\sqrt{\mathsf{w}_{u,c_2(u)}}}|u,c_2(u)\rangle\right).$$

Suppose $\ell \in \{1,\dots,n/2\}$, so u is in the left tree. Then $(p(u),u) \in E_{2\ell}$, so we have

$$\theta(u,p(u)) = -\theta(p(u),u) = -\frac{1}{|E_{2\ell}|} = -2^{-2\ell} \text{ and } \sqrt{\mathsf{w}_{u,p(u)}} = \sqrt{\mathsf{w}_{2\ell}} = 2^{-\lceil 2\ell/2\rceil} = 2^{-\ell}$$

by (32), and for $i \in \{1,2\}$, $(u,c_i(u)) \in E_{2\ell+1}$, so we have

$$\theta(u,c_i(u)) = \frac{1}{|E_{2\ell+1}|} = 2^{-(2\ell+1)} \text{ and } \sqrt{\mathsf{w}_{u,c_i(u)}} = \sqrt{w_{2\ell+1}} = 2^{-\lceil (2\ell+1)/2\rceil} = 2^{-(\ell+1)}$$

also by (32). Thus:

$$\begin{aligned}|w_u\rangle &= \pm\left(-\frac{-2^{-2\ell}}{2^{-\ell}}|u,p(u)\rangle + \frac{2^{-(2\ell+1)}}{2^{-(\ell+1)}}|u,c_1(u)\rangle + \frac{2^{-(2\ell+1)}}{2^{-(\ell+1)}}|u,c_2(u)\rangle\right)\\ &= \pm 2^{-\ell}\left(|u,p(u)\rangle + |u,c_1(u)\rangle + |u,c_2(u)\rangle\right).\end{aligned}$$

On the other hand, if $\ell \in \{n/2+1,\dots,n\}$, so that u is in the right tree, we have $(u,p(u)) \in E_{2\ell+1}$, so:

$$\theta(u,p(u)) = \frac{1}{|E_{2\ell+1}|} = 2^{-(2n+2-2\ell-1)} \text{ and } \sqrt{\mathsf{w}_{u,p(u)}} = \sqrt{\mathsf{w}_{2\ell+1}} = 2^{-(n+2-\lceil (2\ell+1)/2\rceil)} = 2^{-(n+1-\ell)},$$

and for $i \in \{1,2\}$, $(c_i(u),u) \in E_{2\ell}$, so:

$$\theta(u,c_i(u)) = -\theta(c_i(u),u) = -\frac{1}{|E_{2\ell}|} = -2^{-(2n+2-2\ell)} \text{ and } \sqrt{\mathsf{w}_{u,c_i(u)}} = \sqrt{\mathsf{w}_{2\ell}} = 2^{-(n+2-\lceil \frac{2\ell}{2}\rceil)} = 2^{-(n+2-\ell)}.$$

Thus

$$\begin{aligned}|w_u\rangle &= \pm\left(-\frac{2^{-(2n+1-2\ell)}}{2^{-(n+1-\ell)}}|u,p(u)\rangle + \frac{-2^{-(2n+2-2\ell)}}{2^{-(n+2-\ell)}}|u,c_1(u)\rangle + \frac{-2^{-(2n+2-2\ell)}}{2^{-(n+2-\ell)}}|u,c_2(u)\rangle\right)\\ &= \mp 2^{n-\ell}\left(|u,p(u)\rangle + |u,c_1(u)\rangle + |u,c_2(u)\rangle\right).\end{aligned}$$

Thus, letting $\{v_1,v_2,v_3\} = \{p(u),c_1(u),c_2(u)\}$ with $v_1 < v_2 < v_3$, for any $\ell \in \{1,\dots,n\}$, if $u \in V_{2\ell}$, we have: $|w_u\rangle \propto |u,v_1\rangle + |u,v_2\rangle + |u,v_3\rangle$. ■

Then we have the following.

LEMMA 4.7. *Let $\mathsf{w}_M = \mathsf{w}_0$. Suppose $M = \{t\}$, and let $|w\rangle$ be as defined in (41) with respect to the flow defined in (42). Then $|w\rangle$ is a 0-positive witness (see Definition 3.5) with:*

$$\frac{\||w\rangle\|^2}{|\langle w|\psi_0\rangle|^2} = O(\mathsf{w}_0 n).$$

PROOF. To show that $|w\rangle$ is a 0-positive witness, we must show that it is orthogonal to all states in $\Psi^{\mathcal{A}} \cup \Psi^{\mathcal{B}}$. For $(u,v) \in \overrightarrow{E}(G)$, it is clear from the definition of $|w\rangle$, and the definition of $|\psi_{u,v}\rangle = |u,v\rangle - |v,u\rangle$ (see (38)) that $\langle w|\psi_{u,v}\rangle = 0$.

We next check that $\langle w|\psi_\star^{G'}(u)\rangle = 0$ for all $u \in V(G)$, which follows from the fact that θ is a circulation on G'. First, suppose $u \in V(G) \setminus \{s,t\}$:

$$\begin{aligned}
\langle w|\psi_\star^{G'}(u)\rangle &= \sum_{(u',v')\in\overrightarrow{E}(G)} \frac{\theta(u',v')}{\sqrt{\mathsf{w}_{u',v'}}}(\langle u',v'| + \langle v',u'|) \sum_{v\in\Gamma(u)} \sqrt{\mathsf{w}_{u,v}}(-1)^{\Delta_{u,v}}|u,v\rangle \qquad \text{see (34)} \\
&= \sum_{v\in\Gamma^+(u)} \theta(u,v)(-1)^{\Delta_{u,v}}\langle u,v|u,v\rangle + \sum_{v\in\Gamma^-(u)} \theta(v,u)(-1)^{\Delta_{u,v}}\langle u,v|u,v\rangle \\
&= \sum_{v\in\Gamma^+(u)} \theta(u,v) + \sum_{v\in\Gamma^-(u)} (-\theta(u,v))(-1) = \sum_{v\in\Gamma(u)} \theta(u,v),
\end{aligned}$$

where we used the fact that $(-1)^{\Delta_{u,v}} = 1$ when $v \in \Gamma^+(u)$ and (-1) if $v \in \Gamma^-(u)$, and the fact that $\theta(v,u) = -\theta(u,v)$. This is 0 whenever θ is a circulation (see Definition 2.2), so we now simply check that θ, as defined, is a circulation (at least on vertices other than s and t). Suppose $u \in V_k$ for some $k \in \{1,\dots,n\}$. Then u has three neighbours: a parent $p(u) \in V_{k-1}$, and two children $c_1(u), c_2(u) \in V_{k+1}$. We have

$$\theta(u,p(u)) = -\theta(p(u),u) = -\frac{1}{|E_k|}, \text{ and } \theta(u,c_1(u)) = \theta(u,c_2(u)) = \frac{1}{|E_{k+1}|} = \frac{1}{2|E_k|},$$

and thus

$$\theta(u,p(u)) + \theta(u,c_1(u)) + \theta(u,c_2(u)) = 0.$$

The case for $k \in \{n+1,\dots,2n\}$ is nearly identical. We still need to check orthogonality with $|\psi_\star^{G'}(s)\rangle$ and $|\psi_\star^{G'}(t)\rangle$. Suppose t has children u_1 and u_2. Then for $i \in \{1,2\}$, $(u_i,t) \in E_{2n+1}$, so since $2n+1 = 1 \mod 4$ (we are assuming n is even), we have $(u_i,t) \in \overrightarrow{E}(G)$ (see (31)). Thus, referring to (37),

$$\begin{aligned}
\langle w|\psi_\star^{G'}(u)\rangle &= \langle w|\left(\sqrt{\mathsf{w}_{\mathsf{M}}}|t,v_0\rangle - \frac{1}{2}|t,u_1\rangle - \frac{1}{2}|t,u_2\rangle\right) \\
&= \langle t,v_0|t,v_0\rangle - \sum_{(u,v)\in\overrightarrow{E}(G)} \frac{\theta(u,v)}{\sqrt{\mathsf{w}_{u,v}}}(\langle u,v| + \langle v,u|)\left(\frac{1}{2}|t,u_1\rangle + \frac{1}{2}|t,u_2\rangle\right) \\
&= 1 - \frac{\theta(u_1,t)}{\sqrt{\mathsf{w}_{u_1,t}}}\frac{1}{2}\langle t,u_1|t,u_1\rangle - \frac{\theta(u_2,t)}{\sqrt{\mathsf{w}_{u_2,t}}}\frac{1}{2}\langle t,u_2|t,u_2\rangle \\
&= 1 - \frac{1/|E_{2n+1}|}{\sqrt{\mathsf{w}_{2n+1}}}\frac{1}{2} - \frac{1/|E_{2n+1}|}{\sqrt{\mathsf{w}_{2n+1}}}\frac{1}{2} = 1 - \frac{1/2}{\sqrt{2^{-2(n+2-\lceil(2n+1)/2\rceil)}}} = 0
\end{aligned}$$

by (32). This is also simply following from the fact that θ is a circulation. A nearly identical argument works for $|\psi_\star^{G'}(s)\rangle$.

It Thus, only remains to show orthogonality of $|w\rangle$ with the states of $\Psi^{\mathcal{A}}$ that are not star states of G'. The only such states are those in (36) (some of which are also star states of G'). By Claim 4.3, it is sufficient to show orthogonality with the states $|\widehat{\psi^j}(u)\rangle$, for $j \in \{1,2\}$ and

$u \in V_{\text{even}} \setminus \{s\}$. Then letting $v_1 < v_2 < v_3$ be the neighbours of u, and appealing to Claim 4.6:

$$\sqrt{3}\langle w|\psi^i_\star(u)\rangle = \langle w_u| \left(|u,v_1\rangle + \omega_3^j|u,v_2\rangle + \omega_3^{2j}|u,v_3\rangle\right)$$
$$\propto (\langle u,v_1| + \langle u,v_2| + \langle u,v_3|) \left(|u,v_1\rangle + \omega_3^j|u,v_2\rangle + \omega_3^{2j}|u,v_3\rangle\right) \propto 1 + \omega_3^j + \omega_3^{2j} = 0.$$

Since we can also immediately see that: $|\langle w|\psi_0\rangle|^2 = 1/\mathsf{w}_0$, $|w\rangle$ is a positive witness. To complete the analysis of its complexity, we have, using $\mathsf{w}_{\mathsf{M}} = \mathsf{w}_0$, and the fact that all edges in E_k have the same weight, w_k (see (32)), and flow, $\frac{1}{|E_k|}$:

$$\||w\rangle\|^2 = \frac{1}{\mathsf{w}_0} + 2 \sum_{(u,v)\in\overrightarrow{E}(G)} \frac{\theta(u,v)^2}{\mathsf{w}_{u,v}} + \frac{1}{\mathsf{w}_{\mathsf{M}}} = \frac{2}{\mathsf{w}_0} + 2\sum_{k=1}^{2n+1} |E_k| \frac{1}{|E_k|^2}\frac{1}{\mathsf{w}_k}$$
$$= \frac{2}{\mathsf{w}_0} + 2\sum_{k=1}^{n} \frac{1}{2^k}\frac{1}{2^{-2\lceil k/2\rceil}} + 2\sum_{k=n+1}^{2n+1} \frac{1}{2^{2n+2-k}}\frac{1}{2^{-2(n+2-\lceil k/2\rceil)}} = \frac{2}{\mathsf{w}_0} + O(n). \qquad \blacksquare$$

REMARK 4.8. The reader may wonder why the weights change by a factor of 4 every two layers, rather than by a factor of 2 every layer. If we set all the weights to 1, the positive witness size is constant, while the negative witness size is exponential. If we change weights by a factor of two at each layer, the negative witness size is constant, whereas the positive witness size is exponential. With the setting of weights that we have chosen, both witness sizes are linear in n (up to scaling by w_0). This setting of weights and edge directions creates a perfect duality between positive and negative witnesses. For vertices $u \in V_{\text{odd}}$, we include the star state, which is proportional to $|\widehat{\psi}^0(u)\rangle$ (see Claim 4.3) in $\Psi^{\mathcal{A}}$, so the flow through u must be in $\text{span}\{|\widehat{\psi}^1(u)\rangle, |\widehat{\psi}^2(u)\rangle\}$. Conversely, for vertices $u \in V_{\text{even}}$, we include $\text{span}\{|\widehat{\psi}^1(u)\rangle, |\widehat{\psi}^2(u)\rangle\}$ in $\Psi^{\mathcal{A}}$, so the flow through u must be proportional to $|\widehat{\psi}^0(u)\rangle$.

Conclusion of Proof: We now apply Theorem 3.8 to conclude the proof of Theorem 4.1. By Lemma 4.7, there is some constant c such that setting $\mathsf{w}_0 = 1/(cn)$, whenever $M = \{t\}$, there exists a positive witness $|w\rangle$ with

$$\frac{\||w\rangle\|^2}{|\langle w|\psi_0\rangle|^2} \leq c_+ := 50.$$

Then by Lemma 4.5, there is some

$$C_- = O(n/\mathsf{w}_0) = O(n^2)$$

such that whenever $M = \emptyset$, there exists a negative witness with $\||w_{\mathcal{A}}\rangle\|^2 \leq C_-$. Then since the initial state can be prepared in $\mathsf{S}_q = 0$ queries and $\mathsf{S} = O(n)$ time, and by Lemma 4.4, the unitary can be implemented in $O(1)$ query to O_G, and $O(n)$ time, the phase estimation algorithm distinguishes between the cases $M = \emptyset$ and $M = \{t\}$ in

$$O\left(0 + \sqrt{C_-}\right) = O(n) \text{ and } O\left(n + \sqrt{C_-}n\right) = O(n^2)$$

queries and time respectively.

Removing the Assumption that $u \in V_{\text{even}}$ can be Checked: We do not actually require an extra assumption that the algorithm can efficiently check, for a vertex u, if it is in V_{even} or V_{odd}. Intuitively, this is because if a walker starts at u, she can always keep track of the parity of the distance from u, by keeping track of a bit that is initially 0, and flips every time she takes a step. More precisely, we can define a graph G_0 as follows:

$$V(G_0) = V_{\text{even}} \times \{0\} \cup V_{\text{odd}} \times \{1\}$$

$$E(G_0) = \{\{(u,0),(v,1)\} : \{u,v\} \in E(G), u \in V_{\text{even}}\},$$

so that a walk on G_0 is like a walk on G, except that there is a bit indicating which of the two independent sets we are in, which we flip at every step. To find the neighbours of any vertex (u,b), simply query O_G and append $b \oplus 1$ to each of the three returned strings. We let $(s,0)$ and $(t,1)$, which are both in $V(G_0)$, take the places of s and t.

5. k-Distinctness

Fix any constant k. Formally, k-distinctness is defined as follows. Given an input $x \in [q]^n$, for some $q \in \text{poly}(n)$, decide if there exist distinct $a_1, \ldots, a_k \in [n]$ such that $x_{a_1} = \cdots = x_{a_k}$, called a k-collision. A search version of this problem asks that the algorithm find a k-collision if one exists. The search and decision versions are equivalent up to log factors, so we focus on the decision version. The main result of this section is a quantum algorithm that solves k-distinctness in $\widetilde{O}(n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^k-1}})$ time complexity (see Theorem 5.16) for any $k \geq 3$, which is a new result for $k > 3$. As a warm-up, we describe the $k = 3$ case of our algorithm in Section 5.2, before giving the full algorithm in Section 5.3. First, we describe some assumptions on the structure of the input in Section 5.1.

5.1 Assumptions on the Input

We assume that either there is no k-collision, or there is a unique k-collision, $a_1, \ldots, a_k \in [n]$. This is justified by the following lemma, which follows from [4, Section 5].

LEMMA 5.1. *Fix constants $k \geq 2$ and $\lambda \in [1/2, 1)$. Let $\mathcal{A}$ be an algorithm that decides k-distinctness in bounded error with complexity $\widetilde{O}(n^\lambda)$ when there is at most one k-collision. Then there is an algorithm $\mathcal{A}'$ that decides k-distinctness (in the general case) in bounded error in complexity $\widetilde{O}(n^\lambda)$.*

This fact has been exploited in nearly every quantum algorithm for k-distinctness. Another standard trick is to assume that $[n]$ is partitioned as:

$$[n] = A_1 \cup \cdots \cup A_k$$

such that the unique k-collision $(a_1, \ldots, a_k)$, (if it exists) is in $A_1 \times \cdots \times A_k$. Towards fixing **Problem 1** from Section 1.3, we further partition each of $A_2, \ldots, A_{k-1}$ as

$$A_\ell = A_\ell^{(1)} \cup \cdots \cup A_\ell^{(m_\ell)}$$

for some m_ℓ. We will choose these partitions as follows. Fix a d-wise independent permutation $\tau : [n] \to [n]$, for $d = \log^{2^{k-1}}(n)$ that is both efficiently computable, and efficiently invertible (see Definition 2.10 and the discussion below). For $\ell \in [k]$, define:

$$A_\ell = \{\tau(i) : i \in \{(\ell-1)n/k + 1, \ldots, \ell n/k\}\}$$

and for $j \in [m_\ell]$, define:

$$A_\ell^{(j)} = \left\{\tau(i) : i \in \left\{(\ell-1)n/k + (j-1)\frac{n}{km_\ell} + 1, \ldots, (\ell-1)n/k + j\frac{n}{km_\ell}\right\}\right\}. \tag{43}$$

Then we will make use of the following facts:

LEMMA 5.2. *1. For any $i \in [n]$, we can check to which $A_\ell^{(j)}$ it belongs in* polylog(n) *complexity.*

2. *For any $\ell \in [k]$, we can generate a uniform superposition over A_ℓ, and for any $j \in [m_\ell]$, we can generate a uniform superposition over $A_\ell^{(j)}$, in* polylog(n) *complexity.*
3. $\Pr[a_1 \in A_1, \ldots, a_k \in A_k] = \Omega(1)$.

PROOF. Since $d \in$ polylog(n), we can assume (see discussion below Definition 2.10) that both τ and τ^{-1} can be computed in polylog(n) complexity. Then for Item 1, it is enough to compute $\tau^{-1}(i)$.

For Item 2, we describe how to perform a superposition over $\{\tau(i) : i \in \{\ell, \ldots, r\}\}$ for any integers $\ell < r$. First generate the uniform superposition over the set $\{\ell, \ldots, r\}$, and compute τ in a new register, to get (up to normalization) $\sum_{i=\ell}^{r} |i\rangle|\tau(i)\rangle$. Then uncompute the first register by computing τ^{-1} of the second register and adding it (bitwise, mod 2) into the first.

Finally, Item 3 follows from the d-wise independence of τ, since $d > k$. ∎

For any disjoint subsets of $[n]$, $S_1, \ldots, S_\ell$, define:

$$\mathcal{K}(S_1, \ldots, S_\ell) = \{(i_1, \ldots, i_\ell) \in S_1 \times \cdots \times S_\ell : x_{i_1} = \cdots = x_{i_\ell}\}. \tag{44}$$

Then without loss of generality, we can assume that for each $A_j^{(\ell)}$, $\mathcal{K}(A_1, \ldots, A_{j-1}, A_j^{(\ell)}) = \Theta(|A_j^{(\ell)}|)$, because we can simply pad the input with $\Theta(n)$ extra $(k-1)$-collisions, evenly spread across the blocks.

5.2 Warm-up: 3-Distinctness Algorithm

In this section, we prove the following upper bound on the time complexity of 3-distinctness.

THEOREM 5.3. *There is a quantum algorithm that decides 3-distinctness with bounded error in $\widetilde{O}(n^{5/7})$ complexity.*

This upper bound is not new, having been proven in [12], but its proof in our new framework is a useful warm-up for Section 5.3, where we generalise the algorithm to all constants $k > 3$. Throughout this section, $\widetilde{O}$ will surpress polylogarithmic factors in n.

Our algorithm will roughly follow the one described in Section 1.3, but with the modifications, also briefly mentioned in Section 1.3, needed to circumvent the problems with the approach, for which we need our new Mutltidimensional Quantum Walk Framework, Theorem 3.10. We start by setting up these modifications, before formally defining the graph that will be the basis for our quantum walk algorithm, and then performing the necessary analysis to apply Theorem 3.10.

Recall from Section 1.3 that the basic idea of our quantum walk algorithm is to walk on sets $R = (R_1, R_2)$ where $R_1 \subset A_1$ and $R_2 \subset A_2$.

Towards Fixing Problem 1: The first problem identified in Section 1.3 is that $|R_2|$ is larger than the total time we would like our algorithm to spend, meaning we do not want to spend $|R_2|$ steps sampling and writing down the set R_2. To this end, we have partitioned A_2 into equal sized blocks:

$$A_2 = A_2^{(1)} \cup \cdots \cup A_2^{(m_2)},$$

(see Section 5.1 for details of how this partition is chosen). We redefine R_2 as follows: whenever we want to choose a subset of A_2, we do so by selecting $R_2 \subset [m_2]$, which encodes the subset of A_2:

$$\overline{R}_2 := \bigcup_{j \in R_2} A_2^{(j)}.$$

We choose m_2 so that $|A_2^{(j)}| = \frac{n}{3m_2}$ is large enough so that for a random set R_1 of size r_1, the expected size of $\mathcal{K}(R_1, A_2^{(j)})$ is constant, so we set $m_2 = \Theta(r_1)$. Finally, we choose $t_2 = |R_2|$ so that $|\overline{R}_2| = t_2 \frac{n}{3m_2}$ is the desired size of R_2 (denoted r_2 in [9]) and for consistency also define $t_1 = r_1$. We will find that the optimal parameter settings are $t_1 = n^{5/7}$ and $t_2 = n^{4/7}$ (so $m_2 = \Theta(n^{5/7})$).

Towards Fixing Problem 2: In order to solve the second problem discussed in Section 1.3, following a similar construction in [9], each of R_1 and R_2 will be a *tuple* of disjoint sets, as follows. We have $R_1 = (R_1(\{1\}), R_1(\{2\}), R_1(\{1,2\}))$ where $R_1(\{1\})$, $R_1(\{2\})$, and $R_1(\{1,2\})$ are disjoint subsets of A_1 of size t_1; and $R_2 = (R_2(1), R_2(2))$, where $R_2(1)$ and $R_2(2)$ are disjoint subsets of

$[m_2]$ of size t_2 (note that this alters $|R_1|$ and $|R_2|$ by a constant factor), meaning for $s \in \{1,2\}$,

$$\overline{R}_2(s) := \bigcup_{j \in R_2(s)} A_2^{(j)}$$

are disjoint subsets of A_2 of size $t_2 \frac{n}{3m_2}$. We also use R_1 and R_2 to denote the union of sets in the tuple, so for example, $j \in \overline{R}_2$ means $j \in \overline{R}_2(1) \cup \overline{R}_2(2)$. For a vertex labelled by $R = (R_1, R_2)$, we maintain *data* with the following components. We query everything in R_1, so for $S \in 2^{\{1,2\}} \setminus \emptyset$, we define:

$$D_1(R_1(S)) := \{(i_1, x_{i_1}) : i_1 \in R_1(S)\}$$
$$D_1(R) := (D_1(R_1(\{1\})), D_1(R_1(\{2\})), D_1(R_1(\{1,2\})))$$

and for $s \in \{1,2\}$ define

$$D_2(R_2(s)|R_1) := \bigcup_{S \subseteq \{1,2\} : s \in S} \{(i_1, i_2, x_{i_1}) : i_2 \in \overline{R}_2(s), i_1 \in R_1(S), x_{i_1} = x_{i_2}\} \tag{45}$$
$$D_2(R) := (D_2(R_2(1)|R_1), D_2(R_2(2)|R_1)).$$

Finally we let

$$D(R) := (D_1(R), D_2(R)).$$

So to summarise, we query everything in R_1, but we only query those things in $\overline{R}_2$ that have a collision in R_1, and even then, not in every case: if $i_2 \in R_2(s)$, we only query it if it has a collision with $R_1(\{s\})$ or $R_1(\{1,2\})$ (see Figure 2). This partially solves **Problem 2**, because it ensures that if we choose to add a new index i_1 to R_1, we have three choices of where to add it, and either all of those choices are fine (they don't introduce a *fault* in $D_2(R)$), or exactly one of them is fine.

For a finite set $\mathcal{S}$, and positive integers r and ℓ, we will use the notation

$$\binom{\mathcal{S}}{r^{(\ell)}} := \binom{\mathcal{S}}{\underbrace{r, \ldots, r}_{\ell \text{ times}}} \tag{46}$$

to denote the set of all ℓ-tuples of disjoint subsets of $\mathcal{S}$, each of size r. Finally, we define:

$$\binom{\mathcal{S}}{r^{(\ell)}}^+ := \bigcup_{\ell'=1}^{\ell} \left(\binom{\mathcal{S}}{r^{(\ell'-1)}, r+1, r^{(\ell-\ell')}} \right), \tag{47}$$

to be the set of all ℓ-tuples of disjoint sets of $\mathcal{S}$ such that exactly one of the sets has size $r+1$, and all others have size r. We let $\mu(S)$ denote the smallest element of S.

5.2.1 The Graph G

We now define G, by defining disjoint vertex sets $V_0, V_0^+, V_1, V_2, V_3$ whose union will make up $V(G)$, as well as the edges between adjacent sets.

V_0: We first define

$$V_0 := \left\{ v^0_{R_1,R_2} = (0, R_1, R_2, D(R_1,R_2)) : (R_1,R_2) \in \binom{A_1}{t_1^{(3)}} \times \binom{[m_2]}{t_2^{(2)}} \right\} \tag{48}$$

on which the initial distribution will be uniform: $\sigma(v^0_{R_1,R_2}) = \frac{1}{|V_0|}$. We implicitly store all sets including R_1, R_2 and $D(R_1,R_2)$ in a data structure with the properties described in Section 2.3. This will only be important when we analyse the time complexity of the setup and transition subroutines.

V_0^+ and $E_0^+ \subset V_0 \times V_0^+$: Next, each vertex in V_0^+ will be labeled by a vertex in V_0, along with an index $i_1 \notin R_1$ that we have decided to add to one of $R_1(\{1\})$, $R_1(\{1,2\})$ or $R_1(\{2\})$. We have not yet decided to which of the three sets it will be added, nor added it.

$$V_0^+ := \left\{ v^0_{R_1,R_2,i_1} := ((0,+), R_1, R_2, D(R_1,R_2), i_1) : v^0_{R_1,R_2} \in V_0, i_1 \in A_1 \setminus R_1 \right\},$$
$$\text{so } |V_0^+| = |V_0|(n/3 - 3t_1). \tag{49}$$

There is an edge between $v^0_R \in V_0$ and $v^0_{R,i_1} \in V_0^+$ for any $i_1 \in A_1 \setminus R_1$, and for any $v^0_{R,i_1} \in V_0^+$, $v^0_R \in V_0$ is its unique in-neighbour, so we define edge label sets (see Definition 2.3)

$$L^+(v^0_R) := A_1 \setminus R_1 \quad \text{and} \quad L^-(v^0_{R,i_1}) := \{\leftarrow\},$$

and let $f^+_{v^0_R}(i_1) = v^0_{R,i_1}$, and $f^-_{v^0_{R,i_1}}(\leftarrow) = v^0_R$. Here we have added the superscript + (respectively −) to denote the restriction of $f_{v^0_R}$ to $L^+(v^0_R)$ (respectively $L^-(v^0_{R,i_1})$). By writing $f^+_{v^0_R}(i_1)$ and $f^-_{v^0_{R,i_1}}(\leftarrow)$, we emphasise that the index i_1 is an element of $L^+(v^0_R)$ and the index $\leftarrow$ is an element of $L^-(v^0_{R,i_1})$. We stick to this notation convention for the rest of the section.

We let E_0^+ be the set of all edges,

$$E_0^+ := \left\{ \left(v^0_R, v^0_{R,i_1}\right) : v^0_R \in V_0, i_1 \in A_1 \setminus R_1 \right\},$$

and set $\mathsf{w}_e = \mathsf{w}_0^+ = 1$ for all $e \in E_0^+$. This together with (49) implies that

$$|E_0^+| = |V_0^+| = |V_0|(n/3 - 3t_1). \tag{50}$$

Note that we break the move from V_0 to V_1, where we add some i_1 to one of the sets $R_1(\{1\})$, $R_1(\{2\})$ or $R_1(\{1,2\})$, into two steps: First we select an index i_1 to add – that's the step we have just described, from V_0 to V_0^+. Next, we choose one of the three sets and add i_1 there – that's the step we are about to describe, from V_0^+ to V_1. The reason we do this in two steps is that we will use the alternative neighbourhoods trick to ensure we can efficiently implement the second step, only adding i_1 to a set where it won't cause a fault, despite not being able to efficiently decide which sets these are. It is useful to have this somewhat more complicated-to-implement part of the walk isolated, in vertices of constant degree, as the vertices of V_0^+ will be. Note that in defining the graph, as we are currently doing, this complication does not appear, except that the

reader may notice that it looks difficult to implement a step of the walk from V_0^+ – it is indeed more complicated, requiring the use of alternative neighbourhoods later. Let us continue with the description of the graph.

V_1 and $E_1 \subset V_0^+ \times V_1$: Continuing, vertices in V_1 represent having added an additional index to R_1, so we define:

$$V_1(S) := \left\{ v^1_{R_1,R_2} = (1, R_1, R_2, D(R_1,R_2)) : (R_1,R_2) \in \binom{A_1}{t_1^{(3)}}^+ \times \binom{[m_2]}{t_2^{(2)}}, |R_1(S)| = t_1 + 1 \right\},$$

$$V_1 := \bigcup_{S \in 2^{\{1,2\}} \setminus \{\emptyset\}} V_1(S)$$

$$\text{so } |V_1| = 3\binom{n/3}{t_1+1, t_1, t_1}\binom{m_2}{t_2,t_2} = 3\frac{n/3 - 3t_1}{t_1+1}\binom{n/3}{t_1,t_1,t_1}\binom{m_2}{t_2,t_2} = \frac{n-9t_1}{t_1+1}|V_0|. \tag{51}$$

For a vertex $v^0_{R,i_1} \in V_0^+$ we have chosen an index i_1 to add to R_1, but we have not yet decided to which part of R_1 it should be added. A transition to a vertex in V_1 consists of choosing an $S \in 2^{\{1,2\}} \setminus \{\emptyset\}$ and adding i_1 to $R_1(S)$, so

$$L^+(v^0_{R,i_1}) := 2^{\{1,2\}} \setminus \{\emptyset\},$$

and $f^+_{v^0_{R,i_1}}(S) = v^1_{R'}$, where R' is obtained from R by inserting i_1 into $R_1(S)$. Note that not all of these labels represent edges with non-zero weight, as we want to ensure that adding i_1 to $R_1(S)$ does not introduce a *fault*, meaning that adding i_1 to $R_1(S)$ should not require that any collision involving i_1 be added to $D_2(R)$.

Viewing transitions in E_1 from the other direction, a vertex $v^1_{R'} \in V_1(S)$ is connected to a vertex $v^0_{R,i_1} \in V_0^+$ if we can obtain R from R' by removing i_1 from $R'_1(S)$, and if doing so does not require an update to $D_2(R')$, meaning there do not exist any $s \in S$ and $i_2 \in \overline{R}'_2(s)$ such that $x_{i_1} = x_{i_2}$. So for any $v^1_{R'} \in V_1(S)$, we let

$$\begin{aligned} L^-(v^1_{R'}) :&= \{i_1 \in R'_1(S) : \nexists s \in S, i_2 \in \overline{R}'_2(s) \text{ s.t. } x_{i_1} = x_{i_2}\} \\ &= \{i_1 \in R'_1(S) : \nexists i_2 \text{ s.t. } (i_1, i_2, x_{i_1}) \in D_2(R')\}, \end{aligned} \tag{52}$$

and $f^-_{v^1_{R'}}(i_1) = v^0_{R'\setminus\{i_1\},i_1}$. It is currently not clear how to define E_1, the set of (non-zero weight) edges between V_0^+ and V_1, because $|V_0^+| \cdot |L^+(v^0_{R,i_1})| > |V_1| \cdot |L^-(v^1_{R'})|$, so in particular, we cannot assign nonzero weights $\mathsf{w}_{u,i}$ to all $u \in V_0^+$, $i \in L^+(u)$, because that would make E_1 larger than we have labels L^- for. We will instead assign non-zero weights $\mathsf{w}_{v,j}$ to those edges where $v \in V_1$ and $j \in L^-(v)$. That is, define:

$$E_1 := \left\{ \left(f^-_{v^1_{R'}}(i_1), v^1_{R'} \right) = \left(v^0_{R'\setminus\{i_1\},i_1}, v^1_{R'} \right) : v^1_{R'} \in V_1, i_1 \in L^-(v^1_{R'}) \right\}$$

and give weight $\mathsf{w}_e = \mathsf{w}_1 = 1$ to all $e \in E_1$. This means that for $u = v^0_{R,i_1} \in V_0^+$, there are some $S \in 2^{\{1,2\}} \setminus \{\emptyset\}$ with $\mathsf{w}_{u,S} = 0$ – namely those with $f_v^{-1}(u) \notin L^-(v)$ for $v = f_u(S)$. To investigate

which S this applies to, we introduce for all $v^0_{R,i_1} \in V_0^+$:

$$\mathcal{I}(v^0_{R,i_1}) := \left\{ s \in \{1,2\} : \exists i_2 \in \overline{R}_2(s) \text{ s.t. } x_{i_2} = x_{i_1} \right\}, \tag{53}$$

so $\mathcal{I}(v^0_{R,i_1})$ consists of those $s \in \{1,2\}$ where a fault occurs if i_1 is added to $R_1(S)$ such that $s \in S$. Note that since we assume the unique 3-collision has a part in A_3, i_1 can have at most one colliding element in $\overline{R}_2$, and so it cannot be in both $\overline{R}_2(1)$ and $\overline{R}_2(2)$, which are disjoint. Thus, $\mathcal{I}(v^0_{R,i_1}) \subsetneq \{1,2\}$ – so it is $\emptyset$, $\{1\}$, or $\{2\}$ (this heavy-handed notation is overkill here, but we are warming up for k-distinctness, where it is necessary). We now have the following:

LEMMA 5.4. *Let $R^{S\leftarrow i_1}$ be obtained from R by inserting i_1 into $R_1(S)$. Then*

$$E_1 = \left\{ \left(v^0_{R,i_1}, v^1_{R^{S\leftarrow i_1}} \right) : v^0_{R,i_1} \in V_0^+, S \in 2^{\{1,2\}\setminus\mathcal{I}(v^0_{R,i_1})} \setminus \{\emptyset\} \right\}.$$

So for all $v^0_{R,i_1} \in V_0^+$, and $S \in L^+(v^0_{R,i_1})$, $\mathsf{w}_{v^0_{R,i_1},S} = \begin{cases} \mathsf{w}_1 = 1 & \text{if } S \cap \mathcal{I}(v^0_{R,i_1}) = \emptyset \\ 0 & \text{else.} \end{cases}$

PROOF. Let E_1' be the right-hand side of the identity in the theorem statement, so we want to show $E_1 = E_1'$. Fix any $v^0_{R,i_1} \in V_0^+$ and $S \in 2^{\{1,2\}\setminus\mathcal{I}(v^0_{R,i_1})} \setminus \{\emptyset\}$, and let $R' = R^{S\leftarrow i_1}$. Then since $S \cap \mathcal{I}(v_{R,i_1}) = \emptyset$, by definition of $\mathcal{I}(v_{R,i_1})$ there does not exist any $s \in S$ and $i_2 \in \overline{R}_2'(s)$ such that $x_{i_1} = x_{i_2}$. Hence, $L^-(v^1_{R'})$, which implies $E_1' \subseteq E_1$.

For the other direction, fix any $v^1_{R'} \in V_1(S)$ and $i_1 \in L^-(v^1_{R'})$. Since $i_1 \in R_1'(S)$, we have $v^0_{R'\setminus\{i_1\},i_1} \in V_0^+$ and $(R'\setminus\{i_1\})^{S\leftarrow i_1} = R'$. Since by definition of $L^-(v^1_{R'})$ there does not exist $s \in S$ and $i_2 \in \overline{R}_2'(s)$ such that $x_{i_1} = x_{i_2}$, we immediately have $S \cap \mathcal{I}(v_{R'\setminus\{i_1\},i_1}) = \emptyset$. This implies $E_1 \subseteq E_1'$. ■

From (49) we now have:

$$|E_1| \leq 3|V_0^+| = 3|V_0|(n/3 - 3t_1). \tag{54}$$

V_2 and $E_2 \subset V_1 \times V_2$: Vertices $v^1_R \in V_1(S)$ represent having added an additional index i_1 to $R_1(S)$, so $|R_1(S)| = t_1 + 1$. A vertex $v^2_{R_1,R_2'} \in V_2$ is adjacent to v^1_R if R_2' is obtained from R_2 by adding $j_2 \notin R_2$ to $R_2(s)$ for some choice of $s \in \{1,2\}$. We will not let this choice of s be arbitrary though and instead, in order to simplify things in the more complicated k-distinctness setting, we require that j_2 be added to $R_2(\mu(S))$, where $\mu(S)$ denotes the minimum element of S.

$$V_2(S) := \left\{ v^2_R = (2, R, D(R)) : R \in \binom{A_1}{t_1^{(3)}}^+ \times \binom{[m_2]}{t_2^{(2)}}^+, |R_1(S)| = t_1 + 1, |R_2(\mu(S))| = t_2 + 1 \right\},$$

$$V_2 := \bigcup_{S \in 2^{\{1,2\}}\setminus\{\emptyset\}} V_2(S).$$

This means that

$$|V_2| = 3\binom{n/3}{t_1+1, t_1, t_1}\binom{m_2}{t_2+1, t_2} = 3\frac{n/3 - 3t_1}{t_1+1}\binom{n/3}{t_1, t_1, t_1}\frac{m_2 - 2t_2}{t_2+1}\binom{m_2}{t_2, t_2} = O\left(\frac{nm_2}{t_1 t_2}|V_0|\right). \tag{55}$$

We move from $v^1_R \in V_1$ to $v^2_{R'} \in V_2$ by selecting some $j_2 \in [m_2] \setminus R_2$ to add to R_2; and from $v^2_{R'}$ to v^1_R by selecting some j_2 to remove from R_2, so for $v^1_R \in V_1(S)$ and $v^2_{R'} \in V_2(S)$, we let

$$L^+(v^1_R) := [m_2] \setminus R_2 \text{ and } L^-(v^2_{R'}) := R'_2(\mu(S)).$$

The sets $L^+(v^1_R)$ and $L^-(v^1_R)$ (defined in (52)) should be disjoint, but this does not appear to be the case. To ensure this, we implicitly append a label $\leftarrow$ to every label in $L^-(u)$ for any u, and $\rightarrow$ to every label in $L^+(u)$. We let $f^+_{v^1_R}(j_2) = v^2_{R_1,R_2^{\mu(S)\leftarrow j_2}}$ when $v^1_R \in V_1(S)$, and $f^-_{v^2_{R_1,R'_2}}(j_2) = v^1_{R_1,R'_2\setminus\{j_2\}}$. Accordingly we define $E_2(S)$ to be the set of all such edges:

$$\begin{aligned} E_2 :&= \bigcup_{S\in 2^{\{1,2\}}\setminus\{\emptyset\}} \{(v^1_R, v^2_{R_1,R_2^{\mu(S)\leftarrow j_2}}) : v^1_R \in V_1(S), j_2 \in [m_2]\setminus R_2\} \\ &= \bigcup_{S\in 2^{\{1,2\}}\setminus\{\emptyset\}} \left\{\left(v^1_{R_1,R'_2\setminus\{j_2\}}, v^2_{R_1,R'_2}\right) : v^2_{R_1,R'_2} \in V_2(S), j_2 \in R_2(\mu(S))\right\}. \end{aligned}$$

We set $\mathsf{w}_e = \mathsf{w}_2 = \sqrt{n/m_2}$ for all $e \in E_2$, and observe, using (51), that:

$$|E_2| = (m_2 - 2t_2)|V_1| = \frac{(m_2-2t_2)(n-9t_1)}{t_1+1}|V_0|. \tag{56}$$

The Final Stage: V_3 and E_3: The last stage is very simple, as every vertex in V_3 represents having added an additional index to each of R_1, R_2 and chosen some $i_3 \in A_3$:

$$V_3 := \{v^3_{R_1,R_2,i_3} = (3, R_1, R_2, D(R_1,R_2), i_3) : v^2_{R_1,R_2} \in V_2, i_3 \in A_3\}.$$

There is an edge between $v^2_R \in V_2$ and $v^3_{R,i_3} \in V_3$ for any $i_3 \in A_3$, and for any $v^3_{R,i_3} \in V_3$, v^2_R is its unique (in-)neighbour, so we define

$$L^+(v^2_R) := A_3 \text{ and } L(v^3_{R,i_3}) = L^-(v^3_{R,i_3}) := \{\leftarrow\},$$

and let $f^+_{v^2_R}(i_3) = v^3_{R,i_3}$, and $f^-_{v^3_{R,i_3}}(\leftarrow) = v^2_R$. We let E_3 be the set of all such edges,

$$E_3 = \left\{\left(v^2_R, v^3_{R,i_3}\right) : v^2_R \in V_2, i_3 \in A_3\right\},$$

and set $\mathsf{w}_e = \mathsf{w}_3 = 1$ for all $e \in E_3$. Then using (55) we observe

$$|E_3| = \frac{n}{3}|V_2| = O\left(\frac{n^2 m_2}{t_1 t_2}|V_0|\right). \tag{57}$$

The Graph G: The full graph G is defined by:

$$\begin{aligned} V(G) &= V_0 \cup V_0^+ \cup V_1 \cup V_2 \cup V_3 \\ \text{and } \overrightarrow{E}(G) &= \{(u,v) : u \in V(G), i \in L^+(u), \mathsf{w}_{u,i} \neq 0\} = E_0^+ \cup E_1 \cup E_2 \cup E_3, \end{aligned}$$

where the sets $L^+(u)$ are summarised in Table 1, and the condition under which $\mathsf{w}_{u,i} = 0$ can be found in Lemma 5.4. Non-zero edge weights are summarised in Table 2.

u	$j \in L^-(u)$	$f_u^-(j)$	$i \in L^+(u)$	$f_u^+(i)$
$v_R^0 \in V_0$	$\emptyset$		$i_1 \in A_1 \setminus R_1$	v_{R,i_1}^0
$v_{R,i_1}^0 \in V_0^+$	$\leftarrow$	v_R^0	$S \in 2^{\{1,2\}} \setminus \{\emptyset\}$	$v_{R^{S \leftarrow i_1}}^1$
$v_R^1 \in V_1(S)$	$i_1 \in R_1(S) : d_R^{\rightarrow}(i_1) = 0$	$v_{R\setminus\{i_1\},i_1}^0$	$j_2 \in [m_2] \setminus R_2$	$v_{R^{\mu(S)\leftarrow j_2}}^2$
$v_R^2 \in V_2(S)$	$j_2 \in R_2(\mu(S))$	$v_{R\setminus\{j_2\}}^1$	$i_3 \in A_3$	v_{R,i_3}^3
$v_{R,i_3}^3 \in V_3$	$\leftarrow$	v_R^2	$\emptyset$	

Table 1. A summary of the vertex sets and the labels of edges coming into and out of each vertex. Foreshadowing Section 5.3, we here define $d_R^{\rightarrow}(i_1)$ to be 0 if and only if there is no $i_2 \in \bigcup_{s \in S} R_2(s)$ such that $x_{i_1} = x_{i_2}$. Here $R^{\mu(S)\leftarrow j_2}$ is obtained from R by inserting j_2 into $R_2(\mu(S))$, where $\mu(S)$ is the minimum element of S. We remark that $L^-(u)$ and $L^+(u)$ should always be disjoint. To ensure that this holds, we implicitly append a $\leftarrow$ label to all of $L^-(u)$ and a $\rightarrow$ label to all of $L^+(u)$.

Edge set	Weights	Complexity
$E_0^+ \subset V_0 \times V_0^+$	$\mathsf{w}_0^+ = 1$	$\mathsf{T}_0^+ = \widetilde{O}(1)$
$E_1 \subset V_0^+ \times V_1$	$\mathsf{w}_1 = 1$	$\mathsf{T}_1 = \widetilde{O}(1)$
$E_2 \subset V_1 \times V_2$	$\mathsf{w}_2 = \sqrt{n/m_2}$	$\mathsf{T}_2 = \widetilde{O}(\sqrt{n/m_2})$
$E_3 \subset V_2 \times V_3$	$\mathsf{w}_3 = 1$	$\mathsf{T}_3 = \widetilde{O}(1)$

Table 2. A summary of the weights and complexities (see Section 5.2.3) of each edge set.

The Marked Set and Checking Cost: In the notation of Theorem 3.10, we let $V_{\mathsf{M}} = V_3$, and we will define a subset $M \subseteq V_3$ as follows. If $(a_1, a_2, a_3) \in A_1 \times A_2 \times A_3$ is the unique 3-collision (see Section 5.1), we let

$$M = \left\{ v_{R_1,R_2,i_3}^3 \in V_3 : \exists S \in 2^{\{1,2\}} \setminus \{\emptyset\}, \text{ s.t. } a_1 \in R_1(S), a_2 \in \overline{R}_2(\mu(S)), a_3 = i_3 \right\}, \tag{58}$$

and otherwise $M = \emptyset$. Recall that $v_{R_1,R_2,i_3}^3 = (3, R_1, R_2, D(R_1, R_2), i_3)$, where $D(R_1, R_2)$ includes $D_2(R)$, defined in (45), storing all pairs (i_1, i_2, x_{i_1}) such that $x_{i_1} = x_{i_2}$ and $\exists S \in 2^{\{1,2\}}$ and $s \in S$ with $i_1 \in R_1(S)$ and $i_2 \in R_2(s)$. Thus, we can decide if $v_{R_1,R_2,i_3}^3 \in V_3$ is marked by querying i_3 to obtain x_{i_3} and looking it up (see Section 2.3) in $D_2(R)$ to see if we find some (i_1, i_2, x_{i_3}), in which case, it must be that $a_1 = i_1$, $a_2 = i_2$ and $a_3 = i_3$. Thus, the checking cost is at most

$$\mathsf{C} = O(\log n). \tag{59}$$

5.2.2 The Star States and their Generation

We define a set of alternative neighbourhoods for G (see Definition 3.9). For all $u \in V(G) \setminus V_0^+$, we define $\Psi_\star(u) = \{|\psi_\star^G(u)\rangle\}$, which by Table 1 is equal to the following: for $u = v^0_{R_1,R_2} \in V_0$,

$$|\psi_\star^G(u)\rangle = \sum_{i_1 \in A_1 \setminus R_1} \sqrt{\mathsf{w}_0^+}|v^0_{R_1,R_2}, i_1\rangle; \tag{60}$$

for $u = v^1_{R_1,R_2} \in V_1(S)$,

$$|\psi_\star^G(u)\rangle = -\sum_{\substack{i_1 \in R_1(S):\\ \nexists i_2,\ (i_1,i_2,x_{i_1}) \in D_2(R)}} \sqrt{\mathsf{w}_1}|v^1_{R_1,R_2}, \leftarrow, i_1\rangle + \sum_{j_2 \in [m_2] \setminus R_2} \sqrt{\mathsf{w}_2}|v^1_{R_1,R_2}, \rightarrow, j_2\rangle; \tag{61}$$

for $u = v^2_{R_1,R_2} \in V_2(S)$,[16]

$$|\psi_\star^G(u)\rangle = -\sum_{j_2 \in R_2(\mu(S))} \sqrt{\mathsf{w}_2}|v^2_{R_1,R_2}, \leftarrow, j_2\rangle + \sum_{i_3 \in A_3} \sqrt{\mathsf{w}_3}|v^2_{R_1,R_2}, \rightarrow, i_3\rangle; \tag{62}$$

and finally for $u = v^3_{R_1,R_2,i_3} \in V_3$,

$$|\psi_\star^G(u)\rangle = -\sqrt{\mathsf{w}_3}|v^3_{R_1,R_2,i_3}, \leftarrow\rangle. \tag{63}$$

From Table 1, as well as the description of w from Lemma 5.4, we can see that for $u = v^0_{R,i_1} \in V_0^+$,

$$|\psi_\star^G(u)\rangle = -\sqrt{\mathsf{w}_0^+}|u, \leftarrow\rangle + \sum_{S_1 \subseteq \{1,2\} \setminus \mathcal{I}(u) : S_1 \neq \emptyset} \sqrt{\mathsf{w}_1}|u, S_1\rangle.$$

To generate this state, one would have to compute $\mathcal{I}(u)$ (see (53)), which would require finding any $i_2 \in R_2$ such that $x_{i_1} = x_{i_2}$, which is too expensive. Hence, we simply add all three options, for possibilities $\mathcal{I}(u) \in \{\emptyset, \{1\}, \{2\}\}$ (see also Figure 3), to $\Psi_\star(u)$:

$$\begin{aligned}
\Psi_\star(u) := \{|\psi_\star^\emptyset(u)\rangle &:= \sqrt{\mathsf{w}_0^+}|u, \leftarrow\rangle + \sqrt{\mathsf{w}_1}|u, \{1\}\rangle + \sqrt{\mathsf{w}_1}|u, \{1,2\}\rangle + \sqrt{\mathsf{w}_1}|u, \{2\}\rangle, \\
|\psi_\star^{\{1\}}(u)\rangle &:= \sqrt{\mathsf{w}_0^+}|u, \leftarrow\rangle + \sqrt{\mathsf{w}_1}|u, \{2\}\rangle, \\
|\psi_\star^{\{2\}}(u)\rangle &:= \sqrt{\mathsf{w}_0^+}|u, \leftarrow\rangle + \sqrt{\mathsf{w}_1}|u, \{1\}\rangle\} \ni |\psi_\star^G(u)\rangle.
\end{aligned} \tag{64}$$

Note that it is important that each state in $\bigcup_{u \in V_0^+} \Psi_\star(u)$ (and therefore each $|\psi_\star^G(u)\rangle$) have at least one outgoing (i.e. forward) edge. Otherwise, it would be impossible to satisfy **P2** of Theorem 3.10 (or equivalently, Item 2 of Lemma 5.14). This is satisfied because $\mathcal{I}(u)$ is always a proper subset of $\{1,2\}$.

We now describe how to generate the states in $\bigcup_{u \in V(G)} \Psi_\star(u)$ in $\widetilde{O}(1) = \mathrm{polylog}(n)$ complexity (see Definition 3.9). We will make use of the following lemma.

16 Here we explicitly include the $\rightarrow$ and $\leftarrow$ parts of each element of $L^+(u)$ and $L^-(u)$, which are normally left implict, in order to stress that the first and second sum are orthogonal.

LEMMA 5.5. *Let $V' \subseteq V(G) \setminus V_0 \cup V_M$ be such that there exists some constant c such that for all $u \in V'$, $L(u) \subseteq \{0,1\}^c$. Suppose for all $u \in V'$,*

$$\Psi_\star(u) = \{|u\rangle|\phi_\ell\rangle : \ell \in [d']\}$$

for some constant d', and states $|\phi_\ell\rangle \in \mathrm{span}\{|j\rangle : j \in \{0,1\}^c\}$. Then for some $d \le d'$, there is an orthonormal basis $\overline{\Psi}(u) = \{|\overline{\psi}_{u,1}\rangle, \ldots, |\overline{\psi}_{u,d}\rangle\}$ for $\mathrm{span}\{\Psi_\star(u)\}$, for each $u \in V'$, and a map $U'_\star$ that can be implemented in cost $O(1)$ such that for all $u \in V'$ and $\ell \in [d]$, $U'_\star|u,\ell\rangle = |\overline{\psi}_{u,\ell}\rangle$.

PROOF. First note that by the assumptions we are making, $d := \dim \mathrm{span}\{\Psi_\star(u)\}$ for all $u \in V'$, and d is a constant. Fix any orthonormal basis $\{|\overline{\phi}_1\rangle, \ldots, |\overline{\phi}_d\rangle\}$ for $\mathrm{span}\{|\phi_\ell\rangle : \ell \in [d']\}$, which is independent of u. Since the basis lives in a constant-dimensional subspace, the map: $C_\star : |\ell\rangle \mapsto |\overline{\phi}_\ell\rangle$ acts on a constant number of qubits, and so can be implemented in $O(1)$ elementary gates. We complete the proof by letting $U'_\star = I \otimes C_\star$, and observe that: $U'_\star|u,\ell\rangle = |u\rangle|\overline{\phi}_\ell\rangle =: |\overline{\psi}_{u,\ell}\rangle$. ■

LEMMA 5.6. *The states $\Psi_\star = \{\Psi_\star(u)\}_{u \in V(G)}$ can be generated in $\widetilde{O}(1)$ complexity.*

PROOF. The description of a vertex $u \in V(G)$ begins with a label indicating to which of $V_0, V_0^+, V_1, V_2, V_3$ it belongs. Thus, we can define subroutines $U_0, U_{0,+}, U_1, U_2, U_3$ that generate the star states in each vertex set respectively, and then $U_\star = \sum_{\ell=0}^{3} |\ell\rangle\langle\ell| \otimes U_\ell + |0,+\rangle\langle 0,+| \otimes U_{0,+}$ will generate the star states in the sense of Definition 3.9.

We begin with U_0. For $v_R^0 \in V_0$, we have $\Psi_\star(v_R^0) = \{|\psi_\star^G(v_R^0)\rangle\}$, where $|\psi_\star^G(v_R^0)\rangle$ is as in (60). Thus, implementing the map $U_0 : |v_R^0\rangle|0\rangle \mapsto\propto |\psi_\star^G(v_R^0)\rangle$ is as simple as generating a uniform superposition over A_1, and then using $O(\log n)$ rounds of amplitude amplification to get inverse polynomially close to the uniform superposition over $A_1 \setminus R_1$.

For $U_{0,+}$, since all $v_{R,i_1}^0 \in V_0^+$ have the same star states, modulo v_{R,i_1}^0 itself, with constant-sized label set $L = \{\{1\}, \{2\}, \{1,2\}, \leftarrow\}$, we can apply Lemma 5.5, to get a $U_{0,+}$ that costs $O(1)$.

We continue with U_1. For $v_R^1 \in V_1$, we have $\Psi_\star(v_R^1) = \{|\psi_\star^G(v_R^1)\rangle\}$, where $|\psi_\star^G(v_R^1)\rangle$ is as in (61). Thus, to implement the map $U_1 : |u\rangle|0\rangle \mapsto\propto |\psi_\star^G(u)\rangle$, we first compute (referring to Table 2 for the weights):

$$|u,0\rangle \mapsto\propto |u\rangle \left(-\sqrt{w_1}|\leftarrow\rangle + \sqrt{w_2}|\rightarrow\rangle\right)|0\rangle = |u\rangle \left(-|\leftarrow\rangle + (n/m_2)^{1/4}|\rightarrow\rangle\right)|0\rangle,$$

which can be implemented by a $O(1)$-qubit rotation. Then conditioned on $\leftarrow$, generate a uniform superposition over $i_1 \in R_1$, and then use $O(\log n)$ rounds of amplitude amplification to get inverse polynomially close to a superposition over $i_1 \in R_1$ such that there is no $(i_1, i_2, x_{i_1}) \in D_2(R)$. We have used the fact that our data structure supports taking a uniform superposition (see Section 2.3). Finally, conditioned on $\rightarrow$, generate a uniform superposition over $j_2 \in [m_2] \setminus R_2$.

The implementation of U_2 is similar, but instead (see (62)) we perform a single qubit rotation to get $-\sqrt{w_2}|\leftarrow\rangle + \sqrt{w_3}|\rightarrow\rangle$ in the last register, and then conditioned on the value of this register, we either generate a uniform superposition over $R_2(\mu(S))$ or A_3.

Finally, referring to (63), we see that the implementation of U_3 is trivial. We Thus, conclude that $U_\star$ can be implemented in $\widetilde{O}(1) = \text{polylog}(n)$ complexity. ■

5.2.3 The Transition Subroutines

In this section we show how to implement the transition map $|u,i\rangle \mapsto |v,j\rangle$ for $(u,v) \in \overrightarrow{E}(G)$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$ (see Definition 2.3). We do this by exhibiting uniform (in the sense of Lemma 2.6) subroutines $\mathcal{S}_0^+, \mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3$ that implement the transition map for (u,v) in E_0^+, E_1, E_2, E_3 respectively (defined in Section 5.2.1) whose union makes up $\overrightarrow{E}(G)$. In Corollary 5.12, we will combine these to get a quantum subroutine (Definition 2.5) for the full transition map.

LEMMA 5.7. *There is a uniform subroutine $\mathcal{S}_0^+$ such that for all $(u,v) \in E_0^+$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, $\mathcal{S}_0^+$ maps $|u,i\rangle$ to $|v,j\rangle$ with error 0 in complexity $\mathsf{T}_{u,v} = \mathsf{T}_0^+ = \widetilde{O}(1)$.*

PROOF. For $(v_R^0, v_{R,i_1}^0) \in E_0^+$, $\mathcal{S}_0^+$ should implement the map:

$$\begin{aligned} |v_R^0, i_1\rangle &\mapsto |v_{R,i_1}^0, \leftarrow\rangle \\ \equiv\ |(0,R,D(R)),i_1\rangle &\mapsto |((0,+),R,D(R),i_1),\leftarrow\rangle. \end{aligned}$$

It is easy to see that this can be done in $\text{polylog}(n)$ complexity (and is therefore trivially uniform): we just need to do some accounting to move i_1 from the edge label register to the vertex register, and update the first register $|0\rangle \mapsto |(0,+)\rangle$. ■

LEMMA 5.8. *There is a uniform subroutine $\mathcal{S}_1$ such that for all $(u,v) \in E_1$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, $\mathcal{S}_1$ maps $|u,i\rangle$ to $|v,j\rangle$ with error 0 in complexity $\mathsf{T}_{u,v} = \mathsf{T}_1 = \widetilde{O}(1)$.*

PROOF. For $(v_{R_1,R_2,i_1}^0, v_{R_1',R_2}^1) \in E_1$, where $v_{R_1',R_2}^1 \in V_1(S)$, $\mathcal{S}_1$ should implement the map:

$$\begin{aligned} |v_{R_1,R_2,i_1}^0, S\rangle &\mapsto |v_{R_1',R_2}^1, i_1\rangle \\ \equiv\ |((0,+),R_1,R_2,D(R_1,R_2),i_1),S\rangle &\mapsto |(1,R_1',R_2,D(R_1',R_2)),i_1\rangle. \end{aligned}$$

To implement this transition, we need only insert i_1 into $R_1(S)$, query i_1 to obtain x_{i_1} and update the data by inserting (i_1, x_{i_1}) into the $D_1(R)$ part of $D(R_1,R_2) = (D_1(R), D_2(R))$ (see Section 2.3). Note that we *do not* attempt to update the $D_2(R)$ part of the data by searching $\overline{R}_2$ for collisions with i_1. If there is some $s \in S$ and $i_2 \in \overline{R}_2(s)$ such that $x_{i_1} = x_{i_2}$, then by definition of E_1, $(v_{R_1,R_2,i_1}^0, v_{R_1',R_2}^1) \notin E_1$. To finish, we uncompute S by checking which of the three parts of R_1 has size $t_1 + 1$, account for the moving of i_1 from the vertex register to the edge label register, and map $|(0,+)\rangle$ to $|1\rangle$ in the first register. The total cost is $\text{polylog}(n)$. ■

We now move on to $\mathcal{S}_2$, which is somewhat more complicated. For $(v^1_{R_1,R_2}, v^2_{R_1,R_2'}) \in E_2$, where $v^1_R \in V_1(S)$, and $R_2'(\mu(S)) = R_2(\mu(S)) \cup \{j_2\}$ for some $j_2 \in [m_2] \setminus R_2$, $\mathcal{S}_2$ should act as:

$$\begin{aligned} |v^1_{R_1,R_2}, j_2\rangle &\mapsto |v^2_{R_1,R_2'}, j_2\rangle \\ \equiv\ |(1, R_1, R_2, D(R_1,R_2)), j_2\rangle &\mapsto |(2, R_1, R_2', D(R_1,R_2')), j_2\rangle. \end{aligned} \tag{65}$$

The complexity of this map, which we will implement with some error, depends on $|\mathcal{K}(R_1, A_2^{(j_2)})|$ (see (44)), the number of collisions to be found between R_1 and the block $A_2^{(j_2)}$, which is implicitly being added to $\overline{R}_2$ by adding j_2 to R_2. Lemma 5.9 below describes how to implement this transition map as long as there are fewer than $c_{\max} \log n$ collisions to be found for some constant $c_{\max}$. For the case when $|\mathcal{K}(R_1, A_2^{(j_2)})| \geq c_{\max} \log n$, we will let the algorithm fail (so there is no bound on the error for such transitions). That is, we let:

$$\tilde{E} := \left\{ \left(v^1_R, v^2_{R'}\right) \in E_2 : |\mathcal{K}(R_1, A^{(j_2)})| \geq c_{\max} \log n, \text{ where } \{j_2\} = R_2' \setminus R_2 \right\}. \tag{66}$$

LEMMA 5.9. *Fix any constant κ. There is a uniform subroutine $\mathcal{S}_2$ that implements the transition map that maps $|u,i\rangle$ to $|v,j\rangle$ for all $(u,v) \in E_2 \setminus \tilde{E}$, with error $O(n^{-\kappa})$, in complexity $\mathsf{T}_{u,v} = \mathsf{T}_2 = \widetilde{O}(\sqrt{n/m_2})$.*

PROOF. To implement the map in (65), we need to insert j_2 into $R_2(\mu(S))$ to obtain R_2', update $D_2(R)$ to reflect this insertion, and increment the first register. All of these take polylog(n) complexity, except for updating $D_2(R)$. To update $D_2(R)$, we need to search $A_2^{(j_2)}$ – the new block we're adding to $\overline{R}_2$ – to find anything that collides with R_1. Since the number of such collisions is less than $c_{\max} \log n$, we can do this using quantum search, which is uniform, with error $O(n^{-\kappa})$ for any desired constant κ in complexity $O(\sqrt{n/m_2} \log^2 n)$, since $|A_2^{(j_2)}| = \sqrt{n/m_2}$. ■

LEMMA 5.10. *For any constant κ, there exists a choice of constant $c_{\max}$ sufficiently large such that $|\tilde{E}| \leq n^{-\kappa}|E_2|$.*

PROOF. By Lemma 2.8 (or as a special case of Lemma 5.21), for every $j_2 \in [m_2]$, if R_1 is uniformly random from $\binom{A_1}{t_1^{(3)}}$, there exists a constant $c_{\max}$ such that $\Pr[|\mathcal{K}(R_1, A_2^{(j_2)})| \geq c_{\max} \log n] \leq n^{-\kappa}$. It follows that the proportion of edges in E_2 that are in $\tilde{E}$ is at most $n^{-\kappa}$. ■

LEMMA 5.11. *There is a subroutine $\mathcal{S}_3$ such that for all $(u,v) \in E_3$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, $\mathcal{S}_3$ maps $|u,i\rangle$ to $|v,j\rangle$ with error 0 in complexity $\mathsf{T}_{u,v} = \widetilde{O}(1)$.*

PROOF. The proof is identical to that of Lemma 5.7. ■

In order to apply Theorem 3.10, we need to implement the full transition map as a quantum subroutine in the sense of Definition 2.5.

COROLLARY 5.12. *Let κ be any constant. There is a quantum subroutine (in the sense of Definition 2.5) that implements the full transition map with errors $\epsilon_e \leq n^{-\kappa}$ for all $e \in \overrightarrow{E}(G) \setminus \tilde{E}$, and times $\mathsf{T}_e = \widetilde{O}(1)$ for all $e \in \overrightarrow{E}(G) \setminus E_2$, and $\mathsf{T}_e = \mathsf{T}_2 = \widetilde{O}(\sqrt{n/m_2})$ for all $e \in E_2$.*

PROOF. We combine Lemma 5.7, Lemma 5.8, Lemma 5.9 and Lemma 5.11 using Lemma 2.7. ■

5.2.4 Initial State and Setup Cost

The initial state is defined to be the uniform superposition over V_1:

$$|\sigma\rangle := \sum_{v^0_{R_1,R_2} \in V_0} \frac{1}{\sqrt{|V_0|}} |v^0_{R_1,R_2}\rangle.$$

LEMMA 5.13. *The state $|\sigma\rangle$ can be generated with error $n^{-\kappa}$ for any constant κ in complexity*

$$\mathsf{S} = \widetilde{O}\left(t_1 + t_2\sqrt{\tfrac{n}{m_2}}\right).$$

PROOF. We start by taking a uniform superposition over all $R_1 \in \binom{A_1}{t_1^{(3)}}$ and $R_2 \in \binom{[m_2]}{t_2^{(2)}}$ stored in data structures as described in Section 2.3, and querying everything in R_1 to get $D_1(R)$, which altogether costs $\widetilde{O}(t_1 + t_2)$. Next for each $s \in \{1, 2\}$, we search for all elements of $\overline{R}_2(s)$ that collide with an element of $R_1(\{s\})$ or $R_1(\{1, 2\})$. However, we do not want to spend too long on this step, so we stop if we find ct_2 collisions, for some constant c. If we do this before all collisions are found, that part of the state is not correct, but we argue that this only impacts a very small part of the state. The cost of this search is (up to log factors) $\sqrt{t_2|\overline{R}_2|} = t_2\sqrt{n/m_2}$.

For a uniform R_1 and fixed R_2, the expected value of $Z = |\mathcal{K}(R_1, \overline{R}_2)|$, the number of collisions, is

$$\mu = O\left(\frac{|\overline{R}_2|t_1}{n}\right) = O\left(\frac{t_2\frac{n}{m_2}t_1}{n}\right) = O(t_2),$$

since $m_2 = \Theta(t_1)$. Let c' be a constant such that $\mu \leq c't_2$, and choose $c = 7c'$. Since Z is a hypergeometric random variable, we have, by Lemma 2.8, $\Pr[Z \geq ct_2] \leq e^{-ct_2} = o(n^{-\kappa})$ for any κ, since t_2 is polynomial in n. Thus, the state we generate is $n^{-\kappa}$-close to $|\sigma\rangle$. ■

5.2.5 Positive Analysis

For the positive analysis, we must exhibit a flow (see Definition 2.2) from V_0 to M whenever $M \neq \emptyset$.

LEMMA 5.14. *There exists some $\mathcal{R}^{\mathsf{T}} = \widetilde{O}(|V_0|^{-1})$ such that the following holds. Whenever there is a unique 3-collision $(a_1, a_2, a_3) \in A_1 \times A_2 \times A_3$, there exists a flow θ on G that satisfies conditions **P1-P5** of Theorem 3.10. Specifically:*

1. *For all $e \in \tilde{E}$, $\theta(e) = 0$.*
2. *For all $u \in V(G) \setminus (V_0 \cup V_3)$ and $|\psi_\star(u)\rangle \in \Psi_\star(u)$,*

$$\sum_{i \in L^+(u)} \frac{\theta(u, f_u^+(i))\langle\psi_\star(u)|u, i\rangle}{\sqrt{w_{u,i}}} - \sum_{i \in L^-(u)} \frac{\theta(u, f_u^-(i))\langle\psi_\star(u)|u, i\rangle}{\sqrt{w_{u,i}}} = 0.$$

3. $\sum_{u \in V_0} \theta(u) = 1$.

4. $\sum_{u\in V_0} \frac{|\theta(u)-\sigma(u)|^2}{\sigma(u)} \le 1$.
5. $\mathcal{E}^{\mathsf{T}}(\theta) \le \mathcal{R}^{\mathsf{T}}$.

PROOF. Recall that M is the set of $v^3_{R,i_3} \in V_3$ such that for some $S \subseteq \{1,2\}$, $a_1 \in R_1(S)$, $a_2 \in \overline{R}_2(\mu(S))$ and $a_3 = i_3$. Let $j^* \in [m_2]$ be the unique block label such that $a_2 \in A_2^{(j^*)}$. Then $a_2 \in \overline{R}_2(\mu(S))$ if and only if $j^* \in R_2(\mu(S))$. Assuming $M \neq \emptyset$, we define a flow θ on G with all its sinks in M. It will have sources in both V_1 and M, but all other vertices will conserve flow. This will imply **Item 2** for all *correct* star states of G, $|\psi^G_\star(u)\rangle$, but we will have to take extra care to ensure that **Item 2** is satisfied for the additional states in $\Psi_\star(u) : u \in V_0^+$.

To satisfy condition **P5** of Theorem 3.10, we must upper bound $\mathcal{E}^{\mathsf{T}}(\theta) = \mathcal{E}(\theta^{\mathsf{T}})$ (see Definition 2.4), which is the energy of the flow θ extended to a graph G^{T}, in which each edge of G in E_2 has been replaced by a path of length $\mathsf{T}_2 = \widetilde{O}(\sqrt{n/m_2})$, and all other edges have been replaced by paths of length $\widetilde{O}(1)$ (see Corollary 5.12). We define θ on E_0^+, E_1, E_2 and E_3 stage by stage, and upper bound the contribution to $\mathcal{E}^{\mathsf{T}}(\theta)$ for each stage.

$\mathcal{R}_0^+$, Item 3, and Item 4: Let M_0 be the set of $v^0_{R_1,R_2} \in V_0$ such that $a_1 \notin R_1$, $j^* \notin R_2$, and for $c_{\max}$ as in Lemma 5.10, $|\mathcal{K}(R_1, A_1^{(j^*)})| < c_{\max} \log n$ (see (44)). This latter condition is because we will later send flow down edges that add j^* to R_2, and we don't want to have flow on edges in $\tilde{E}$. For all $v^0_R \in M_0$, let $\theta(v^0_R, v^1_{R,a_1}) = |M_0|^{-1}$. For all other edges in E_0^+, let $\theta(e) = 0$. Note that we can already see that $\theta(u) = |M_0|^{-1}$ for all $u \in M_0$, so we satisfy **Item 3**. By Lemma 5.10, we know that the proportion of R_1 that are excluded because $|\mathcal{K}(\overline{R}_1, A_1^{(j^*)})| \ge c_{\max} \log n$ is $o(1)$, so we can conclude:

$$\frac{|V_0|}{|M_0|} = (1+o(1))\left(1+O\left(\frac{t_1}{n}\right)\right)\left(1+O\left(\frac{t_2}{m_2}\right)\right) = 1+o(1). \tag{67}$$

Since $\sigma(u) = \frac{1}{|V_0|}$, we can conclude with **Item 4** of the lemma statement:

$$\sum_{u\in V_0} \frac{|\theta(u)-\sigma(u)|^2}{\sigma(u)} = |V_0|^2\left(\frac{1}{|M_0|}-\frac{1}{|V_0|}\right)^2 = \left(\frac{|V_0|}{|M_0|}-1\right)^2 = o(1).$$

Using $\mathsf{w}_0^+ = 1$ and $\mathsf{T}_e = \widetilde{O}(1)$ for all $e \in E_0^+$ (see Table 2), the contribution of the edges in E_0^+ to the energy of the flow can be computed as:

$$\mathcal{R}_0^+ = \sum_{e\in E_0^+} \mathsf{T}_e \frac{\theta(e)^2}{\mathsf{w}_0^+} = \widetilde{O}\left(\sum_{u\in M_0} \frac{1}{|M_0|^2}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right), \tag{68}$$

since each vertex in M_0 has a unique outgoing edge with flow, and the flow is uniformly distributed.

$\mathcal{R}_1$ and Item 2: Let M_0^+ be the set of $v^0_{R,i_1} \in V_0^+$ such that $v^0_R \in M_0$ and $i_1 = a_1$, so $|M_0^+| = |M_0|$. These are the only vertices in V_0^+ that have flow coming in from V_0, and specifically, the incoming flow from V_0 to a vertex in M_0^+ is $\frac{1}{|M_0|}$.

The only way there could be a fault adding a_1 to R_1 would be if $a_2 \in \overline{R}_2$, but we have ensured that that is not the case. Thus, for each $u \in M_0^+$, $\mathcal{I}(u) = \emptyset$, so there are three edges going into V_1 (labelled by $\{1\}$, $\{2\}$, and $\{1,2\}$, all disjoint from $\mathcal{I}(u)$) to which we can assign flow.

Item 2 is satisfied for all $|\psi_\star^G(u)\rangle : u \in V(G) \setminus (V_0 \cup V_3)$ by virtue of θ conserving flow at all vertices in $V(G) \setminus (V_0 \cup V_3)$ (we have not finished defining θ, but it will be defined so that this holds). However, for $u \in V_0^+$, $\Psi_\star(u) = \{|\psi_\star^{\mathcal{I}}(u)\rangle\}_{\mathcal{I} \subsetneq \{1,2\}}$ (see (64)) contains more than just $|\psi_\star^G(u)\rangle$. When $u \in V_0^+ \setminus M_0^+$, there is no flow through u, so Item 2 is easily seen to be satisfied for all states in $\Psi_\star(u)$. For $u \in M_0^+$, $|\psi_\star^G(u)\rangle = |\psi_\star^\emptyset(u)\rangle$, so the additional constraints we need to take additional care to satisfy are those for $|\psi_\star^{\{s\}}(u)\rangle$ with $s \in \{1,2\}$:

$$\begin{aligned}
&\sum_{i \in L^+(u)} \theta(u, f_u^+(i)) \frac{\langle \psi_\star^{\{s\}}(u) | u, i \rangle}{\sqrt{\mathsf{w}_1}} - \sum_{j \in L^-(u)} \theta(u, f_u^-(j)) \frac{\langle \psi_\star^{\{s\}}(u) | u, i \rangle}{\sqrt{\mathsf{w}_0^+}} \\
&= \sum_{S \in 2^{\{1,2\}} \setminus \{\emptyset\}} \theta(u, f_u^+(S)) \frac{\langle \psi_\star^{\{s\}}(u) | u, S \rangle}{\sqrt{\mathsf{w}_1}} - \theta(u, f_u^-(\leftarrow)) \frac{\langle \psi_\star^{\{s\}}(u) | u, \leftarrow \rangle}{\sqrt{\mathsf{w}_0^+}} && \text{see Table 1} \\
&= \theta(u, f_u^+(\{3-s\})) \frac{\sqrt{\mathsf{w}_1}}{\sqrt{\mathsf{w}_1}} - \theta(u, f_u^-(\leftarrow)) \frac{-\sqrt{\mathsf{w}_0^+}}{\sqrt{\mathsf{w}_0^+}} && \text{see (64)} \\
&= \theta(u, v_{\{3-s\}}) + \theta(u, v^0),
\end{aligned}$$

where $v^0 = f_u^-(\leftarrow)$ is the neighbour of u in V_0, and $v_{\{3-s\}} = f_u^+(\{3-s\})$ is the neighbour of u in V_1 with edge labelled by $\{3-s\}$ (see Figure 8). So for $s' \in \{1,2\}$, we must have:

$$0 = \theta(u, v_{\{s'\}}) + \theta(u, v^0) = \theta(u, v_{\{s'\}}) - \frac{1}{|M_0|},$$

since $\theta(u, v^0) = -\theta(v^0, u) = -\frac{1}{|M_0|}$. To satisfy this, we set:

$$\theta(u, v_{\{1\}}) = \theta(u, v_{\{2\}}) = \frac{1}{|M_0|},$$

meaning that all the flow that comes into u along edge (v^0, u) must leave u along edge $(u, v_{\{1\}})$, but it must also all leave along edge $(u, v_{\{2\}})$. However, we have now assigned twice as much outgoing flow as incoming flow, so the only way for flow to be conserved at u is to also have $\frac{1}{|M_0|}$ flow coming into u along edge $(v_{\{1,2\}}, u)$, so we set:

$$\theta(u, v_{\{1,2\}}) = -\frac{1}{|M_0|}.$$

This is shown visually in Figure 8. Using $\mathsf{w}_1 = 1$ and $\mathsf{T}_1 = \widetilde{O}(1)$, we can compute the contribution of edges in E_1 to the energy of the flow as:

$$\mathcal{R}_1 = \sum_{u \in M_0^+} \mathsf{T}_1 \frac{3(1/|M_0|)^2}{\mathsf{w}_1} = \widetilde{O}\left(\frac{|M_0^+|}{|M_0|^2}\right) = \widetilde{O}\left(\frac{|M_0|}{|M_0|^2}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right). \tag{69}$$

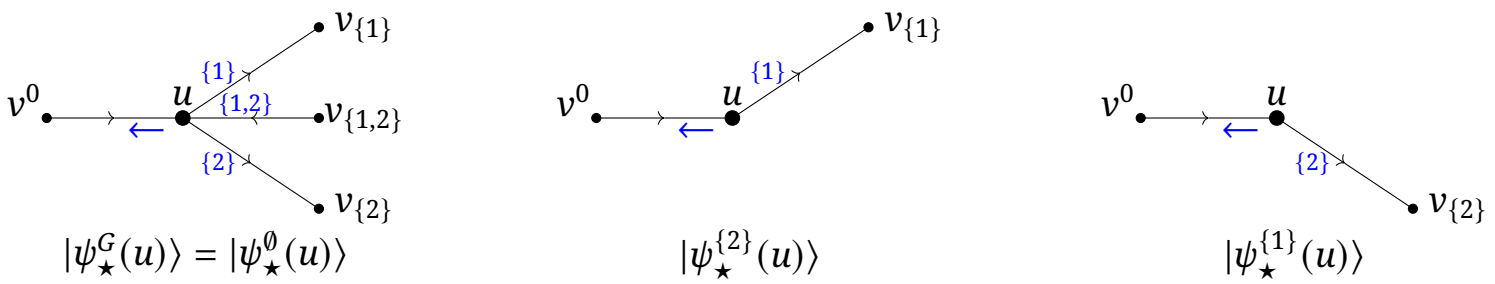


Figure 8. The three star states in $\Psi_\star(u)$, for $u \in V_0^+$. Edge labels from $L(u)$ are shown in blue. Arrows in edges indicate the direction of flow. We have chosen the flow so that flow is conserved at u in G, which can be seen by the fact that flow comes in on two edges, and leaves by two edges in the figure for $|\psi^G_\star(u)\rangle$; but flow is still conserved if we restrict to either of the other two neighbourhoods, which is necessary to satisfy Item 2 of Lemma 5.14.

$\mathcal{R}_2$ **and Item 1:** Let $M_1(S)$ be the set of $v_R^1 \in V_1(S)$ such that $a_1 \in R_1(S)$ and $j^* \notin R_2$, and let $M_1 = M_1(\{1\}) \cup M_1(\{2\}) \cup M_1(\{1,2\})$, so $|M_1| = 3|M_0|$. These are exactly the vertices of V_1 that have non-zero flow coming in from V_0^+, and in particular, for $v_R^1 \in M_1(S)$, the amount of flow coming in from V_0^+ is $(-1)^{|S|}\frac{3}{|M_1|}$, and we will send it along the edge $(v_R^1, v_{R'}^2) \in E_2$ that adds j^* to the set $R_2(\mu(S))$ to obtain R':

$$\theta\left(v_R^1, v_{R'}^2\right) = (-1)^{|S|+1}\frac{3}{|M_1|} = (-1)^{|S|+1}\frac{1}{|M_1(S)|}.$$

All other edges of E_2 will have $\theta(e) = 0$. Using $\mathsf{w}_2 = \sqrt{n/m_2}$ and $\mathsf{T}_2 = \widetilde{O}(\sqrt{n/m_2})$, we can compute the contribution of edges in E_2 to the energy $\mathcal{E}^{\mathsf{T}}$ of the flow:

$$\mathcal{R}_2 = \frac{\mathsf{T}_2}{\mathsf{w}_2}|M_1|\frac{9}{|M_1|^2} = \widetilde{O}\left(\frac{1}{|M_0|}\right). \tag{70}$$

We also note that by ensuring that there is only flow on $v_R^1 \in V_1$ when $\mathcal{K}(R_1, A_2^{(j^*)})$ is not too big, we have ensured that the flow on the edges in $\tilde{E}$ is 0, satisfying **Item 1**.

$\mathcal{R}_3$**:** Finally, let $M_2(S)$ be the set of $v_R^2 \in V_2(S)$ such that $a_1 \in R_1(S)$ and $j^* \in R_2(\mu(S))$, and let $M_2 = M_2(\{1\}) \cup M_2(\{2\}) \cup M_2(\{1,2\})$. These are exactly the vertices of V_2 that have non-zero flow coming in from V_1, in the amount of $(-1)^{|S|+1}|M_2(S)|^{-1}$. We send this flow along the unique edge from v_R^2 into V_3 that adds $i_3 = a_3$:

$$\theta(v_R^2, v_{R,a_3}^3) = (-1)^{|S|+1}\frac{1}{|M_2(S)|} = (-1)^{|S|+1}O\left(\frac{1}{|M_0|}\right).$$

Using $\mathsf{w}_3 = 1$ and $\mathsf{T}_3 = \widetilde{O}(1)$, the total contribution of edges in E_3 to the energy of the flow is:

$$\mathcal{R}_3 = \frac{\mathsf{T}_3}{\mathsf{w}_3}|M_2|O\left(\frac{1}{|M_2|^2}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right). \tag{71}$$

Item 5: It remains only to upper bound the energy of the flow by adding up the 4 contributions in (68) to (71), and applying $|V_0| = (1 + o(1))|M_0|$ from (67):

$$\mathcal{E}^{\mathsf{T}}(\theta) \leq \mathcal{R}_0^+ + \mathcal{R}_1 + \mathcal{R}_2 + \mathcal{R}_3 = \widetilde{O}\left(\frac{1}{|M_0|}\right) = \widetilde{O}\left(\frac{1}{|V_0|}\right). \qquad \blacksquare$$

5.2.6 Negative Analysis

For the negative analysis, we need to upper bound the total weight of the graph, taking into account the subroutine complexities, $\mathcal{W}^{\mathsf{T}}(G)$ (see Definition 2.4).

LEMMA 5.15. *There exists $\mathcal{W}^{\mathsf{T}}$ such that:*

$$\mathcal{W}^{\mathsf{T}}(G) \leq \mathcal{W}^{\mathsf{T}} \leq \widetilde{O}\left(\left(n + \frac{n^2}{t_1} + \frac{n^2}{t_2}\right)|V_1|\right).$$

PROOF. Recall that $\mathcal{W}^{\mathsf{T}}(G) = \mathcal{W}(G^{\mathsf{T}})$ is the total weight of the graph G^{T}, where we replace each edge e of G, with weight w_e, by a path of T_e edges of weight w_e, where T_e is the complexity of the edge transition e. Thus, $\mathcal{W}^{\mathsf{T}}(G) = \sum_{e \in E(G)} \mathsf{T}_e \mathsf{w}_e$. By Corollary 5.12, for all $e \in \overrightarrow{E}(G) \setminus E_2$, $\mathsf{T}_e = \widetilde{O}(1)$, and $\mathsf{w}_e = 1$ (see Table 2). Thus, using (50), the total contribution to the weight from the edges in E_0^+ is:

$$\mathcal{W}_0^+ := \mathsf{w}_0^+ |E_0^+| \mathsf{T}_0^+ = \widetilde{O}\left(n|V_0|\right). \tag{72}$$

Using (54), the total contribution from the edges in E_1 is:

$$\mathcal{W}_1 := \mathsf{w}_1 |E_1| \mathsf{T}_1 = \widetilde{O}\left(n|V_0|\right). \tag{73}$$

The edges $e \in E_2$ have $\mathsf{T}_e = \mathsf{T}_2 = \widetilde{O}(\sqrt{n/m_2})$, by Corollary 5.12, so using $\mathsf{w}_2 = \sqrt{n/m_2}$ and (56), the total contribution from the edges in E_2 is:

$$\mathcal{W}_2 := \mathsf{w}_2 |E_2| \mathsf{T}_2 = \widetilde{O}\left(\sqrt{\frac{n}{m_2}} \frac{2(m_2 - t_2)(n - 9t_1)}{t_1 + 1} |V_0| \sqrt{\frac{n}{m_2}}\right) = \widetilde{O}\left(\frac{n^2}{t_1}|V_0|\right). \tag{74}$$

Finally, using (57) and the fact that $m_2 = \Theta(t_1)$, the total contribution from the edges in E_3 is:

$$\mathcal{W}_3 := \mathsf{w}_3 |E_3| \mathsf{T}_3 = \widetilde{O}\left(\frac{n^2}{t_2}|V_0|\right). \tag{75}$$

Combining (72) to (75), we get total weight:

$$\mathcal{W}^{\mathsf{T}}(G) = \mathcal{W}_0^+ + \mathcal{W}_1 + \mathcal{W}_2 + \mathcal{W}_3 = \widetilde{O}\left(\left(n + \frac{n^2}{t_1} + \frac{n^2}{t_2}\right)|V_0|\right). \qquad \blacksquare$$

5.2.7 Conclusion of Proof of Theorem 5.3

We can now conclude with the proof of Theorem 5.3, showing an upper bound of $\widetilde{O}(n^{5/7})$ on the bounded error quantum time complexity of 3-distinctness.

PROOF OF THEOREM 5.3. We apply Theorem 3.10 to G (Section 5.2.1), M ((58)), σ the uniform distribution on V_0 ((48)), and $\Psi_\star$ (Section 5.2.2), with

$$\mathcal{W}^\mathsf{T} = \widetilde{O}\left(\left(n + \frac{n^2}{t_1} + \frac{n^2}{t_2}\right)|V_0|\right) \text{ and } \mathcal{R}^\mathsf{T} = \widetilde{O}\left(|V_0|^{-1}\right).$$

Then we have

$$\mathcal{W}^\mathsf{T}\mathcal{R}^\mathsf{T} = \widetilde{O}\left(n + \frac{n^2}{t_1} + \frac{n^2}{t_2}\right) = o(n^2).$$

We have shown the following:

Setup Subroutine: By Lemma 5.13, the state $|\sigma\rangle$ can be generated in cost $\mathsf{S} = \widetilde{O}\left(t_1 + t_2\sqrt{\frac{n}{m_2}}\right)$.
Star State Generation Subroutine: By Lemma 5.6, the star states $\Psi_\star$ can be generated in $\widetilde{O}(1)$ complexity.
Transition Subroutine: By Corollary 5.12, there is a quantum subroutine that implements the transition map with errors $\epsilon_{u,v}$ and costs $\mathsf{T}_{u,v}$ such that

TS1 For all $(u,v) \in \overrightarrow{E}(G) \setminus E_2$, $\epsilon_{u,v} = 0$. For all $(u,v) \in E_2 \setminus \tilde{E}$ (see (66)), taking $\kappa > 2$ in Lemma 5.9, we have $\epsilon_{u,v} = O(n^{-\kappa}) = o(1/(\mathcal{R}^\mathsf{T}\mathcal{W}^\mathsf{T}))$.

TS2 By Lemma 5.10, using $\mathsf{w}_2 = \sqrt{n/m_2}$ and $\kappa > 2$:

$$\sum_{e\in\tilde{E}} \mathsf{w}_e = \mathsf{w}_2|\tilde{E}| \le \sqrt{\frac{n}{m_2}}n^{-\kappa}|E_2| = \sqrt{\frac{n}{m_2}}n^{-\kappa}\frac{2(m_2-t_2)(n-9t_1)}{t_1+1}|V_0| \qquad \text{by (56)}$$

$$= O\left(\sqrt{n}n^{-\kappa}n\frac{1}{\mathcal{R}^\mathsf{T}}\right) = o(1/\mathcal{R}^\mathsf{T}).$$

since $m_2 = \Theta(t_1)$.

Checking Subroutine: By (59), for any $u \in V_\mathsf{M} = V_3$, we can check if $u \in M$ in cost $\widetilde{O}(1)$.
Positive Condition: By Lemma 5.14, there exists a flow satisfying conditions **P1**-**P5** of Theorem 3.10, with $\mathcal{E}^\mathsf{T}(\theta) \le \mathcal{R}^\mathsf{T} = \widetilde{O}\left(|V_0|^{-1}\right)$.
Negative Condition: By Lemma 5.15, $\mathcal{W}^\mathsf{T}(G) \le \mathcal{W}^\mathsf{T} = \widetilde{O}\left(\left(n + \frac{n^2}{t_1} + \frac{n^2}{t_2}\right)|V_0|\right)$.

Thus, by Theorem 3.10, there is a quantum algorithm that decides if $M = \emptyset$ in bounded error in complexity:

$$\widetilde{O}\left(\mathsf{S} + \sqrt{\mathcal{R}^\mathsf{T}\mathcal{W}^\mathsf{T}}\right) = \widetilde{O}\left(t_1 + t_2\sqrt{\frac{n}{m_2}} + \sqrt{n + \frac{n^2}{t_1} + \frac{n^2}{t_2}}\right) = \widetilde{O}\left(t_1 + t_2\sqrt{\frac{n}{t_1}} + \sqrt{n} + \frac{n}{\sqrt{t_1}} + \frac{n}{\sqrt{t_2}}\right).$$

Choosing the optimal values of $t_1 = n^{5/7}$ and $t_2 = n^{4/7}$, we get an upper bound of $\widetilde{O}(n^{5/7})$. Since $M \neq \emptyset$ if x has a unique 3-collision, and $M = \emptyset$ if x has no 3-collision, the algorithm distinguishes these two cases. By Lemma 5.1, this is enough to solve 3-distinctness in general. ■

5.3 k-Distinctness Algorithm

In this section, we generalise the 3-distinctness algorithm from Section 5.2 to prove the following.

THEOREM 5.16. *Let k be any constant. There is a quantum algorithm that decides k-distinctness with bounded error in $\widetilde{O}\left(n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^k-1}}\right)$ complexity.*

Throughout this section, $\widetilde{O}$ will surpress polylogarithmic factors in n. We use the assumptions on the input defined in Section 5.1, including partitioning $[n]$ into $A_1 \cup \cdots \cup A_k$, and each A_ℓ, for $\ell \in \{2, \ldots, k-1\}$ into blocks $A_\ell^{(1)} \cup \cdots \cup A_\ell^{(m_\ell)}$ of size $\frac{n}{km_\ell}$. Additionally, for the uniformity of our notation in this section, we choose to also partition A_1 into blocks $A_1^{(1)} \cup \cdots \cup A_\ell^{(m_1)}$ of size $\frac{n}{km_1}$. By choosing $m_1 = \frac{n}{k} = |A_1|$, this becomes the trivial partition, where each block is of size 1. A summary of the parameters of the algorithm appears in Table 3.

Tuples of Sets: Fix constants $c_1, \ldots, c_{k-1}$ and parameters $t_1, \ldots, t_{k-1}$ as in Table 3. The vertices of our graph are labelled by sets $R = (R_1, \ldots, R_{k-1})$, where each R_ℓ is a tuple of $c_1 \ldots c_{\ell-1}(2^{c_\ell} - 1)$ disjoint subsets of $[m_\ell]$ of size t_ℓ:

$$R_\ell = (R_\ell(s_1, \ldots, s_{\ell-1}, S_\ell))_{s_1 \in [c_1], \ldots, s_{\ell-1} \in [c_{\ell-1}], S_\ell \in 2^{[c_\ell]} \setminus \{\emptyset\}}.$$

We define:

$$\overline{R}_\ell(s_1, \ldots, s_{\ell-1}, S_\ell) := \bigcup_{j_\ell \in R_\ell(s_1, \ldots, s_{\ell-1}, S_\ell)} A_\ell^{(j_\ell)}.$$

and

$$\overline{R}_\ell := (\overline{R}_\ell(s_1, \ldots, s_{\ell-1}, S_\ell))_{s_1 \in [c_1], \ldots, s_{\ell-1} \in [c_{\ell-1}], S_\ell \in 2^{[c_\ell]} \setminus \{\emptyset\}}.$$

If we let $r_\ell = |\overline{R}_\ell| \approx t_\ell \frac{n}{m_\ell}$ for $\ell \in [k-1]$, we get the set sizes r_ℓ from [9]. We will not use these variables, but we note that the values we get for $\{r_\ell\}_{\ell=1}^{k-1}$ (from the values of $\{t_\ell\}_{\ell=1}^{k-1}$) are the same as those obtained in [9], as our algorithm can be seen as an algorithmic version of the combinatorial construction used in [9]. Finally, we choose the number of blocks in each A_ℓ, m_ℓ, so that $m_\ell = \Theta(t_{\ell-1})$ for each $\ell \in \{2, \ldots, k-1\}$. This ensures that the expected size of $\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j_\ell)})$ is constant. These values are summarised in Table 3.

Data: With any R defined as above, we keep track of some input-dependent data as follows. First, we query everything in $\overline{R}_1$, so we define:

$$\begin{aligned} \forall S_1 \in 2^{[c_1]} \setminus \{\emptyset\}, D_1(R_1(S_1)) &:= \{(i_1, x_{i_1}) : i_1 \in \overline{R}_1(S_1)\} \\ D_1(R) &:= (D_1(R_1(S_1)))_{S_1 \in 2^{[c_1]} \setminus \{\emptyset\}}. \end{aligned} \tag{76}$$

Next, for $\ell \in \{2, \ldots, k-1\}$, and $(s_1, \ldots, s_{\ell-1}, S_\ell) \in [c_1] \times \cdots \times [c_{\ell-1}] \times (2^{[c_\ell]} \setminus \{\emptyset\})$, we only query some of the indices in $\overline{R}_\ell$, and which ones we query depends on R, specifically on

$\ell \in \{1,\dots,k-1\}, t_\ell$	$= n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^k-1}-\sum_{\ell'=2}^{\ell}\frac{2^{k-1-\ell'}}{2^k-1}}$
m_1	$= \frac{n}{k}$
$\ell \in \{2,\dots,k-1\}$ m_ℓ	$= \Theta(t_{\ell-1})$
c_1	$= k-1$
$\ell \in \{2,\dots,k-2\}, c_\ell$	$= O(1)$ large enough for Corollary 5.27
c_{k-1}	$= 1$
$\ell \in \{2,\dots,k-1\}, p_\ell$	$= \text{polylog}(n)$ large enough for Corollary 5.27.

Table 3. A summary of the (asymptotic) values of variables used in this section.

$R_1,\dots,R_{\ell-1}$:

$$D_\ell(R_\ell(s_1,\dots,s_{\ell-1},S_\ell)|R) := \bigcup_{\substack{S_{\ell-1}\subseteq[c_{\ell-1}]:\\ s_{\ell-1}\in S_{\ell-1}}} \Big\{(i_1,\dots,i_\ell,x_{i_1}) : x_{i_\ell}=x_{i_1}, i_\ell \in \overline{R}_\ell(s_1,\dots,s_{\ell-1},S_\ell),$$
$$(i_1,\dots,i_{\ell-1},x_{i_1}) \in D_{\ell-1}(R_{\ell-1}(s_1,\dots,s_{\ell-2},S_{\ell-1})|R)\Big\}. \quad (77)$$

We will sometimes omit "$|R$" when the context is clear. We can group these together to get:

$$D_\ell(R) := (D_\ell(R_\ell(s_1,\dots,s_{\ell-1},S_\ell)))_{(s_1,\dots,s_{\ell-1},S_\ell)\in[c_1]\times\dots[c_{\ell-1}]\times(2^{[c_\ell]}\setminus\{\emptyset\})}. \quad (78)$$

In addition to this data, we want to keep track of a number for each $j_\ell \in R_\ell$ that we call the *forward collision degree*. Loosely speaking, for some $i_\ell \in \overline{R}_\ell$, a forward collision is an element $(i_1,\dots,i_\ell,\dots,i_{\ell'},x_{i_1}) \in D_{\ell'}(R)$, for some $\ell' > \ell$, and some $i_1,\dots,i_{\ell-1},i_{\ell+1},\dots,i_{\ell'}$. This can only exist if $(i_1,\dots,i_\ell,i_{\ell+1},x_{i_1}) \in D_{\ell+1}(R)$, so the *forward collision degree of* i_ℓ, $\overline{d}^{\rightarrow}_\ell(i_\ell)$, counts these. Concretely, for $\ell \in \{1,\dots,k-2\}$ it is defined as:

$$\overline{d}^{\rightarrow}_R(i_\ell) := \left|\left\{(i_1,\dots,i_{\ell-1},i_{\ell+1}) \in \overline{R}_1\times\dots\times\overline{R}_{\ell-1}\times\overline{R}_{\ell+1} : (i_1,\dots,i_\ell,i_{\ell+1},x_{i_1}) \in D_{\ell+1}(R)\right\}\right|. \quad (79)$$

For consistency, we also define $\overline{d}^{\rightarrow}_R(i_{k-1}) := 0$ for $i_{k-1} \in \overline{R}_{k-1}$. Then we can define the forward collision degree of $j_\ell \in R_\ell$ for any $\ell \in \{1,\dots,k-1\}$ as:

$$d^{\rightarrow}_R(j_\ell) := \sum_{i_\ell \in A_\ell^{(j_\ell)}} \overline{d}^{\rightarrow}_\ell(i_\ell). \quad (80)$$

When our quantum walk removes some j_ℓ from R_ℓ, we will want to make sure that $d_R^{\rightarrow}(j_\ell) = 0$, because otherwise we will have to uncompute all forward collisions from the data, which could be expensive. Thus, we also keep a database of forward collision degrees:

$$\forall \ell \in \{1, \ldots, k-2\},\ C_\ell^{\rightarrow}(R) := \{(j_\ell, d_R^{\rightarrow}(j_\ell)) : j_\ell \in R_\ell, d_R^{\rightarrow}(j_\ell) > 0\}. \tag{81}$$

To summarise, the data we keep track of at a vertex v_R includes:

$$D(R) := (D_1(R), \ldots, D_{k-1}(R), C_1^{\rightarrow}(R), \ldots, C_{k-2}^{\rightarrow}(R)). \tag{82}$$

5.3.1 Intuition about the Combinatorial Structure

The way we partition each $\overline{R}_\ell$ (by partitioning R_ℓ) precisely follows the combinatorial structure of [9], but in this section, we try to give some intuition about why this is done. This section is not technically necessary, and may be skipped without impacting correctness. We will be imprecise for the sake of intuition; for precision see the remainder of this paper.
A vertex v_R is labelled by a tuple of tuples of sets:

$$\begin{aligned} R &= (R_1, \ldots, R_{k-1}) \\ &= \left((R_1(S_1))_{S_1 \in 2^{[c_1]} \setminus \{\emptyset\}}, \ldots, (R_{k-1}(s_1, \ldots, s_{k-2}, S_{k-1}))_{s_1 \in [c_1], \ldots, s_{k-2} \in [c_{k-2}], S_{k-1} \in 2^{[c_{k-1}]} \setminus \{\emptyset\}} \right). \end{aligned}$$

Let $\bar{t}_\ell := t_\ell c_1 c_2 \ldots c_{\ell-1}(2^{c_\ell} - 1)$ be the size of R_ℓ if each part $R_\ell(s_1, \ldots, s_{\ell-1}, S_\ell)$ has size t_ℓ. The set of such vertices where $|R_\ell| = \bar{t}_\ell$ for all $\ell \in \{1, \ldots, k-1\}$ is called V_0. Starting from such a vertex in V_0, we may add an index to R_1 to get a vertex where now $|R_1| = \bar{t}_1 + 1$ – call the set of such vertices V_1. Then we may add something to R_2 to get a vertex with $|R_2| = \bar{t}_2 + 1$ – call vertices of that form V_2. We can continue until we get a vertex where $|R_\ell| = \bar{t}_\ell + 1$ for all $\ell \in \{1, \ldots, k-1\}$, the set of which is called V_{k-1}.

Let us give some more detail on the process of moving from V_0 to V_{k-1}. For any vertex $v_R = v_R^0 \in V_0$, we first choose an index j_1 to add to R_1 (but don't yet add it), to get a vertex v_{R,j_1}^0 (we call the set of such vertices V_0^+). Next, we want to actually add j_1 to R_1 to get a vertex in V_1, but before we can do that, we need to choose *where* in R_1 to add j_1, so we first choose some non-empty $S_1 \subseteq [c_1]$ and then add j_1 to $R_1(S_1)$, completing the transition to V_1. The new vertex does not remember which j_1 was most recently added to $R_1(S_1)$, but it remembers where it was added (i.e. S_1), because it has $|R_1(S_1)| = t_1 + 1$.

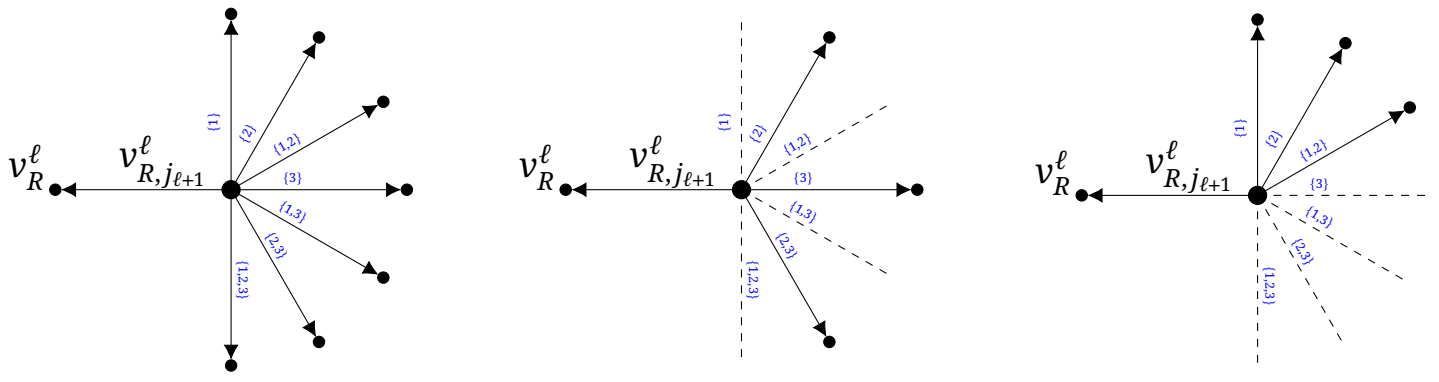
Next, to move from $v_R^1 \in V_1$ to some vertex in V_2, we again start by choosing the j_2 that we will eventually add to R_2, to get an intermediate vertex $v_{R,j_2}^1 \in V_1^+$. Then we need to choose some (s_1, S_2) and add j_2 to $R_2(s_1, S_2)$. We will ensure that we choose $s_1 \in S_1$ (we remember S_1, because $|R_1(S_1)| = t_1 + 1$), but we do this deterministically by taking the minimum element of S_1. The choice of S_2 however is random, which we discuss more below. The reason we choose $s_1 \in S_1$ is that in the analysis, we will construct a flow that goes along edges from V_0 to V_1 that add the unique j_1 of the block containing a_1; and then along edges that add the unique j_2 of

the block containing a_2, etc. We need $s_1 \in S_1$ to ensure that $(a_1, a_2, x_{a_1}) \in D_2(R)$, so that this flow eventually goes into the set of vertices in V_{k-1} that not only contain $a_1, \ldots, a_{k-1}$, but have noticed that they form a $(k-1)$-collision.

We continue moving from V_ℓ to $V_{\ell+1}$, making some choice $S_{\ell+1}$, and adding a new index to $R_{\ell+1}(s_1, \ldots, s_\ell, S_{\ell+1})$ (the $s_1, \ldots, s_\ell$ are chosen deterministically), so that a vertex in $V_{\ell+1}$ has associated sets $S_1, \ldots, S_{\ell+1}$. In this way, the choices of sets made in moving from V_0 to V_{k-1} give rise to a kind of tree, of depth k, with the degree at level ℓ being $s^{c_{\ell+1}} - 1$. (The nodes of this tree correspond to *sets* of vertices of the graph we're walking on, and the edges of the tree correspond to sets of edges in our graph).

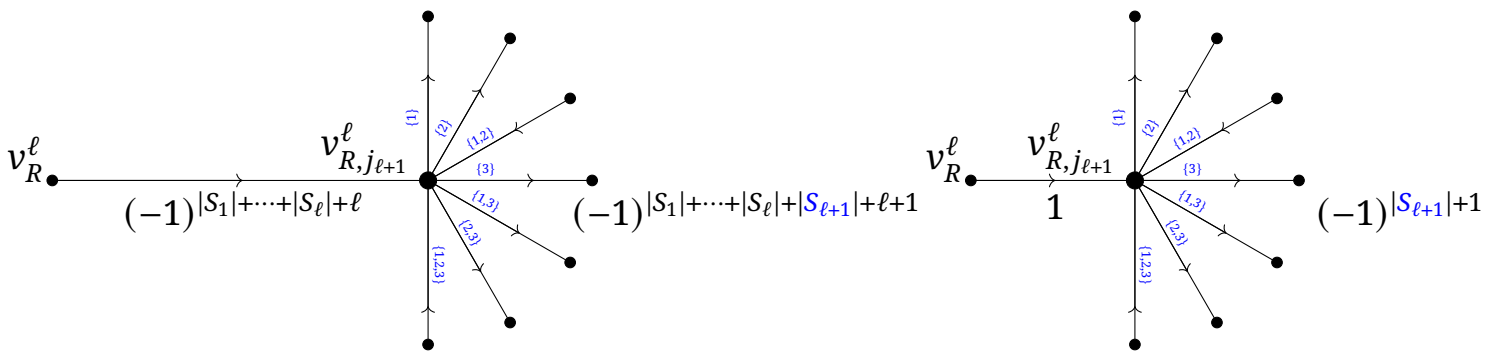
But the choice of $S_{\ell+1}$ requires care, because if, in our quantum walk, we add $j_{\ell+1}$ to a bad choice of $R_{\ell+1}(s_1, \ldots, s_\ell, S_{\ell+1})$, we might find that we have introduced a *fault*, rendering our data incorrect. It turns out that we can avoid faults precisely by making a choice of $S_{\ell+1}$ that avoids a certain set $\mathcal{I} = \mathcal{I}(v^\ell_{R,j_{\ell+1}}) \subset [c_{\ell+1}]$ – this ensures that if we add an index to $\overline{R}_{\ell+1}(s_1, \ldots, s_\ell, S_{\ell+1})$ that has a match in some part $\overline{R}_{\ell+2}(s_1, \ldots, s_{\ell+1}, S_{\ell+2})$, then $\overline{R}_{\ell+2}(s_1, \ldots, s_{\ell+1}, S_{\ell+2})$ is exempt from requiring matches with $\overline{R}_{\ell+1}(s_1, \ldots, s_\ell, S_{\ell+1})$ to be queried (because $s_{\ell+1} \notin S_{\ell+1}$).

Since we don't know $\mathcal{I}$, we use alternative neighbourhoods for all the different possibilities $\mathcal{I} \subsetneq [c_{\ell+1}]$. The resulting alternative neighbourhoods for $v^\ell_{R,j_{\ell+1}}$ are as follows: They all have a backwards edge to v^ℓ_R. The possible sets of forward edges are those labelled by $S_{\ell+1} \in L^+(v^\ell_{R,j_{\ell+1}}) = 2^{[c_{\ell+1}]} \setminus \emptyset$ such that $S_{\ell+1} \cap \mathcal{I} = \emptyset$ – that is non-empty $S_{\ell+1} \subset [c_{\ell+1}] \setminus \mathcal{I}$.[17] Below are just some of the alternative neighbourhoods for $c_{\ell+1} = 3$, for the choices $\mathcal{I} = \emptyset$, $\mathcal{I} = \{1\}$ and $\mathcal{I} = \{3\}$. Dotted lines indicate missing edges (or edges of weight 0).



The reason we have done this in such an involved way is that we need to be able to design a flow that is orthogonal to *all* of these stars. Later, we will see that such a flow exists. Up to scaling by some positive real number, the flow is a sign that depends on the sizes of the sets $S_1, \ldots, S_{\ell+1}$ that have been chosen so far, as shown in the following figure:

17 It's possible that $\mathcal{I} = [c_{\ell+1}]$, but this is sufficiently unlikely that we can treat this case separately.



An edge labelled by the set $S_{\ell+1}$ has flow $(-1)^{|S_1|+\cdots+|S_\ell|+|S_{\ell+1}|+\ell+1}$, also indicated by the direction of the edge. Up to scaling by $(-1)^{|S_1|+\cdots+|S_\ell|+\ell}$, the left-hand side is the same as the simplified picture on the right-hand-side above.

Consider the inner product of this flow state and a star state for a set $\mathcal{I}$. The incoming flow from v_R^ℓ always contributes (-1) (because in the star it's pointing out, and recall that switching edge direction switches the sign). For every non-empty $S_{\ell+1} \subset [c_{\ell+1}] \setminus \mathcal{I}$, the corresponding edge contributes 1 if it has outgoing flow (direction of flow matches the star state), and (-1) if it has incoming flow (direction of edge is opposite to star state). Then the inner product is:

$$-1 + \sum_{S_{\ell+1} \in 2^{[c_{\ell+1}]\setminus\mathcal{I}} \setminus \{\emptyset\}} (-1)^{|S_{\ell+1}|+1} = 0,$$

so flow is conserved with respect to all possible alternative neighbourhoods. This is precisely what we need, and why we use this complex combinatorial structure.

5.3.2 The Graph: Vertex Sets

To define G, we begin by defining disjoint vertex sets V_0, V_0^+, $(V_\ell)_{\ell=1}^{k-1}$, $(V_\ell^+)_{\ell=1}^{k-2}$, and V_k, whose union makes up $V(G)$. We will use the notation in (46) and (47) for tuples of disjoint sets throughout this section. Table 4 summarises G.

V_0: We define

$$V_0 = \left\{ v^0_{R_1,\ldots,R_{k-1}} := (0, R_1, \ldots, R_{k-1}, D(R_1, \ldots, R_{k-1})) : R_\ell \in \binom{[m_\ell]}{t_\ell^{(c_1\cdots c_{\ell-1}(2^{c_\ell}-1))}} \right\}. \quad (83)$$

Our initial distribution is uniform on V_0: $\sigma(u) = \frac{1}{|V_0|}$ for all $u \in V_0$. We implicitly store all sets including those making up $R_1, \ldots, R_{k-1}$ and $D(R_1, \ldots, R_{k-1})$ in a data structure with the properties described in Section 2.3. This will only be important when we analyse the time complexity of the setup and transition subroutines.

V_0^+: At a vertex in V_0^+, we suppose we have chosen a new element j_1 to add to R_1, but not yet added it. Thus, we label such a vertex by a tuple of sets R, and an index $j_1 \notin R_1$:

$$V_0^+ := \left\{ v^0_{R_1,\ldots,R_{k-1},j_1} := ((0,+), R_1, \ldots, R_{k-1}, D(R_1, \ldots, R_{k-1}), j_1) : v^0_{R_1,\ldots,R_{k-1}} \in V_0, j_1 \in [m_1] \setminus R_1 \right\},$$

$$\text{so } \left|V_0^+\right| = |V_0|\,|[m_1] \setminus R_1| = O\left(n\,|V_0|\right). \tag{84}$$

V_ℓ for $\ell \in \{1,\dots,k-1\}$: At a vertex in V_ℓ, we suppose we have added a new element to each of $R_1,\dots,R_\ell$, meaning that for each $\ell' \in [\ell]$, there is some $(s_1,\dots,s_{\ell'-1},S_{\ell'}) \in [c_1]\times\cdots\times[c_{\ell'-1}]\times(2^{[c_{\ell'}]}\setminus\{\emptyset\})$ such that $|R_{\ell'}(s_1,\dots,s_{\ell'-1},S_{\ell'})| = t_{\ell'}+1$. However, we will not let the choices of $s_1,\dots,s_{\ell'-1}$ for different ℓ' be arbitrary. Instead, we define the following sets of vertices, for $(S_1,\dots,S_\ell) \in (2^{[c_1]}\setminus\{\emptyset\})\times\cdots\times(2^{[c_\ell]}\setminus\{\emptyset\})$, where $\mu(S)$ denotes the minimum element of a set S:

$$\begin{aligned} V_\ell(S_1,\dots,S_\ell) := \Bigg\{ v_R^\ell = (\ell,R,D(R)) : \forall \ell' \in [\ell], R_{\ell'} \in \binom{[m_\ell]}{t_\ell^{(c_1\dots c_{\ell-1}(2^{c_\ell}-1))}}^+; \\ \forall \ell' \in \{\ell+1,\dots,k-1\}, R_{\ell'} \in \binom{[m_\ell]}{t_\ell^{(c_1\dots c_{\ell-1}(2^{c_\ell}-1))}}; \\ \forall \ell' \in \{1,\dots,\ell\},\ |R_{\ell'}(\mu(S_1),\dots,\mu(S_{\ell'-1}),S_{\ell'})| = t_{\ell'}+1 \Bigg\}. \end{aligned} \tag{85}$$

This is the set of vertices labelled by sets R where we have added elements to each of $R_1,\dots,R_\ell$, not yet added elements to $R_{\ell+1},\dots,R_{k-1}$, and for $\ell' \in \{1,\dots,\ell\}$, the choice of *where* the new element was added to $R_{\ell'}$ is determined by $S_1,\dots,S_\ell$. Then we can define:

$$V_\ell := \bigcup_{(S_1,\dots,S_\ell)\in(2^{[c_1]}\setminus\{\emptyset\})\times\cdots\times(2^{[c_\ell]}\setminus\{\emptyset\})} V_\ell(S_1,\dots,S_\ell). \tag{86}$$

Using the fact that for all $\ell' \in \{2,\dots,k-2\}$, $m_{\ell'} = \Theta(t_{\ell'-1})$, we have:

$$|V_\ell| = O\left(|V_0|\prod_{\ell'=1}^{\ell}\frac{m_{\ell'}}{t_{\ell'}}\right) = O\left(|V_0|\prod_{\ell'=1}^{\ell}\frac{t_{\ell'-1}}{t_{\ell'}}\right) = O\left(\frac{n}{t_\ell}\,|V_0|\right). \tag{87}$$

V_ℓ^+ : for $\ell \in \{1,\dots,k-2\}$: At a vertex in V_ℓ^+, we suppose, as in V_ℓ, that we have already added an element to each of the sets $R_1,\dots,R_\ell$, but now have also selected an element $j_{\ell+1} \in [m_{\ell+1}]$ to add to $R_{\ell+1}$:

$$V_\ell^+(S_1,\dots,S_\ell) := \left\{ v_{R,j_{\ell+1}}^\ell := ((\ell,+),R,D(R),j_{\ell+1}) : v_R^\ell \in V_\ell(S_1,\dots,S_\ell), j_{\ell+1} \in [m_{\ell+1}]\setminus R_{\ell+1} \right\}$$

$$V_\ell^+ := \bigcup_{(S_1,\dots,S_\ell)\in(2^{[c_1]}\setminus\{\emptyset\})\times\cdots\times(2^{[c_\ell]}\setminus\{\emptyset\})} V_\ell^+(S_1,\dots,S_\ell),$$

so together with (87) and $m_{\ell+1} = \Theta(t_\ell)$, this implies

$$\left|V_\ell^+\right| = |V_\ell|\,|[m_{\ell+1}] \setminus T_{\ell+1}| = O\left(n\,|V_0|\right). \tag{88}$$

The Final Stage, V_k: At a vertex in V_k, we have added a new element to each of $R_1,\dots,R_{k-1}$, as in V_{k-1}, and also selected some $i_k \in A_k$, which we can view as a candidate for completing one

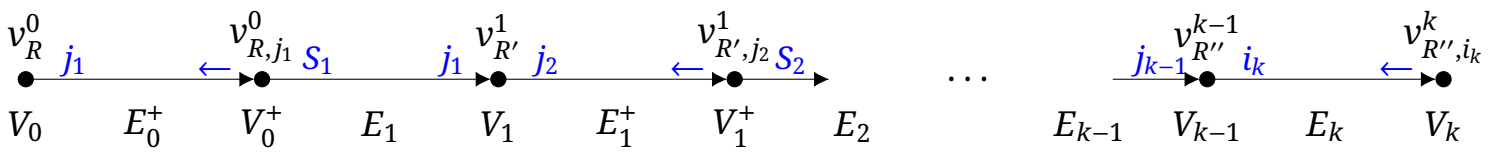


Figure 9. A path from V_0 to V_k, with edge labels shown in blue. R' is obtained from R by inserting j_1 into $R_1(S_1)$. R'' is obtained from R' by inserting j_2 into $R_2(\mu(S_1), S_2)$, and for some choice of $S_3, \ldots, S_{k-1}$, inserting, for each $\ell \in \{3, \ldots, k-1\}$, some j_ℓ into $R_\ell(\mu(S_1), \ldots, \mu(S_{\ell-1}), S_\ell)$.

u	$j \in L^-(u)$	$f_u^-(j)$	$i \in L^+(u)$	$f_u^+(i)$
$v_R^0 \in V_0$	$\emptyset$		$j_1 \in [m_1] \setminus R_1$	v_{R,j_1}^0
$v_{R,j_1}^0 \in V_0^+$	$\leftarrow$	v_R^0	$S_1 \in 2^{[c_1]} \setminus \{\emptyset\}$	$v_{R^{S_1 \leftarrow j_1}}^1$
$v_R^\ell \in V_\ell(S)$	$j_\ell \in R_\ell(\hat{\mu}(S)) : d_R^{\rightarrow}(j_\ell) = 0$	$v_{R \setminus \{j_\ell\}, j_\ell}^{\ell-1}$	$j_{\ell+1} \in [m_{\ell+1}] \setminus R_{\ell+1}$	$v_{R,j_{\ell+1}}^\ell$
$v_{R,j_{\ell+1}}^\ell \in V_\ell^+$	$\leftarrow$	v_R^ℓ	$S_{\ell+1} \in 2^{[c_{\ell+1}]} \setminus \{\emptyset\}$	$v_{R^{S_{\ell+1} \leftarrow j_{\ell+1}}}^{\ell+1}$
$v_R^{k-1} \in V_{k-1}(S)$	$j_{k-1} \in R_{k-1}(\hat{\mu}(S))$	$v_{R \setminus \{j_{k-1}\}, j_{k-1}}^{k-2}$	$i_k \in A_k$	v_{R,i_k}^k
$v_{R,i_k}^k \in V_k$	$\leftarrow$	v_R^{k-1}	$\emptyset$	

Table 4. The sets labeling incoming (L^-) and outgoing (L^+) edges of each vertex $u \in V(G)$, and the neighbouring vertices at the end of every such edge. $\ell \in \{1, \ldots, k-2\}$, $S = (S_1, \ldots, S_\ell)$, and for brevity we use $\hat{\mu}(S) := (\mu(S_1), \ldots, \mu(S_{\ell-1}), S_\ell)$, where μ is the minimum. $R^{S_1 \leftarrow j_1}$ is obtained from R by inserting j_1 into $R_1(S_1)$, and for $v_{R,j_{\ell+1}}^\ell \in V_{\ell+1}^+(S)$, $R^{S_{\ell+1} \leftarrow j_{\ell+1}}$ is obtained from R by inserting $j_{\ell+1}$ into $R_{\ell+1}(\hat{\mu}(S))$. To ensure that $L^-(u)$ and $L^+(u)$ are always disjoint, we implicitly append a $\leftarrow$ label to all of $L^-(u)$ and a $\rightarrow$ label to all of $L^+(u)$.

of the $(k-1)$-collisions in $D_{k-1}(R)$ to a k-collision:

$$V_k := \left\{ v_{R,i_k}^k := (k, R, D(R), i_k) : v_R^{k-1} \in V_{k-1}, i_k \in A_k. \right\},$$
$$\text{so } |V_k| = |V_{k-1}|\,|A_k| = O\left(\frac{n^2}{t_{k-1}}\,|V_0|\right). \tag{89}$$

5.3.3 The Graph: Edge Sets

We now define the sets of edges that make up $\overrightarrow{E}(G)$, as well as the edge label sets $L(u)$ (see Definition 2.3) for each $u \in V(G)$. These are also summarised in Table 4.

$E_\ell^+ \subset V_\ell \times V_\ell^+$ for $\ell \in \{0, \ldots, k-2\}$: There is an edge between $v_R^\ell \in V_\ell$ and $v_{R,j_{\ell+1}}^\ell \in V_\ell^+$ for any $j_{\ell+1} \in [m_{\ell+1}] \setminus R_{\ell+1}$, so we define

$$L^+(v_R^\ell) := [m_{\ell+1}] \setminus R_{\ell+1} \text{ and } L^-(v_{R,j_{\ell+1}}^\ell) := \{\leftarrow\},$$

and let $f^{+}_{v^{\ell}_{R}}(j_{\ell+1}) = v^{\ell}_{R,j_{\ell+1}}$ and $f^{-}_{v^{\ell}_{R,j_{\ell+1}}}(\leftarrow) = v^{\ell}_{R}$. We let E^{+}_{ℓ} be the set of all such edges

$$E^{+}_{\ell} := \left\{(v^{\ell}_{R}, v^{\ell}_{R,j_{\ell+1}}) : v^{\ell}_{R} \in V_{\ell}, j_{\ell+1} \in [m_{\ell+1}] \setminus R_{\ell+1}\right\}$$

and set $\mathsf{w}_e = \mathsf{w}^{+}_{\ell} = 1$ for all $e \in E^{+}_{\ell}$. This together with (88) implies that

$$\left|E^{+}_{\ell}\right| = \left|V^{+}_{\ell}\right| = O\left(n\left|V_0\right|\right). \tag{90}$$

Faults: Fix $\ell \in \{1, \ldots, k-1\}$. As in the case of 3-distinctness, if we add a new block index j_ℓ to certain parts of R_ℓ, to get R', such that $d^{\rightarrow}_{R'}(j_\ell) > 0$, this introduces a *fault* in the data, which our quantum walk will want to avoid. The case for $k > 3$ is slightly more complicated, so we examine exactly when a fault is introduced before describing the remaining edge sets.

Suppose $v^{\ell-1}_{R} \in V_{\ell-1}(S^{*}_{1}, \ldots, S^{*}_{\ell-1})$ (see (85)) and we add some j_ℓ to $R_\ell(\mu(S^{*}_{1}), \ldots, \mu(S^{*}_{\ell-1}), S_\ell)$, for some $S_\ell \subseteq [c_\ell]$. For $\ell \in \{2, \ldots, k-2\}$, this introduces a fault if the following conditions are satisfied, by some $i_\ell \in A^{(j_\ell)}_{\ell}$, which is added to $\overline{R}_\ell(\mu(S^{*}_{1}), \ldots, \mu(S^{*}_{\ell-1}), S_\ell)$, (we use $[\mathcal{E}]$ to denote the logical value of an event $\mathcal{E}$):

$$\begin{aligned}
\mathbf{C}^{\leftarrow}(i_\ell, R, S_\ell) := \big[&\exists (i_1, \ldots, i_{\ell-1}) \in R_1 \times \overline{R}_2 \times \cdots \times \overline{R}_{\ell-1} \text{ s.t.} \\
&(i_1, \ldots, i_{\ell-1}, i_\ell, x_{i_1}) \in D_\ell(R_\ell(\mu(S^{*}_{1}), \ldots, \mu(S^{*}_{\ell-1}), S_\ell))\big] \\
\mathbf{C}^{\rightarrow}(i_\ell, R, S_\ell) := \Bigg[&\exists s_\ell \in S_\ell \text{ s.t.} \\
&\exists i_{\ell+1} \in \bigcup_{S_{\ell+1} \in 2^{[c_{\ell+1}]} \setminus \{\emptyset\}} \overline{R}_{\ell+1}(\mu(S^{*}_{1}), \ldots, \mu(S^{*}_{\ell-1}), s_\ell, S_{\ell+1}) \text{ s.t. } x_{i_{\ell+1}} = x_{i_\ell}\Bigg].
\end{aligned} \tag{91}$$

In words $\mathbf{C}^{\leftarrow}$ is the condition that i_ℓ forms a collision $(i_1, \ldots, i_\ell, x_{i_1})$ that would be stored in $D_\ell(R)$, and $\mathbf{C}^{\rightarrow}$ is the condition that i_ℓ collides with something in $\overline{R}_{\ell+1}$ such that if $\mathbf{C}^{\rightarrow}$ holds, $(i_1, \ldots, i_{\ell+1}, x_{i_1})$ would be stored in $D_{\ell+1}(R)$. For $\ell = 1$, $\mathbf{C}^{\rightarrow}$ is also defined, and i_1 introduces a fault whenever $\mathbf{C}^{\rightarrow}$ is true. For $\ell = k-1$, $\mathbf{C}^{\rightarrow}$ can never be true, so there is never a fault. We set $c_{k-1} = 1$ (see Table 3).

Then for any $\ell \in \{1, \ldots, k-2\}$, $v^{\ell-1}_{R} \in V_{\ell-1}(S^{*}_{1}, \ldots, S^{*}_{\ell-1})$, $i_\ell \in A_\ell \setminus \overline{R}_\ell$, and $S_\ell \in 2^{[c_\ell]} \setminus \{\emptyset\}$, condition $\mathbf{C}^{\rightarrow}$ is false if and only if S_ℓ is disjoint from the following set:

$$\mathcal{I}(v^{\ell-1}_{R}, i_\ell) := \left\{ s_\ell \in [c_\ell] : \exists i_{\ell+1} \in \bigcup_{S_{\ell+1} \in 2^{[c_{\ell+1}]} \setminus \{\emptyset\}} \overline{R}_{\ell+1}(\mu(S^{*}_{1}), \ldots, \mu(S^{*}_{\ell-1}), s_\ell, S_{\ell+1}) \text{ s.t. } x_{i_{\ell+1}} = x_{i_\ell} \right\}. \tag{92}$$

For $\ell = k-1$, we define $\mathcal{I}(v^{k-2}_{R}, i_{k-1}) := \emptyset$. When $\ell = 1$ we can define, for $v^{0}_{R,j_1} \in V^{+}_{0}$:

$$\mathcal{I}(v^{0}_{R,j_1}) := \bigcup_{i_1 \in A^{(j_1)}_{1}} \mathcal{I}(v^{0}_{R}, i_1) \tag{93}$$

As long as we choose some S_1 that avoids this set, we will not introduce a fault. For $\ell > 1$, examining condition $\mathbf{C}^{\leftarrow}$ above, although it appears to depend on S_ℓ, it does not. Referring to

(77), we can rewrite $\mathbf{C}^{\leftarrow}$ as:

$$\mathbf{C}^{\leftarrow}(i_\ell, R, S_\ell) \Leftrightarrow \mathbf{C}^{\leftarrow}(i_\ell, R) := \Big[\exists S_{\ell-1} \subseteq [c_{\ell-1}] \text{ s.t. } \mu(S^*_{\ell-1}) \in S_{\ell-1},$$
$$\exists (i_1, \ldots, i_{\ell-1}, x_{i_1}) \in D_{\ell-1}(R_{\ell-1}(\mu(S^*_1), \ldots, \mu(S^*_{\ell-2}), S_{\ell-1})) \text{ s.t. } x_{i_\ell} = x_{i_1}\Big].$$

Thus, for $\ell \in \{2, \ldots, k-2\}$, for any $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}$, we can define:

$$\mathcal{I}(v^{\ell-1}_{R,j_\ell}) := \bigcup_{i_\ell \in A^{(j_\ell)}_\ell : \mathbf{C}^{\leftarrow}(i_\ell, R)} \mathcal{I}(v^{\ell-1}_R, i_\ell). \tag{94}$$

LEMMA 5.17. *For any $\ell \in \{1, \ldots, k-1\}$, fix $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}(S^*_1, \ldots, S^*_{\ell-1})$, and non-empty $S_\ell \subseteq [c_\ell]$, and let R' be obtained from R by inserting j_ℓ into $R_\ell(\mu(S^*_1), \ldots, \mu(S^*_{\ell-1}), S_\ell)$. Then $d^{\rightarrow}_{R'}(j_\ell) = 0$ if and only if $S_\ell \cap \mathcal{I}(v^{\ell-1}_{R,j_\ell}) = \emptyset$.*

PROOF. For $\ell = k-1$, $d^{\rightarrow}_{R'}(j_{k-1}) = 0$ and $\mathcal{I}(v^{k-2}_{R,j_{k-1}}) = \emptyset$ always hold, by definition. For $\ell \in \{1, \ldots, k-2\}$,

$$\begin{aligned}
\bar{d}^{\rightarrow}_{R'}(i_\ell) &= |\{(i_1, \ldots, i_\ell, i_{\ell+1}, x_{i_1}) \in D_{\ell+1}(R')\}| && \text{see (79)} \\
&= \sum_{\substack{(s_1,\ldots,s_\ell,S_{\ell+1}) \in \\ [c_1]\times\cdots\times[c_\ell]\times(2^{[c_{\ell+1}]}\setminus\{\emptyset\})}} |\{(i_1, \ldots, i_{\ell+1}, x_{i_1}) \in D_{\ell+1}(R(s_1, \ldots, s_\ell, S_{\ell+1}))\}| && \text{see (78)} \\
&= \sum_{\substack{(s_1,\ldots,s_\ell,S_{\ell+1}) \in \\ [c_1]\times\cdots\times[c_\ell]\times(2^{[c_{\ell+1}]}\setminus\{\emptyset\})}} \sum_{\substack{S'_\ell \subseteq [c_\ell]: \\ s_\ell \in S_\ell}} |\{(i_1, \ldots, i_{\ell+1}, x_{i_1}) : i_{\ell+1} \in \overline{R}_{\ell+1}(s_1, \ldots, s_\ell, S_{\ell+1}), \\
&\qquad x_{i_{\ell+1}} = x_{i_\ell}, (i_1, \ldots, i_\ell, x_{i_1}) \in D_\ell(R'_\ell(s_1, \ldots, s_{\ell-1}, S'_\ell))\}| && \text{see (77).}
\end{aligned}$$

Suppose $i_\ell \in A^{(j_\ell)}_\ell$, meaning we have $i_\ell \in \overline{R}_\ell(\mu(S^*_1), \ldots, \mu(S^*_{\ell-1}), S_\ell)$. By (77), ℓ-collisions of the form $(i_1, \ldots, i_\ell, x_{i_1})$ can only occur in $D_\ell(R_\ell(\mu(S^*_1), \ldots, \mu(S^*_{\ell-1}), S_\ell))$, so we continue:

$$\bar{d}^{\rightarrow}_{R'}(i_\ell) = \sum_{s_\ell \in S_\ell, S_{\ell+1} \in 2^{[c_{\ell+1}]}\setminus\{\emptyset\}} |\{(i_1, \ldots, i_{\ell+1}, x_{i_1}) : i_{\ell+1} \in \overline{R}_{\ell+1}(\mu(S^*_1), \ldots, \mu(S^*_{\ell-1}), s_\ell, S_{\ell+1}),$$
$$x_{i_{\ell+1}} = x_{i_\ell}, (i_1, \ldots, i_\ell, x_{i_1}) \in D_\ell(R'_\ell(\mu(S^*_1), \ldots, \mu(S^*_{\ell-1}), S_\ell))\}|,$$

and thus $\bar{d}^{\rightarrow}_{R'}(i_\ell) > 0$ if and only if:

$$\exists s_\ell \in S_\ell, S_{\ell+1} \in 2^{[c_{\ell+1}]} \setminus \{\emptyset\} \text{ s.t. } \exists i_{\ell+1} \in \overline{R}_{\ell+1}(\mu(S^*_1), \ldots, \mu(S^*_{\ell-1}), s_\ell, S_{\ell+1}) \text{ s.t. } x_{i_{\ell+1}} = x_{i_\ell} \tag{95}$$
$$\text{and } \exists (i_1, \ldots, i_{\ell-1}, i_\ell, x_{i_1}) \in D_\ell(R'_\ell(\mu(S^*_1), \ldots, \mu(S^*_{\ell-1}), S_\ell)). \tag{96}$$

Suppose $d^{\rightarrow}_{R'}(j_\ell) > 0$. By (80), this happens if and only if there exists $i_\ell \in A^{(j_\ell)}_\ell$ such that $\bar{d}^{\rightarrow}_{R'}(i_\ell) > 0$, which holds if and only if (95) and (96) are true. We know (95) if and only if $\mathbf{C}^{\rightarrow}(i_\ell, R, S_\ell)$ holds, if and only if $S_\ell \cap \mathcal{I}(v^{\ell-1}_R, i_\ell) \neq \emptyset$. For (96), we make a distinction based on the value of ℓ.

In the case $\ell = 1$, (96) is just $(i_1, x_{i_1}) \in D_1(R'_1(S_1))$, which is true by (76), since we just added j_1 to $R_1(S_1)$ to get $R'(S_1)$. For the other direction, if $S_1 \cap \mathcal{I}(v^0_{R,j_1}) \neq \emptyset$, then by (93), $\exists i_1 \in A^{(j_1)}_1$ satisfying (96). This completes the $\ell = 1$ case.

Continuing with the case $\ell \in \{2, \dots, k-2\}$, by (77), using the fact that $R'_{\ell-1} = R_{\ell-1}$, we have (96) if and only if $\mathbf{C}^{\leftarrow}(i_\ell, R)$. Thus, we have:

$$\left[d^{\rightarrow}_{R'}(j_\ell) > 0\right] \Leftrightarrow \underbrace{\exists i_\ell \in A_\ell^{(j_\ell)} \text{ s.t } \left[\left[S_\ell \cap \mathcal{I}(v_R^{\ell-1}, i_\ell) \neq \emptyset\right] \wedge \mathbf{C}^{\leftarrow}(i_\ell, R)\right]}_{=:\mathbf{C}}.$$

If $\mathbf{C}$ holds, then by (94), $\mathcal{I}(v_R^{\ell-1}, i_\ell) \subseteq \mathcal{I}(v_{R,j_\ell}^{\ell-1})$, and so, also by $\mathbf{C}$, $S_\ell \cap \mathcal{I}(v_{R,j_\ell}^{\ell-1}) \neq \emptyset$. For the other direction, if $S_\ell \cap \mathcal{I}(v_{R,j_\ell}^{\ell-1}) \neq \emptyset$, then by (94), $\exists i_\ell \in A_\ell^{(j_\ell)}$ satisfying both conditions of $\mathbf{C}$. ■

$E_1 \subset V_0^+ \times V_1$: Recall that V_0^+ is the set of vertices v^0_{R,j_1} in which we have chosen an index j_1 to add to R_1, but not yet decided to which part of R_1 it should be added. A transition in E_1 represents selecting some $S_1 \in 2^{[c_1]} \setminus \{\emptyset\}$ and then adding j_1 to $R_1(S_1)$, so we have

$$L^+(v^0_{R,j_1}) := 2^{[c_1]} \setminus \{\emptyset\},$$

and $f^+_{v^0_{R,j_1}}(S_1) = v^1_{R'}$, where R' is obtained from R by inserting j_1 into $R_1(S_1)$. As in the case of 3-distinctness, not all of these labels represent edges with non-zero weight. To go from a vertex $v^1_{R'} \in V_1(S_1^*)$, we choose some j_1 to remove from $R'_1(S_1^*)$, the part of R_1 that has had an index added, to get some R such that $v^0_R \in V_0$ (and $v^0_{R,j_1} \in V_0^+$). However, we make sure to choose an j_1 with no forward collisions – i.e. $d^{\rightarrow}_{R'}(j_1) = 0$ – so we let

$$L^-(v^1_{R'}) := \{j_1 \in R'_1(S_1^*) : d^{\rightarrow}_{R'}(j_1) = 0\},$$

and then set $f^-_{v^1_{R'}}(j_1) = v^0_{R,j_1}$ where $R = R' \setminus j_1$ is obtained from R' by removing j_1. Importantly, given $v^1_{R'}$, we can take a superposition over this set, because we store the set $C_1^{\rightarrow}(R)$ defined in (81) (this is necessary in Section 5.3.4).

As in the case of 3-distinctness, it is not yet clear how to define E_1, the set of (non-zero weight) edges between V_0^+ and V_1, because $|V_0^+| \cdot |L^+(v^0_{r,j_1})| > |V_1| \cdot |L^-(v^1_{R'})|$. We define it as follows.

$$E_1 := \left\{\left(f^-_{v^1_{R'}}(j_1), v^1_{R'}\right) = \left(v^0_{R'\setminus\{j_1\},j_1}, v^1_{R'}\right) : v^1_{R'} \in V_1, j_1 \in L^-(v^1_{R'})\right\}$$

and give weight $\mathsf{w}_1 = 1$ to all edges in E_1. Then we have the following.

LEMMA 5.18. *Let $R^{S_1 \leftarrow j_1}$ be obtained from R by inserting j_1 into $R_1(S_1)$. Then*

$$E_1 = \left\{\left(v^0_{R,j_1}, v^1_{R^{S_1\leftarrow j_1}}\right) : v^0_{R,j_1} \in V_0^+, S_1 \in 2^{[c_1]\setminus\mathcal{I}(v^0_{R,j_1})} \setminus \{\emptyset\}\right\}.$$

So for all $v^0_{R,j_1} \in V_0^+$, and $S_1 \in L^+(v^0_{R,j_1})$, $\mathsf{w}_{v^0_{R,j_1},S_1} = \begin{cases} \mathsf{w}_1 = 1 & \text{if } S_1 \cap \mathcal{I}(v^0_{R,j_1}) = \emptyset \\ 0 & \text{else.} \end{cases}$

PROOF. Let E'_1 be the right-hand side of the identity in the theorem statement, so we want to show $E_1 = E'_1$. Fix any $v^0_{R,j_1} \in V_0^+$ and non-empty $S_1 \subseteq [c_1] \setminus \mathcal{I}(v_{R,j_1})$, and let $R' = R^{S_1 \leftarrow j_1}$. Then since $S_1 \cap \mathcal{I}(v_{R,j_1}) = \emptyset$, by Lemma 5.17, $d^{\rightarrow}_{R'}(j_1) = 0$. This implies $E'_1 \subseteq E_1$.

For the other direction, fix any $v^1_{R'} \in V_1(S^*_1)$ and $j_1 \in L^-(v^1_{R'})$. Since $j_1 \in R_1(S^*_1)$, we have $v^1_{R'\setminus\{j_1\}} \in V^+_0$ (that is, we have removed an index from the set that had size t_1+1) and $(R' \setminus \{j_1\})^{S^*_1 \leftarrow j_1} = R'$. Since $d^{\rightarrow}_{R'}(j_1) = 0$, by Lemma 5.17, $S^*_1 \cap \mathcal{I}(v^0_{R'\setminus\{j_1\},j_1}) = \emptyset$. This implies $E_1 \subseteq E'_1$. ■

We remark that for any $j_1 \in [m_1]$, $d^{\rightarrow}_R(i_1)$ is always at most $k-2$. Otherwise, there are at least $k-1$ elements $i_2 \in \overline{R}_2 \subset A_2$ such that $x_{i_1} = x_{i_2}$ (where i_1 is the unique element in $A^{(j_1)}_1$), and together with i_1 these form a k-collision, which contradicts our assumption that the unique k-collision is in $A_1 \times \cdots \times A_k$. Thus, if we set $c_1 = k-1$, we have for any $v^0_{R,j_1} \in V^+_0$, $\mathcal{I}(v^0_{R,j_1}) \subsetneq [c_1]$, which will be important in Section 5.3.4.

Finally, it follows from (84), that

$$|E_1| \leq \left|L^+(v^0_{R,j_1})\right| \left|V^+_0\right| = O\left(n\,|V_0|\right). \tag{97}$$

$E_\ell \subset V^+_{\ell-1} \times V_\ell$ for $\ell \in \{2,\ldots,k-1\}$: For E_ℓ, we generalise the construction of E_1. Similar to the definition E_1, we define, for any $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}$, and $v^\ell_{R'} \in V_\ell(S^*_1,\ldots,S^*_\ell)$:

$$L^+(v^{\ell-1}_{R,j_\ell}) := 2^{[c_\ell]} \setminus \{\emptyset\} \text{ and } L^-(v^\ell_{R'}) := \{j_\ell \in R_\ell(\mu(S^*_1),\ldots,\mu(S^*_{\ell-1}),S_\ell) : d^{\rightarrow}_{R'}(j_\ell) = 0\}.$$

We set $f^+_{v^{\ell-1}_{R,j_\ell}}(S_\ell) = v^\ell_{R'}$ where if $v^{\ell-1}_R \in V_{\ell-1}(S^*_1,\ldots,S^*_{\ell-1})$, R' is obtained from R by inserting j_ℓ into $R_\ell(\mu(S^*_1),\ldots,\mu(S^*_{\ell-1}),S_\ell)$. We set $f^-_{v^\ell_{R'}}(j_\ell) = v^{\ell-1}_{R'\setminus\{j_\ell\},j_\ell}$. Similar to E_1, we define:

$$E_\ell := \left\{\left(f^-_{v^\ell_{R'}}(j_\ell), v^\ell_{R'}\right) = \left(v^{\ell-1}_{R'\setminus\{j_\ell\},j_\ell}, v^\ell_{R'}\right) : v^\ell_{R'} \in V_\ell, j_\ell \in L^-(v^\ell_{R'})\right\}, \tag{98}$$

and give weight $w_\ell := \sqrt{n/m_{\ell-1}}$ to all edges in E_ℓ. Then we have the following.

LEMMA 5.19. *For any $(S_1,\ldots,S_{\ell-1}) \in (2^{[c_1]} \setminus \{\emptyset\}) \times \cdots \times (2^{[c_{\ell-1}]} \setminus \{\emptyset\})$, define:*

$$E_\ell(S_1,\ldots,S_{\ell-1}) = \left\{\left(v^{\ell-1}_{R,j_\ell}, v^\ell_{R'}\right) : v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}(S_1,\ldots,S_{\ell-1}),\right.$$
$$\left.\exists S_\ell \in 2^{[c_\ell]\setminus\mathcal{I}(v^{\ell-1}_{R,j_\ell})} \setminus \{\emptyset\},\ R' = R^{(\mu(S_1),\ldots,\mu(S_{\ell-1}),S_\ell)\leftarrow j_\ell}\right\},$$

where $R^{(\mu(S_1),\ldots,\mu(S_{\ell-1}),S_\ell)\leftarrow j_\ell}$ is obtained from R by inserting j_ℓ into $R_\ell(\mu(S_1),\ldots,\mu(S_{\ell-1}),S_\ell)$. Then

$$E_\ell = \bigcup_{(S_1,\ldots,S_{\ell-1})\in(2^{[c_1]}\setminus\{\emptyset\})\times\cdots\times(2^{[c_{\ell-1}]}\setminus\{\emptyset\})} E_\ell(S_1,\ldots,S_{\ell-1}).$$

PROOF. Fix $S_1,\ldots,S_{\ell-1}$ and suppose $(v^{\ell-1}_{R,j_\ell}, v^\ell_{R'}) \in E_\ell(S_1,\ldots,S_{\ell-1})$. Then by Lemma 5.17, since S_ℓ is chosen so that $S_\ell \cap \mathcal{I}(v^{\ell-1}_{R,j_\ell}) = \emptyset$, $d^{\rightarrow}_{R'}(j_\ell) = 0$, and Thus, $j_\ell \in L^-(v^\ell_{R'})$, so $(v^{\ell-1}_{R,j_\ell}, v^\ell_{R'}) \in E_\ell$.

For the other direction, suppose $\left(v^{\ell-1}_{R'\setminus\{j_\ell\},j_\ell}, v^\ell_{R'}\right) \in E_\ell$, and let $S^*_1,\ldots,S^*_\ell$ be such that $v^\ell_{R'} \in V_\ell(S^*_1,\ldots,S^*_\ell)$. Then R' is obtained from $R' \setminus \{j_\ell\}$ by adding j_ℓ to $R_\ell(\mu(S^*_1),\ldots,\mu(S^*_{\ell-1}),S^*_\ell)$. Then since $j_\ell \in L^-(v^\ell_{R'})$, $d^{\rightarrow}_{R'}(j_\ell) = 0$, so by Lemma 5.17, $S^*_\ell \in 2^{[c_\ell]\setminus\mathcal{I}(v^{\ell-1}_{R,j_\ell})} \setminus \{\emptyset\}$, and so $\left(v^{\ell-1}_{R'\setminus\{j_\ell\},j_\ell}, v^\ell_{R'}\right) \in E_\ell(S^*_1,\ldots,S^*_{\ell-1})$. ■

While E_ℓ represents all edges between $V^+_{\ell-1}$ and V_ℓ, we now define two sets of edges $\tilde{E}_\ell \subset E_\ell$, and $\tilde{E}'_\ell$ disjoint from $V^+_{\ell-1} \times V_\ell$, that each solve a different technical issue. First, in Section 5.3.6, we will see that the complexity of transitions in E_ℓ depends on the number of collisions between the new block $A_\ell^{(j_\ell)}$ being added, and $(\ell-1)$-collisions already stored in $D_{\ell-1}(R)$, so we will only attempt to implement the transition subroutine correctly when this set is not too large. In anticipation of this, we define:

$$\tilde{E}_\ell := \left\{(v^{\ell-1}_{R,j_\ell}, v^\ell_{R'}) \in E_\ell : |\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j_\ell)})| \geq p_\ell\right\} \subset E_\ell, \tag{99}$$

where $p_\ell \in \text{polylog}(n)$, which will be part of $\tilde{E}$, the set of edges whose transitions we fail to implement. Second, if any $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}$ has no neighbour in V_ℓ, which happens exactly when $\mathcal{I}(v^{\ell-1}_{R,j_\ell}) = [c_\ell]$, then its correct star state would simply have one incoming edge from V_0, which, as discussed in Section 5.2.2, would make it impossible to define a flow satisfying all star state constraints (**P2** of Theorem 3.10). Unlike in the case of E_1, there is no constant c_ℓ such that we can assume $\mathcal{I}(v^{\ell-1}_{R,j_\ell}) \subsetneq [c_\ell]$ for all $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}$. That is because while each $i_\ell \in A_\ell$ can have at most $k-2$ collisions in $R_{\ell+1}$, the total number of such collisions for all $i_\ell \in A_\ell^{(j_\ell)}$ may be linear in $|A_\ell^{(j_\ell)}|$. Fortunately this happens for only a very small fraction of $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}$. Thus, we define (choosing $\{1\}$ arbitrarily):

$$\tilde{E}'_\ell := \left\{\left(v^{\ell-1}_{R,j_\ell}, f^+_{v^{\ell-1}_{R,j_\ell}}(\{1\})\right) : \mathcal{I}(v^{\ell-1}_{R,j_\ell}) = [c_\ell]\right\}, \tag{100}$$

which is disjoint from E_ℓ. Note that for $\ell = k-1$, we always have $\mathcal{I}(v^{k-2}_{R,j_{k-1}}) = \emptyset$ and $[c_{k-1}] = [1]$, so $\tilde{E}'_{k-1} = \emptyset$. Since $\tilde{E}'_\ell$ will be part of $\tilde{E}$, we assume its endpoints $f^+_{v^{\ell-1}_{R,j_\ell}}(\{1\})$ are just otherwise isolated vertices that we do not consider a part of $V(G)$ (see Remark 3.11). As with E_ℓ, we set all edges in $\tilde{E}'_\ell$ to have weight w_ℓ. Thus, from this discussion as well as Lemma 5.19 we have, for all $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}$ and $S_\ell \in L^+(v^{\ell-1}_{R,j_\ell})$,

$$\mathsf{w}_{v^{\ell-1}_{R,j_\ell}, S_\ell} = \begin{cases} \mathsf{w}_\ell = \sqrt{n/m_\ell} & \text{if } S_\ell \cap \mathcal{I}(v^{\ell-1}_{R,j_\ell}) = \emptyset \\ \mathsf{w}_\ell = \sqrt{n/m_\ell} & \text{if } \mathcal{I}(v^{\ell-1}_{R,j_\ell}) = [c_\ell] \text{ and } S_\ell = \{1\} \\ 0 & \text{else.} \end{cases} \tag{101}$$

We can see from (84), that

$$|E_\ell| + \left|\tilde{E}'_\ell\right| \leq \left|L^+(v^{\ell-1}_{R,j_\ell})\right| \left|V^+_{\ell-1}\right| = O\left(n\,|V_0|\right). \tag{102}$$

$E_k \subset V_{k-1} \times V_k$: Finally, there is an edge between $v^{k-1}_R \in V_{k-1}$ and $v^k_{R,i_k} \in V_k$ for any $i_k \in A_k$, so we define

$$L^+(v^{k-1}_R) := A_k \text{ and } L^-(v^k_{R,i_k}) := \{\leftarrow\},$$

Edge Set	(u,v)	(u,i)	(v,j)	$\mathsf{w}_{u,v}$	$\mathsf{T}_{u,v}$
$E_0^+ \subset V_0 \times V_0^+$	(v_R^0, v_{R,j_1}^0)	(v_R^0, j_1)	$(v_{R,j_1}^0, \leftarrow)$	$\mathsf{w}_0^+ = 1$	$\mathsf{T}_0^+ = \widetilde{O}(1)$
$E_1 \subset V_0^+ \times V_1$	$(v_{R,j_1}^0, v_{R'}^1)$	(v_{R,j_1}^0, S_1)	$(v_{R'}^1, j_1)$	$\mathsf{w}_1 = 1$	$\mathsf{T}_1 = \widetilde{O}(1)$
$\{E_\ell^+ \subset V_\ell \times V_\ell^+\}_{\ell=1}^{k-2}$	$(v_R^1, v_{R,j_{\ell+1}}^1)$	$(v_R^1, j_{\ell+1})$	$(v_{R,j_{\ell+1}}^1, \leftarrow)$	$\mathsf{w}_\ell^+ = 1$	$\mathsf{T}_\ell^+ = \widetilde{O}(1)$
$\{E_\ell \subset V_{\ell-1}^+ \times V_\ell\}_{\ell=1}^{k-1}$	$(v_{R,j_\ell}^{\ell-1}, v_{R'}^\ell)$	$(v_{R,j_\ell}^{\ell-1}, S_\ell)$	$(v_{R'}^\ell, j_\ell)$	$\mathsf{w}_\ell = \sqrt{\frac{n}{m_\ell}}$	$\mathsf{T}_\ell = \widetilde{O}\left(\sqrt{\frac{n}{m_\ell}}\right)$
$E_k \subset V_{k-1}^+ \times V_k$	(v_R^{k-1}, v_{R,i_k}^k)	(v_R^{k-1}, i_k)	$(v_{R,i_k}^k, \leftarrow)$	$\mathsf{w}_k = 1$	$\mathsf{T}_k = \widetilde{O}(1)$

Table 5. For each edge in $\overrightarrow{E}(G)$, we can describe it in three ways: as a pair of vertices (u,v); as a vertex u and forward label $i = f_u^{-1}(v)$; and as a vertex v and backward label $j = f_v^{-1}(u)$ (see Definition 2.3). We summarise these three descriptions for the edge sets that make up $\overrightarrow{E}(G) \setminus \tilde{E}$, along with the edge weights, and transitions costs (see Corollary 5.29). The edge labels i and j range across (sometimes strict) subsets of $L^+(u)$ and $L^-(u)$ (see Table 4). For example, for $u \in V_0^+$, $i = S_1 \in L^+(u) = 2^{[c_1]} \setminus \{\emptyset\}$, (u,i) only represents an edge of $\overrightarrow{E}(G)$ when $S_1 \cap \mathcal{I}(u) = \emptyset$ (see Lemma 5.18 and (101)).

and let $f^+_{v_R^{k-1}}(i_k) = v_{R,i_k}^k$, and $f^-_{v_{R,i_k}^k}(\leftarrow) = v_R^{k-1}$. We let E_k be the set of such edges:

$$E_k := \left\{(v_R^{k-1}, v_{R,i_k}^k) : v_R^{k-1} \in V_{k-1}, i_k \in A_k\right\},$$

and we set $\mathsf{w}_e = \mathsf{w}_k = 1$ for all $e \in E_k$. This, together with (89), implies that

$$|E_k| = O\left(\frac{n^2}{t_{k-1}}|V_0|\right). \tag{103}$$

The Graph G: The full graph G is defined by:

$$V(G) = \bigcup_{\ell=0}^{k} V_\ell \cup \bigcup_{\ell=0}^{k-2} V_\ell^+$$

$$\overrightarrow{E}(G) = \{(u,v) : u \in V(G), i \in L^+(u) : \mathsf{w}_{u,i} \neq 0\} = \bigcup_{\ell=0}^{k-2} E_\ell^+ \cup E_1 \cup \bigcup_{\ell=2}^{k-1} (E_\ell \cup \tilde{E}'_\ell) \cup E_k,$$

where the edge label sets $L^+(u)$ are summarised in Table 4, and weights are summarised in Table 5. We define (recall that $\tilde{E}_\ell \subset E_\ell$):

$$\tilde{E} := \bigcup_{\ell=2}^{k-1} (\tilde{E}_\ell \cup \tilde{E}'_\ell). \tag{104}$$

The Marked Set and Checking Cost: In the notation of Theorem 3.10, we let $V_{\mathsf{M}} = V_k$, and we define a subset $M \subseteq V_{\mathsf{M}}$ as follows. If $(a_1, \ldots, a_k) \in A_1 \times \cdots \times A_k$ is the unique k-collision,

we let

$$\begin{aligned} M &= \left\{ v^k_{R_1,\ldots,R_{k-1},i_k} \in V_k : \exists (i_1,\ldots,i_{k-1},x_{i_1}) \in D_{k-1}(R) \text{ s.t. } x_{i_1} = x_{i_k} \right\} \\ &= \Big\{ v^k_{R_1,\ldots,R_{k-1},i_k} : \exists S_1 \subseteq [c_1],\ldots,S_{k-1} \subseteq [c_{k-1}], s_1 \in S_1,\ldots,s_{k-1} \in S_{k-1} \text{ s.t.} \\ &\qquad\qquad \forall \ell \in \{1,\ldots,k-1\},\ a_\ell \in \overline{R}_\ell(s_1,\ldots,s_{\ell-1},S_\ell) \text{ and } i_k = a_k \Big\}, \end{aligned} \tag{105}$$

and otherwise, if there is no k-collision, $M = \emptyset$. We can decide whether $v^k_{R,i_k} \in V_k$ is marked by querying i_k to obtain the value x_{i_k} and looking it up in $D_{k-1}(R)$ to see if we find some $(i_1,\ldots,i_{k-1},x_{i_k})$, in which case, it must be that $a_1 = i_1,\ldots,a_k = i_k$. Thus, the checking cost is at most

$$\mathsf{C} = O(\log n). \tag{106}$$

5.3.4 The Star States and their Generation

We define the set of alternative neighbourhoods (Definition 3.9) with which we will apply Theorem 3.10. For $\ell \in \{0,\ldots,k\}$, for all $v^\ell_R \in V_\ell$, we add a single star state to $\Psi_\star(u)$, which has one of three forms, depending on ℓ (refer to Table 4): for $v^0_R \in V_0$,

$$\Psi_\star(v^0_R) := \left\{ |\psi^G_\star(v^0_R)\rangle = \sum_{j_1 \in [m_1]\setminus R_1} \sqrt{\mathsf{w}^+_0}\, |v^0_R, j_1\rangle \right\}; \tag{107}$$

for $\ell \in \{1,\ldots,k-1\}$, and $v^\ell_R \in V_\ell(S_1,\ldots,S_\ell)$,[18]

$$\Psi_\star(v^\ell_R) := \left\{ |\psi^G_\star(v^\ell_R)\rangle = - \sum_{\substack{j_\ell \in R_\ell(\mu(S_1),\ldots,\mu(S_{\ell-1}),S_\ell): \\ d^{\rightarrow}_R(j_\ell)=0}} \sqrt{\mathsf{w}_\ell}\, |v^\ell_R, \leftarrow, j_\ell\rangle + \sum_{j_{\ell+1} \in [m_{\ell+1}]\setminus R_{\ell+1}} \sqrt{\mathsf{w}^+_\ell}\, |v^\ell_R, \rightarrow, j_{\ell+1}\rangle \right\}; \tag{108}$$

and finally, for $v^k_{R,i_k} \in V_k$,

$$\Psi_\star(v^k_{R,i_k}) := \left\{ |\psi^G_\star(v^k_{R,i_k})\rangle = -\sqrt{\mathsf{w}_k}\, |v^k_{R,i_k}, \leftarrow\rangle \right\}. \tag{109}$$

From Table 4, along with the description of w in Lemma 5.18, we can see that for $v^0_{R,j_1} \in V^+_0$,

$$|\psi^G_\star(v^0_{R,j_1})\rangle = -\sqrt{\mathsf{w}^+_0}\, |v^0_{R,j_1}, \leftarrow\rangle + \sum_{S_1 \in 2^{[c_1]\setminus \mathcal{I}(v^0_{R,j_1})}\setminus\{\emptyset\}} \sqrt{\mathsf{w}_1}\, |v^0_{R,j_1}, S_1\rangle.$$

To generate this state, one would have to compute $\mathcal{I}(v^0_{R,j_1})$ (see (92)), which would require determining the locations of all forward collisions of j_1, which is far too expensive. Hence, we

18 Here we explicitly include the $\rightarrow$ and $\leftarrow$ parts of each element of L^+ and L^-, which are normally left implicit, in order to stress that the two sum are orthogonal.

simply add all options to $\Psi_\star(v^0_{R,j_1})$ (we see in Lemma 5.20 that generating this *set* is not difficult):

$$\Psi_\star(v^0_{R,j_1}) := \left\{ |\psi^{\mathcal{I}_1}_\star(v^0_{R,j_1})\rangle := -\sqrt{\mathsf{w}^+_0}|v^0_{R,j_1},\leftarrow\rangle + \sum_{S_1\in 2^{[c_1]\setminus\mathcal{I}_1}\setminus\{\emptyset\}} \sqrt{\mathsf{w}_1}|v^0_{R,j_1},S_1\rangle : \mathcal{I}_1 \subsetneq [c_1] \right\}. \quad (110)$$

Thus, since we always have $\mathcal{I}(v^0_{R,j_1}) \subsetneq [c_1]$, $|\psi^G_\star(v^0_{R,j_1})\rangle = |\psi^{\mathcal{I}(v^0_{R,j_1})}_\star(v^0_{R,j_1})\rangle \in \Psi_\star(v^0_{R,j_1})$.

Similarly, for $\ell \in \{2,\dots,k-1\}$ and $v^{\ell-1}_{R,j_\ell} \in V^+_{\ell-1}$ define:

$$\Psi_\star(v^{\ell-1}_{R,j_\ell}) := \left\{ |\psi^{\mathcal{I}_\ell}_\star(v^{\ell-1}_{R,j_\ell})\rangle := -\sqrt{\mathsf{w}^+_{\ell-1}}|v^{\ell-1}_{R,j_\ell},\leftarrow\rangle + \sum_{S_\ell\in 2^{[c_\ell]\setminus\mathcal{I}_\ell}\setminus\{\emptyset\}} \sqrt{\mathsf{w}_\ell}|v^{\ell-1}_{R,j_\ell},S_\ell\rangle : \mathcal{I}_\ell \subsetneq [c_\ell] \right\}. \quad (111)$$

Then from (101), we have:

$$|\psi^G_\star(v^\ell_{R,j_\ell})\rangle = \begin{cases} |\psi^{\mathcal{I}(v^\ell_{R,j_\ell})}_\star(v^\ell_{R,j_\ell})\rangle & \text{if } \mathcal{I}(v^\ell_{R,j_\ell}) \subsetneq [c_\ell] \\ |\psi^{[c_\ell]\setminus\{1\}}_\star(v^\ell_{R,j_\ell})\rangle & \text{if } \mathcal{I}(v^\ell_{R,j_\ell}) = [c_\ell], \end{cases} \quad (112)$$

where $\mathcal{I}(v^\ell_{R,j_\ell})$ is defined in (94).

We now describe how to generate the states in $\bigcup_{u\in V(G)}\Psi_\star(u)$ in $\widetilde{O}(1)$ complexity (see Definition 3.9):

LEMMA 5.20. *The states $\Psi_\star = \{\Psi_\star(u)\}_{u\in V(G)}$ can be generated in $\widetilde{O}(1)$ complexity.*

PROOF. The description of a vertex $u \in V(G)$ begins with a label indicating to which of $V_0,\dots,V_k$ or $V^+_0,\dots,V^+_{k-2}$ it belongs, so we can define subroutines $U_0,\dots,U_k,U_{0,+},\dots,U_{k-2,+}$ that generate the star states in each vertex set respectively, and then

$$U_\star = \sum_{\ell=0}^{k} |\ell\rangle\langle\ell| \otimes U_\ell + \sum_{\ell=0}^{k-2} |\ell,+\rangle\langle\ell,+| \otimes U_{\ell,+}$$

will generate the star states in the sense of Definition 3.9.

We begin with U_0. For $v^0_R \in V_0$, we have $\Psi_\star(v^0_R) = \{|\psi^G_\star(v^0_R)\rangle\}$, where $|\psi^G_\star(v^0_R)\rangle$ is as in (107). Thus, implementing the map $U_0 : |u\rangle|0\rangle \mapsto\propto |\psi^G_\star(u)\rangle$ is as simple as generating a uniform superposition over $[m_1]$, and then using $O(\log n)$ rounds of amplitude amplification to get inverse polynomially close to the uniform superposition over $[m_1]\setminus R_1$.

For $\ell \in \{1,\dots,k-1\}$, and $v^\ell_R \in V_\ell$, we again have $\Psi_\star(v^\ell_R) = \{|\psi^G_\star(v^\ell_R)\rangle\}$, where $|\psi^G_\star(v^\ell_R)\rangle$ is as in (108). To implement $U_\ell : |u\rangle|0\rangle \mapsto\propto |\psi^G_\star(u)\rangle$, we first compute (referring to Table 5 for the weights):

$$|u,0\rangle \mapsto\propto |u\rangle\left(-\sqrt{\mathsf{w}_\ell}|\leftarrow\rangle + \sqrt{\mathsf{w}^+_\ell}|\rightarrow\rangle\right)|0\rangle = |u\rangle\left(-(n/m_\ell)^{1/4}|\leftarrow\rangle + |\rightarrow\rangle\right)|0\rangle,$$

which can be implemented by a $O(1)$-qubit rotation. Then conditioned on $\leftarrow$, generate a uniform superposition over $j_\ell \in R_\ell(\mu(S^*_1),\dots,\mu(S^*_{\ell-1}),S^*_\ell)$ (we can learn the sets $S^*_1,\dots,S^*_\ell$ by seeing which sets are bigger, or assume we simply keep track of these values in some convenient

way), and then using $O(\log n)$ rounds of amplitude amplification to get inverse polynomially close to the superposition over such j_ℓ such that $d_R^{\rightarrow}(j_\ell) = 0$, which we can check by looking up j_ℓ in $C_\ell^{\rightarrow}(R)$. We have used the fact that our data structure supports taking a uniform superposition (see Section 2.3). Finally, conditioned on $\rightarrow$, generate a uniform superposition over $j_{\ell+1} \in [m_{\ell+1}]$, and use $O(\log n)$ rounds of amplitude amplification to get inverse polynomially close to a superposition over $j_{\ell+1} \in [m_{\ell+1}] \setminus R_{\ell+1}$.

For $\ell \in \{2, \ldots, k-1\}$ (the case for $\ell = 0$ is nearly identical), and $v_{R,j_\ell}^{\ell-1} \in V_{\ell-1}^+$, $\Psi_\star(v_{R,j_\ell}^{\ell-1})$ is a set of multiple states, as in (111). To implement $U_{\ell-1,+} : |v_{R,j_\ell}^{\ell-1}\rangle|\mathcal{I}_\ell\rangle \mapsto |\psi_\star^{\mathcal{I}_\ell}(v_{R,j_\ell}^{\ell-1})\rangle$ for all $\mathcal{I}_\ell \subsetneq [c_\ell]$, we note that each of these states is just $|v_{R,j_\ell}^{\ell-1}\rangle$ tensored with a constant-sized state depending only on $\mathcal{I}_\ell$, so we can implement $U_{\ell-1,+}$ in $O(1)$ time, by Lemma 5.5.

Finally, we implement U_k. For $v_{R,i_k}^k \in V_k$, we have $\Psi_\star(v_{R,i_k}^k) = \{|\psi_\star^G(v_{R,i_k}^k)\rangle\}$ as in (109), so implementing the map $U_k : |v_{R,i_k}^k\rangle|0\rangle \mapsto\propto |\psi_\star^G(v_{R,i_k}^k)\rangle \propto |v_{R,i_k}^k\rangle| \leftarrow\rangle$ is trivial.
We Thus, conclude that $U_\star$ can be implemented in polylog$(n) = \widetilde{O}(1)$ complexity. ■

5.3.5 Tail Bounds on Number of Collisions

If $\overline{R}_1, \ldots, \overline{R}_{k-1}$ were uniform random subsets of $A_1, \ldots, A_{k-1}$ respectively, it would be simple to argue that, for example, the number of collisions stored in $D_\ell(R)$ for any ℓ, which is a subset of $\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_\ell)$, is within a constant of the average, with high probability. Since $\overline{R}_\ell$ is instead chosen from A_ℓ by taking t_ℓ blocks of A_ℓ, and these blocks themselves are not uniform random, but rather chosen by a d-wise independent permutation for some $d = \text{polylog}(n)$, proving the necessary bounds, which are needed to upper bound the setup and transitions costs, is somewhat more subtle.

LEMMA 5.21. *For any $\ell', \ell \in \{1, \ldots, k-1\}$ where $\ell > \ell'$ and for any constant κ there exists a constant c such that the following holds. If $v_R^{\ell-1}$ is chosen uniformly at random from $V_{\ell-1}$, then for any fixed (non-random) $j \in [m_\ell]$ we have*

$$\Pr\left[\left|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'}, A_\ell^{(j)})\right| \geq c\frac{t_{\ell'}}{m_\ell}\log^{2^{\ell'-1}}(n)\right] \leq n^{-\kappa}.$$

PROOF. The proof proceeds by induction on ℓ'.
Base case: For $\ell' = 1$, we have R_1 is a uniform random subset of $[m_1]$ of size $\Theta(t_1)$ (for this proof, we ignore the partition of R_1 into $(R_1(S_1))_{S_1}$). Since A_1 has been partitioned the trivial way, where each block contains a single element, this means that $\overline{R}_1$ is a uniform random subset of A_1 of size $\Theta(t_1)$ as well. Hence, for any $\ell > 1$ and fixed $A_\ell^{(j)}$, $Z = |\mathcal{K}(\overline{R}_1, A_\ell^{(j)})|$ is a hypergeometric random variable, where we draw R_1 from $[m_1]$ and where we consider any $j_1 \in R_1$ marked whenever the unique element in $A_1^{(j_1)}$ is part of a collision in $\mathcal{K}(\overline{R}_1, A_\ell^{(j)})$. This means Z has parameters $N = m_1 = \Theta(n)$, $K = |\mathcal{K}(A_1, A_\ell^{(j)})|$ and $d = |R_1| = \Theta(t_1)$. As mentioned in Section 5.1, we may assume for any $A_\ell^{(j)}$ that $K = \Theta\left(\left|A_\ell^{(j)}\right|\right) = \Theta(n/m_\ell)$. Then for any constant c, we have

$c(t_1/m_\ell)\log n \geq 7Kd/N$ for sufficiently large n, so by Lemma 2.8:

$$\Pr\left[\left|\mathcal{K}(\overline{R}_1, A_\ell^{(j)})\right| \geq c\frac{t_1}{m_\ell}\log n\right] \leq e^{-c\frac{t_1}{m_\ell}\log n} = n^{-ct_1/m_\ell}.$$

Referring to Table 3, we have

$$\frac{t_1}{m_\ell} = \Theta\left(\frac{t_1}{t_{\ell-1}}\right) \geq \Omega\left(\frac{t_1}{t_1}\right) = \Omega(1).$$

Hence, we can choose c sufficiently large so that $n^{-ct_1/m_\ell} \leq n^{-\kappa}$, completing the base case.
Induction step: Suppose that for some $\ell' - 1 \in \{1, \ldots, k-2\}$ and all $\ell \in \{\ell', \ldots, k-1\}$ the lemma holds; i.e. for any fixed $j \in [m_\ell]$ and for any κ' there exists a constant c' such that

$$\Pr\left[\left|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'-1}, A_\ell^{(j)})\right| \geq c'\frac{t_{\ell'-1}}{m_\ell}\log^{2^{\ell'-2}}(n)\right] \leq n^{-\kappa'},$$

where R_1 is a uniform random subset of $[m_1]$ of size $\Theta(t_1)$, and for all $\ell'' \in \{2, \ldots, \ell'-1\}$, $R_{\ell''}$ is a uniform random subset of $[m_{\ell''}]$ of size $\Theta(t_{\ell''})$.

Now consider $Z = |\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'}, A_\ell^{(j)})|$, where $\ell > \ell'$ and where $R_{\ell'}$ is a uniform random subset of $[m_{\ell'}]$ of size $\Theta(r_{\ell'})$, and $A_\ell^{(j)}$ is still fixed. It is important to remark that this does not imply that $\overline{R}_{\ell'}$ is uniformly random, so instead we look at the blocks of $A_{\ell'}$. We say that any block $A_{\ell'}^{(j')}$ of $A_{\ell'}$ is *marked* if it collides with an ℓ'-collision in $\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'-1}, A_\ell^{(j)})$, and we let B_1 be the random variable that counts the number of such marked blocks in $A_{\ell'}$. Let $\mathcal{E}_1$ be the event that $|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'-1}, A_\ell^{(j)})| < c'\frac{t_{\ell'-1}}{m_\ell}\log^{2^{\ell'-2}}$, which happens with probability at least $1 - n^{-\kappa'}$, by the induction hypothesis. Assuming $\mathcal{E}_1$ directly implies an upper bound on B_1 of the form

$$B_1 \leq c'\frac{t_{\ell'-1}}{m_\ell}\log^{2^{\ell'-2}}(n). \tag{113}$$

Next we introduce a hypergeometric random variable B_2 that counts the number of marked blocks of $R_{\ell'}$, which we draw uniformly from $\{A_{\ell'}^{(j')}\}_{j' \in [m_{\ell'}]}$, which has B_1 marked blocks. Conditioned on event $\mathcal{E}_1$, this means that B_2 has parameters $N = m_{\ell'}$, $K = B_1 \leq c'\frac{t_{\ell'-1}}{m_\ell}\log^{2^{\ell'-2}}(n)$ (see (113)) and $d = |R_{\ell'}| = \Theta(t_{\ell'})$.

To relate $Z = |\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'}, A_\ell^{(j)})|$ to B_2, we need to analyse what the effect is of any marked block in $R_{\ell'}$ on the number of collisions in $|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'}, A_\ell^{(j)})|$. By the induction hypothesis we know that for any block $A_{\ell'}^{(j')}$ of $A_{\ell'}$ and each κ'', there exists a constant c'' such that:

$$\Pr\left[\left|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'-1}, A_{\ell'}^{(j')})\right| \geq c''\log^{2^{\ell'-2}}(n)\right] \leq n^{-\kappa''}.$$

If $|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'-1}, A_{\ell'}^{(j')})| < c''\log^{2^{\ell'-2}}(n)$ for every block $A_{\ell'}^{(j')}$ of $A_{\ell'}$, which we denote by event $\mathcal{E}_2$, then any marked block that gets added to $R_{\ell'}$ results in at most $c''\log^{2^{\ell'-2}}(n)$ collisions in $\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell'}, A_\ell^{(j)})$, so $Z \leq B_2 c''\log^{2^{\ell'-2}}(n)$. This implies that

$$\Pr\left[Z \geq cc''\frac{t_{\ell'}}{m_\ell}\log^{2^{\ell'-1}}(n)\middle|\mathcal{E}_1 \wedge \mathcal{E}_2\right] \leq \Pr\left[B_2 \geq c\frac{t_{\ell'}}{m_\ell}\log^{2^{\ell'-1}-2^{\ell'-2}} n\middle|\mathcal{E}_1\right]. \tag{114}$$

Since B_2 is a hypergeometric random variable, and $c\frac{t_{\ell'}}{m_\ell}\log^{2^{\ell'-1}-2^{\ell'-2}} n \geq 7dK/N$ for sufficiently large c, we can use Lemma 2.8 and $m_{\ell'} = \Theta\left(t_{\ell'-1}\right)$ to derive:

$$\Pr\left[B_2 \geq c\frac{t_{\ell'}}{m_\ell}\log^{2^{\ell'-1}-2^{\ell'-2}} n \middle| \mathcal{E}_1\right] \leq e^{-c\frac{t_{\ell'}}{m_\ell}\log^{2^{\ell'-1}-2^{\ell'-2}} n} \leq n^{-ct_{\ell'}/m_\ell}, \tag{115}$$

since $2^{\ell'-1} - 2^{\ell'-2} \geq 1$ whenever $\ell' \geq 2$. By Table 3, we have

$$\frac{t_{\ell'}}{m_\ell} = \Theta\left(\frac{t_{\ell'}}{t_{\ell-1}}\right) \geq \Omega\left(\frac{t_{\ell'}}{t_{\ell'}}\right) = \Omega(1).$$

Hence, by (114) and (115), we can choose c sufficiently large such that:

$$\Pr\left[Z \geq cc''\frac{t_{\ell'}}{m_\ell}\log^{2^{\ell'-1}}(n) \middle| \mathcal{E}_1 \wedge \mathcal{E}_2\right] \leq n^{-ct_{\ell'}/m_\ell} \leq n^{-\kappa'}. \tag{116}$$

The only thing left to do is to use the union bound to upper bound

$$\Pr\left[\neg\left(\mathcal{E}_1 \wedge \mathcal{E}_2\right)\right] \leq n^{-\kappa'} + m_{\ell'}n^{-\kappa''} \leq n^{-\kappa'} + n^{-\kappa''+1}, \tag{117}$$

where in the final inequality we have used the assumption from the lemma that $m_{\ell'} \leq n$. We can now combine (116) and (117), to conclude that for any κ we can choose $\kappa' > \kappa$ and $\kappa'' > \kappa+1$ and a constant $c_{\ell'}$ large enough such that:

$$\begin{aligned}\Pr\left[Z \geq c_\ell\frac{t_{\ell'}}{m_\ell}\log^{2^{\ell'-1}}(n)\right] &\leq \Pr\left[Z \geq c_\ell\frac{t_{\ell'}}{m_\ell}\log^{2^{\ell'-1}}(n) \middle| \mathcal{E}_1 \wedge \mathcal{E}_2\right] + \Pr\left[\neg\left(\mathcal{E}_1 \wedge \mathcal{E}_2\right)\right] \\ &\leq n^{-\kappa'} + n^{-\kappa'} + n^{-\kappa''+1} \leq n^{-\kappa}.\end{aligned}$$

■

COROLLARY 5.22. *For $\ell \in \{2, \ldots, k-2\}$, let $v_R^{\ell-1}$ be chosen uniformly at random from $V_{\ell-1}$. If the partition of $A_{\ell+1}$ into $\bigcup_{j\in[m_{\ell+1}]} A_{\ell+1}^{(j)}$ is chosen by a d-wise independent permutation for $d = \log^{2^{k-1}}(n)$ (see Section 5.1), then for all $j \in [m_\ell]$,*

$$\Pr\left[\left|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j)}, \overline{R}_{\ell+1})\right| \geq 1\right] \leq o(1),$$

and for any constant κ there exists a constant c such that:

$$\Pr\left[\left|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j)}, \overline{R}_{\ell+1})\right| \geq c\right] \leq n^{-\kappa},$$

where we note that $|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j)}, \overline{R}_{\ell+1})|$ is an upper bound on the number of potential faults if we add the block $A_\ell^{(j)}$ to $\overline{R}_\ell$.

PROOF. By Lemma 5.21, for any κ' there exists a c' large enough such that for each fixed $j \in [m_\ell]$ we have

$$\Pr\left[\left|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j)})\right| \geq c'\log^{2^{\ell-1}}(n)\right] \leq n^{-\kappa'}.$$

Let $i^1, \ldots, i^K \in A_{\ell+1}$ be all the points that collide with $\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j)})$, so w.h.p. there are

$$K < c'k\log^{2^{\ell-1}}(n) \leq \log^{2^{k-1}}(n) = d$$

of them, since each tuple in $\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j)})$ can collide with less than k other indices. Let $\tau : [n] \to [n]$ be the d-wise independent permutation used to choose the partitions of $[n]$, as in (43). Then $\{\tau(i^1), \ldots, \tau(i^K)\}$ is distributed as uniform set of size K, as long as $K \leq d$. In that case, the number of elements of $\{i^1, \ldots, i^K\}$ that are included in $\overline{R}_{\ell+1}$, Z, is a hypergeometric random variable with $N = |A_{\ell+1}| = \Theta(n)$, $K < c'k \log^{2^{\ell-1}}(n)$ as above, and $d = |\overline{R}_{\ell+1}| = \Theta\left(\frac{nt_{\ell+1}}{m_{\ell+1}}\right)$ draws. From Table 3, we have

$$\frac{dK}{N} = \Theta\left(\frac{t_{\ell+1}}{m_{\ell+1}} \log^{2^{\ell-1}}(n)\right) = \Theta\left(n^{-\frac{2^k-\ell-2}{2^k-1}} \log^{2^\ell-1}(n)\right) \leq n^{-\Omega(1)}.$$

Thus, there is some constant ϵ such that $dK/N \leq n^{-\epsilon}$, so for any constant c, we can use the union bound and Corollary 2.9 to get:

$$\Pr[Z \geq c] \leq 2e^c \left(cn^\epsilon\right)^{-c} = 2\left(\frac{e}{c}\right)^c n^{-\epsilon c} + n^{-\kappa'}.$$

Hence, by choosing $\kappa' > \kappa$ and c large enough we obtain $\Pr[Z \geq c] \leq n^{-\kappa}$. Moreover, we also see that for $c = 1$:

$$\Pr[Z \geq 1] \leq O(n^{-\epsilon} + n^{-\kappa'}) \leq o(1).$$ ■

COROLLARY 5.23. *Let v_R^ℓ be chosen uniformly at random from V_ℓ. For any constant κ, there exist a constant c such that:*

$$\Pr\left[|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_\ell)| \geq ct_\ell \log^{2^{\ell-2}}(n)\right] \leq n^{-\kappa}.$$

PROOF. By Lemma 5.21, for each fixed $j \in [m_\ell]$ and constant $\kappa' > \kappa + 1$ there exists a c large enough such that

$$\Pr\left[|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j)})| \geq c \log^{2^{\ell-2}}(n)\right] \leq n^{-\kappa'}.$$

By the union bound, the probability of this bad event happening for *any* $j \in R_\ell$ is at most $t_\ell n^{-\kappa'} \leq n^\kappa$, since $t_\ell < n$, from which the statement follows. ■

5.3.6 The Transition Subroutines

In this section we show how to implement the transition map $|u, i\rangle \mapsto |v, j\rangle$ for $(u, v) \in \overrightarrow{E}(G)$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$ (see Definition 2.3). We do this by exhibiting uniform (see Lemma 2.6) subroutines $\mathcal{S}_1, \ldots, \mathcal{S}_k, \mathcal{S}_{0,+}, \ldots, \mathcal{S}_{k-2,+}$ that implement the transitions in each of the edge sets $E_1, \ldots, E_k, E_0^+, \ldots, E_{k-2}^+$ defined in Section 5.3.3, whose union is $\overrightarrow{E}(G) \setminus \tilde{E}$. In Corollary 5.29, we will combine these to get a quantum subroutine (Definition 2.5) for the full transition map.

LEMMA 5.24. *For $\ell \in \{0, \ldots, k-2\}$, there is a subroutine $\mathcal{S}_{\ell,+}$ such that for all $(u, v) \in E_\ell^+$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, $\mathcal{S}_{\ell,+}$ maps $|u, i\rangle$ to $|v, j\rangle$ with error 0 in complexity $\mathsf{T}_{u,v} = \mathsf{T}_\ell^+ = \widetilde{O}(1)$.*

PROOF. The proof is identical to that of Lemma 5.7. ■

LEMMA 5.25. *There is a uniform subroutine $\mathcal{S}_1$ such that for all $(u,v) \in E_1$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, $\mathcal{S}_1$ maps $|u,i\rangle$ to $|v,j\rangle$ with error 0 in complexity $\mathsf{T}_{u,v} = \mathsf{T}_1 = \widetilde{O}(1)$.*

PROOF. The proof is identical to that of Lemma 5.8. ■

We now move on to $\mathcal{S}_\ell$, for $\ell \in \{2,\dots,k-1\}$, which is somewhat more complicated. For $(v_{R,j_\ell}^{\ell-1}, v_{R'}^{\ell}) \in E_\ell$, where $v_{R,j_\ell}^{\ell-1} \in V_{\ell-1}^{+}(S_1^*,\dots,S_{\ell-1}^*)$, meaning that R' is obtained from R by inserting j_ℓ into $R_\ell(\mu(S_1^*),\dots,\mu(S_{\ell-1}^*),S_\ell)$ for some S_ℓ, $\mathcal{S}_\ell$ should act as:

$$\begin{aligned} |v_{R,j_\ell}^{\ell-1}, S_\ell\rangle &\mapsto |v_{R'}^{\ell}, j_\ell\rangle \\ \equiv |((\ell-1,+),R,D(R),j_\ell),S_\ell\rangle &\mapsto |(\ell,R',D(R')),j_\ell\rangle. \end{aligned} \tag{118}$$

The complexity of this map depends on $|\mathcal{K}(\overline{R}_1,\dots,\overline{R}_{\ell-1},A_\ell^{(j_\ell)})|$, which is less than $p_\ell \in \mathrm{polylog}(n)$ whenever $(v_{R,j_\ell}^{\ell-1}, v_{R'}^{\ell}) \in E_\ell \setminus \tilde{E}_\ell = E_\ell \setminus \tilde{E}$ (see (99) and (104)). Lemma 5.26 below describes how to implement this transition map, up to some error, in that case. For the case when $(v_{R,j_\ell}^{\ell-1}, v_{R'}^{\ell}) \in \tilde{E}_\ell \subset \tilde{E}$ we let the algorithm fail.

LEMMA 5.26. *Fix any constant κ. For each $\ell \in \{2,\dots,k-1\}$, there is a uniform subroutine $\mathcal{S}_\ell$ that implements the transition map that maps $|u,i\rangle$ to $|v,j\rangle$ for all $(u,v) \in E_\ell \setminus \tilde{E}$ with error $O(n^{-\kappa})$, in complexity $\mathsf{T}_{u,v} = \mathsf{T}_\ell = \widetilde{O}(\sqrt{n/m_\ell})$.*

PROOF. Suppose $u = v_{R,j_\ell}^{\ell} \in V_{\ell-1}^{+}(S_1^*,\dots,S_{\ell-1}^*)$. We can compute the values $S_1^*,\dots,S_{\ell-1}^*$ by checking which sets are larger, or just keeping track of these values in some convenient way, as they are chosen. Then to implement the map in (118), we need to insert j_ℓ into $R_\ell(\mu(S_1^*),\dots,\mu(S_{\ell-1}^*),S_\ell)$ to obtain R'_ℓ, update $D(R)$ to obtain $D(R')$, uncompute S_ℓ by checking which part of R_ℓ has size $t_\ell+1$, and increment the first register by mapping $|\ell-1,+\rangle \mapsto |\ell\rangle$. All of these take $\mathrm{polylog}(n)$ complexity, except for updating $D(R)$, which we now describe.

By (82), we know that $D(R)$ consists of sets $\{D_{\ell'}(R)\}_{\ell'=1}^{k-1}$ where each $D_{\ell'}(R)$ contains a subset of $\mathcal{K}(\overline{R}_1,\dots,\overline{R}_{\ell'})$ (see (77)). When we go from R to R', we need to update each of these to account for any collisions involving indices $i_\ell \in A_\ell^{(j_\ell)}$ that should be recorded in $D_{\ell'}(R')$. For $\ell' < \ell$, we can see that $D_{\ell'}(R) = D_{\ell'}(R')$, since $D_{\ell'}$ only depends on $R_1,\dots,R_{\ell'}$, which are unchanged. For $\ell' > \ell$, the existence of any $(i_1,\dots,i_{\ell-1},i_\ell,i_{\ell+1},\dots,i_{\ell'},x_{i_1}) \in D_{\ell'}(R')$ (which we would now need to find and add) implies $d_R^{\rightarrow}(j_\ell) > 0$ (see (80)), and this is not true for any $(v_{R,j_\ell}^{\ell-1}, v_{R'}^{\ell}) \in E_\ell$ (see (98)). Thus, we only need to find any tuples $(i_1,\dots,i_\ell,x_{i_1})$ such that $i_\ell \in A_\ell^{(j_\ell)}$ that belong in $D_\ell(R')$. By (77), such a tuple should be added to $D_\ell(R')$ if and only if $(i_1,\dots,i_{\ell-1},x_{i_1}) \in D_{\ell-1}(R_{\ell-1}(\mu(S_1^*),\dots,\mu(S_{\ell-2}^*),S_{\ell-1}))$ such that $x_{i_\ell} = x_{i_1}$ for some $S_{\ell-1}$ such that $\mu(S_{\ell-1}^*) \in S_{\ell-1}$.

We search for $i_\ell \in A_\ell^{(j_\ell)}$ such that if we look up x_{i_ℓ} in $D_{\ell-1}(R_{\ell-1}(\mu(S_1^*),\dots,\mu(S_{\ell-2}^*),S_{\ell-1}))$ for some $S_{\ell-1}$ containing $\mu(S_{\ell-1}^*)$, we get back a non-empty set of values $(i_1,\dots,i_{\ell-1},x_{i_\ell})$. For any such value found, we add $(i_1,\dots,i_\ell,x_{i_\ell})$ to $D_\ell(R_\ell(\mu(S_1^*),\dots,\mu(S_{\ell-1}^*),S_{\ell-1}))$. This increments the value of $\bar{d}_{R'}^{\rightarrow}(i_{\ell-1})$, and so if $j_{\ell-1} \in R_{\ell-1}$ is such that $i_{\ell-1} \in A_{\ell-1}^{(j_{\ell-1})}$ (we can compute $j_{\ell-1}$

from $i_{\ell-1}$ in $\widetilde{O}(1)$, see Section 5.1), we have incremented the forward collision degree of $j_{\ell-1}$, $d_{R'}^{\rightarrow}(j_{\ell-1})$. We must therefore update the entry in $C_{\ell-1}^{\rightarrow}$ for $j_{\ell-1}$. We look up $j_{\ell-1}$, and if nothing is returned, insert $(j_{\ell-1}, 0)$. If $(j_{\ell-1}, N)$ is returned, remove it and insert $(j_{\ell-1}, N+1)$. We repeat this quantum search procedure, which is uniform, until we find $p_\ell = \mathrm{polylog}(n)$ values i_ℓ, or no new i_ℓ is returned for $\kappa \log n$ times. Since we are assuming that the number of such collisions is less than p_ℓ, since $(u, v) \in E_\ell \setminus \tilde{E}_\ell$, this finds all collisions with error $O(n^{-\kappa})$, in complexity $\widetilde{O}\left(\sqrt{|A_\ell^{(j_\ell)}|}\right) = \widetilde{O}(\sqrt{n/m_\ell})$. ■

We have the following corollary of the results in Section 5.3.5.

COROLLARY 5.27. *For any constant κ, there exists a choice of constants $\{c_\ell\}_{\ell=2}^{k-2}$ in the definition of $\tilde{E}'_\ell$ ((100)) and polylogarithmic functions $\{p_\ell\}_{\ell=1}^{k-1}$ in the definition of $\tilde{E}_\ell$ ((99)) large enough such that*

$$\widetilde{\mathcal{W}} := \sum_{e \in \tilde{E}} \mathsf{w}_e = O(n^{-\kappa} \mathcal{W}(G)).$$

PROOF. Fix $\ell \in \{2, \ldots, k-1\}$. Let $v_R^{\ell-1}$ be uniform random on $V_{\ell-1}$ (so R is uniform on its support, see (85) and (86)). Then by Lemma 5.21, for any $j_\ell \in [m_\ell]$, if $p_\ell \in \mathrm{polylog}(n)$ is sufficiently large,

$$\Pr[|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j_\ell)})| \geq p_\ell] \leq n^{-\kappa}. \tag{119}$$

Referring to (99), this implies that

$$|\tilde{E}_\ell| \leq n^{-\kappa} |\{(u, v) : u \in V_{\ell-1}^+, v \in L^+(u)\}| = n^{-\kappa} |V_{\ell-1}^+|.$$

For $\ell \in \{2, \ldots, k-2\}$, by Corollary 5.22, if c_ℓ is a sufficiently large constant,

$$\Pr\left[|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_{\ell-1}, A_\ell^{(j_\ell)}, \overline{R}_{\ell+1})| \geq c_\ell\right] \leq n^{-\kappa},$$

which implies that $|\mathcal{I}(v_{R,j_\ell})| \geq c_\ell$ with probability at most $n^{-\kappa}$ (see (92) and (94)). Referring to (100), this implies that

$$|\tilde{E}'_\ell| \leq n^{-\kappa} |\{(u, v) : u \in V_{\ell-1}^+, v \in L^+(u)\}| = n^{-\kappa} |V_{\ell-1}^+|.$$

Since $\tilde{E}_{k-1} = \emptyset$, the above also holds for $\ell = k-1$.

Combining these, and using the definition of $\tilde{E}$ in (104), and that $|V_{\ell-1}^+| = \Theta(|E_\ell| + |\tilde{E}'_\ell|)$, since each vertex in $V_{\ell-1}^+$ has constant out-degree, we have:

$$\widetilde{\mathcal{W}} = \sum_{\ell=2}^{k-1} \mathsf{w}_\ell |V_{\ell-1}^+| \leq 2n^{-\kappa} \sum_{\ell=2}^{k-1} \mathsf{w}_\ell O(|E_\ell| + |\tilde{E}'_\ell|) = O\left(n^{-\kappa} \sum_{e \in \overrightarrow{E}(G)} \mathsf{w}_e\right) = O(n^{-\kappa} \mathcal{W}(G)).$$ ■

LEMMA 5.28. *There is a uniform subroutine $\mathcal{S}_k$ such that for all $(u, v) \in E_k$ with $i = f_u^{-1}(v)$ and $j = f_v^{-1}(u)$, $\mathcal{S}_k$ maps $|u, i\rangle$ to $|v, j\rangle$ with error 0 in complexity $\mathsf{T}_{u,v} = \mathsf{T}_k = \widetilde{O}(1)$.*

PROOF. The proof is identical to that of Lemma 5.7. ■

We combine the results of this section into the following.

COROLLARY 5.29. *Let κ be any constant. There is a quantum subroutine (Definition 2.5) that implements the full transition map with errors $\epsilon_e \leq n^{-\kappa}$ for all $e \in \overrightarrow{E}(G) \setminus \tilde{E}$, and times: $\mathsf{T}_e = \mathsf{T}_1 = \widetilde{O}(1)$ for all $e \in E_1$; $\mathsf{T}_e = \mathsf{T}^+_\ell = \widetilde{O}(1)$ for all $e \in E^+_\ell$, for all $\ell \in \{0, \ldots, k-2\}$; $\mathsf{T}_e = \mathsf{T}_\ell = \widetilde{O}(\sqrt{n/m_\ell})$ for all $e \in E_\ell$, for all $\ell \in \{2, \ldots, k-1\}$; and $\mathsf{T}_e = \mathsf{T}_k = \widetilde{O}(1)$ for all $e \in E_k$.*

PROOF. This follows from combining Lemma 5.24, Lemma 5.25, Lemma 5.26 and Lemma 5.28 using Lemma 2.7. ■

5.3.7 Initial State and Setup Cost

The initial state will be the uniform superposition over V_0:

$$|\sigma\rangle := \sum_{v^0_R \in V_0} \frac{1}{\sqrt{|V_0|}} |v^0_R\rangle.$$

LEMMA 5.30. *The state $|\sigma\rangle$ can be generated with error $O(n^{-\kappa})$ for any constant κ in complexity*

$$\mathsf{S} = \widetilde{O}\left(t_1 + t_2\sqrt{\frac{n}{t_1}} + \cdots + t_{k-1}\sqrt{\frac{n}{t_{k-2}}}\right).$$

PROOF. Fix $p \in \mathrm{polylog}(n)$ and a constant c. We start by taking a uniform superposition over all $R_1 \in \binom{[m_1]}{t_1(2^{c_1}-1)}$ and querying each $\overline{R}_1$ to get $D_1(R)$, which costs $\widetilde{O}(t_1)$ (with log factors coming from the cost of inserting everything into data structures as in Section 2.3). For $\ell \in \{2, \ldots, k-1\}$, we take a uniform superposition over all sets $R_\ell \in \binom{[m_\ell]}{t_\ell(c_1 \cdots c_{\ell-1}(2^{c_\ell}-1))}$. The total cost so far is $\widetilde{O}(t_1 + t_2 \cdots + t_{k-1})$.

Next, we need to populate the rest of the data structure:

For each $\ell \in \{2, \ldots, k-1\}$, do the following.

For each $(s_1, \ldots, s_{\ell-1}, S_\ell) \in [c_1] \times \cdots \times [c_{\ell-1}] \times (2^{[c_\ell]} \setminus \{\emptyset\})$, do the following.

Repeat until pt_ℓ values i_ℓ have been found, or $c \log n$ repetitions have passed in which no i_ℓ was found:

Search for a new value $i_\ell \in R_\ell(s_1, \ldots, s_{\ell-1}, S_\ell)$ such that there exists

$$(i_1, \ldots, i_{\ell-1}, x_{i_\ell}) \in D_{\ell-1}(R_{\ell-1}(s_1, \ldots, s_{\ell-2}, S_{\ell-1}))$$

for some $S_{\ell-1}$ containing $s_{\ell-1}$. If such an i_ℓ is found, insert the tuple $(i_1, \ldots, i_\ell, x_{i_\ell})$ into $D_\ell(R_\ell(s_1, \ldots, s_{\ell-1}, S_\ell)$, and increment the forward collision degree of $j_{\ell-1}$ such that $i_{\ell-1} \in A^{(j_{\ell-1})}_{\ell-1}$ stored in $C^{\rightarrow}_{\ell-1}(R)$, as described in the proof of Lemma 5.26.

If the inner loop finds $Y \in [pt_\ell]$ values, so $Y = \widetilde{O}(t_\ell)$, it costs at most (up to polylogarithmic factors):

$$\sum_{y=0}^{Y-1} \sqrt{\frac{|R_\ell|}{Y-y}} = \sqrt{\frac{t_\ell n}{m_\ell}} \sum_{y=1}^{Y} \frac{1}{\sqrt{y}} = \Theta\left(\sqrt{\frac{t_\ell n Y}{m_\ell}}\right) = \widetilde{O}\left(t_\ell \sqrt{\frac{n}{m_\ell}}\right),$$

since $|R_\ell| = \Theta(t_\ell n/m_\ell)$. Since $k, c_1, \ldots, c_{k-1}$ are all constant, there are $\widetilde{O}(1)$ loops in total, so the total cost of this procedure is $O(\sqrt{n/m_\ell})$ for a total cost of:

$$\widetilde{O}\left(\sum_{\ell=1}^{k-1} t_\ell + \sum_{\ell=2}^{k-1} t_\ell \sqrt{\frac{n}{m_\ell}}\right) = \widetilde{O}\left(t_1 + \sum_{\ell=1}^{k-2} t_{\ell+1} \sqrt{\frac{n}{t_\ell}}\right)$$

since $t_\ell = \Theta(m_{\ell+1})$ and for all $\ell > 1$, $t_\ell = o(t_1)$ (see Table 3).

In parts of the superposition in which there are more than pt_ℓ collisions to be found in some inner loop, we have failed to correctly populate the data $D(R)$, and so the state is not correct. We now argue that this represents a very small part of the state. For uniform random sets $\overline{R}_1, \ldots, \overline{R}_{k-1}$, we could argue that the expected number of ℓ-collisions in $\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_\ell)$ is $\Theta(t_\ell)$, and use a hypergeometric tail inequality to upper bound the proportion of R for which this failure occurs. Things are more complicated, since the sets $\overline{R}_\ell$ for $\ell > 1$ are not uniform on all possible sets – they are composed instead of blocks. However, by Corollary 5.23, for every $\ell \in \{2, \ldots, k-1\}$, if v_R^ℓ is uniform random on V_ℓ, meaning $R_1, \ldots, R_{k-1}$ are uniform random sets, but $\overline{R}_1, \ldots, \overline{R}_{k-1}$ have limited support, we still have the necessary tail bound, when c' is a sufficiently large constant:

$$\Pr\left[|\mathcal{K}(\overline{R}_1, \ldots, \overline{R}_\ell)| \geq t_\ell c' \log^{2^{\ell-2}}(n)\right] \leq n^{-\kappa}.$$

Thus, choosing $p = c' \log^{2^{\ell-1}}(n)$, the state we generate is $O(n^{-\kappa})$-close to $|\sigma\rangle$. ■

5.3.8 Positive Analysis

For the positive analysis, we must exhibit a flow (see Definition 2.2) on G whenever $M \neq \emptyset$.

LEMMA 5.31. *There exists some $\mathcal{R}^{\mathsf{T}} = O(|V_0|^{-1})$ such that the following holds. Whenever there is a unique k-collision $(a_1, \ldots, a_k) \in A_1 \times \cdots \times A_k$, there exists a flow θ on G that satisfies conditions **P1-P5** of Theorem 3.10. Specifically:*

1. *For all $e \in \tilde{E}$, $\theta(e) = 0$.*
2. *For all $u \in V(G) \setminus (V_0 \cup V_k)$ and $|\psi_\star(u)\rangle \in \Psi_\star(u)$,*
$$\sum_{i \in L^+(u)} \frac{\theta(u, f_u^+(i))\langle\psi_\star(u)|u,i\rangle}{\sqrt{\mathsf{w}_{u,i}}} - \sum_{i \in L^-(u)} \frac{\theta(u, f_u^-(i))\langle\psi_\star(u)|u,i\rangle}{\sqrt{\mathsf{w}_{u,i}}} = 0.$$
3. $\sum_{u \in V_0} \theta(u) = 1$.
4. $\sum_{u \in V_0} \frac{|\theta(u) - \sigma(u)|^2}{\sigma(u)} \leq 1$.
5. $\mathcal{E}^{\mathsf{T}}(\theta) \leq \mathcal{R}^{\mathsf{T}}$.

PROOF. Recall the definition of M from (105). For $\ell \in \{2,\dots,k-1\}$, let $j_\ell^* \in [m_\ell]$ be the unique block label such that $a_\ell \in A_\ell^{(j_\ell^*)}$. Then $a_\ell \in \overline{R}_\ell(s_1,\dots,s_{\ell-1},S_\ell)$ if and only if $j_\ell^* \in R_\ell(s_1,\dots,s_{\ell-1},S_\ell)$.

Assuming $M \neq \emptyset$, we define a flow θ on G with all its sinks in M. It will have sources in both V_0 and M, but all other vertices will conserve flow. This will imply **Item 2** for all *correct* star states of G, but we take extra case to ensure that **Item 2** is satisfied for the additional star states in $\Psi_\star(u) : u \in \bigcup_{\ell=0}^{k-2} V_\ell^+$. We define θ from V_0 to V_k as follows.

$\mathcal{R}_0^+$, Item 3, and Item 4: We define M_0 as the set of $v_R^0 \in V_0$ such that for all $\ell \in \{2,\dots,k-1\}$, we have $|\mathcal{K}(\overline{R}_1,\dots,\overline{R}_{\ell-1},A_\ell^{(j_\ell^*)})| < p_\ell$, where p_ℓ is as in Corollary 5.27, and for all $\ell \in \{2,\dots,k-2\}$, $|\mathcal{K}(\overline{R}_1,\dots,\overline{R}_{\ell-1},A_\ell^{(j_\ell^*)},\overline{R}_{\ell+1})| = 0$. We define the flow θ over the edges in E_0^+ as

$$\theta\left(v_R^0, v_{R,j_1}^0\right) = \begin{cases} \frac{1}{|M_0|} & \text{if } v_R^0 \in M_0 \text{ and } A^{(j_1)} = \{a_1\}, \\ 0 & \text{else.} \end{cases}$$

That is, each vertex in M_0 has a unique outgoing edge with flow, and the flow is uniformly distributed. From this construction we immediately satisfy **Item 3**.

By Corollary 5.22, we know that the proportion of vertices $v_R^0 \in V_0$ that are excluded from M_0 because $|\mathcal{K}(\overline{R}_1,\dots,\overline{R}_{\ell-1},A_\ell^{(j_\ell^*)},\overline{R}_{\ell+1})| \geq 1$ is $o(1)$. By (119), the proportion of vertices excluded because $|\mathcal{K}(\overline{R}_1,\dots,\overline{R}_{\ell-1},A_\ell^{(j_\ell^*)})| \geq p_\ell$ is also $o(1)$. Hence, we can compute:

$$\frac{|V_0|}{|M_0|} = \left(1 + O\left(\frac{t_1}{n}\right)\right)\Pi_{\ell=2}^{k-1}\left(1 + O\left(\frac{t_\ell}{m_\ell}\right)\right)(1+o(1)) = 1 + o(1).$$

Since $\sigma(u) = \frac{1}{|V_0|}$, we can conclude with **Item 4** of the theorem statement:

$$\sum_{u\in V_0} \frac{|\theta(u)-\sigma(u)|^2}{\sigma(u)} = |V_0|^2\left(\frac{1}{|M_0|} - \frac{1}{|V_0|}\right)^2 = \left(\frac{|V_0|}{|M_0|} - 1\right)^2 = o(1).$$

Recall we want to compute $\mathcal{E}^{\mathsf{T}}(\theta) = \mathcal{E}(\theta^{\mathsf{T}})$ (see Definition 2.4), which treats an edge e as a path of length T_e. Using $\mathsf{T}_0^+ = \widetilde{O}(1)$ and $\mathsf{w}_0^+ = 1$ (refer to Table 5), the contribution of the edges in E_0^+ to the energy of the flow can be computed as:

$$\mathcal{R}_0^+ = \sum_{e\in E_0^+} \mathsf{T}_0^+ \frac{\theta(e)^2}{\mathsf{w}_0^+} = \widetilde{O}\left(\sum_{u\in M_0} \frac{1}{|M_0|^2}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right), \tag{120}$$

since each vertex in M_0 has a unique outgoing edge with flow and the flow is uniformly distributed.

$\mathcal{R}_1$ and Item 2 (partially): Let M_0^+ be the set of $v_{R,j_1}^0 \in V_0^+$ such that $v_R^0 \in M_0$ and $A^{(j_1)} = \{a_1\}$, so $\left|M_0^+\right| = |M_0|$. These are the only vertices in V_0^+ that have incoming flow, which is equal to $\frac{1}{|M_0|}$. Note that no fault can occur when we add a_1 to R_1 because we have ensured that $a_2 \notin \overline{R}_2$; that is $\mathcal{I}(v_{R,a_1}^0) = \emptyset$, and so by Lemma 5.18, $\mathsf{w}_{v_{R,a_1}^0,S_1} = \mathsf{w}_1 = 1$ for all $S_1 \in 2^{[c_1]} \setminus \{\emptyset\}$. To ensure that

we satisfy **Item 2** we define the flow as

$$\theta(v^0_{R,j_1}, v^1_{R'}) = \begin{cases} (-1)^{|S_1|+1}\frac{1}{|M_0|} & \text{if } v^0_{R,j_1} \in M_0^+ \text{ and } v^1_{R'} = f^+_{v^0_{R,j_1}}(S_1), \\ 0 & \text{else} \end{cases}$$

where we recall that $v^1_{R'} = f^+_{v^0_{R,j_1}}(S_1)$ if and only if R' is obtained from R by inserting j_1 into $R_1(S_1)$. We verify that indeed for each $u = v^0_{R,a_1} \in M_0^+$ and $|\psi^{\mathcal{I}_1}_\star(u)\rangle \in \Psi_\star(u)$ (see (110)) Item 2 holds:

$$\begin{aligned} \Theta_\star(\mathcal{I}_1, u) &:= \sum_{i \in L^+(u)} \theta(u, f^+_u(i)) \frac{\langle \psi^{\mathcal{I}_1}_\star(u) | u, i\rangle}{\sqrt{\mathsf{w}_1}} - \sum_{i \in L^-(u)} \theta(u, f^-_u(i)) \frac{\langle \psi^{\mathcal{I}_1}_\star(u) | u, i\rangle}{\sqrt{\mathsf{w}^+_0}} \\ &= \sum_{S_1 \in 2^{[c_1]\setminus \mathcal{I}_1}\setminus\{\emptyset\}} \theta(u, f^+_u(S_1)) \frac{\sqrt{\mathsf{w}_1}}{\sqrt{\mathsf{w}_1}} - \theta(u, f^-_u(\leftarrow)) \frac{-\sqrt{\mathsf{w}^+_0}}{\sqrt{\mathsf{w}^+_0}}. \end{aligned} \tag{121}$$

We have $f^-_u(\leftarrow) = v^0_R \in V_0$, and $\theta(v^0_{R,a_1}, v^0_R) = -\theta(v^0_R, v^0_{R,a_1}) = -|M_0|^{-1}$, and $\theta(u, f^+_u(S_1)) = (-1)^{|S_1|}|M_0|^{-1}$, so we continue from above:

$$\begin{aligned} \Theta_\star(\mathcal{I}_1, u) &= \sum_{S_1 \in 2^{[c_1]\setminus \mathcal{I}_1}\setminus\{\emptyset\}} (-1)^{|S_1|+1}|M_0|^{-1} - (-|M_0|^{-1})(-1) \\ &= -|M_0|^{-1}\left(\sum_{S_1 \in 2^{[c_1]\setminus \mathcal{I}_1}} (-1)^{|S_1|} - 1 + 1\right) = 0, \end{aligned} \tag{122}$$

since $\sum_{S_1 \in 2^{[c_1]\setminus \mathcal{I}_1}} (-1)^{|S_1|} = 0$ (i.e. for any set S, exactly half of its subsets have even size). Using $\mathsf{T}_1 = \widetilde{O}(1)$ and $\mathsf{w}_1 = 1$, the contribution of the edges in E_1 to the energy of the flow can be upper bounded as:

$$\mathcal{R}_1 = \sum_{e \in E_1} \mathsf{T}_1 \frac{\theta(e)^2}{\mathsf{w}_1} = \widetilde{O}\left(\sum_{u \in M_0^+, S_1 \in 2^{[c_1]}\setminus\{\emptyset\}} \frac{1}{|M_0|^2}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right). \tag{123}$$

$\mathcal{R}^+_\ell$ for $\ell \in \{1, \ldots, k-2\}$: Let $M_\ell(S_1, \ldots, S_\ell)$ be the set of $v^\ell_R \in V_\ell(S_1, \ldots, S_\ell)$ (see (85)) such that $a_1 \in R_1(S_1)$, for all $\ell' \in \{2, \ldots, \ell\}$, $j^*_\ell \in R_\ell(\mu(S_1), \ldots, \mu(S_{\ell-1}), S_\ell)$, and

$$v^0_{R_1\setminus\{a_1\}, R_2\setminus\{j^*_1\}, \ldots, R_\ell\setminus\{j^*_\ell\}, R_{\ell+1}, \ldots, R_{k-1}} \in M_0.$$

Then letting M_ℓ be the union of all $M_\ell(S_1, \ldots, S_\ell)$, we have $|M_\ell| = \Theta(|M_0|)$. We will define θ so that M_ℓ are exactly the vertices of V_ℓ that have non-zero flow coming in from $V^+_{\ell-1}$, and specifically, we will ensure that the amount of incoming flow for each $v^\ell_R \in M_\ell(S_1, \ldots, S_\ell)$ is $(-1)^{|S_1|+\cdots+|S_\ell|+\ell}|M_0|^{-1}$. So far this can only be verified for $\ell = 1$ due to the flow that we constructed on E_1, but it will follow for all $\ell \in \{2, \ldots, k-1\}$ when we define the flow on E_ℓ (see (125)). For now, we define the flow θ over the edges in E^+_ℓ as

$$\theta(v^\ell_R, v^\ell_{R,j_\ell}) = \begin{cases} (-1)^{|S_1|+\cdots+|S_\ell|+\ell}\frac{1}{|M_0|} & \text{if } v^\ell_R \in M_\ell(S_1, \ldots, S_\ell) \text{ and } j_\ell = j^*_\ell, \\ 0 & \text{else,} \end{cases}$$

so we are just forwarding all flow from v_R^ℓ to a unique neighbour $v_{R,j_\ell^*}^\ell$. Using $\mathsf{T}_\ell^+ = \widetilde{O}(1)$ and $\mathsf{w}_\ell^+ = 1$, the contribution of the edges in E_ℓ^+ to the energy of the flow can be upper bounded as:

$$\mathcal{R}_\ell^+ = \sum_{e\in E_\ell^+} \mathsf{T}_\ell^+ \frac{\theta(e)^2}{\mathsf{w}_\ell^+} = \widetilde{O}\left(\sum_{u\in M_\ell^+} \frac{1}{|M_0|^2}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right). \tag{124}$$

$\mathcal{R}_\ell$ for $\ell \in \{2,\dots,k-1\}$ and Item 2 (continued): Let $M_{\ell-1}^+(S_1,\dots,S_{\ell-1})$ be the set of $v_{R,j_\ell^*}^{\ell-1} \in V_{\ell-1}^+(S_1,\dots,S_{\ell-1})$ such that $v_R^{\ell-1} \in M_{\ell-1}(S_1,\dots,S_{\ell-1})$, so letting $M_{\ell-1}^+$ be the union over all the sets $M_{\ell-1}^+(S_1,\dots,S_{\ell-1})$, $\left|M_{\ell-1}^+\right| = O\left(|M_{\ell-1}|\right) = O\left(|M_0|\right)$. $M_{\ell-1}^+(S_1,\dots,S_{\ell-1})$ are exactly the vertices of $V_{\ell-1}^+$ that have non-zero flow coming in from $M_{\ell-1}(S_1,\dots,S_\ell)$. For any $v_{R,j_\ell^*}^{\ell-1} \in M_{\ell-1}^+(S_1,\dots,S_{\ell-1})$, this flow is equal to $(-1)^{|S_1|+\cdots+|S_{\ell-1}|+(\ell-1)}|M_0|^{-1}$. Note that no fault can occur when we add j_ℓ^* to R, because we have ensured in our definition of M_0 that $\mathcal{K}(\overline{R}_1,\dots,\overline{R}_{\ell-1},A_\ell^{(j_\ell^*)},\overline{R}_{\ell+1}) = \emptyset$, so we have $\mathcal{I}\left(v_{R,j_\ell^*}^{\ell-1}\right) = \emptyset$, so by (101), there is an edge for each $S_\ell \in 2^{[c_\ell]} \setminus \{\emptyset\}$ to which we can assign flow. To ensure that we satisfy **Item 2** we define the flow as

$$\theta(v_{R,j_\ell}^{\ell-1}, v_{R'}^\ell) = \begin{cases} (-1)^{|S_1|+\cdots+|S_\ell|+\ell}\frac{1}{|M_0|} & \text{if } v_{R,j_\ell} \in M_{\ell-1}^+(S_1,\dots,S_{\ell-1}) \text{ and } v_{R'}^\ell = f^+_{v_{R,j_\ell}^{\ell-1}}(S_\ell), \\ 0 & \text{else,} \end{cases} \tag{125}$$

where we recall that for $v_{R,j_\ell}^{\ell-1} \in V_{\ell-1}^+(S_1,\dots,S_{\ell-1})$, $v_{R'}^\ell = f^+_{v_{R,j_\ell}^{\ell-1}}(S_\ell)$ if and only if R' is obtained from R by inserting j_ℓ into $R_\ell(\mu(S_1),\dots,\mu(S_{\ell-1}),S_\ell)$. Note that this is consistent with the incoming flow we assumed when defining θ on the edges in $E_{\ell-1}^+$, above. We verify that for each $u = v_{R,j_\ell^*}^{\ell-1} \in M_{\ell-1}^+(S_1,\dots,S_{\ell-1})$ and $|\psi_\star^{\mathcal{I}_\ell}(u)\rangle \in \Psi_\star(u)$ (see (111)), Item 2 holds. By a computation nearly identical to (121) and (122), we obtain:

$$\begin{aligned} &\sum_{i\in L^+(u)} \theta(u, f_u^+(i)) \frac{\langle\psi_\star^{\mathcal{I}_\ell}(u)|u,i\rangle}{\sqrt{\mathsf{w}_\ell}} - \sum_{i\in L^-(u)} \theta(u, f_u^-(i)) \frac{\langle\psi_\star^{\mathcal{I}_\ell}(u)|u,i\rangle}{\sqrt{\mathsf{w}_{\ell-1}^+}} \\ &= (-1)^{|S_1|+\cdots+|S_{\ell-1}|+\ell}\left(\sum_{S_\ell\in 2^{[c_\ell]}\setminus\mathcal{I}_\ell} (-1)^{|S_\ell|} - 1 - (-1)\right) = 0. \end{aligned}$$

Using $\mathsf{T}_\ell = \widetilde{O}(\sqrt{n/m_\ell})$ and $\mathsf{w}_\ell = \sqrt{n/m_\ell}$ (see Table 5), we can upper bound the contribution of the edges in E_ℓ to the energy of the flow:

$$\mathcal{R}_\ell = \sum_{e\in E_\ell} \mathsf{T}_\ell \frac{\theta(e)^2}{\mathsf{w}_\ell} = \widetilde{O}\left(\sum_{u\in M_{\ell-1}^+, S_\ell\in 2^{[c_\ell]}\setminus\{\emptyset\}} \frac{1}{|M_0|}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right). \tag{126}$$

$\mathcal{R}_k$: Finally, let $M_{k-1}(S_1,\dots,S_{k-1})$ be the set of $v_R^{k-1} \in V_{k-1}(S_1,\dots,S_{k-1})$ (see (85)) such that, firstly, $a_1 \in R_1(S_1)$; secondly, for all $\ell \in \{2,\dots,k-1\}$, $j_\ell^* \in R_\ell(\mu(S_1),\dots,\mu(S_{\ell-1}),S_\ell)$; and finally, $v^0_{R_1\setminus\{a_1\},R_2\setminus\{j_2^*\},\dots,R_{k-1}\setminus\{j_{k-1}^*\}} \in M_0$. We let M_{k-1} be the union of all $M_{k-1}(S_1,\dots,S_{k-1})$. These are exactly the vertices of V_{k-1} that have non-zero incoming flow, with the amount of incoming

flow equal to $(-1)^{|S_1|+\cdots+|S_{k-1}|+(k-1)}|M_0|^{-1}$. We define the flow θ on the edges in E_k as

$$\theta(v_R^{k-1}, v_{R,i_k}^k) = \begin{cases} (-1)^{|S_1|+\cdots+|S_{k-1}|+(k-1)} \frac{1}{|M_0|} & \text{if } v_R^k \in M_{k-1}(S_1,\ldots,S_{k-1}) \text{ and } i_k = a_k \\ 0 & \text{else.} \end{cases}$$

It is easy to verify that the only vertices $v_{R,i_k}^k \in V_k$ that have non-zero flow are those in M, and thus all sources and sinks are in $V_0 \cup M$ (M contains some sources, because some vertices have negative flow coming in). Using $\mathsf{T}_k = \widetilde{O}(1)$ and $\mathsf{w}_k = 1$, the contribution of the edges in E_k to the energy of the flow is:

$$\mathcal{R}_k = \sum_{e\in E_k} \mathsf{T}_k \frac{\theta(e)^2}{\mathsf{w}_k} = \widetilde{O}\left(\sum_{u\in M_k} \frac{1}{|M_k|^2}\right) = \widetilde{O}\left(\frac{1}{|M_0|}\right). \tag{127}$$

Item 1: Recall that $\tilde{E} := \bigcup_{\ell=2}^{k-1}(\tilde{E}_\ell \cup \tilde{E}'_\ell)$ ((104)), where $\tilde{E}_\ell \subset E_\ell$ ((99)). By ensuring that there is only flow on $v^\ell_{R_1,\ldots,R_{k-1}}$ whenever $\mathcal{K}(\overline{R}_1,\ldots,\overline{R}_{\ell-1}, A_\ell^{(j_\ell^*)})$ is not too big, we have ensured that the flow on the edges in $\bigcup_{\ell=2}^{k-1}\tilde{E}_\ell$ is 0, and by only sending flow down edges that are part of E_ℓ, which is disjoint from $\tilde{E}_{\ell'}$ ((100)), the flow on $\bigcup_{\ell=2}^{k-1}\tilde{E}_{\ell'}$ is 0 as well, which implies that the flow on all of $\tilde{E}$ is 0.

Item 5: It remains only to upper bound the energy of the flow by adding up the contributions in (120), (123), (124), (126) and (127):

$$\mathcal{E}^{\mathsf{T}}(\theta) = \mathcal{R}_0^+ + \mathcal{R}_1 + \sum_{\ell=1}^{k-2}\mathcal{R}_\ell^+ + \sum_{\ell=2}^{k-1}\mathcal{R}_\ell + \mathcal{R}_k = \widetilde{O}\left(\frac{1}{|M_0|}\right).$$

Substituting $|M_0| = \Theta(|V_0|)$ yields the desired upper bound. ∎

5.3.9 Negative Analysis

For the negative analysis, we need to upper bound the total weight of the graph, taking into account the subroutine complexities: $\mathcal{W}^{\mathsf{T}}(G)$.

LEMMA 5.32. *There exists $\mathcal{W}^{\mathsf{T}}$ such that*

$$\mathcal{W}^{\mathsf{T}}(G) \le \mathcal{W}^{\mathsf{T}} \le \widetilde{O}\left(\left(n + \sum_{\ell=1}^{k-1}\frac{n^2}{t_\ell}\right)|V_0|\right).$$

PROOF. Recall that $\mathcal{W}^{\mathsf{T}}(G) = \mathcal{W}(G^{\mathsf{T}})$ is the total weight of the graph G^{T}, where we replace each edge e of G, with weight w_e, by a path of T_e edges of weight w_e, where T_e is the complexity of the edge transition e (see Definition 2.4 and **TS1-2** of Theorem 3.10). Thus, $\mathcal{W}^{\mathsf{T}}(G) = \sum_{e\in E(G)} \mathsf{T}_e\mathsf{w}_e$. By Corollary 5.29 (see also Table 5) $\mathsf{T}_e = \widetilde{O}(1)$ for all $e \in E_1 \cup E_k \cup \bigcup_{\ell\in\{0,\ldots,k-2\}} E_\ell^+$ and $\mathsf{T}_e = \widetilde{O}(\sqrt{n/m_\ell})$ for all $e \in E_\ell$ for $\ell \in \{2,\ldots,k-1\}$. We have defined the weight function (see Table 5) so that $\mathsf{w}_e = 1$ for all $e \in E_1 \cup E_k \cup \bigcup_{\ell\in\{0,\ldots,k-2\}} E_\ell^+$ and $\mathsf{w}_e = \sqrt{n/m_\ell}$ for all $e \in E_\ell$ for $\ell \in \{2,\ldots,k-1\}$.

Thus, using (97), the total contribution to the weight from the edges in E_0^+ is:

$$\mathcal{W}_0^+ := \mathsf{T}_0^+ \mathsf{w}_0^+ \left|E_0^+\right| = \widetilde{O}\left(n\left|V_0\right|\right). \tag{128}$$

For $\ell \in \{1, \ldots, k-2\}$, we can use (90) to compute the total contribution to the weight from the edges in E_ℓ^+:

$$\mathcal{W}_\ell^+ := \mathsf{T}_\ell^+ \mathsf{w}_\ell^+ \left|E_\ell^+\right| = \widetilde{O}\left(n\left|V_0\right|\right). \tag{129}$$

Using (97), the total contribution from the edges in E_1 is:

$$\mathcal{W}_1 := \mathsf{T}_1 \mathsf{w}_1 \left|E_1\right| = \widetilde{O}\left(n\left|V_0\right|\right). \tag{130}$$

For $\ell \in \{2, \ldots, k-1\}$ using (102) the total contribution from the edges in E_ℓ is:

$$\mathcal{W}_\ell := \mathsf{T}_\ell \mathsf{w}_\ell \left|E_\ell\right| = \widetilde{O}\left(\frac{n}{m_\ell}\right)\left|E_\ell\right| = \widetilde{O}\left(\frac{n^2}{m_\ell}\left|V_0\right|\right). \tag{131}$$

Finally, using (103), the total contribution from the edges in E_k is:

$$\mathcal{W}_k := \mathsf{T}_k \mathsf{w}_k \left|E_k\right| = \widetilde{O}\left(\frac{n^2}{t_{k-1}}\left|V_0\right|\right). \tag{132}$$

Combining (128) to (132), we get total weight:

$$\mathcal{W}^{\mathsf{T}}(G) = \widetilde{O}\left(\left(n + \sum_{\ell=1}^{k-2} n + n + \sum_{\ell=2}^{k-1} \frac{n^2}{m_\ell} + \frac{n^2}{t_{k-1}}\right)\left|V_0\right|\right) = \widetilde{O}\left(\left(n + \sum_{\ell=1}^{k-1} \frac{n^2}{t_\ell}\right)\left|V_0\right|\right),$$

using $m_\ell = \Theta(t_{\ell-1})$ for all $\ell \in \{2, \ldots, k-1\}$. ■

5.3.10 Conclusion of Proof of Theorem 5.16

We can now conclude with the proof of Theorem 5.16, showing an upper bound of $\widetilde{O}\left(n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^k-1}}\right)$ on the bounded error quantum time complexity of k-distinctness.

PROOF OF THEOREM 5.16. We apply Theorem 3.10 to G (Section 5.3.3 and Section 5.3.2), M ((105)), σ the uniform distribution on V_0 ((83)), and $\Psi_\star$ (Section 5.3.4), with

$$\mathcal{W}^{\mathsf{T}} = \widetilde{O}\left(\left(n + \sum_{\ell=1}^{k-1} \frac{n^2}{t_\ell}\right)\left|V_0\right|\right) \text{ and } \mathcal{R}^{\mathsf{T}} = \widetilde{O}\left(\left|V_0\right|^{-1}\right).$$

Then we have, referring to Table 3,

$$\mathcal{W}^{\mathsf{T}}\mathcal{R}^{\mathsf{T}} = \widetilde{O}\left(n + \sum_{\ell=1}^{k-1} \frac{n^2}{t_\ell}\right) = o(n^2).$$

We have shown the following:

Setup Subroutine: By Lemma 5.30, the state $|\sigma\rangle$ can be generated in cost

$$\mathsf{S} = \widetilde{O}\left(t_1 + \sum_{\ell=1}^{k-2} t_{\ell+1}\sqrt{\frac{n}{t_\ell}}\right).$$

Star State Generation Subroutine: By Lemma 5.20, the star states $\Psi_\star$ can be generated in $\widetilde{O}(1)$ complexity.

Transition Subroutine: By Corollary 5.29, there is a quantum subroutine that implements the transition map with errors $\epsilon_{u,v}$ and costs $\mathsf{T}_{u,v}$, such that

TS1 For all $(u,v) \in \overrightarrow{E}(G) \setminus \tilde{E}$ (defined in (104)), taking $\kappa > 2$ in Lemma 5.26, we have $\epsilon_{u,v} = O(n^{-\kappa}) = o(1/(\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}}))$.

TS2 By Corollary 5.27, Lemma 5.32 and using $\kappa > 2$:

$$\widetilde{\mathcal{W}} = O(n^{-\kappa}\mathcal{W}^{\mathsf{T}}(G)) = o(1/\mathcal{R}^{\mathsf{T}}).$$

Checking Subroutine: By (106), for any $u \in V_{\mathsf{M}} = V_k$, we can check if $u \in M$ in cost $\widetilde{O}(1)$.

Positive Condition: By Lemma 5.31, there exists a flow satisfying conditions **P1**-**P5** of Theorem 3.10, with $\mathcal{E}^{\mathsf{T}}(\theta) \leq \mathcal{R}^{\mathsf{T}} = \widetilde{O}\left(|V_0|^{-1}\right)$.

Negative Condition: By Lemma 5.32, $\mathcal{W}^{\mathsf{T}}(G) \leq \mathcal{W}^{\mathsf{T}} = \widetilde{O}\left(\left(n + \sum_{\ell=1}^{k-1} \frac{n^2}{t_\ell}\right)|V_0|\right)$.

Thus, by Theorem 3.10, there is a quantum algorithm that decides if $M = \emptyset$ in bounded error in complexity:

$$\widetilde{O}\left(\mathsf{S} + \sqrt{\mathcal{R}^{\mathsf{T}}\mathcal{W}^{\mathsf{T}}}\right) = \widetilde{O}\left(t_1 + \sum_{\ell=1}^{k-2} t_{\ell+1}\sqrt{\frac{n}{t_\ell}} + \sqrt{n} + \sum_{\ell=1}^{k-1}\frac{n}{\sqrt{t_\ell}}\right) = \widetilde{O}\left(t_1 + \sum_{\ell=1}^{k-2} t_{\ell+1}\sqrt{\frac{n}{t_\ell}} + \sqrt{n} + \frac{n}{\sqrt{t_{k-1}}}\right)$$

since $t_1 > t_2 > \cdots > t_{k-1}$. Choosing the optimal values of $t_\ell = n^{\frac{3}{4} - \frac{1}{4}\frac{1}{2^k-1} - \sum_{\ell'=2}^{\ell}\frac{2^{k-1-\ell'}}{2^k-1}}$ for $\ell \in \{1,\ldots,k-1\}$, as in Table 3, we get an upper bound of $\widetilde{O}\left(n^{\frac{3}{4}-\frac{1}{4}\frac{1}{2^k-1}}\right)$. Since $M \neq \emptyset$ if x has a unique k-collision, and $M = \emptyset$ if x has no k-collision, the algorithm distinguishes these two cases. By Lemma 5.1, this is enough to decide k-distinctness in general. ■

Acknowledgements

We thank Arjan Cornelissen and Maris Ozols for discussions on the early ideas of this work; and Simon Apers, Aleksandrs Belovs, Shantanav Chakraborty, Andrew Childs, Frédéric Magniez and Ronald de Wolf for helpful comments and discussions about these results.

References

[1] **Scott Aaronson and Yaoyun Shi**. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, 2004. DOI (2)

[2] **Miklós Ajtai**. A non-linear time lower bound for Boolean branching programs. *Theory of Computing*, 1:149–176, 2005. DOI (1)

[3] **Andris Ambainis**. Quantum search with variable times. *Theory of Computing Systems*, 47:786–807, 2010. DOI ePrint (15)

[4] **Andris Ambainis**. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007. Earlier version in FOCS'04. DOI ePrint (2, 3, 55)

[5] **Andris Ambainis**, **András Gilyén**, **Stacey Jeffery**, **and Martins Kokainis**. Quadratic speedup for finding marked vertices by quantum walks. *Proceedings of the 52nd ACM Symposium on the Theory of Computing (STOC)*, pages 412–424, 2020. DOI ePrint (4)

[6] **Simon Apers**, **Shantanav Chakraborty**, **Leonardo Novo**, **and Jérémie Roland**. Quadratic speedup for spatial search by continuous-time quantum walk. *Physical review letters*, 129(16):160502, 2022. DOI (3, 44)

[7] **Simon Apers**, **András Gilyén**, **and Stacey Jeffery**. A unified framework of quantum walk search. *Proceedings of the 38th Symposium on Theoretical Aspects of Computer Science (STACS)*, 6:1–6:13, 2020. DOI ePrint (2, 4, 6, 7)

[8] **Yosi Atia and Shantanav Chakraborty**. Improved upper bounds for the hitting times of quantum walks. *Physical Review A*, 104:032215, 2021. DOI ePrint (8, 43)

[9] **Aleksandrs Belovs**. Learning-graph-based quantum algorithm for k-distinctness. *Proceedings of the 53rd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 207–216, 2012. DOI ePrint (2, 10, 57, 74, 76)

[10] **Aleksandrs Belovs**. Quantum walks and electric networks, 2013. DOI (2, 23)

[11] **Aleksandrs Belovs**. Span programs for functions with constant-sized 1-certificates. *Proceedings of the 44th ACM Symposium on the Theory of Computing (STOC)*, pages 77–84, 2012. DOI (6)

[12] **Aleksandrs Belovs**, **Andrew M. Childs**, **Stacey Jeffery**, **Robin Kothari**, **and Frédéric Magniez**. Time-efficient quantum walks for 3-distinctness. *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 105–122, 2013. DOI (2, 57)

[13] **Ethan Bernstein and Umesh Vazirani**. Quantum complexity theory. *SIAM Journal on Computing*, 26:1411–1473, 5, 1997. DOI (43)

[14] **Gilles Brassard**, **Peter Høyer**, **Michele Mosca**, **and Alain Tapp**. Quantum amplitude amplification and estimation, *Quantum Computation and Quantum Information: A Millennium Volume*. Volume 305, Contemporary Mathematics Series, pages 53–74. AMS, 2002. DOI ePrint (1)

[15] **Gilles Brassard**, **Peter Høyer**, **and Alain Tapp**. Quantum algorithm for the collision problem. *ACM SIGACT News*, 28:14–19, 1997. DOI ePrint (1)

[16] **Harry Buhrman**, **Christoph Dürr**, **Mark Heiligman**, **Peter Høyer**, **Frédéric Magniez**, **Miklos Santha**, **and Ronald de Wolf**. Quantum algorithms for element distinctness. *SIAM Journal on Computing*, 34(6):1324–1330, 2005. Earlier version in CCC'01. DOI ePrint (1)

[17] **Harry Buhrman**, **Bruno Loff**, **Subhasree Patro**, **and Florian Speelman**. Limits of quantum speed-ups for computational geometry and other problems: fine-grained complexity via quantum walks. *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022. DOI (16)

[18] **Mark Bun**, **Robin Kothari**, **and Justin Thaler**. The polynomial method strikes back: tight quantum query bounds via dual polynomials. *Proceedings of the 50th ACM Symposium on the Theory of Computing (STOC)*, 2018. DOI ePrint (2)

[19] **Ashok K. Chandra**, **Prabhakar Raghavan**, **Walter L. Ruzzo**, **Roman Smolensky**, **and Prasoon Tiwari**. The electrical resistance of a graph captures its commute and cover times. *Computational Complexity*, 6(4):312–340, 1996. DOI (4)

[20] **Andrew M Childs and Jason M Eisenberg**. Quantum algorithms for subset finding. *Quantum Information & Computation*, 5(7):593–604, 2005. DOI (2)

[21] **Andrew M. Childs**, **Richard Cleve**, **Enrico Deotto**, **Edward Farhi**, **Sam Gutmann**, **and Daniel A. Spielman**. Exponential algorithmic speedup by a quantum walk. *Proceedings of the 35th ACM Symposium on the Theory of Computing (STOC)*, pages 59–68, 2003. DOI ePrint (3, 7, 8, 42, 43)

[22] **Andrew M. Childs**, **Stacey Jeffery**, **Robin Kothari**, **and Frédéric Magniez**. A time-efficient quantum walk for 3-distinctness using nested updates, 2013. DOI (2)

[23] **Arjan Cornelissen**, **Stacey Jeffery**, **Maris Ozols**, **and Alvaro Piedrafita**. Span programs and quantum time complexity. *Proceedings of the 45th International Symposium on Mathematical Foundations of Computer Science (MFCS)*, 21:1–26:14, 2020. DOI ePrint (15)

[24] **Craig Costello**, **Patrick Longa**, **and Michael Naehrig**. Efficient algorithms for supersingular isogeny diffie-hellman. *Advances in Cryptology (CRYPTO 2016)*, pages 572–601, 2016. DOI (2)

[25] **Svante. Janson**, **Tomasz. Luczak**, **and Andrzej Rucinski**. Random Graphs. Wiley Series in Discrete Mathematics and Optimization. John Wiley & Sons, 2011. DOI (17)

[26] **Stacey Jeffery**. Frameworks for Quantum Algorithms. PhD thesis, University of Waterloo, 2014. URL (2)

[27] **Stacey Jeffery and Sebastian Zur**. Multidimensional quantum walks. *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, pages 1125–1130. ACM, 2023. DOI (1)

[28] **Eyal Kaplan**, **Moni Naor**, **and Omer Reingold**. Derandomized constructions of k-wise (almost) independent permutations. *Algorithmica*, 55(1):113–133, 2009. DOI (17)

[29] **Alexei Kitaev**. Quantum np. *Talk at AQIP*, 99, 1999. (36)

[30] **Alexei Y. Kitaev**. Quantum measurements and the Abelian stabilizer problem. *ECCC*, TR96-003, 1996. DOI (18, 22)

[31] **Troy Lee**, **Rajat Mittal**, **Ben W. Reichardt**, **Robert Špalek**, **and Mário Szegedy**. Quantum query complexity of state conversion. *Proceedings of the 52nd IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 344–353, 2011. DOI ePrint (18)

[32] **Frédéric Magniez**, **Ashwin Nayak**, **Jérémie Roland**, **and Miklos Santha**. Search via quantum walk. *SIAM Journal on Computing*, 40(1):142–164, 2011. Earlier version in STOC'07. DOI ePrint (2, 4)

[33] **Nikhil S. Mande**, **Justin Thaler**, **and Shuchen Zhu**. Improved Approximate Degree Bounds for k-Distinctness. *Proceedings of the 15th Conference on the Theory of Quantum Computation, Communication, and Cryptography (TQC)*, volume 158, 2:1–2:22, 2020. DOI ePrint (2)

[34] **Mario Szegedy**. Quantum speed-up of Markov chain based algorithms. *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 32–41, 2004. DOI ePrint (2, 3)

[35] **Seiichiro Tani**. Claw finding algorithms using quantum walk. *Theoretical Computer Science*, 410(50):5285–5297, 2009. DOI (2)