A Quantum Time-Space Tradeoff for Directed st-Connectivity

Stacey Jeffery[1,2] and Galina Pass[2]

[1]CWI, Amsterdam
[2]QuSoft & University of Amsterdam

Abstract

Directed st-connectivity (DSTCON) is the problem of deciding if there exists a directed path between a pair of distinguished vertices s and t in an input directed graph. This problem appears in many algorithmic applications, and is also a fundamental problem in complexity theory, due to its NL-completeness. We show that for any $S \geq \log^2(n)$, there is a quantum algorithm for DSTCON using space S and time $T \leq 2^{\frac{1}{2} \log(n) \log(n/S) + o(\log^2(n))}$, which is an (up to quadratic) improvement over the best classical algorithm for any $S = o(\sqrt{n})$. Of the S total space used by our algorithm, only $O(\log^2(n))$ is quantum space – the rest is classical. This effectively means that we can trade off classical space for quantum time.

1 Introduction

In the directed st-connectivity problem (DSTCON), the input is a directed graph on n vertices with two distinguished vertices s and t, and the goal is to decide if there is a directed path from s to t. This fundamental problem underlies a wide range of applications in (e.g.) logistics, databases [AHV95, Ull89], compilers [ALSU06, NNH99], and model checking [CGP99, BK08].

In addition to its practical applications, DSTCON plays a central role in space-bounded complexity theory. It is NL-complete (under NC^1 reductions), where NL is the class of problems decidable by nondeterministic logspace machines. Consequently, understanding its complexity has broad implications for space-bounded computation. For example, a classical (or quantum) algorithm for DSTCON using $O(\log(n))$ space would show that L, the class of problems solvable in $O(\log(n))$ space (or its quantum analogue) contains NL – a major breakthrough in either case.

Currently, the smallest space complexity of any classical or quantum algorithm for DSTCON is $S = O(\log^2(n))$, achieved by Savitch's (classical) algorithm. However, Savitch's algorithm achieves this low space complexity at the expense of a large *quasipolynomial* time complexity, using

$$T \leq 2^{\log^2(n) + O(\log(n))}$$

steps of computation. In contrast, a simple breadth-first search (BFS) algorithm solves this problem in just $T = O(n^3)$ steps (in the adjacency-matrix model), but at the expense of a much larger $S = \widetilde{O}(n)$ space requirement. The fundamental nature of this problem, as well as its many applications, motivates understanding the best possible *tradeoff* between the time and space needed to solve it. Progress was made by Barnes, Buss, Ruzzo and Schieber [BBRS98], who gave a classical algorithm that runs in time

$$T \leq 2^{\log^2(\frac{n}{S}) + O(\log n \log \log n)} \tag{1}$$

given any space $S \geq \log^2(n)$. Their approach combines a breadth-first search with a clever recursive algorithm.

For quantum algorithms, the study of space-bounded complexity is even more well motivated, as quantum memories are expected to be limited in size for the foreseeable future. It is an important question in which memory regimes we can still achieve speedups over classical algorithms, and time-space tradeoffs are a key part of this picture. We can make such a tradeoff even more useful by

distinguishing between the *quantum* space and *classical* (or total) space needed by the algorithm, as quantum space is the scarce resource.

For DSTCON, quantum speedups are known only at the two extreme regimes of space. In the high-space setting, Dürr, Heiligman, Høyer, and Mhalla [DHHM06] gave an $\widetilde{O}(n^{1.5})$-time, $\widetilde{O}(n)$-space algorithm using quantum search to build a spanning tree. In the low-space setting, Ref. [JP25] recently gave a quadratic quantum speedup over Savitch's algorithm, running in time $T \leq 2^{\frac{1}{2}\log^2 n + O(\log n)}$ with $S = O(\log^2 n)$ space. Between these two extremes, however, no quantum improvements over classical tradeoffs were known.

For the *undirected* variant (USTCON), the picture is much clearer, at least as far as quantum algorithms go: in both the adjacency-matrix and edge-list models of graph access[1], quantum algorithms achieve optimal time and space simultaneously [BR12, AJPW23].

However, existing classical and quantum approaches face significant obstacles in the directed case. A random walk on a directed graph, starting from s, may fail to find t even when t is reachable from s, if the walk leads from s to some part of the graph from which t is not reachable. Quantum walks, which are powerful tools for studying undirected graphs, do not generalize to directed graphs. Moreover, known classical algorithms for DSTCON such as the time-space tradeoff algorithm in [BBRS98] cannot be directly quantized. This stems from the fact that these classical algorithms are not reversible, and making them reversible using standard methods increases the space complexity. These difficulties highlight why directed connectivity remains a challenging and subtle problem. They also show that progress requires algorithmic ideas that go beyond random or quantum walks, and straightforward adaptations of classical tradeoffs.

Our Contribution: We present the first nontrivial quantum time-space tradeoff for DSTCON, by designing a new quantum algorithm for this problem that, for any space bound $S \geq \log^2(n)$, runs in time
$$T \leq 2^{\frac{1}{2}\log(n)\log(\frac{n}{S}) + O(\log n \log \log n)}.$$

In particular, our result yields a quantum speedup over the best known classical algorithm (see (1)) in the regime $S = o(n^{1/2})$. We also show that, of the S space, the required *quantum* space is always $O(\log^2(n))$, which makes this result much more applicable to quantum computers with a limited number of qubits. This effectively means that we can tradeoff classical space for quantum time. We formally state this result in Theorem 3.6.

Our Techniques: The classical algorithm of Barnes et al. [BBRS98] achieves a time-space tradeoff for DSTCON by combining breadth-first search with a recursive subroutine that decides, for any pair of vertices u and v, if there is a directed path from u to v of length at most L. We call this problem DIST_L. The breadth-first search saves on space by not traversing the whole graph, but only those vertices at a distance from s that is a multiple of L (the space then decreases as L increases). The subroutine is used to find these vertices in a manner that is more space efficient, but less time efficient, than BFS.

The BFS portion of this classical algorithm could be sped up using quantum search techniques, but this saves at most a polynomial factor in the time (and nothing in the space), which is not very interesting in the small-space regime (where we get our quantum improvement) where such polynomial factors are hidden by the $O(\log(n))$ in the exponent. On the other hand, a direct quantization of their subroutine for DIST_L fails, in large part because this subroutine is not reversible, and making it so would increase the space complexity. Moreover, the subroutine calls itself recursively, with significant depth of recursion. A quantum speedup of this subroutine would most likely have bounded error. Naively composing bounded-error quantum subroutines to depth d results in $\log^d$ factors, which can be significant. Recent techniques for composing bounded error quantum algorithms without log-factor overhead [BJY24, BJ25] could reduce this overhead to c^d for some constant, but this could still be significant, if $c > 1$. Overcoming these limitation requires a fundamentally different approach.

---

[1]In this work, unless otherwise stated, we use the adjacency-matrix model, which assumes the input is given via queries to an adjacency matrix. This distinction is only significant in high-space regimes where the time complexity is polynomial.
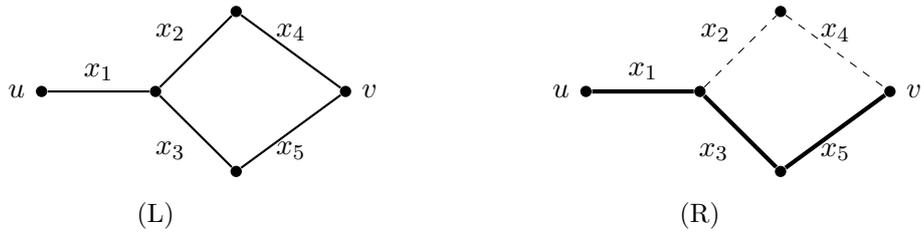
Figure 1: (L) An example of a switching network. (R) The same switching network, with edges switched on (thick) or off (dashed) by the assignment $x = 10101$. In this example, u and v are connected by a path of edges labelled by variables that are true under x, and so the switching network accepts x.

To address this, we design a new quantum algorithm for DIST_L that is based on a recursively constructed *switching network*. A switching network (Definition 2.5) is an undirected graph with *terminals* u and v, whose edges are labeled by Boolean variables $\{x_1, \ldots, x_m\}$, that can switch the edges "on" or "off", by their truth value. Such a network naturally defines a Boolean function $f : \{0,1\}^m \to \{0,1\}$, with $f(x) = 1$ (we then say the switching network *accepts x*) if and only if u and v are connected by a path consisting of edges whose labels are true under the assignment x. An example is given in Figure 1.

Switching networks are a natural and well-motivated model in classical computing, and have been used to study classical space-bounded complexity (see, e.g., [Pot15]). They also give a simple way of designing quantum algorithms, as any switching network can be compiled into a quantum algorithm for the associated function, whose time and space complexity depend on certain properties of the switching network. Quantum algorithms for evaluating switching networks were developed in [JK17] using span program techniques inspired by [BR12], and a subsequent work [JJKP18] provided a tight analysis of these span program algorithms for arbitrary switching networks, in terms of *query complexity*. These quantum algorithms were only explicitly related to the classical model of switching networks in [JP25], where more detailed techniques were developed for efficiently implementing this type of quantum algorithm, and rigorously analyzing their time complexity.

Switching networks lend themselves well to quantum algorithms in part because they are a naturally reversible structure, being defined by undirected graphs. Moreover, they give a natural way of defining recursive quantum algorithms: by defining a switching network recursively, and turning the final product into a quantum algorithm, bounded error is introduced only once, when turning the switching network into a quantum algorithm, and there are no factors of c^d or $\log^d$ on the complexity of the algorithm, even when the depth of recursion in the switching network is d.

Because we keep the more space-intensive outer BFS algorithm classical, most of the space needed by the algorithm is classical. For any S, we only need $O(\log^2(n))$ qubits of *quantum* space to implement our quantum subroutine for DIST_L, the remainder of the S space is classical.

Our algorithm for DIST_L provides a compelling example of how switching networks can be used within quantum algorithms. It exploits the full power of the classical switching network model, while at the same time extending it into a setting where time-space tradeoffs become essential. It is the first time switching networks have been used to prove a quantum time-space tradeoff. Beyond its role in our algorithm for DSTCON, our algorithm for DIST_L may be of independent interest as a quantum primitive for other space-efficient graph algorithms.

Open Problems: Our speedup over the best known classical algorithm is quadratic when $S = \log^2(n)$, and gets worse as S increases towards $\sqrt{n}$, at which point we no longer achieve any speedup. However, we do know there is a quadratic speedup at the other extreme, $S = n$. It is thus very natural to hope that we might get a quadratic speedup over the algorithm of [BBRS98] for *all* S. One difficulty in this is that our algorithm is not just a quantization of the algorithm of [BBRS98], but actually does something different.

As mentioned earlier, for the undirected variant (USTCON), quantum algorithms achieve optimal time and space simultaneously [BR12, AJPW23]. This naturally raises the question of whether a

3

similar result is possible in the directed case.

We have already mentioned the compelling open problem of showing a $O(\log(n))$-space quantum algorithm for DSTCON. This would necessarily be a polynomial-time algorithm, and so it would, in some rough sense, achieve the above goal of have optimal space and time simultaneously. It would also show that the quantum analogue of L (logspace) is contained in NL (nondeterministic logspace). Ref. [AE25] made some progress on this question by showing that a promise version of DSTCON in which the input has few paths can be solved in $O(\log(n))$ quantum space. We remark that any $o(\log^2(n))$ upper bound on the quantum space complexity of DSTCON would be interesting.

Organization: The remainder of this paper is organized as follows. In Section 2, we give the necessary preliminaries on graph theory, switching networks, and quantum algorithms. In Section 3, we present our main result, a quantum algorithm for DSTCON with a parameter L that can be used to tune the classical space complexity of the algorithm. The key technical building block of this algorithm is a quantum subroutine for the problem DIST_L, which we describe in Section 4.

2 Preliminaries

Here we state preliminaries. In Section 2.1, we state some results about efficiently generating quantum superpositions. In Section 2.2, we define both directed and undirected graphs, as well as the notation we will use to talk about them. In Section 2.3, we formally define switching networks, and state results about quantum algorithms for evaluating them. Part of instantiating such a quantum algorithm requires generating the states of a basis for a certain graph-theoretic space called the *cut space*, so in Section 2.4, we develop some theory that helps with this.

First, we state a few simple conventions and notational definitions. Unless otherwise specified, logarithms are with respect to base 2. We let $\bar{0}$ denote the all-zeros string, which throughout this paper will always have length $\log n$. We use $\widetilde{O}(f(n))$ to suppress $\text{poly}(\log n)$ factors, where n is always the number of vertices in the input digraph. So, for example, $\widetilde{O}(1)$ is $\text{polylog}(n)$.

2.1 Generating superpositions

As part of our algorithms, we will often require a subroutine that makes a superposition over strings, with non-uniform amplitudes. The following lemma gives conditions under which we can do that efficiently.

Lemma 2.1 ([GR02])**.** *Fix $\alpha \in \mathbb{R}^{\{0,1\}^m}$, and suppose there is a subroutine that can compute, given any prefix $p \in \{0,1\}^{\leq m}$, the partial sum of entries of α with prefix p, $S(p) = \sum_{s \in \{0,1\}^m : s \text{ has prefix } p} \alpha_s^2$, in time T. Then the state $\frac{1}{\sqrt{\sum_s \alpha_s^2}} \sum_s \alpha_s |s\rangle$ can be prepared in time $O(Tm)$.*

Remark 2.2 (Larger alphabets)**.** *The result of Lemma 2.1 can be easily generalized to the case when indices are length-m strings over an alphabet A of constant size $d = |A|$ (e.g., $A = \{0,1,2\}$). In this case, p is a prefix over A, and the controlled binary rotations from the proof of [GR02] are replaced by d-way branching rotations. Since d is constant, the total complexity remains $O(Tm)$.*

2.2 Graphs

A *directed graph* $G = (V(G), E(G))$, or *digraph*, consists of a vertex set $V(G) = \{v_1, \ldots, v_n\}$ and an edge set $E(G) \subseteq \{(u,v) \in V(G)^2 : u \neq v\}$. We say v is *reachable* from u in G if there exists a path $u = u_0, \ldots, u_\ell = v$ such that for all $i \in [\ell]$, $(u_{i-1}, u_i) \in E(G)$. When G is clear from context, we just write V and E.

We are interested in solving the *st-connectivity* – or reachability – problem on directed graphs, but we will also make use of undirected graphs, in the context of switching networks, described shortly.

An undirected graph $\mathcal{N}$ is just a directed graph, where we don't care about the direction of the edge – and in fact, we will ultimately want to assign to each edge an arbitrary orientation, for convenience, so we could even take the same definition we used for directed graphs, and just change notions like

reachability. However, for our purpose an undirected graph will ultimately define a quantum algorithm on a space spanned by the *edges* of $\mathcal{N}$. We therefore take a somewhat edge-centric definition, which has the added bonus of accommodating multigraphs.

Definition 2.3. *An* undirected graph $\mathcal{N} = (V(\mathcal{N}), E(\mathcal{N}))$ *is specified by a pair of finite sets of* vertex labels, V *and* edge labels, E; *as well as incidence sets* $E_\mathcal{N}(\mathsf{u}) = E_\mathcal{N}^\leftarrow(\mathsf{u}) \sqcup E_\mathcal{N}^\rightarrow(\mathsf{u})$ *for each* $\mathsf{u} \in V$, *such that for all* $e \in E$, *there are unique distinct vertices* $\mathsf{u}, \mathsf{v} \in V$ *such that* $e \in E^\rightarrow(\mathsf{u}) \cap E^\leftarrow(\mathsf{v})$. *We think of* $E^\rightarrow(\mathsf{u})$ *as the set of edges* coming out *of* u, *and* $E^\leftarrow(\mathsf{u})$ *as the set of edges* going into u. *When* $\mathcal{N}$ *is clear from context, we omit it from the notation, as we have just demonstrated.*

This definition gives us the freedom to specify $e \in E$ however we want to, not necessarily by its endpoints, as $e = (\mathsf{u}, \mathsf{v})$. To recover this usual edge set, we can map e to (u, v) such that $e \in E^\rightarrow(\mathsf{u}) \cap E^\leftarrow(\mathsf{v})$. When this holds for some edge, we will abuse notation by writing $(\mathsf{u}, \mathsf{v}) \in E$, to mean "there is an edge from u to v in $\mathcal{N}$." Note that while each edge does have an orientation, these are arbitrary and for convenience only. We say that u is *connected to* v in $\mathcal{N}$ (or *reachable from* v) if there is a $\mathsf{u}\mathsf{v}$-*path* $\mathsf{u} = \mathsf{u}_0, \ldots, \mathsf{u}_\ell = \mathsf{v}$ such that for all $i \in [\ell]$, $(\mathsf{u}_{i-1}, \mathsf{u}_i) \in E$ or $(\mathsf{u}_i, \mathsf{u}_{i-1}) \in E$.

Finally, we will often build a (undirected) graph $\mathcal{N}$ out of graphs $\mathcal{N}_1$ and $\mathcal{N}_2$ by "gluing" (identifying) some of the vertices in $\mathcal{N}_1$ with some vertices in $\mathcal{N}_2$. The following definition makes precise what we mean by this.

Definition 2.4. *Let* $\mathcal{N}_1$ *and* $\mathcal{N}_2$ *be graphs, with* $\mathsf{u}_{1,1}, \ldots, \mathsf{u}_{1,r} \in V(\mathcal{N}_1)$ *and* $\mathsf{u}_{2,1}, \ldots, \mathsf{u}_{2,r} \in V(\mathcal{N}_2)$. *The graph* $\mathcal{N}$ *obtained from these graphs by* gluing *vertex* $\mathsf{u}_{1,i}$ *with* $\mathsf{u}_{2,i}$ *for all* $i \in [r]$ *is defined by the following vertex and edge (label) sets:*

$$V(\mathcal{N}) = V(\mathcal{N}_1) \sqcup (V(\mathcal{N}_2) \setminus \{\mathsf{u}_{2,1}, \ldots, \mathsf{u}_{2,r}\})$$
$$and \; E(\mathcal{N}) = E(\mathcal{N}_1) \sqcup E(\mathcal{N}_2)$$

and incidence sets:

$$\forall \mathsf{u} \in V(\mathcal{N}_1) \setminus \{\mathsf{u}_{1,1}, \ldots, \mathsf{u}_{1,r}\}, \; E_\mathcal{N}^\rightarrow(\mathsf{u}) = E_{\mathcal{N}_1}^\rightarrow(\mathsf{u}) \; and \; E_\mathcal{N}^\leftarrow(\mathsf{u}) = E_{\mathcal{N}_1}^\leftarrow(\mathsf{u}),$$
$$\forall \mathsf{u} \in V(\mathcal{N}_2) \setminus \{\mathsf{u}_{2,1}, \ldots, \mathsf{u}_{2,r}\}, \; E_\mathcal{N}^\rightarrow(\mathsf{u}) = E_{\mathcal{N}_2}^\rightarrow(\mathsf{u}) \; and \; E_\mathcal{N}^\leftarrow(\mathsf{u}) = E_{\mathcal{N}_2}^\leftarrow(\mathsf{u}),$$
$$and \; \forall i \in [r], \; E_\mathcal{N}^\rightarrow(\mathsf{u}_{1,i}) = E_{\mathcal{N}_1}^\rightarrow(\mathsf{u}_{1,i}) \sqcup E_{\mathcal{N}_2}^\rightarrow(\mathsf{u}_{i,2}) \; and \; E_\mathcal{N}^\leftarrow(\mathsf{u}_{1,i}) = E_{\mathcal{N}_1}^\leftarrow(\mathsf{u}_{1,i}) \sqcup E_{\mathcal{N}_2}^\leftarrow(\mathsf{u}_{i,2}).$$

We made the arbitrary choice to let the glued vertices inherit their names from $\mathcal{N}_1$ rather than $\mathcal{N}_2$, but since in practice, we will mainly work on the edges of the graph, this detail doesn't matter so much. What is more important is that the edge set is simply a disjoint union of the edges sets of $\mathcal{N}_1$ and $\mathcal{N}_2$.

2.3 Switching Networks

A switching network on $\{0, 1\}^m$ (see e.g. [Pot15]) is an undirected graph $\mathcal{N}$ with two distinct *boundary* vertices s and t, in which each edge is labeled by a literal from $\{x_1, \ldots, x_m, \neg x_1, \ldots, \neg x_m, 1\}$. To simplify things slightly, in our case, we only have edges labeled with positive literals, $\{x_1, \ldots, x_m\}$, so we can just use the label set $[m]$ (or more specifically, in our case, the labels will be pairs $i, j \in [n]$, with $x_{i,j} = 1$ if and only if $(v_i, v_j) \in E(G)$ for G the input digraph). For $x \in \{0, 1\}^m$, letting $\mathcal{N}(x)$ denote the subgraph of $\mathcal{N}$ that includes only those edges whose labels are true under the string x, we say that $\mathcal{N}$ *accepts* x if and only if s is connected to t in $\mathcal{N}(x)$. In [JP25], formalizing [JK17], switching networks were additionally equipped with some associated subspaces, as we describe in the following definition.

Definition 2.5. *A* switching network $\mathcal{N}$ *on* $\{0, 1\}^m$ *consists of:*

1. *an undirected multigraph* $\mathcal{N} = (V, E)$ *with a* source $\mathsf{s} \in V$ *and* sink $\mathsf{t} \in V \setminus \{\mathsf{s}\}$;

2. *for each edge* $e \in E$, *a* query label $\varphi_e \in [m]$.

For $x \in \{0, 1\}^m$, *we define* $\mathcal{N}(x)$ *by restricted* $\mathcal{N}$ *to those edges* $e \in E$ *such that* $x_{\varphi_e} = 1$. *We associate the following spaces with* $\mathcal{N}$, *and an input* x:

1. $\Xi_{\mathsf{s}}^{\mathcal{A}} = \mathrm{span}\{|\mathsf{s}\rangle + |\leftarrow,\mathsf{s}\rangle\}$ *and* $\Xi_{\mathsf{t}}^{\mathcal{A}} = \mathrm{span}\{|\rightarrow,\mathsf{t}\rangle + |\mathsf{t}\rangle\}$

2. *for all* $\mathsf{v} \in V \setminus \{\mathsf{s},\mathsf{t}\}$, $\mathcal{V}_{\mathsf{v}} = \mathrm{span}\{|\psi_\star(\mathsf{v})\rangle := \sum_{e \in E^{\rightarrow}(\mathsf{u})} |\rightarrow,e\rangle + \sum_{e \in E^{\leftarrow}(\mathsf{u})} |\leftarrow,e\rangle\}$

3. $\mathcal{V}_{\mathsf{s}} = \mathrm{span}\{|\psi_\star(\mathsf{s})\rangle := \sum_{e \in E^{\rightarrow}(\mathsf{s})} |\rightarrow,e\rangle + \sum_{e \in E^{\leftarrow}(\mathsf{s})} |\leftarrow,e\rangle + |\leftarrow,\mathsf{s}\rangle\}$

4. $\mathcal{V}_{\mathsf{t}} = \mathrm{span}\{|\psi_\star(\mathsf{t})\rangle := \sum_{e \in E^{\rightarrow}(\mathsf{t})} |\rightarrow,e\rangle + \sum_{e \in E^{\leftarrow}(\mathsf{t})} |\leftarrow,e\rangle + |\rightarrow,\mathsf{t}\rangle\}$

5. *for all* $e \in E$, $\Xi_e = \mathrm{span}\{|\rightarrow,e\rangle, |\leftarrow,e\rangle\}$, $\Xi_e^{\mathcal{A}}(x) = \mathrm{span}\{|\rightarrow,e\rangle + (-1)^{x_{\varphi_e}}|\leftarrow,e\rangle\}$, $\Xi_e^{\mathcal{B}} = \mathrm{span}\{|\rightarrow,e\rangle + |\leftarrow,e\rangle\}$.

Then we let $H_{\mathcal{N}} = \bigoplus_{e \in E} \Xi_e \oplus \mathrm{span}\{|\mathsf{s}\rangle, |\mathsf{t}\rangle, |\leftarrow,\mathsf{s}\rangle, |\rightarrow,\mathsf{t}\rangle\}$ *and define the following two important subspaces of* $H_{\mathcal{N}}$:

$$\mathcal{A}(x) = \Xi_{\mathsf{s}}^{\mathcal{A}} \oplus \Xi_{\mathsf{t}}^{\mathcal{A}} \oplus \bigoplus_{e \in E} \Xi_e^{\mathcal{A}}(x) \quad and \quad \mathcal{B} = \bigoplus_{\mathsf{u} \in V(G)} \mathcal{V}_{\mathsf{u}} + \bigoplus_{e \in E} \Xi_e^{\mathcal{B}} \oplus \mathrm{span}\{|\leftarrow,\mathsf{s}\rangle + |\rightarrow,\mathsf{t}\rangle\}. \quad (2)$$

A single switching network can potentially compute different functions for different possible values of s or t. Thus, we will later find it convenient to equip a *set* of sinks[2], which, together with the single source s, will form a *boundary* for $\mathcal{N}$.

Remark 2.6. *It is also possible to assign weights to the edges of* $\mathcal{N}$ *in the definition of a switching network, which impacts the complexity of the quantum algorithm for evaluating it, but in this work we will assume all edges have weight 1.*

A switching network, along with the spaces defined in Definition 2.5, is a special case of a *subspace graph*, defined in [JP25]. As there, we can define *working bases* for a switching network as a pair of bases $\Psi^{\mathcal{A}}$ and $\Psi^{\mathcal{B}}$ for $\mathcal{A}$ and $\mathcal{B}$ respectively. For switching networks, the natural working basis for $\mathcal{A}$ is simply

$$\Psi_{\mathcal{A}}(x) = \{|\rightarrow,e\rangle + (-1)^{x_{\varphi_e}}|\leftarrow,e\rangle : e \in E(\mathcal{N})\} \cup \{|\mathsf{s}\rangle + |\leftarrow,\mathsf{s}\rangle, |\rightarrow,\mathsf{t}\rangle + |\mathsf{t}\rangle\}.$$

It's not difficult to see that this basis can be generated using a query to x, and $O(1)$ basic operations, by which we mean the following.

Definition 2.7 (Basis Generation). *We say an orthonormal basis* $\Psi = \{|b_\ell\rangle\}_{\ell \in L} \subset H_{\mathcal{N}}$ *can be generated in time* T *if:*

1. *The reflection around the subspace* $\mathrm{span}\{|\ell\rangle : \ell \in L\}$ *of* $H_{\mathcal{N}}$ *can be implemented in time* T.

2. *There is a map that acts as* $|\ell\rangle \mapsto |b_\ell\rangle$ *for all* $\ell \in L$ *that can be implemented in time* T.

For the space $\mathcal{B}$, we construct a basis for the orthogonal complement and invoke the following lemma from [JP25, Corollary 2.11], which guarantees that this suffices.

Lemma 2.8. *If* Ψ *is a basis that can be generated in time* T, *then there is a basis* Ψ' *for* $\mathrm{span}\{\Psi\}^\perp$ *that can be generated in time* T.

The following is a special case of [JP25, Theorem 3.13], which is proven by analyzing a phase estimation algorithm on the product of reflections around $\mathcal{A}$, and $\mathcal{B}$. It is also similar to [JK17, Theorem 13], but with a different implementation of reflection around $\mathcal{B}$ by generating a basis directly, rather than via a quantum walk, which is potentially more expensive.

Theorem 2.9. *For* $f : \{0,1\}^m \rightarrow \{0,1\}$, *let* $\mathcal{N}$ *be a switching network that accepts* $x \in \{0,1\}^m$ *if and only if* $f(x) = 1$. *Suppose that for all* x *accepted by* $\mathcal{N}$, *there is an* st-*path in* $\mathcal{N}(x)$ *of length at most* W_+. *Suppose the space* $\mathcal{B}^\perp$ *of* $\mathcal{N}$ *has a basis* $\Psi_{\mathcal{B}}^\perp$ *that can be generated in time* T_B. *Then there is a quantum algorithm that decides* f *with bounded error in time* $O(T_B\sqrt{W_+|E(\mathcal{N})|})$ *and space* $O(\log |E(\mathcal{N})|)$.

---

[2]We could also allow multiple sources instead of just a single s, but we don't need that for our constructions.

We can get a stronger version of this theorem by replacing path length with effective resistance, and cut size with capacitance (see [JJKP18]), but the theorem as stated suffices for our purposes.

We will use the following statement to find the orthogonal complement of $\mathcal{B}$. This is a slight generalization of [JP25, Lemma 3.15].

Lemma 2.10. *Fix a switching network, and define, for any $u \in V(\mathcal{N})$,*

$$|\psi_\star^-(\mathsf{u})\rangle := \sum_{e \in E^\to(\mathsf{u})} \frac{1}{2}(|\to, e\rangle - |\leftarrow, e\rangle) + \sum_{e \in E^\leftarrow(\mathsf{u})} \frac{1}{2}(|\leftarrow, e\rangle - |\to, e\rangle).$$

Define the cut space *of $\mathcal{N}$ as*

$$\mathcal{B}^- = \mathrm{span}\{|\psi_\star^-(\mathsf{u})\rangle : \mathsf{u} \in V(\mathcal{N}) \setminus \{\mathsf{s}, \mathsf{t}\}\} \cup \{|\leftarrow, \mathsf{s}\rangle + |\psi_\star^-(\mathsf{s})\rangle, |\to, \mathsf{t}\rangle + |\psi_\star^-(\mathsf{t})\rangle\}.$$

Then if $\Psi_{\mathcal{B}}^-$ is an orthonormal basis for $\mathcal{B}^-$, the following is an orthonormal basis for $\mathcal{B}$:

$$\Psi_{\mathcal{B}} = \Psi_{\mathcal{B}}^- \cup \left\{|b_e\rangle := \frac{1}{\sqrt{2}}(|\to, e\rangle + |\leftarrow, e\rangle) : e \in E(\mathcal{N})\right\}.$$

Since the map $|e\rangle|0\rangle \mapsto |b_e\rangle$ can be implemented with a single Hadamard gate, it is sufficient to be able to generate an orthonormal basis for $\mathcal{B}^-$. In the final section of the preliminaries, we give some results that further simplify this task.

2.4 Flows, Circulations, and the Cut Space

In this section, we will study the structure of *cut spaces*, of undirected graphs, with respect to the boundary $\{\mathsf{s}, \mathsf{t}\}$:

$$\mathcal{B}^- = \mathrm{span}\{|\psi_\star^-(\mathsf{u})\rangle : \mathsf{u} \in V(\mathcal{N}) \setminus \{\mathsf{s}, \mathsf{t}\}\} \cup \{|\leftarrow, \mathsf{s}\rangle + |\psi_\star^-(\mathsf{s})\rangle, |\to, \mathsf{t}\rangle + |\psi_\star^-(\mathsf{t})\rangle\}.$$

This is called a cut space, because it is intuitively the span of *cuts*, or sets of edges whose removal leaves the graph disconnected. We first show the following simple fact.

Lemma 2.11. $\dim \mathcal{B}^- = |V(\mathcal{N})|$.

Proof. We show this by proving that the states in the definition of $\mathcal{B}^-$ are independent. Let

$$|\psi_\mathsf{u}\rangle = \begin{cases} |\psi_\star^-(\mathsf{u})\rangle & \text{if } \mathsf{u} \in V \setminus \{\mathsf{s}, \mathsf{t}\} \\ |\leftarrow, \mathsf{s}\rangle + |\psi_\star^-(\mathsf{s})\rangle & \text{if } \mathsf{u} = \mathsf{s} \\ |\to, \mathsf{t}\rangle + |\psi_\star^-(\mathsf{t})\rangle & \text{if } \mathsf{u} = \mathsf{t} \end{cases}$$

and for simplicity, write $|e\rangle = |\to, e\rangle - |\leftarrow, e\rangle$. Suppose towards a contradiction that for some $\mathsf{u} \in V$,

$$|\psi_\mathsf{u}\rangle = \sum_{\mathsf{v} \in V \setminus \{\mathsf{u}\}} \alpha_\mathsf{v} |\psi_\mathsf{v}\rangle.$$

We will show by induction on distance $d \geq 1$ from u, that for all u' holds $|\alpha_{\mathsf{u}'}| = 1$. For the base case, if u' and u share an edge e, then

$$1 = |\langle e | \psi_\mathsf{u}\rangle| = \left| \sum_{\mathsf{v} \in V \setminus \{\mathsf{u}\}} \alpha_\mathsf{v} \langle e | \psi_\mathsf{v}\rangle \right| = |\alpha_{\mathsf{u}'}|,$$

since $|\langle e | \psi_\mathsf{v}\rangle|$ is 1 if e is incident to v and 0 otherwise.

For the induction step, if u' is at distance d from u, it has a neighbour u'' at distance $d-1$ from u, to which the induction hypothesis applies, so letting e be the edge incident to u' and u'', we have:

$$0 = \left| \sum_{\mathsf{v} \in V \setminus \{\mathsf{u}\}} \alpha_\mathsf{v} \langle e | \psi_\mathsf{v}\rangle \right| = |\alpha_{\mathsf{u}'} + \alpha_{\mathsf{u}''}|$$

so $|\alpha_{u'}| = 1$, since $|\alpha_{u''}| = 1$ by the induction hypothesis.

The contradiction arises because there is at least one boundary vertex that is not u – without loss of generality, suppose $s \neq u$. Then since $| \leftarrow, s \rangle$ only appears in $|\psi_s\rangle$, we have:

$$0 = |\langle \leftarrow, s | \psi_u \rangle| = \left| \sum_{v \in V \setminus \{u\}} \alpha_v \langle \leftarrow, s | \psi_v \rangle \right| = |\alpha_s| = 1.$$

This is clearly a contradiction. $\qquad\square$

Definition 2.12. *A* flow *on $\mathcal{N}$ is a real-valued function θ on $E(\mathcal{N})$. For $u \in V(\mathcal{N})$, define*

$$\theta(u) := \sum_{e \in E^{\rightarrow}(u)} \theta(e) - \sum_{e \in E^{\leftarrow}(u)} \theta(e).$$

We call θ a circulation *on $\mathcal{N}$ if for all $u \in V(\mathcal{N})$, $\theta(u) = 0$. We say θ has* boundary B *if for all $u \in V(\mathcal{N}) \setminus B$, $\theta(u) = 0$. We call θ an* st*-flow if it has boundary $\{s, t\}$ (so in particular, every circulation is an* st*-flow). We call θ a* unit st*-flow if it is an* st*-flow, and $\theta(s) = -\theta(t) = 1$. We call θ an* optimal unit st*-flow if it minimizes the expression $\sum_{e \in E(\mathcal{N})} \theta(e)^2$.*

We can naturally view any function θ on $E(\mathcal{N})$ as a vector on

$$\mathrm{span}\{|e\rangle := | \rightarrow, e \rangle - | \leftarrow, e \rangle : e \in E(\mathcal{N})\},$$

which we denote

$$|\bar{\theta}\rangle := \sum_{e \in E(\mathcal{N})} \theta(e)(| \rightarrow, e \rangle - | \leftarrow, e \rangle) = \sum_{e \in E(\mathcal{N})} \theta(e)|e\rangle. \tag{3}$$

For an edge e oriented from u to v, if $\theta(e)$ is a positive real number, we interpret θ as sending $\theta(e)$ flow from u to v along the edge e. If $\theta(e)$ is negative, we interpret θ as sending $|\theta(e)|$ flow from v to u. In this way, we can interpret $-|e\rangle$ as the edge e oriented in the opposite direction – that is, vector negation changes the orientation of edges.

When θ is an st-flow, we will sometimes want to additionally include flow entering and exiting the boundary vertices, via "boundary edges", so that the total flow is conserved on boundary vertices as well, and this is represented by the vector

$$|\theta\rangle = -\theta(s)| \leftarrow, s \rangle + |\bar{\theta}\rangle - \theta(t)| \rightarrow, t \rangle. \tag{4}$$

Whereas states of the form $|\bar{\theta}\rangle$ enable seamless combination of flows in graphs obtained from gluing other graphs together (see Definition 2.4), states of the form $|\theta\rangle$ capture the boundary of the final graph we use in our switching network.

It is not difficult to see that the set of flows on a particular (possibly empty) boundary is closed under linear combinations, so we can define the following vectors spaces.

Definition 2.13. *We call*

$$\mathcal{F}(\mathcal{N}) = \left\{ |\theta\rangle : \forall u \in V(\mathcal{N}) \setminus \{s, t\}, \ \theta(u) = 0 \right\}$$

the space of st*-flows;*

$$\mathcal{C}(\mathcal{N}) = \left\{ |\theta\rangle : \forall u \in V(\mathcal{N}), \ \theta(u) = 0 \right\}$$

the space of circulations;

$$\mathcal{F}^{\mathrm{OPT}}(\mathcal{N}) = \mathrm{span}\left\{ |\theta\rangle \in \mathcal{F}(\mathcal{N}) : |\theta\rangle \text{ is an optimal unit } st\text{-flow} \right\}$$

the space of optimal st*-flows.*

In the following lemmas, we establish fundamental properties of the space of st-flows together with its subspaces. These results will serve as the foundation for our later analysis. We start with stating some properties of optimal flows and their orthogonality to circulations, see [Cor23, Lemma 7.2.2] for the proof.

Lemma 2.14. *The optimal unit* st*-flow θ is unique and its corresponding vectors, $|\theta\rangle$ and $|\bar{\theta}\rangle$ (see (3) and (4)) are orthogonal to the space of circulations $\mathcal{C}(\mathcal{N})$. Consequently, the space of optimal* st*-flows $\mathcal{F}^{\text{OPT}}(\mathcal{N})$ is one-dimensional and orthogonal to the space of circulations $\mathcal{C}(\mathcal{N})$.*

Next, we show that the space of st-flows admits a direct sum decomposition into optimal flows and circulations.

Lemma 2.15. $\mathcal{F}(\mathcal{N}) = \mathcal{F}^{\text{OPT}}(\mathcal{N}) \oplus \mathcal{C}(\mathcal{N})$

Proof. By Lemma 2.14, $\mathcal{F}^{\text{OPT}}(\mathcal{N}) \perp \mathcal{C}(\mathcal{N})$. It is immediate that $\mathcal{F}(\mathcal{N}) \supseteq \mathcal{F}^{\text{OPT}}(\mathcal{N}) \oplus \mathcal{C}(\mathcal{N})$. Thus, it remains to prove the reverse inclusion $\mathcal{F}(\mathcal{N}) \subseteq \mathcal{F}^{\text{OPT}}(\mathcal{N}) \oplus \mathcal{C}(\mathcal{N})$. Let $|\theta\rangle \in \mathcal{F}(\mathcal{N})$. If θ is a circulation, then we're done, so assume not. Without loss of generality, assume θ is scaled so that $\theta(\mathsf{s}) = -\theta(\mathsf{t}) = 1$. Then, by Lemma 2.14, there exists a unique optimal unit st-flow $|\theta'\rangle \in \mathcal{F}^{\text{OPT}}$. It is easy to check that $|\theta\rangle - |\theta'\rangle$ is a circulation, which establishes $|\theta\rangle \in \mathcal{F}^{\text{OPT}}(\mathcal{N}) \oplus \mathcal{C}(\mathcal{N})$, completing the proof. $\square$

Next, we establish the relationship between the space of st-flows of a switching network and its associated space $\mathcal{B}$.

Lemma 2.16. *Let $\mathcal{N}$ be a switching network, with $\mathcal{B}$ as in (2). Then $\mathcal{B}^{\perp} = \mathcal{F}(\mathcal{N}) \oplus \text{span}\{|\mathsf{s}\rangle, |\mathsf{t}\rangle\}$.*

Proof. First, we observe that $\mathcal{F}(\mathcal{N})$ is orthogonal to $\text{span}\{|\mathsf{s}\rangle, |\mathsf{t}\rangle\}$, because, by definition, flow states have no overlap with $|\mathsf{s}\rangle$ and $|\mathsf{t}\rangle$. By Lemma 2.10 we can write:

$$\mathcal{B}^{\perp} = \left(\mathcal{B}^{-} \oplus \text{span}\left\{ \frac{1}{\sqrt{2}}(| \to, e\rangle + | \leftarrow, e\rangle) : e \in E(G) \right\} \right)^{\perp}$$
$$= (\mathcal{B}^{-})^{\perp} \cap \left(\text{span}\left\{ \frac{1}{\sqrt{2}}(| \to, e\rangle + | \leftarrow, e\rangle) : e \in E(G) \right\} \right)^{\perp}.$$

Note that

$$\left(\text{span}\left\{ \frac{1}{\sqrt{2}}(| \to, e\rangle + | \leftarrow, e\rangle) : e \in E(G) \right\} \right)^{\perp}$$
$$= \text{span}\left\{ \frac{1}{\sqrt{2}}(| \to, e\rangle - | \leftarrow, e\rangle) : e \in E(G) \right\} \oplus \text{span}\{|\mathsf{s}\rangle, |\mathsf{t}\rangle, | \leftarrow, \mathsf{s}\rangle, | \to, \mathsf{t}\rangle\}.$$

Since $\mathcal{B}^{-} \subseteq \left(\text{span}\left\{ \frac{1}{\sqrt{2}}(| \to, e\rangle + | \leftarrow, e\rangle) : e \in E(G) \right\} \right)^{\perp}$, we can take the orthogonal complement of $\mathcal{B}^{-}$ in $\left(\text{span}\left\{ \frac{1}{\sqrt{2}}(| \to, e\rangle + | \leftarrow, e\rangle) : e \in E(G) \right\} \right)^{\perp}$. That is, we can assume that all edge spaces are one-dimensional and are spanned by $|e\rangle = | \to, e\rangle - | \leftarrow, e\rangle$, for $e \in E$. We can take an arbitrary state $|\psi\rangle \in \left(\text{span}\left\{ \frac{1}{\sqrt{2}}(| \to, e\rangle + | \leftarrow, e\rangle) : e \in E(G) \right\} \right)^{\perp}$ and write it as $|\psi\rangle = |\hat{\theta}\rangle + |b\rangle$, where

$$|\hat{\theta}\rangle = -\theta_{\mathsf{s}}| \leftarrow, \mathsf{s}\rangle + \sum_{e \in E} \theta(e)|e\rangle - \theta_{\mathsf{t}}| \to, \mathsf{t}\rangle,$$

for some function θ on E and values $\theta_{\mathsf{s}}, \theta_{\mathsf{t}}$, and $|b\rangle \in \text{span}\{|\mathsf{s}\rangle, |\mathsf{t}\rangle\}$. By definition of $|\psi^{-}_{\star}(\mathsf{u})\rangle$ (see Lemma 2.10), it is orthogonal to $|b\rangle$ for any $\mathsf{u} \in V$, and so by inspection, $|b\rangle$ is orthogonal to $\mathcal{B}^{-}$.

Next, we compute the inner product of $|\hat{\theta}\rangle$ with the states in $\mathcal{B}^{-}$ to determine precisely when $|\psi\rangle$ is orthogonal to all of them.

First, for *any* $\mathsf{u} \in V$,

$$\langle \hat{\theta} | \psi^{-}_{\star}(\mathsf{u})\rangle = \sum_{e \in E^{\to}(\mathsf{u})} \frac{1}{2}\theta(e)\langle e|e\rangle - \sum_{e \in E^{\leftarrow}(\mathsf{u})} \frac{1}{2}\theta(e)\langle e|e\rangle$$
$$= \sum_{e \in E^{\to}(\mathsf{u})} \theta(e) - \sum_{e \in E^{\leftarrow}(\mathsf{u})} \theta(e) = \theta(\mathsf{u}). \tag{5}$$

9

This inner product vanishes for all $u \in V \setminus \{s, t\}$ if and only if θ is an st-flow. Next, we compute:

$$\langle \hat{\theta}|(|\leftarrow, s\rangle + |\psi_\star^-(s)\rangle) = -\theta_s + \langle \hat{\theta}|\psi_\star^-(s)\rangle = -\theta_s + \theta(s), \tag{6}$$

by (5). This inner product vanishes if and only if $\theta_s = \theta(s)$. Finally, we compute:

$$\langle \theta|(|\rightarrow, t\rangle + |\psi_\star^-(t)\rangle) = -\theta_t + \theta(t), \tag{7}$$

again by (5). This inner product vanishes if and only if $\theta_t = \theta(t)$.

Combining (5), (6) and (7), we can see that $|\psi\rangle \in \mathcal{B}^-$ if and only if $|\hat{\theta}\rangle = |\theta\rangle$ (as defined in (4)) for some st-flow θ, which is if and only if $|\hat{\theta}\rangle \in \mathcal{F}(\mathcal{N})$. □

Finally, we compute the dimensions of the space of flows and circulations of $\mathcal{N}$ and its subspaces.

Lemma 2.17. *Let $\mathcal{F}(\mathcal{N})$ be the space of boundary flows of $\mathcal{N}$; $\mathcal{C}(\mathcal{N})$ the space of circulations of $\mathcal{N}$; and $\mathcal{F}^{\mathrm{OPT}}(\mathcal{N})$ the span of optimal boundary flows of $\mathcal{N}$ (see Definition 2.13). Then*

$$\dim \mathcal{F}(\mathcal{N}) = |E(\mathcal{N})| + 2 - |V(\mathcal{N})|$$
$$\dim \mathcal{F}^{\mathrm{OPT}}(\mathcal{N}) = 1$$
$$\dim \mathcal{C}(\mathcal{N}) = |E(\mathcal{N})| - |V(\mathcal{N})| + 1.$$

Proof. The dimension of $\mathcal{F}(\mathcal{N})$ can be computed using Lemma 2.16:

$$\dim \mathcal{F}(\mathcal{N}) + \dim \mathrm{span}\{|s\rangle, |t\rangle\} = \dim H_\mathcal{N} - \dim \mathcal{B}.$$

The dimension of $H_\mathcal{N}$ is equal to $2|E(\mathcal{N})| + 4$ by Definition 2.5. By Lemma 2.10 and Lemma 2.11, $\dim \mathcal{B} = \dim \mathcal{B}^- + |E(\mathcal{N})| = |V(\mathcal{N})| + |E(\mathcal{N})|$. Combining these observations, we obtain

$$\dim \mathcal{F}(\mathcal{N}) = \dim H_\mathcal{N} - \dim \mathcal{B} - 2 = |E(\mathcal{N})| + 2 - |V(\mathcal{N})|.$$

Next, we note that it follows from Lemma 2.14 that $\dim \mathcal{F}^{\mathrm{OPT}}(\mathcal{N}) = 1$. Finally, the dimension of $\mathcal{C}(\mathcal{N})$ is straightforward to determine, as it follows from the direct sum decomposition $\mathcal{F}(\mathcal{N}) = \mathcal{F}^{\mathrm{OPT}}(\mathcal{N}) \oplus \mathcal{C}(\mathcal{N})$ from Lemma 2.15. □

3 Quantum Algorithm for DSTCON

In this section, we prove our main result, by describing a quantum algorithm for DSTCON that works for any space bound $S \geq \log^2(n)$. We begin by stating the main technical result of this paper, which we prove in Section 4. Specifically, we present a quantum algorithm for deciding directed st-connectivity under an additional constraint on the path length (i.e. solving PATH$_L$). While this subroutine is quantum, the algorithm we decide in the remainder of this section is otherwise classical.

Theorem 3.1. *Let $G = (V, E)$ be a directed graph such that $V = \{v_1, \ldots, v_n\}$. Assume that G can be accessed via a quantum oracle $\mathcal{O}_G$ that can be implemented in time $O(1)$, where for any $i, j \in [n]$, $b \in \{0, 1\}$*

$$\mathcal{O}_G : |i\rangle|j\rangle|b\rangle \mapsto \begin{cases} |i\rangle|j\rangle|b \oplus 1\rangle & \text{if } (v_i, v_j) \in E \\ |i\rangle|j\rangle|b\rangle & \text{otherwise.} \end{cases}$$

Let $L \leq n$ be a power of 2. Then there is a bounded-error quantum algorithm, $\mathsf{D}_L(G, u, v)$, that decides for any $u, v \in V$ whether there is a directed path from u to v in G of length at most L, in time

$$\widetilde{O}\left(\left(L^{\log 3}(2n + 1)^{\log L} n\right)^{1/2}\right)$$

and space $O(\log(L) \log(n))$.

In Section 4, we prove the statement for L a power of 2, using a recursive structure, but a simple corollary extends this result to any L.
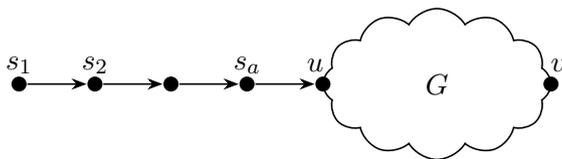
Figure 2: The graph G' constructed from G. It is clear that there is a uv-path of length at most L in G if and only if there is a s_1v-path of length at most $L + a$ in G'.

Corollary 3.2. *Let G be as in Theorem 3.1, and let L be any positive integer. Then there is a bounded-error quantum algorithm, $\mathtt{Dist}_L(G, u, v)$, that decides for any $u, v \in V$ whether there is a directed path from u to v in G of length at most L, in time*

$$\widetilde{O}\left(\left(L^{\log 3}(2n+1)^{\log L}n\right)^{1/2}\right)$$

and space $O\left(\log(L)\log(n)\right)$.

Proof. To prove the statement, we exhibit a quantum algorithm, Algorithm 1 that uses one call to the subroutine D from Theorem 3.1, on a graph G' that is constructed from G by adding a directed path from some new vertex s_1 into u of length such that any uv-path of length at most L corresponds to a s_1v-path of length at most $2^{\lceil \log L \rceil}$ (see Figure 2). Since G' can be queried using at most one query to G, the result follows. □

Algorithm 1 $\mathtt{Dist}_L(G, u, v)$

Parameter: a positive integer $L \leq n$

Input: a directed graph $G = (V, E)$ and a pair of vertices $u, v \in V$

Output: 0 or 1 indicating there is a path of length at most L between u and v in G

---

1. Let $\ell = \lceil \log L \rceil$ and $a = 2^\ell - L$

2. Let G' be the graph G with a new vertices $s_1, \ldots, s_a$ and a new edges $(s_1, s_2), \ldots, (s_{a-1}, s_a), (s_a, u)$. Then G' is just G with a directed path of length a coming into u, and can easily be queried using queries to G.

3. Return $\mathtt{D}_{2^\ell}(G, s_1, v)$.

3.1 BFS algorithm that calls the quantum short path subroutine

To get a time-space tradeoff for DSTCON, and prove our main result, we describe a classical BFS-based algorithm, Algorithm 2, from [BBRS98], that makes calls to a subroutine for the problem $\mathrm{DIST}_L(G, u, v)$, of deciding whether there is a path of length at most L from u to v in a directed graph G. Our algorithm for DSTCON is obtained by instantiating that subroutine with the quantum algorithm $\mathtt{Dist}_L$ from Corollary 3.2.

Ref. [BBRS98] show that this algorithm correctly decides DSTCON whenever $\mathtt{Dist}_L$ decides PATH_L. From [BBRS98], or by inspecting Algorithm 2, we get the following.

Lemma 3.3. *Let $\mathtt{Dist}_L(G, u, v)$ be a bounded-error algorithm for DIST_L, with time complexity $D_T(n, L)$, and the space complexity is $D_S(n, L)$. Then the time complexity of Algorithm 2 is*

$$\widetilde{O}\left(\frac{n^3}{L}D_T(n, L)\right)$$

and its space complexity is

$$O\left(\frac{n \log L}{L} + D_S(n, L)\right).$$

Algorithm 2 $\text{DSTCON}_L(G, s, t)$ [BBRS98]

Parameter: a positive integer $L \leq n$

Input: a directed graph $G = (V, E)$ and a pair of vertices $s, t \in V$

Output: CONNECTED if there is a directed path from s to t in G, NOT CONNECTED otherwise

```
 1: for j = 0, . . . , L − 1 do
 2:     S ← {s}
 3:     for all vertices v ∈ V do
 4:         if Dist_j(s, v) = 1 ∧ Dist_{j−1}(s, v) = 0 then
 5:             if |S| > n/L then try next j
 6:             else add v to S
 7:             end if
 8:         end if
 9:     end for
10:     for i = 1, . . . , ⌊n/L⌋ do
11:         S' = ∅
12:         for all vertices v ∈ V do
13:             if ∃u ∈ S : Dist_L(u, v) = 1 ∧ ∀u ∈ S : Dist_{L−1}(u, v) = 0 then
14:                 if |S| + |S'| > n/L then try next j
15:                 else add v to S'
16:                 end if
17:             end if
18:         end for
19:         S = S ∪ S'
20:     end for
21:     if t within distance L of a vertex in S then return (CONNECTED)
22:     else return (NOT CONNECTED)
23:     end if
24: end for
```

If Dist_L is implemented by a quantum algorithm, then the quantum space complexity is at most $O(D_S(n, L))$, and any remaining space is classical.

Substituting the result of Corollary 3.2 yields the following theorem, which combines our quantum subroutine with the classical BFS algorithm. Although the algorithm in Corollary 3.2 has bounded error, its success probability can be boosted high enough that the outer algorithm will not notice, using majority voting, at the cost of an overhead of $\log \widetilde{O}(n^3/L)$, which is hidden in the $\widetilde{O}$ of the final complexity.

Theorem 3.4. *Let $G = (V, E)$ be a directed graph such that $V = \{v_1, \ldots, v_n\}$, and $s, t \in V$. Assume that G can be accessed via a quantum oracle $\mathcal{O}_G$ that can be implemented in time $O(1)$, where for any $i, j \in [n]$, $b \in \{0, 1\}$,*

$$\mathcal{O}_G : |i\rangle|j\rangle|b\rangle \mapsto \begin{cases} |i\rangle|j\rangle|b \oplus 1\rangle & \text{if } (v_i, v_j) \in E \\ |i\rangle|j\rangle|b\rangle & \text{otherwise.} \end{cases}$$

Then there is a quantum algorithm that decides whether there is a directed path from s to t in G with bounded error in time

$$\widetilde{O}\left(n^{3.5} L^{.5 \log(3) - 1}(2n + 1)^{0.5 \log L}\right)$$

and total space

$$\widetilde{O}\left(\left(\frac{n}{L} + \log L\right) \log n\right)$$

of which $O(\log(L) \log(n))$ is quantum space.

3.2 Complexity comparison: classical vs. quantum

Theorem 3.5 ([BBRS98]). *Let $G = (V, E)$ be a directed graph such that $V = \{v_1, \ldots, v_n\}$, and $s, t \in V$. Assume that G can be accessed via a classical oracle $\mathcal{O}_G$ that can be implemented in time $O(1)$, where for any $i, j \in [n]$,*

$$\mathcal{O}_G(i, j) = \begin{cases} 1 & \text{if } (v_i, v_j) \in E \\ 0 & \text{otherwise.} \end{cases}$$

Then for any $S \geq \log^2(n)$, there is a classical algorithm that decides whether there is a directed path from s to t in G using space $O(S)$ and time

$$T \leq 2^{\log^2 \frac{n}{S} + O(\log n \log \log n)}.$$

In [BBRS98], they state a bound of $T \leq 2^{O(\log^2(n/S))}$, but using their choice of parameters, and a slightly more precise analysis of their algorithm, we can compute the more fine-grained upper bound we have stated above. We improve on their result in our main theorem, which is the following.

Theorem 3.6. *Let $G = (V, E)$ be a directed graph such that $V = \{v_1, \ldots, v_n\}$, and $s, t \in V$. Assume that G can be accessed via a quantum oracle $\mathcal{O}_G$ that can be implemented in time $O(1)$, where for any $i, j \in [n]$, $b \in \{0, 1\}$,*

$$\mathcal{O}_G : |i\rangle|j\rangle|b\rangle \mapsto \begin{cases} |i\rangle|j\rangle|b \oplus 1\rangle & \text{if } (v_i, v_j) \in E \\ |i\rangle|j\rangle|b\rangle & \text{otherwise.} \end{cases}$$

Then for any $S \geq \log^2(n)$, there is a quantum algorithm that decides for any $s, t \in V$ whether there is a directed path from s to t in G with bounded error using space $O(S)$ and time

$$T \leq 2^{\frac{1}{2} \log n \log \frac{n}{S} + O(\log n \log \log n)}.$$

This algorithm uses $O(\log^2(n))$ quantum space.

Proof. We analyze the time-space tradeoff of the quantum algorithm of Theorem 3.4. Assuming $L \log L = O(n)$, its space complexity becomes $S = O\left(\frac{n}{L} \log n\right)$, with only $O(\log(L) \log(n)) = O(\log^2(n))$ quantum space. That is, we can express $L = \Theta\left(\frac{n}{S} \log n\right)$. Substituting this into the time complexity $T = \widetilde{O}\left(n^{3.5} L^{.5 \log(3)-1}(2n+1)^{0.5 \log L}\right)$ and taking logarithms, we obtain

$$\log T = 3.5 \log n + \log L \left(\frac{\log(2n+1)}{2} + .5 \log(3) - 1\right) + O(\log \log n)$$

$$= \left(\log \frac{n}{S} + \log \log n + O(1)\right)\left(\frac{\log n}{2} + O(1)\right) + O(\log n)$$

$$= \frac{1}{2} \log(\frac{n}{S}) \log(n) + O\left(\log(n) \log \log(n)\right). \qquad \square$$

Remark 3.7. *For $S = o(n^{1/2})$, the quantum algorithm of Theorem 3.6 achieves a better time-space tradeoff than the classical time-space tradeoff stated in Theorem 3.5.*

4 Quantum short path subroutine

In this section, we prove Theorem 3.1 by describing and analyzing a quantum algorithm for $\text{DIST}_L(G, s, t)$, for $G = (V, E)$ a directed graph with $V = \{v_1, \ldots, v_n\}$, $s, t \in V$ any pair of vertices, and $L \in \mathbb{N}$ a power of 2. The algorithm is designed by exhibiting a switching network (Definition 2.5), and then applying Theorem 2.9.

In Section 4.1, we describe the switching network, through a recursive construction. In order to apply Theorem 2.9, we need to analyze the number of edges in the switching network, and upper bound the distance between its source s and sink t, which we do in Section 4.2; and describe a basis for the space $\mathcal{B}^{\perp}$, and a procedure for generating it, which we do in Section 4.3. Finally, in Section 4.4, we put it all together to prove Theorem 3.1.

4.1 Switching network

The switching networks we will work with will have vertices represented by a tuple $[u_1, \ldots, u_k] \in V^k$, of some number k of vertices of G, as well as possibly some additional information. We will not actually care so much about naming conventions for the vertices, but the important detail is that each vertex of a switching network has an associated subset $\{u_1, \ldots, u_k\} \subseteq V$ (so the order of the tuple actually doesn't matter). In particular, we will construct switching networks by gluing together switching networks of this form (see Definition 2.4), and it will be important that any pair of vertices we glue together have the same associated set, so there is no ambiguity.

The way we construct our switching networks, we will only have an edge between a pair of vertices where the associated sets are of the form $\{u_1, \ldots, u_k\}$ and $\{u_1, \ldots, u_k, u_{k+1}\}$, and the query label for that edge (see Definition 2.5) is (u_i, u_{k+1}) for some $i \in [k]$. Such switching networks were first studied in [Pot14]. This structure ensures that a vertex $[u]$ can only be connected to a vertex $[u_1, \ldots, u_k]$ by a path of "on" edges if each u_i is reachable from u in G – a property that will be crucial for our analysis in Section 4.1.2.

In this section, we will describe and analyze a switching network $\mathcal{N}_L(s)$ of the above described form. This will be built inductively from switching networks $\mathcal{N}_{2^\ell}(u)$ for $\ell \in \{0, \ldots, \log L\}$, and $u \in V$, called the *root*. $\mathcal{N}_{2^\ell}(u)$ has a single source $[u]$ and n sinks $\{[u, v_i] : v_i \in V\}$. With respect to the i-th sink $[u, v_i]$, the switching network computes whether v_i is reachable from u by a path of length at most 2^ℓ, for every $v_i \in V$ simultaneously (i.e. v_i is reachable from u by a path of length at most 2^ℓ in G if and only if $[u]$ and $[u, v_i]$ are connected in $\mathcal{N}_{2^\ell}(u)(G)$). This "extended boundary" is used solely for the recursive construction in Section 4.1.1. The final construction $\mathcal{N}_L(s)$ has source $\mathsf{s} = [s]$ and a single sink $\mathsf{t} = [s, t]$. We write $\mathcal{N}_L(s, t)$ when we want to emphasize this.

4.1.1 Graph construction

Define $\Sigma = \{(0, \bar{0})\} \cup \{(1, i) : i \in \{0, 1\}^{\log n}\} \cup \{(2, j) : j \in \{0, 1\}^{\log n}\}$, an alphabet of size $2n + 1$. We have ensured that all symbols in this alphabet have an obvious representation as a string in $\{0, 1, 2\} \times \{0, 1\}^{\log n}$, but for convenience we will sometimes use 0 to denote $(0, \bar{0})$, and $1i$ or $2j$ to denote $(1, i)$ or $(2, j)$. For any $\sigma \in \Sigma^*$, let $|\sigma|$ denote its length, and define:

$$f_1(\sigma) := \begin{cases} \max\{i \in \{1, \ldots, |\sigma|\} : \sigma_i \in \{1\} \times \{0, 1\}^{\log n}\} & \text{if } \exists i : \sigma_i \in \{1\} \times \{0, 1\}^{\log n} \\ 0 & \text{else.} \end{cases}$$

We now define $\mathcal{N}_{2^\ell}(u)$ for $\ell \in [\log L]$, by a recursive construction.

Base construction. For the base case, $\ell = 0$, the switching network $\mathcal{N}_1(u)$ consists of a source vertex $[u]$ connected to n sinks $[u, v_i], v_i \in V$. The edges have query labels (u, v_i), each "checking" whether there is an edge (u, v_i) in G (see Figure 3). More precisely, we formally define the sets of vertices and edges as follows.

$$V_1 = \{[u]\} \cup \{[u, v_i] : v_i \in V\}, \quad \text{and} \quad E_1 = \{|e_i\rangle = |i\rangle : i \in [n]\}.$$

Above, we put edge labels in a ket, to emphasize that they form an orthonormal basis of some inner product space. The incidence of edges and vertices is defined (see Definition 2.3 for a reminder of how undirected graphs are specified):

$$E_1^{\rightarrow}([u]) = E_1 \quad \text{and} \quad E_1^{\leftarrow}([u]) = \emptyset$$

$$\forall i \in [n], \ E_1^{\rightarrow}([u, v_i]) = \emptyset \quad \text{and} \quad E_1^{\leftarrow}([u, v_i]) = \{|e_i\rangle\}.$$
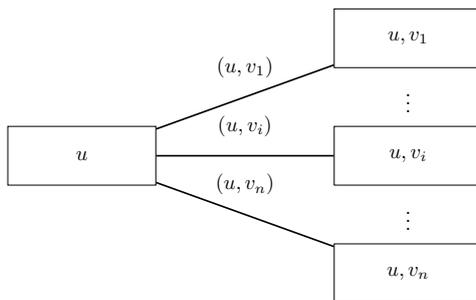
The query label of the edge e_i is (u, v_i).

Figure 3: Graph construction for the switching network $\mathcal{N}_1(u)$ that decides whether each vertex of a graph $G = (V, E)$ is reachable from a vertex $u \in V$ by a path of length 1. Each edge $([u], [u, v_i])$ of $\mathcal{N}_1(u)$ has query label (u, v_i) and is "on" in $\mathcal{N}_1(u)(G)$ if and only if $(u, v_i) \in E$. Therefore, it holds that v_i is reachable from u by a path of length at most 1 in G if and only if $[u]$ and $[u, v_i]$ are connected in $\mathcal{N}_1(u)(G)$.

Recursive construction. Next, we describe the construction of $\mathcal{N}_{2^\ell}(u)$, assuming that a construction of $\mathcal{N}_{2^{\ell-1}}(u)$ is given. Let $\mathcal{N}_{2^{\ell-1}}^v(u)$ denote a copy of $\mathcal{N}_{2^{\ell-1}}(u)$ in which each vertex-tuple of the switching network is augmented with an additional vertex $v \in V$ (although the order in the tuple doesn't matter, for clarity, assume we append v to the front of each tuple).

The construction of $\mathcal{N}_{2^\ell}(u)$ uses $2n + 1$ copies $\mathcal{N}_{2^{\ell-1}}(u')$ for some u', some of them augmented by additional vertices. Specifically, define:

$$\mathcal{N}_{2^{\ell-1}}^0 = \mathcal{N}_{2^{\ell-1}}(u)$$
$$\forall i \in [n], \; \mathcal{N}_{2^{\ell-1}}^{(1,i)} = \mathcal{N}_{2^{\ell-1}}^u(v_i)$$
$$\forall j \in [n], \; \mathcal{N}_{2^{\ell-1}}^{(2,j)} = \mathsf{Rev}(\mathcal{N}_{2^{\ell-1}}^{v_j}(u)).$$

Above, we used the notation $\mathsf{Rev}(\mathcal{N})$ to be the switching network $\mathcal{N}$ except with the orientation of every edge reversed. As we will see shortly, this ensures that all edges of $\mathcal{N}_{2^\ell}(u)$ have a logical left-to-right orientation. Define $\mathcal{N}_{2^\ell}(u)$ from these $2n + 1$ copies of $\mathcal{N}_{2^{\ell-1}}$ by gluing (as made precise in Definition 2.4) the i-th sink of $\mathcal{N}_{2^\ell}^0$ – which encodes $[u, v_i]$ – to the source of $\mathcal{N}_{2^\ell}^{(1,i)}$ – which also encodes $[u, v_i]$ – (for all $i \in [n]$); and gluing the j-th sink of $\mathcal{N}_{2^\ell}^{(1,i)}$ – which encodes $[u, v_i, v_j]$ – to the i-th sink of $\mathcal{N}_{2^\ell}^{(2,j)}$ – which also encodes $[v_j, u, v_i] \equiv [u, v_i, v_j]$ – (for all $i, j \in [n]$), as in Figure 4. Note that the source of $\mathcal{N}_{2^\ell}^{(2,j)}$, which encodes $[v_j, u] \equiv [u, v_j]$, is the j-th sink of $\mathcal{N}_{2^\ell}(u)$. The source $[u]$ of $\mathcal{N}_{2^{\ell-1}}^0$ is the source of $\mathcal{N}_{2^\ell}(u)$.

The edges E_{2^ℓ} of $\mathcal{N}_{2^\ell}$ should be the disjoint union of the edge sets of the $2n + 1$ copies of $\mathcal{N}_{2^{\ell-1}}$. We use the elements of Σ labeling each copy to make this union disjoint:

$$E_{2^\ell} = \bigsqcup_{\sigma \in \Sigma} E(\mathcal{N}_{2^{\ell-1}}^\sigma) = \Sigma \times E_{2^{\ell-1}} = \Sigma^\ell \times E_1. \tag{8}$$

For $\sigma \in \Sigma^\ell$ and $i \in \{0, 1\}^{\log n}$, we will sometimes denote the edge $|\sigma, e_i\rangle = |\sigma, i\rangle$ using

$$(-1)^{|\sigma|_2} |e_i^\sigma\rangle := |\sigma, e_i\rangle, \tag{9}$$

where $|\sigma|_2$ denotes the number of occurrences of $(2, j)$ for some j in σ. The reason for the sign is that we always want $|\sigma, e_i\rangle$ to represent the i-th edge in the σ-labeled copy of $\mathcal{N}_1$ *oriented from left-to-right*[3] (i.e. from source towards sinks), and we therefore need to reverse the edge orientations every time we use a copy of $\mathcal{N}_{2^\ell}$ in the $(2, j)$-th position for some j. Generally, if $|\psi\rangle \in \mathrm{span}\{|e\rangle : e \in E_{2^\ell}\}$ – equivalently, ψ is a function on E_{2^ℓ} – and $\sigma \in \Sigma^{\ell'}$, we will let

$$|\psi^\sigma\rangle = (-1)^{|\sigma|_2} |\sigma\rangle |\psi\rangle, \tag{10}$$

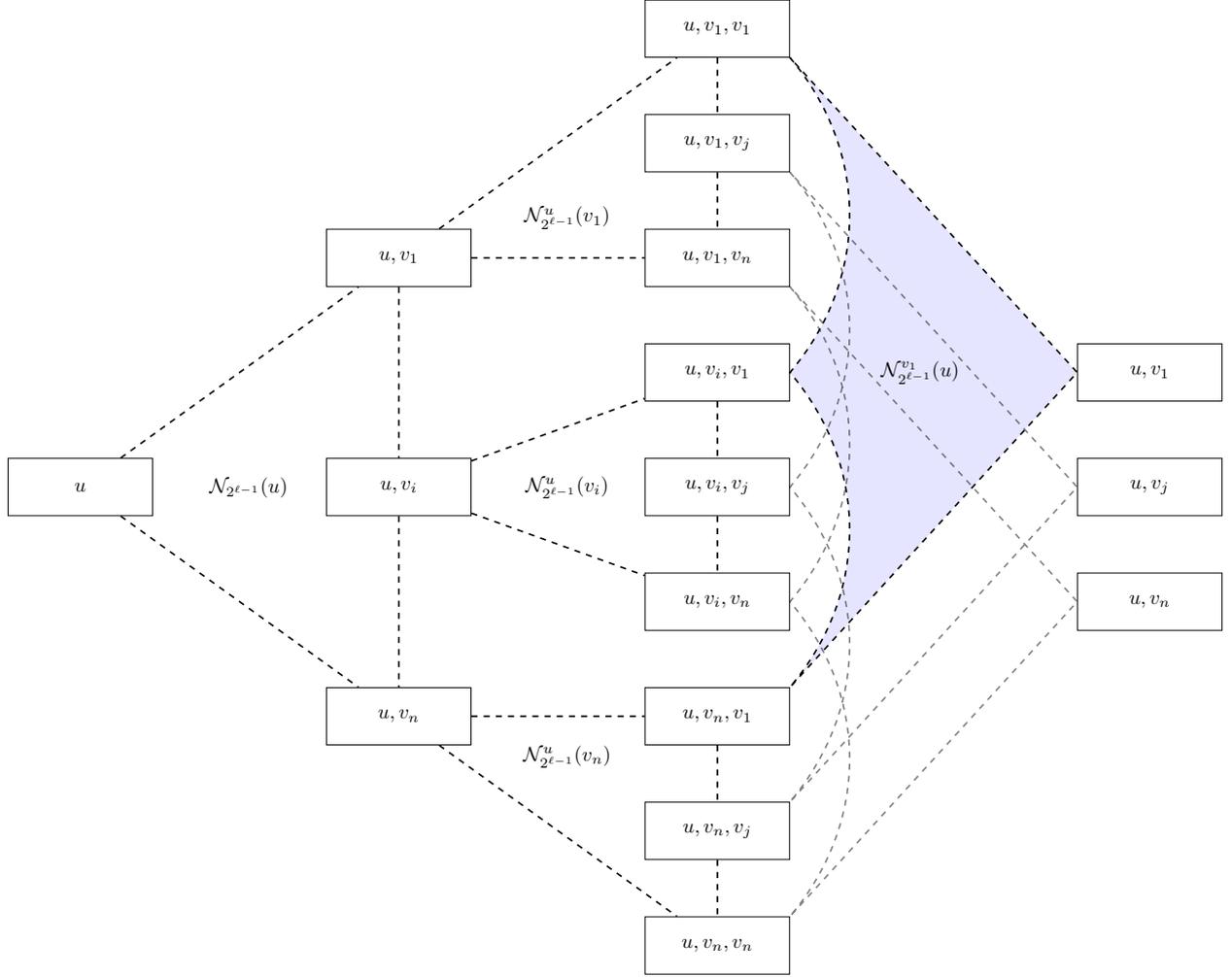[3]Such an orientation is not strictly necessary, but is more intuitive.

Figure 4: Graph construction for the switching network $\mathcal{N}_{2^\ell}(u)$ that decides whether each vertex of a graph $G = (V, E)$ is reachable from a vertex $u \in V$ by a path of length 2^ℓ.

which is a state in $\mathrm{span}\{|e\rangle : e \in E_{2^{\ell+\ell'}}\}$ that is only supported on the σ-labeled copy of $\mathcal{N}_{2^\ell}$ (which we may denote $\mathcal{N}_{2^\ell}^\sigma$) in $\mathcal{N}_{2^{\ell+\ell'}}$.

Each of the $2n + 1$ copies of $\mathcal{N}_{2^{\ell-1}}$ in $\mathcal{N}_{2^\ell}$ has a unique label $\sigma \in \Sigma$. From this label, and the (global) root u of $\mathcal{N}_{2^\ell}(u)$, we can extract the root of the specific copy $\mathcal{N}_{2^{\ell-1}}^\sigma$, and the vertex that is additionally stored in all of its vertices.

Inductively, we assign to each copy of $\mathcal{N}_{2^{\ell-k}}, k \in [\ell]$ a label $\sigma \in \Sigma^k$, from which we can extract the root of the specific copy $\mathcal{N}_{2^{\ell-k}}^\sigma$ and the set that is additionally stored in its vertices, knowing the global root $u \in V$. In particular, each copy of $\mathcal{N}_1$ has a label $\sigma \in \Sigma^\ell$ and consists of n edges. We can extract the root of each copy of $\mathcal{N}_1$ from its label σ and the global root $u \in V$ and, hence, recover the edge query labels as well. In the following lemma, we show how exactly the query label can be extracted from the label of an edge.

Lemma 4.1. *Let $\sigma \in \Sigma^\ell$ and $i \in [n]$ encode an edge in $\mathcal{N}_{2^\ell}(u)$. Then its edge is labeled by the query $((v_{\sigma_{(f_1(\sigma))_2}}), v_i)$, if $f_1(\sigma) \neq 0$, and (u, v_i) otherwise.*

Proof. We prove the statement by induction. In the base case of $\mathcal{N}_1(u)$, the claim is trivial. Since $\sigma \in \Sigma^0$ is an empty string, we have $f_1(\sigma) = 0$. The edges are encoded as $|e_i\rangle = |i\rangle$, and the corresponding query labels are (u, v_i).

For the induction step, assume that the claim holds for $\mathcal{N}_{2^{\ell-1}}$. The switching network $\mathcal{N}_{2^\ell}(u)$ consists of $2n + 1$ copies of $\mathcal{N}_{2^{\ell-1}}$ labeled by $0, (1, i), (2, j)$, where $i, j \in [n]$. The edges in each of these switching networks are encoded by $|\tilde{\sigma}, k\rangle$, where $\tilde{\sigma} \in \Sigma^{\ell-1}$ and $k \in [n]$. First, consider the copies of

$\mathcal{N}_{2^{\ell-1}}$ labeled by 0 and $(2, j)$ for $j \in [n]$. These are the switching networks $\mathcal{N}_{2^{\ell-1}}(u)$ and $\mathcal{N}_{2^{\ell-1}}^{v_j}(u)$ for $j \in [n]$, all of which have u as root, which is the same as the global root of $\mathcal{N}_{2^\ell}(u)$. By the induction hypothesis, the query label of an edge $|\tilde{\sigma}, k\rangle$ in one of these switching networks is $((v_{\tilde{\sigma}_{(f_1(\tilde{\sigma}))_2}}), v_k)$, if $f_1(\tilde{\sigma}) \neq 0$, and (u, v_k) otherwise. In the global switching network $\mathcal{N}_{2^\ell}(u)$, the same edge is encoded by $|\sigma, k\rangle$, where $\sigma = 0\tilde{\sigma}$ or $\sigma = (2, j)\tilde{\sigma}$, depending on the copy of $\mathcal{N}_{2^{\ell-1}}(u)$. Note that, in both cases, $f_1(\sigma) = f_1(\tilde{\sigma})$, which proves the statement for the copies of $\mathcal{N}_{2^{\ell-1}}$ labeled by 0 and $(2, j)$ for $j \in [n]$.

Next, consider the copies of $\mathcal{N}_{2^{\ell-1}}^{(1,i)}$ for $i \in [n]$. These are the switching networks $\mathcal{N}_{2^{\ell-1}}^u(v_i)$ for $i \in [n]$ that have v_i as their roots. By the induction hypothesis, the query label of an edge $(\tilde{\sigma}, k)$ in one of these switching networks is $((v_{\tilde{\sigma}_{(f_1(\tilde{\sigma}))_2}}), v_k)$, if $f_1(\tilde{\sigma}) \neq 0$, and (v_i, v_k) otherwise. In the global switching network $\mathcal{N}_{2^\ell}(u)$, the same edge is encoded by $|\sigma, k\rangle$, where $\sigma = (1, i)\tilde{\sigma}$. Note that $f_1(\sigma) = f_1(\tilde{\sigma})$ if $f_1(\tilde{\sigma}) \neq 0$ and $f_1(\sigma) = 1$ otherwise. This proves the statement for the copies of $\mathcal{N}_{2^{\ell-1}}$ labeled by $(1, i)$ with $i \in [n]$, and thus concludes the proof. □

Finally, we describe a top-down approach to building up $\mathcal{N}_{2^{\ell+1}}$.

Lemma 4.2. *Let $\mathcal{N}'$ be a switching network obtained from $\mathcal{N}_{2^\ell}$ by replacing each $\mathcal{N}_1$ block with an $\mathcal{N}_2$ block with the same boundary. Then $\mathcal{N}' = \mathcal{N}_{2^{\ell+1}}$.*

Proof. We prove the statement by induction. The base case is trivial, $\mathcal{N}_1$ is entirely replaced with $\mathcal{N}_2$. For the induction step, assume that the statement holds for every $\mathcal{N}_{2^{\ell'}}$ such that $\ell' < \ell$. To show it for $\mathcal{N}_{2^\ell}$, we observe that it consists of $\mathcal{N}_{2^{\ell-1}}$ blocks. For each such block, if we replace every $\mathcal{N}_1$ block with $\mathcal{N}_2$, it becomes $\mathcal{N}_{2^\ell}$ by the induction hypothesis. Therefore, the whole switching network becomes $\mathcal{N}_{2^{\ell+1}}$. □

4.1.2 Correctness of the construction

Next, we show that this construction is indeed a switching network that simultaneously decides the connectivity of all v_i to u. The proofs in this section follow the general ideas of [Pot15, Chapter 3], adapted to the setting of our switching network.

Lemma 4.3. *For every $\ell \in \{0, \ldots, \log L\}$ and $u, v_i \in V$, v_i is reachable from u by a path of length at most 2^ℓ in G if and only if source $[u]$ and sink $[u, v_i]$ are connected in $\mathcal{N}_{2^\ell}(u)(G)$.*

To prove the statement of Lemma 4.3, we define the following *pebbling game* on the input graph G.

- Initially, there is one pebble on the vertex u;

- For vertices $v, v' \in V$ such that $(v, v') \in E$, if there is a pebble on v, it is legal to put a pebble on v' or remove a pebble from v'.

In such a game, various choices of legal moves give rise to different "pebblings" – sets of vertices containing pebbles – of the graph. Clearly no vertex not reachable from u can ever be pebbled (i.e., contain a pebble). Restricting the number of pebbles available may further restrict the possible configurations achievable, as the following two lemmas show.

Lemma 4.4 ([LV96])**.** *Let $D(\ell)$ be the maximal distance from u on which a vertex can be pebbled if ℓ pebbles are available in the game. Then $D(\ell) \leq 2^{\ell-1} - 1$.*

Lemma 4.5. *Let $D_r(\ell)$ be the maximal distance from u such that it is possible to obtain pebbling configuration $[u, v]$ for some $v \in V$ using only ℓ pebbles. Then $D_r(\ell) \leq 2^{\ell-2}$*

Proof. We prove the statement by showing that $D_r(\ell + 1) \leq D(\ell) + 1$. The claim then follows from Lemma 4.4. Assume for contradiction that it is possible to obtain a configuration $[u, u_{D(\ell)+2}]$ for some $u_{D(\ell)+2} \in V(G)$ such that the distance from u to $u_{D(\ell)+2}$ is at least $D(\ell) + 2$. Before the last time a pebble is placed on $u_{D(\ell)+2}$, there is a pebble on some $u_{D(\ell)+1} \in V(G)$ such that the distance from u to $u_{D(\ell)+1}$ is at least $D(\ell) + 1$. After this, the pebble is removed from $u_{D(\ell)+1}$ using only ℓ pebbles. Consider the sequence of moves that accomplishes this in reverse. It is a sequence of moves that allows to put a pebble on $u_{D(\ell)+1}$ using only ℓ pebbles, since one pebble always stays on $u_{D(\ell)+2}$. This is a contradiction, since the distance from u to $u_{D(\ell)+1}$ is at least $D(\ell) + 1$. □

17

Lemma 4.6. *If a vertex $v \in V(G)$ is reachable from u by a path of length at most L in G, then it is possible to obtain pebbling configuration $[u, v]$ using $\log L + 2$ pebbles and $L^{\log 3}$ moves of the pebbling game on G.*

Proof. Let $u = u_0, u_1, \ldots, u_L = v$ be the path from u to v in G. Without loss of generality, its length is L, which is a power of 2. We show by induction how to put pebbles on vertices of the path to obtain the configuration $[u, v]$. Assume that there is a sequence of M_k moves that ends in the configuration $[u, u_k]$, $k \leq L/2$. Then we can obtain $[u, u_{2k}]$ as follows, using $M_{2k} = 3T_k$ moves.

1. Perform the sequence of T_k moves to obtain $[u, v_k]$;

2. Perform the same sequence of T_k moves with respect to v_k to obtain $[u, v_k, v_{2k}]$;

3. Perform step 1 in reverse to obtain $[u, v_{2k}]$.

The base case is putting a pebble on v_1 while there is a pebble on u, which is a valid pebbling game move. In the base case, we use 2 pebbles while traversing distance 1 in $M_1 = 1$ moves. In the induction step, we double the distance from u and use one additional pebble for this. Therefore, we need $\log(L) + 2$ pebbles in order to obtain $[u, v]$, and $M_L = 3M_{L/2} = 3^{\log L} = L^{\log 3}$. $\square$

Lemma 4.7. *Let S be a list of vertices of G. If pebbling configuration $[S]$ can be obtained in the pebbling game on G then every $v \in S$ is reachable from u in G.*

Proof. We prove this statement by induction. The base case is the starting configuration $[u]$, the starting vertex u is reachable from itself. For the induction step we assume that $[S]$ is a pebbling configuration that can be obtained in the pebbling game and each vertex in S is reachable from u in G. Then, we consider the two types of possible pebbling game moves.

- If we remove a pebble from a vertex in S, then all remaining vertices remain reachable from u.

- If we put a pebble on a new vertex v', then it is reachable by some vertex $v \in S$. Since v is reachable from u by the induction hypothesis, we can combine paths form u to v and from v to v' and conclude that v' is reachable from u. $\square$

Lemma 4.8. *It is possible to obtain pebbling configuration $[u, v]$ in the pebbling game on G if and only if there is a path from u to v in G.*

Proof. If there is a path from u to v in G, then, by Lemma 4.6, configuration $[u, v]$ can be obtained in the pebbling game.

If it is possible to obtain pebbling configuration $[u, v]$ in the pebbling game on G, then, by Lemma 4.7, v is reachable from u in G. $\square$

Lemma 4.9. *Every path in $\mathcal{N}_{2^\ell}(u)(G)$ corresponds to a sequence of moves in the pebbling game on G transforming the corresponding configurations.*

Proof. Every vertex of the switching network contains a list of vertices of G and represents a configuration of pebbles. By construction, every edge that is "on" is of the form $\{[S], [S, v']\}$ labeled by $(v, v') \in E(G)$ for some $v \in S$. Therefore, depending on the direction, an edge of a path in the switching network corresponds either to adding a pebble to a vertex of G or removing a pebble from a vertex of G. $\square$

Proof of Lemma 4.3. We prove by induction that if v_i is reachable from u by a path of length 2^ℓ, then the source $[u]$ and the sink $[u, v_i]$ are connected in $\mathcal{N}_{2^\ell}(u)(G)$.

Base case. For $\ell = 0$, consider $\mathcal{N}_1(u)(G)$. If the edge (u, v_i) is present in G, then the edge with query label (u, v_i) and endpoints $[u]$ and $[u, v_i]$ is present in $\mathcal{N}_1(u)(G)$. Consequently, $[u]$ and $[u, v_i]$ are connected.

Induction step. Assume the claim holds for $2^{\ell-1}$. Let p be a path of length at most 2^ℓ from u to v_i in G, and let u' denote the midpoint of p. Then u' is reachable from u by a path of length at most $2^{\ell-1}$, and v_i is reachable from u' by a path of length at most $2^{\ell-1}$. By the induction hypothesis:

- $[u]$ and $[u,u']$ are connected in $\mathcal{N}_{2^{\ell-1}}(u)(G)$;

- $[u,u']$ and $[u,u',v_i]$ are connected in $\mathcal{N}_{2^{\ell-1}}^u(u')(G)$;

- $[u,u',v_i]$ and $[u,v_i]$ are connected in $\mathsf{Rev}(\mathcal{N}_{2^{\ell-1}}^{v_i}(u)(G))$.

Concatenating these paths yields a path from $[u]$ to $[u,v_i]$ in $\mathcal{N}_{2^\ell}(u)(G)$. This completes the induction.

If source $[u]$ and sink $[u,v_i]$ are connected in $\mathcal{N}_{2^\ell}(u)(G)$, then, by Lemma 4.9, there is a sequence of pebbling game moves on G that transforms $[u]$ into $[u,v_i]$. Therefore, by Lemma 4.8, there is a path from u to v_i in G. Note that the configurations in the sequence contain at most $\ell + 2$ vertices of G, by construction of $\mathcal{N}_{2^\ell}(u)(G)$. Hence, by Lemma 4.5, the path in G is of length at most 2^ℓ. $\qquad\square$

4.2 Complexity analysis of the switching network

Lemma 4.10. *Fix any $\ell \in \{0, \ldots, \log L\}$ and $u, v \in V$ such that there is a path from u to v in G of length at most 2^ℓ. Then there is a path connecting the source $[u]$ and the sink $[u,v]$ in $\mathcal{N}_{2^\ell}(u)(G)$ of length at most $2^{\ell \log 3}$.*

Proof. The statement of this lemma follows directly from the proof of Lemma 4.3. $\qquad\square$

Lemma 4.11. *For any $\ell \in \{0, \ldots, \log L\}$, and $u \in V$, $|E(\mathcal{N}_{2^\ell}(u))| \leq (2n+1)^\ell n$.*

Proof. $|E(\mathcal{N}_{2^\ell}(u))| = (2n+1) |E(\mathcal{N}_{2^{\ell-1}}(u))| = (2n+1)^\ell |E(\mathcal{N}_1(u))| = (2n+1)^\ell n$. $\qquad\square$

4.3 Basis of the space of st-flows

In order to apply Theorem 2.9, we need to describe a basis for $\mathcal{B}^\perp$, and give an efficient procedure for generating it. By Lemma 2.16, it is enough to describe a basis for the st-flow space $\mathcal{F}(\mathcal{N}_L)$ (see Definition 2.13), and by Lemma 2.15, we can further decompose this task into finding a basis for the circulation space $\mathcal{C}(\mathcal{N}_L)$ (see Definition 2.13) and an optimal st-flow. We first define the basis, and then describe a procedure for generating it.

4.3.1 Basis definition

In this section, we prove the following theorem, which, if we take $u = s$ and $v_j = t$, gives us a working basis of $\mathcal{B}^\perp$ for the switching network $\mathcal{N}_L(s,t)$.

Theorem 4.12. *For $j \in \{0,1\}^{\log n}$ and $\ell \in [\log(L)]$, let $\theta_j(2^{\ell-1})$ be the optimal unit $[u], [u, v_j]$-flow in $\mathcal{N}_{2^{\ell-1}}$, and let $|\bar{\theta}_j(2^{\ell-1})\rangle$ be as in (3). For $i, j \in \{0,1\}^{\log n}$, let*

$$|p_{ij}(2^\ell)\rangle = |\bar{\theta}_i^0(2^{\ell-1})\rangle + |\bar{\theta}_j^{1i}(2^{\ell-1})\rangle - |\bar{\theta}_i^{2j}(2^{\ell-1})\rangle = |0,\bar{0}\rangle|\bar{\theta}_i(2^{\ell-1})\rangle + |1,i\rangle|\bar{\theta}_j(2^{\ell-1})\rangle + |2,j\rangle|\bar{\theta}_i(2^{\ell-1})\rangle,$$

where the superscript encodes the copy of $\mathcal{N}_{2^{\ell-1}}$ in $\mathcal{N}_{2^\ell}$ (see Figure 5), and for each $x, z \in \{0,1\}^{\log n}$, let

$$|\psi_{z,x}(2^\ell)\rangle = \sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} \sum_{i \in \{0,1\}^{\log n}} (-1)^{z \cdot i} |p_{ij}(2^\ell)\rangle.$$

Finally, let $|\theta_j(L)\rangle$ be as in (4). Then

$$\bigcup_{\ell=1}^{\log(L)} \bigcup_{\sigma \in \Sigma^{\log(L)-\ell}} \left\{ \underbrace{|\sigma\rangle|\psi_{z,x}(2^\ell)\rangle}_{=\pm|\psi_{z,x}^\sigma(2^\ell)\rangle} : z, x \in \{0,1\}^{\log n}, z \neq \bar{0} \right\} \cup \{|\theta_j(L)\rangle, |[u]\rangle, |[u,v_j]\rangle\}$$

is an orthogonal basis of $\mathcal{B}^\perp$.

This result follows directly from Lemma 4.13 (below), which recursively constructs an orthogonal basis for the circulation space, $\mathcal{C}(\mathcal{N}_{2^\ell})$, of $\mathcal{N}_{2^\ell}$ (see Definition 2.13); Lemma 4.15 (below), which recursively constructs an optimal flow state, which is a basis for the space of optimal flows; and Lemma 2.15, which states that the space of flows decomposes as the direct sum of the circulation space and the (one-dimensional) space of optimal flows.
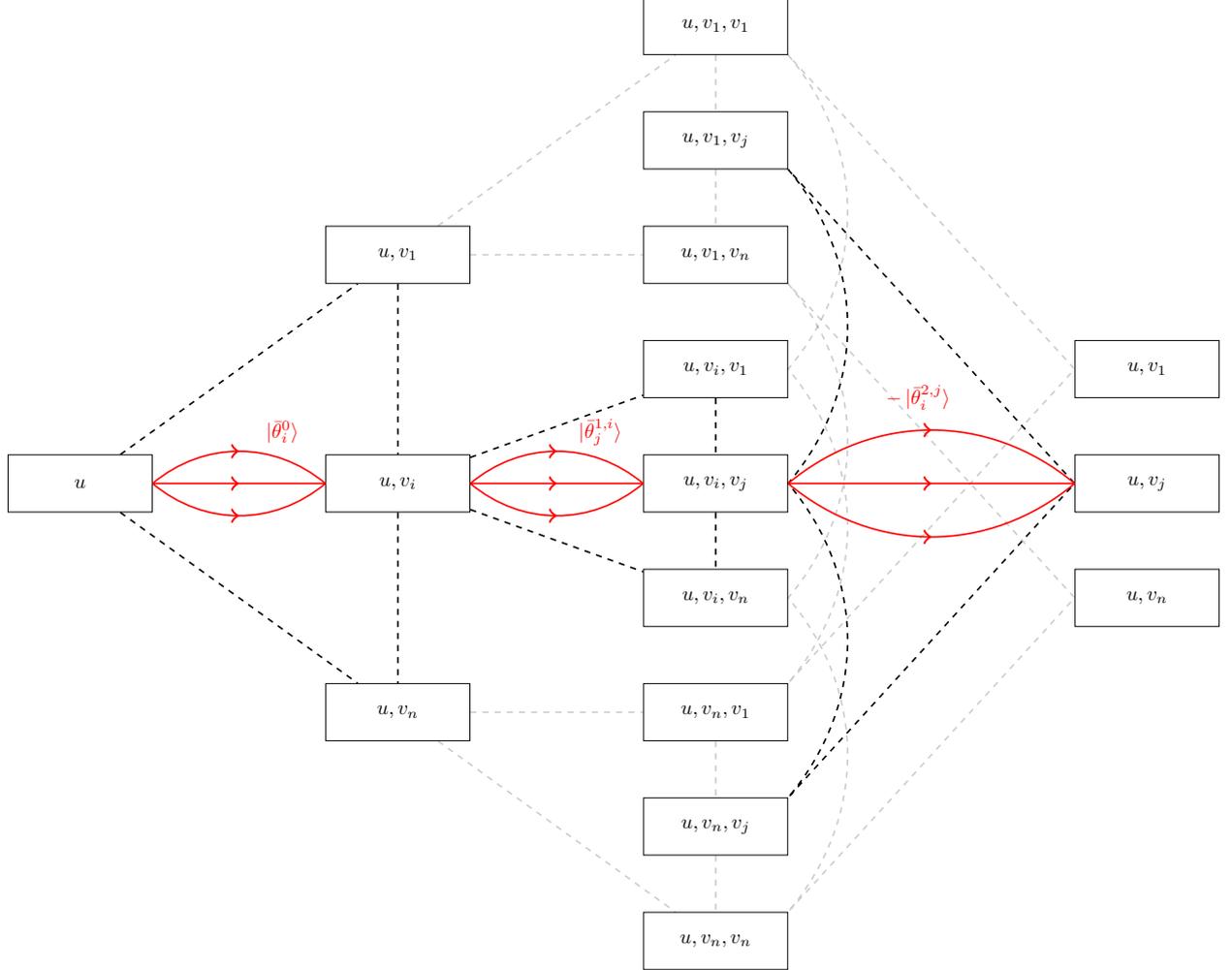

Figure 5: Visualization of the state $|p_{ij}\rangle = |\bar{\theta}_i^0\rangle + |\bar{\theta}_j^{1i}\rangle - |\bar{\theta}_i^{2j}\rangle = |0,\bar{0}\rangle|\bar{\theta}_i\rangle + |1,i\rangle|\bar{\theta}_j\rangle + |2,j\rangle|\bar{\theta}_i\rangle$ in $\mathcal{N}_{2^\ell}$, where $j \in \{0,1\}^{\log n}$ and $\ell \in [\log(L)]$. Here θ_j denotes the optimal unit $[u],[u,v_j]$-flow in $\mathcal{N}_{2^{\ell-1}}$, and $|\bar{\theta}_j\rangle$ is as in (3). The superscript encodes the copy of $\mathcal{N}_{2^{\ell-1}}$ in $\mathcal{N}_{2^\ell}$.

Lemma 4.13. *For $j \in \{0,1\}^{\log n}$ and $\ell \in [\log(L)]$, let θ_j be the optimal unit $[u],[u,v_j]$-flow in $\mathcal{N}_{2^{\ell-1}}$, and let $|\bar{\theta}_j\rangle$ be as in (3). For $i,j \in \{0,1\}^{\log n}$, let $|p_{ij}\rangle = |p_{ij}(2^\ell)\rangle$ and $|\psi_{z,x}\rangle = |\psi_{z,x}(2^\ell)\rangle$ be as in Theorem 4.12. Let $\{|b_1\rangle, \ldots, |b_D\rangle\}$ be an orthogonal basis for $\mathcal{C}(\mathcal{N}_{2^{\ell-1}})$. Then*

$$\left\{|\sigma\rangle|b_d\rangle : \sigma \in \Sigma, d \in [D]\right\} \cup \left\{|\psi_{z,x}\rangle : z,x \in \{0,1\}^{\log n}, z \neq \bar{0}\right\},$$

is an orthogonal basis of the space of circulations $\mathcal{C}(\mathcal{N}_{2^\ell})$.

Proof. Clearly, the vectors $\{|\sigma\rangle|b_d\rangle\}_{\sigma,d}$ are pairwise orthogonal. Moreover, for each $\sigma \in \Sigma$, $\text{span}\{|\sigma\rangle|b_d\rangle : d \in [D]\}$ is contained in $\mathcal{C}(\mathcal{N}_{2^\ell})$, since any circulation on a $\mathcal{N}_{2^{\ell-1}}$ block – in this case, the one labeled by σ – is a circulation on the full graph $\mathcal{N}_{2^\ell}$. Finally, each state $|\psi_{z,x}\rangle$ is orthogonal to every subspace $\text{span}\{|\sigma\rangle|b_d\rangle : d \in [D]\}$, as it is constructed from optimal flows within the $\mathcal{N}_{2^{\ell-1}}$ blocks that are orthogonal to their respective circulation subspaces, by Lemma 2.14, and don't overlap other blocks.

20

Next, we compute the dimension of $\mathcal{C}(\mathcal{N}_{2\ell})$. By Lemma 2.17, $\dim \mathcal{C}(\mathcal{N}_{2\ell}) = |E(\mathcal{N}_{2\ell})| - |V(\mathcal{N}_{2\ell})| + 1$. From the construction of $\mathcal{N}_{2\ell}$, $|E(\mathcal{N}_{2\ell})| = (2n+1)|E(\mathcal{N}_{2\ell-1})|$, and $|V(\mathcal{N}_{2\ell})| = (2n+1)|V(\mathcal{N}_{2\ell-1})| - n^2 - n$. Therefore,

$$
\begin{aligned}
\dim \mathcal{C}(\mathcal{N}_{2\ell}) &= |E(\mathcal{N}_{2\ell})| - |V(\mathcal{N}_{2\ell})| + 1 \\
&= (2n+1)|E(\mathcal{N}_{2\ell-1})| - (2n+1)|V(\mathcal{N}_{2\ell-1})| + n^2 + n + 1 \\
&= (2n+1)\underbrace{(|E(\mathcal{N}_{2\ell-1})| - |V(\mathcal{N}_{2\ell-1})| + 1)}_{\dim \mathcal{C}(\mathcal{N}_{2\ell-1})} + n^2 - n.
\end{aligned}
$$

This implies that a basis for $\mathcal{C}(\mathcal{N}_{2\ell})$ can be constructed by taking orthogonal bases of the circulation spaces of each $\mathcal{N}_{2\ell-1}$ block within $\mathcal{N}_{2\ell}$, and adding $n^2 - n$ additional states that are orthogonal to these subspace bases.

Next, we argue that $|\psi_{z,x}\rangle$ are indeed circulations. We can interpret $\psi_{z,x}$ as a function on the edges in the natural way: $\psi_{z,x}(e) = \langle \psi_{z,x}|e\rangle$. Then, as usual, $\psi_{z,x}(\mathsf{u})$ denotes the total flow on u: $\psi_{z,x}(\mathsf{u}) = \sum_{e \in E^{\to}(\mathsf{u})} \psi_{z,x}(e) - \sum_{e \in E^{\leftarrow}(\mathsf{u})} \psi_{z,x}(e)$.

Note that each p_{ij} is a $[u], [u, v_j]$-flow in $\mathcal{N}_{2\ell}$, as shown in Figure 5, so in particular, it has boundary $B = \{[u], [u, v_1], \ldots, [u, v_n]\}$. Hence, each $\psi_{z,x}$ is a flow in $\mathcal{N}_{2\ell}$ with boundary B as well. Therefore, $\psi_{z,x}(\mathsf{v}) = 0$ for any $\mathsf{v} \in V(\mathcal{N}_{2\ell}) \setminus B$. It remains to show that $\psi_{z,x}(\mathsf{v}) = 0$ for any $\mathsf{v} \in B$ as well. Since each $|p_{ij}\rangle$ is a unit $[u], [u, v_j]$-flow and $z \neq \bar{0}$, we get

$$
\psi_{z,x}([u]) = \sum_{j \in \{0,1\}^n} (-1)^{x \cdot j} \sum_{i \in \{0,1\}^n} (-1)^{z \cdot i} \underbrace{p_{ij}([u])}_{=1} = 0
$$

$$
\psi_{z,x}([u, v_{j'}]) = \sum_{j \in \{0,1\}^n} (-1)^{x \cdot j} \sum_{i \in \{0,1\}^n} (-1)^{z \cdot i} \underbrace{p_{i,j}([u, v_{j'}])}_{=\delta_{j,j'}} = (-1)^{x \cdot j'} \sum_{i \in \{0,1\}^n} (-1)^{z \cdot i} = 0.
$$

Hence, each $|\psi_{z,x}\rangle$ is a circulation. To complete the proof, it remains to show that the states $|\psi_{z,x}\rangle$ are pairwise orthogonal, towards which the following claim is helpful.

Claim 4.14. *There exist real numbers c, c' such that for all $i, j, x, z \in \{0,1\}^n$ such that $z \neq \bar{0}$,*

$$
\langle p_{ij}|\psi_{z,x}\rangle = (-1)^{z \cdot i}(c(-1)^{x \cdot j} + c' A_{x,j}),
$$

where $A_{x,j} = \sum_{j' \in \{0,1\}^{\log n}: j' \neq j} (-1)^{x \cdot j'}$.

Proof. First, we observe that, due to symmetry of the graph $\mathcal{N}_{2\ell-1}$, inner products between optimal flows $|\bar{\theta}_j\rangle$ in this graph are constant. More precisely, there exist $c_0, c_1 \in \mathbb{R}$ such that

$$
\langle \bar{\theta}_j|\bar{\theta}_{j'}\rangle = \begin{cases} c_0, & \text{if } j \neq j' \\ c_1, & \text{if } j = j'. \end{cases}
$$

Therefore, the inner product $\langle p_{ij}|p_{i'j'}\rangle$ only depends on whether $i = i'$ and $j = j'$. Indeed,

$$
\begin{aligned}
\langle p_{ij}|p_{i'j'}\rangle &= \langle \bar{\theta}_i^0|\bar{\theta}_{i'}^0\rangle + \langle \bar{\theta}_j^{1i}|\bar{\theta}_{j'}^{1i'}\rangle + \langle \bar{\theta}_i^{2j}|\bar{\theta}_{i'}^{2j'}\rangle \\
&= \begin{cases} c_{00} := c_0 + 0 + 0, & \text{if } i \neq i', j \neq j' \\ c_{01} := c_0 + 0 + c_0, & \text{if } i \neq i', j = j' \\ c_{10} := c_1 + c_0 + 0, & \text{if } i = i', j \neq j' \\ c_{11} := c_1 + c_1 + c_1, & \text{if } i = i', j = j'. \end{cases}
\end{aligned}
\tag{11}
$$

Since $z \neq \bar{0}$,

$$
\sum_{i' \in \{0,1\}^{\log n}: i' \neq i} (-1)^{z \cdot i'} = \sum_{i' \in \{0,1\}^{\log n}} (-1)^{z \cdot i'} - (-1)^{z \cdot i} = -(-1)^{z \cdot i}.
$$

We use this to compute:

$$
\begin{aligned}
\langle p_{ij}|\psi_{z,x}\rangle &= \sum_{i',j'\in\{0,1\}^{\log n}} (-1)^{x\cdot j'+z\cdot i'}\langle p_{ij}|p_{i'j'}\rangle \\
&= (-1)^{x\cdot j+z\cdot i}\langle p_{ij}|p_{ij}\rangle + \sum_{i'\in\{0,1\}^{\log n}:i'\neq i}(-1)^{x\cdot j+z\cdot i'}\langle p_{ij}|p_{i'j}\rangle \\
&\quad + \sum_{j'\in\{0,1\}^{\log n}:j'\neq j}(-1)^{x\cdot j'+z\cdot i}\langle p_{ij}|p_{ij'}\rangle + \sum_{i',j'\in\{0,1\}^{\log n}:i'\neq i,j'\neq j}(-1)^{x\cdot j'+z\cdot i'}\langle p_{ij}|p_{i'j'}\rangle \\
&= (-1)^{x\cdot j+z\cdot i}c_{11} - (-1)^{x\cdot j+z\cdot i}c_{01} + \big((-1)^{z\cdot i}c_{10}-(-1)^{z\cdot i}c_{00}\big)\sum_{j'\in\{0,1\}^{\log n}:j'\neq j}(-1)^{x\cdot j'},
\end{aligned}
$$

from which the result follows by taking $c = c_{11}-c_{01}$ and $c' = c_{10}-c_{00}$. $\qquad\square$

This allows us to compute the inner product $\langle\psi_{z,x}|\psi_{z',x'}\rangle$ with $(z,x)\neq(z',x')$ and $z,z'\neq\bar 0$:

$$
\begin{aligned}
\langle\psi_{z',x'}|\psi_{z,x}\rangle &= \sum_{i,j\in\{0,1\}^{\log n}}(-1)^{x'\cdot j+z'\cdot i}\langle p_{ij}|\psi_{z,x}\rangle \\
&= \sum_{i,j\in\{0,1\}^{\log n}}(-1)^{x'\cdot j+z'\cdot i}(-1)^{z\cdot i}(c(-1)^{x\cdot j}+c'A_{x,j}) \qquad\text{by Claim 4.14} \\
&= \sum_{i\in\{0,1\}^{\log n}}(-1)^{(z'+z)\cdot i}\cdot\sum_{j\in\{0,1\}^{\log n}}(c(-1)^{(x'+x)\cdot j}+c'(-1)^{x'\cdot j}A_{x,j}).
\end{aligned}
$$

If $z\neq z'$, then the first product term is 0, and so $\langle\psi_{z',x'}|\psi_{z,x}\rangle = 0$. Suppose $z=z'$, but $x\neq x'$, which is the only other way to have $(z,x)\neq(z',x')$. Then

$$
\langle\psi_{z',x'}|\psi_{z,x}\rangle = n\cdot\sum_{j\in\{0,1\}^{\log n}}(c(-1)^{(x'+x)\cdot j}+c'(-1)^{x'\cdot j}A_{x,j}) = c'n\sum_{j\in\{0,1\}^{\log n}}(-1)^{x'\cdot j}A_{x,j}.
$$

Using

$$
A_{x,j} = \sum_{j'\in\{0,1\}^{\log n}:j'\neq j}(-1)^{x\cdot j'} = \begin{cases} n-1 & \text{if } x=\bar 0 \\ -(-1)^{x\cdot j} & \text{else,}\end{cases}
$$

we see that if $x\neq\bar 0$, then

$$
\langle\psi_{z',x'}|\psi_{z,x}\rangle = -c'n\sum_{j\in\{0,1\}^{\log n}}(-1)^{(x'+x)\cdot j} = 0,
$$

and otherwise, we must have $x'\neq\bar 0$, so

$$
\langle\psi_{z',x'}|\psi_{z,x}\rangle = c'n(n-1)\sum_{j\in\{0,1\}^{\log n}}(-1)^{x'\cdot j} = 0,
$$

completing the proof. $\qquad\square$

We now describe the form of optimal flows in $\mathcal{N}_{2^\ell}$, by recursively combining the optimal flows in $\mathcal{N}_{2^{\ell-1}}$. For the base case $\mathcal{N}_1$, the optimal $[u],[u,v_j]$-flow simply assigns a unit of flow to the single edge between $[u]$ and $[u,v_j]$.

Lemma 4.15. *For $i,j\in\{0,1\}^{\log n}$, let $|p_{ij}\rangle = |p_{ij}(2^\ell)\rangle$ be as in Theorem 4.12. Then for any j, the optimal unit $[u],[u,v_j]$-flow in $\mathcal{N}_{2^\ell}(u,v_j)$ is*

$$
|\theta_j(2^\ell)\rangle = -|\leftarrow,[u]\rangle + \frac{1}{n}\sum_{i\in\{0,1\}^{\log n}}|p_{ij}\rangle + |\rightarrow,[u,v_j]\rangle.
$$

Proof. Note that each p_{ij} is a $[u], [u, v_j]$-flow in $\mathcal{N}_{2^\ell}$, as shown in Figure 5, though it is not optimal, since it doesn't spread out through the $(1, i')$ copies of $\mathcal{N}_{2^{\ell-1}}$ for $i' \neq i$. Intuitively, $\theta_j(2^\ell)$, which is also easily seen to be a unit $[u], [u, v_j]$-flow on $\mathcal{N}_{2^\ell}$, is optimal because it spreads the flow across all values of i. We prove that it is optimal by showing that $|\theta_j(2^\ell)\rangle$ is orthogonal to the circulation subspace $\mathcal{C}(\mathcal{N}_{2^\ell})$. By Lemma 2.14 and Lemma 2.15, this orthogonality implies that $|\theta_j(2^\ell)\rangle$ is the optimal unit flow.

By Lemma 4.13, the space $\mathcal{C}(\mathcal{N}_{2^\ell})$ has the following orthogonal basis:

$$\{|\sigma\rangle|b_d\rangle : \sigma \in \Sigma, d \in [D]\} \cup \left\{|\psi_{z,x}\rangle : z, x \in \{0,1\}^{\log n}, z \neq \bar{0}\right\},$$

where $|\psi_{z,x}\rangle = |\psi_{z,x}(2^\ell)\rangle$ is as in Theorem 4.12,

and $\{|b_1\rangle, \ldots, |b_D\rangle\}$ is an orthogonal basis for the circulation space $\mathcal{C}(\mathcal{N}_{2^{\ell-1}})$. All $|\theta_j(2^\ell)\rangle$ are composed of optimal unit flows of the $\mathcal{N}_{2^{\ell-1}}$ blocks and boundary states, and therefore orthogonal to all vectors $|\sigma\rangle|b_d\rangle$, which are circulations on these blocks. Hence, it remains to show that $\langle\theta_j(2^\ell)|\psi_{z,x}\rangle = 0$ for every $j, z, x \in \{0,1\}^{\log n}$ such that $z \neq \bar{0}$. We apply Claim 4.14 to get:

$$n \cdot \langle\theta_j(2^\ell)|\psi_{z,x}\rangle = \sum_{i \in \{0,1\}^{\log n}} \langle p_{ij}|\psi_{z,x}\rangle = \sum_{i \in \{0,1\}^{\log n}} (-1)^{z \cdot i}(c(-1)^{x \cdot j} + c' A_{x,j}) = 0. \qquad \square$$

4.3.2   Basis Generation

We show in this section how to prepare the basis from Theorem 4.12 in $\widetilde{O}(1)$ time. All states considered in this section lie in the subspace in which each edge $e \in E(\mathcal{N}_{2^\ell})$ is represented by $\frac{1}{\sqrt{2}}(|\rightarrow, e\rangle - |\leftarrow, e\rangle)$. Without loss of generality, we may therefore describe states in terms of the canonical edge states $\{|e\rangle = |\sigma, i\rangle : e \in E(\mathcal{N}_{2^\ell}) = \Sigma^\ell \times \{0,1\}^{\log n}\}$, and obtain the isomorphism with the above subspace by appending the auxiliary state $|-\rangle = \frac{1}{\sqrt{2}}(|\rightarrow\rangle - |\leftarrow\rangle)$ in an additional register.

Recall from (8) that edges E_{2^ℓ} of $\mathcal{N}_{2^\ell}$ are labeled $(\sigma, e_i) = (\sigma, i)$ for $\sigma \in \Sigma^\ell$ and $i \in \{0,1\}^{\log n}$. For any $\sigma \in \Sigma^\ell$, letting $\bar{\sigma} \in \{0,1,2\}^\ell$ be such that for all $t \in [\ell]$, $\sigma_t = (\bar{\sigma}_t, i)$ for some $i \in \{0,1\}^{\log n}$ (that is, $\bar{\sigma}$ encodes the first part of each entry of σ), we can break E_{2^ℓ} into 3^ℓ *layers*, defined, for each $\tau \in \{0,1,2\}^\ell$:

$$E_\tau := \{(\sigma, i) \in E_{2^\ell} : \bar{\sigma} = \tau\} = \{(\sigma, i) \in \Sigma^\ell \times \{0,1\}^{\log n} : \bar{\sigma} = \tau\}. \tag{12}$$

A crucial step in our basis generation subroutine will be taking uniform superpositions over these layers. We first prove two lemmas about the size of these layers that will enable this.

Lemma 4.16. *Fix $\ell \in [\log(L)]$. Let $\tau \in \{0,1,2\}^\ell$ encode a layer E_τ of edges in $\mathcal{N}_{2^\ell}$. Then $|E_\tau| = n^{1+|\tau|-|\tau|_0}$, where $|\tau|_0$ denotes the number of zeros in τ.*

Proof. We prove the statement by induction on ℓ. For the base case, let $\ell = 1$, so we have for any $\tau \in \{0,1,2\}$:

$$|E_\tau| = |\{(\sigma, i) \in \Sigma \times \{0,1\}^{\log n} : \bar{\sigma} = \tau\}| = n|\{\sigma \in \Sigma : \bar{\sigma} = \tau\}|.$$

If $\tau = 0$, the only $\sigma \in \Sigma = \{(0, \bar{0}), (1, i), (2, j) : i, j \in \{0,1\}^{\log n}\}$ such that $\bar{\sigma} = \tau$ is $\sigma = (0, \bar{0})$, and so

$$|E_\tau| = n \cdot 1 = n^{1+|\tau|-|\tau|_0}$$

since $|\tau| = |\tau|_0 = 1$. Otherwise, $\tau \in \{1,2\}$, and we have:

$$|E_\tau| = n|\{(\tau, i) : i \in \{0,1\}^{\log n}\}| = n^2 = n^{1+|\tau|-|\tau|_0}$$

since $|\tau| = 1$ and $|\tau|_0 = 0$.

For the induction step, assume $\ell > 1$. Referring to (12), we have, for any $\tau \in \{0,1,2\}^{\ell-1}$:

$$E_{0\tau} = \{((0, \bar{0})\sigma, i) \in E_{2^\ell} : \bar{\sigma} = \tau\} = \{((0, \bar{0}), e) : e \in E_\tau\}$$

and for $a \in \{1,2\}$, $E_{a\tau} = \{((a, j)\sigma, i) \in E_{2^\ell} : \bar{\sigma} = \tau\} = \{((a, j), e) : e \in E_\tau, j \in \{0,1\}^{\log n}\}.$ (13)

From this, and the induction hypothesis, we have:

$$|E_{0\tau}| = |E_\tau| = n^{1+|\tau|-|\tau|_0} = n^{1+|\tau|+1-(|\tau|_0+1)} = n^{1+|0\tau|-|0\tau|_0}$$

and for $a \in \{1,2\}$, $|E_{a\tau}| = n|E_\tau| = n \cdot n^{1+|\tau|-|\tau|_0} = n^{1+(1+|\tau|)-|\tau|_0} = n^{1+|a\tau|-|a\tau|_0}$.

Thus, for any $\tau' \in \{0,1,2\}^\ell$, we can conclude $|E_{\tau'}| = n^{1+|\tau'|-|\tau'|_0}$. $\qquad\square$

Lemma 4.17. *For all $\ell \in [\log(L)]$, $\sum_{\tau \in \{0,1,2\}^\ell} \frac{1}{|E_\tau|} = \frac{(n+2)^\ell}{n^{\ell+1}} = \frac{1}{n}\left(1+\frac{2}{n}\right)^\ell$.*

Proof. We first use Lemma 4.16 to compute:

$$\sum_{\tau \in \{0,1,2\}^\ell} \frac{1}{|E_\tau|} = \sum_{\tau \in \{0,1,2\}^\ell} \frac{1}{n^{1+|\tau|-|\tau|_0}} = \sum_{t=0}^{\ell} \binom{\ell}{t} 2^{\ell-t} n^{t-1-\ell}.$$

Above, we used the fact that for any $t \in \{0,\ldots,\ell\}$, there are $\binom{\ell}{t} 2^{\ell-t}$ strings in $\{0,1,2\}^\ell$ with exactly t 0s. Continuing, we have:

$$\sum_{\tau \in \{0,1,2\}^\ell} \frac{1}{|E_\tau|} = \frac{1}{n^{\ell+1}} \sum_{t=0}^{\ell} \binom{\ell}{t} 2^{\ell-t} n^t = \frac{1}{n^{\ell+1}}(n+2)^\ell,$$

by the binomial theorem. The result follows. $\qquad\square$

Next, we describe a subroutine for generating superpositions over all optimal flows, which will be a key subroutine in our basis generation.

Lemma 4.18. *For all $\ell \in [\log(L)]$, a map that acts as*

$$|0\rangle \mapsto\propto \sum_{i \in \{0,1\}^{\log n}} |\bar{\theta}_i(2^\ell)\rangle$$

can be implemented in $\widetilde{O}(1)$ steps.

Proof. We first show by induction that

$$\sum_{j \in \{0,1\}^{\log n}} |\bar{\theta}_j(2^\ell)\rangle = n \cdot \sum_{\tau \in \{0,1,2\}^\ell} \frac{1}{|E_\tau|} \sum_{e \in E_\tau} |e\rangle. \tag{14}$$

Using the definition in the statement of Theorem 4.12, we can write

$$\sum_{j \in \{0,1\}^{\log n}}^{n} |\bar{\theta}_j(2^\ell)\rangle = \frac{1}{n} \sum_{j \in \{0,1\}^{\log n}} \sum_{i \in \{0,1\}^{\log n}} |p_{ij}(2^\ell)\rangle.$$

In the base case of $\mathcal{N}_2$ ($\ell = 1$), referring to the definition of $|p_{ij}\rangle$ in Theorem 4.12, we have

$$|p_{ij}(2)\rangle = |\bar{\theta}_i^0(1)\rangle + |\bar{\theta}_j^{1i}(1)\rangle - |\bar{\theta}_i^{2j}(1)\rangle = |0,\bar{0}\rangle|\bar{\theta}_i\rangle + |1,i\rangle|\bar{\theta}_j\rangle + |2,j\rangle|\bar{\theta}_i\rangle$$
$$= |0,\bar{0}\rangle|e_i\rangle + |1,i\rangle|e_j\rangle + |2,j\rangle|e_i\rangle$$

by (10), and the fact that an optimal $[u], [u,v_i]$-flow in $\mathcal{N}_1$ simply assigns a unit of flow to e_i.

$$|p_{ij}(2)\rangle = |e_i^0\rangle + |e_j^{1i}\rangle + |e_i^{2j}\rangle.$$

We thus get

$$\sum_{j \in \{0,1\}^{\log n}} |\bar{\theta}_j(2)\rangle = \frac{1}{n} \sum_{j \in \{0,1\}^{\log n}} \sum_{i \in \{0,1\}^{\log n}} |p_{ij}(2)\rangle$$

$$= n \cdot \frac{1}{n} \sum_{i \in \{0,1\}^{\log n}} |0,\bar{0},e_i\rangle + n \cdot \frac{1}{n^2} \sum_{i,j \in \{0,1\}^{\log n}} |1,i,e_j\rangle + n \cdot \frac{1}{n^2} \sum_{i,j \in \{0,1\}^{\log n}} |2,j,e_i\rangle.$$

Since the layer encoded by 0 consists of n edges $E_0 = \{((0, \bar{0}), e_i) : i \in \{0,1\}^{\log n}\}$, and the layers encoded by 1 and 2 consist of n^2 edges $E_1 = \{((1, i), e_j) : i, j \in \{0,1\}^{\log n}\}$ and $E_2 = \{((2, j), e_i) : i, j \in \{0,1\}^{\log n}\}$ respectively, we conclude that the equality holds in the base case.

Assume, for the induction step, that the equality from (14) holds for $\mathcal{N}_{2^{\ell-1}}$. We have

$$\sum_{j \in \{0,1\}^{\log n}} |\bar{\theta}_j(2^\ell)\rangle = \frac{1}{n} \sum_{i,j \in \{0,1\}^{\log n}} |p_{ij}(2^\ell)\rangle = \frac{1}{n} \sum_{i,j \in \{0,1\}^{\log n}} \left(|\bar{\theta}_i^0(2^{\ell-1})\rangle + |\bar{\theta}_j^{1i}(2^{\ell-1})\rangle - |\bar{\theta}_i^{2j}(2^{\ell-1})\rangle \right)$$

$$= \sum_{i \in \{0,1\}^{\log n}} |0, \bar{0}\rangle |\bar{\theta}_i(2^{\ell-1})\rangle + \frac{1}{n} \sum_{i,j \in \{0,1\}^{\log n}} |1, i\rangle |\bar{\theta}_j(2^{\ell-1})\rangle + \frac{1}{n} \sum_{i,j \in \{0,1\}^{\log n}} |2, j\rangle |\bar{\theta}_i(2^{\ell-1})\rangle,$$

where we again used (10). We can use the induction hypothesis on each of the three terms to get:

$$\sum_{j \in \{0,1\}^{\log n}} |\bar{\theta}_j(2^\ell)\rangle = n|0, \bar{0}\rangle \sum_{\tau \in \{0,1,2\}^{\ell-1}} \frac{1}{|E_\tau|} \sum_{e \in E_\tau} |e\rangle + \sum_{i \in \{0,1\}^{\log n}} |1, i\rangle \sum_{\tau \in \{0,1,2\}^{\ell-1}} \frac{1}{|E_\tau|} \sum_{e \in E_\tau} |e\rangle$$

$$+ \sum_{j \in \{0,1\}^{\log n}} |2, j\rangle \sum_{\tau \in \{0,1,2\}^{\ell-1}} \frac{1}{|E_\tau|} \sum_{e \in E_\tau} |e\rangle.$$

From (13), and Lemma 4.16, it follows that

$$\sum_{j \in \{0,1\}^{\log n}} |\bar{\theta}_j(2^\ell)\rangle = \sum_{\tau \in \{0,1,2\}^{\ell-1}} \frac{1}{n^{|\tau|-|\tau|_0}} \sum_{e \in E_\tau} |(0, \bar{0}), e\rangle$$

$$+ \sum_{a \in \{1,2\}} \sum_{\tau \in \{0,1,2\}^{\ell-1}} \frac{1}{n^{1+|\tau|-|\tau|_0}} \sum_{i \in \{0,1\}^{\log n}, e \in E_\tau} |(a, i), e\rangle$$

$$= \sum_{\tau \in \{0,1,2\}^{\ell-1}} \frac{1}{n^{|0\tau|-|0\tau|_0}} \sum_{e' \in E_{0\tau}} |e'\rangle + \sum_{a \in \{1,2\}} \sum_{\tau \in \{0,1,2\}^{\ell-1}} \frac{1}{n^{|a\tau|-|a\tau|_0}} \sum_{e' \in E_{a\tau}} |e'\rangle$$

$$= n \sum_{\tau' \in \{0,1,2\}^\ell} \frac{1}{|E_{\tau'}|} \sum_{e' \in E_{\tau'}} |e'\rangle,$$

as desired. Next, we show how to prepare this state in two steps.

Step 1. Prepare a superposition of layer names with the correct amplitudes:

$$\frac{1}{\sqrt{\sum_{\tau' \in \{0,1,2\}^\ell} \frac{1}{|E_{\tau'}|}}} \sum_{\tau \in \{0,1,2\}^\ell} \frac{1}{\sqrt{|E_\tau|}} |\tau\rangle.$$

By Remark 2.2, this superposition can be prepared in $\widetilde{O}(1)$ time if the amplitudes

$$\frac{1}{\sqrt{\sum_{\tau' \in \{0,1,2\}^\ell} \frac{1}{|E_{\tau'}|}}} \times \frac{1}{\sqrt{|E_\tau|}}$$

and partial sums

$$S(p) = \sum_{\tau : \tau \text{ has prefix } p} \frac{1}{|E_\tau|}$$

can be computed in $\widetilde{O}(1)$ time. We find closed-form expressions for these quantities, which shows that they can indeed be evaluated efficiently. We start by applying Lemma 4.16 and Lemma 4.17 to compute the amplitudes:

$$\frac{1}{\sqrt{\sum_{\tau' \in \{0,1,2\}^\ell} \frac{1}{|E_{\tau'}|}}} \frac{1}{\sqrt{|E_\tau|}} = \left(\frac{(n+2)^\ell}{n^{\ell+1}} \right)^{-1/2} \left(n^{1+\ell-|\tau|_0} \right)^{-1/2} = \sqrt{\frac{n^{|\tau|_0}}{(n+2)^\ell}}.$$

25

Finally, we compute the partial sums, again using [Lemma 4.16] and [Lemma 4.17]. Let $p \in \{0,1,2\}^k$ for $k \in [\ell]$ be a fixed prefix. Then

$$
\begin{aligned}
S(p) &= \sum_{\tau:\tau \text{ has prefix } p} \frac{1}{|E_\tau|} = \sum_{\tau' \in \{0,1,2\}^{\ell-k}} \frac{1}{|E_{p\tau'}|} = \sum_{\tau' \in \{0,1,2\}^{\ell-k}} \frac{1}{n^{1+\ell-(|p|_0+|\tau'|_0)}} \\
&= \frac{1}{n^{k-|p|_0}} \sum_{\tau' \in \{0,1,2\}^{\ell-k}} \frac{1}{n^{1+(\ell-k)-|\tau'|_0}} = \frac{1}{n^{k-|p|_0}} \sum_{\tau' \in \{0,1,2\}^{\ell-k}} \frac{1}{|E_{\tau'}|} \\
&= \frac{1}{n^{k-|p|_0+1}} \left(1 + \frac{2}{n}\right)^{\ell-k} = \frac{(n+2)^{\ell-k}}{n^{\ell+1-|p|_0}}.
\end{aligned}
$$

Step 2. Map each $|\tau\rangle$ to the uniform superposition over the edges in the layer E_τ. Since these edges are encoded as $|\sigma, i\rangle$ with $\sigma \in \Sigma^\ell$, $i \in \{0,1\}^{\log n}$, and $\bar{\sigma} = \tau$, this mapping can be performed in $\widetilde{O}(1)$ time, as follows. For each $k \in [\ell]$, if $\tau_k \in \{1,2\}$, generate a uniform superposition over $j \in \{0,1\}^{\log n}$, to get a superposition over σ_k such that $\bar{\sigma}_k = \tau_k$. Finally, generate a uniform superposition over $i \in \{0,1\}^{\log n}$. Letting $S(\tau_k) = \{\bar{0}\}$ if $\tau_k = 0$ and $\{0,1\}^{\log n}$ otherwise, this mapping acts as:

$$
\begin{aligned}
|\tau\rangle = |\tau_1\rangle \otimes \cdots \otimes |\tau_\ell\rangle &\mapsto \left(\bigotimes_{k=1}^{\ell} |\tau_k\rangle \sum_{j \in S(\tau_k)} \frac{1}{\sqrt{|S(\tau_k)|}} |j\rangle\right) \otimes \sum_{i \in \{0,1\}^{\log n}} \frac{1}{\sqrt{n}} |i\rangle \\
&= \frac{1}{\sqrt{n^{1+\ell-|\tau|_0}}} \sum_{e \in E_\tau} |e\rangle = \frac{1}{\sqrt{|E_\tau|}} \sum_{e \in E_\tau} |e\rangle.
\end{aligned}
$$

We thus end up with a state proportional to

$$
\sum_{\tau \in \{0,1,2\}^\ell} \frac{1}{\sqrt{|E_\tau|}} \sum_{e \in E_\tau} \frac{1}{\sqrt{|E_\tau|}} |e\rangle = \sum_{\tau \in \{0,1,2\}^\ell} \frac{1}{|E_\tau|} \sum_{e \in E_\tau} |e\rangle, \tag{15}
$$

which is proportional to $\sum_{j \in \{0,1\}^{\log n}} |\bar{\theta}_j(2^\ell)\rangle$ by (14). $\qquad \square$

Building on the previous lemma, we describe how to generate states that will shortly be shown to make up part of the states $|\psi_{z,x}(2^\ell)\rangle$ from the basis in [Theorem 4.12].

Lemma 4.19. *For all $\ell > 0$, there is a circuit C_{2^ℓ} that acts, for all $x \in \{0,1\}^{\log n}$, as*

$$
|x\rangle \mapsto \propto \sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |\bar{\theta}_j(2^\ell)\rangle
$$

and uses $\widetilde{O}(1)$ gates.

Proof. We start by noticing that if $x = \bar{0}$ then the desired state is $\sum_{j \in \{0,1\}^{\log n}} |\bar{\theta}_j(2^\ell)\rangle$. By [Lemma 4.18], a state proportional to this one can be prepared in $\widetilde{O}(1)$. Hence, we assume that $x \neq \bar{0}$ for the rest of the proof. We prove the statement by induction. Consider $\mathcal{N}_2$ for the base case. For a fixed $x \neq \bar{0}$, we can write the following.

$$
\begin{aligned}
\sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |\bar{\theta}_j(2)\rangle &= \sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} \sum_{i \in \{0,1\}^{\log n}} \left(|e_i^0\rangle + |e_j^{1i}\rangle - |e_i^{2j}\rangle\right) \\
&= \underbrace{\sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} \sum_{i \in \{0,1\}^{\log n}} |e_i^0\rangle}_{=0} + \sum_{i,j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |e_j^{1i}\rangle - \sum_{i,j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |e_i^{2j}\rangle \\
&= \sum_{i,j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |1, i\rangle |j\rangle + \sum_{i,j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |2, j\rangle |i\rangle \\
&\propto |1\rangle H^{\otimes \log n} |\bar{0}\rangle H^{\otimes \log n} |x\rangle + |2\rangle H^{\otimes \log n} |x\rangle H^{\otimes \log n} |\bar{0}\rangle.
\end{aligned}
\tag{16}
$$

26

We used the fact that edges are encoded as $|e_j^{1i}\rangle = |1, i\rangle|j\rangle$ and $|e_i^{2j}\rangle = -|2, j\rangle|i\rangle$ (see (9)). Thus, to prepare this state from $|x\rangle$, first make a uniform superposition over $|1\rangle$ and $|2\rangle$, and then swap the second and third register controlled on the first register:

$$|0, \bar{0}\rangle|x\rangle \mapsto \frac{1}{\sqrt{2}}|1, \bar{0}\rangle|x\rangle + \frac{1}{\sqrt{2}}|2, \bar{0}\rangle|x\rangle \mapsto \frac{1}{\sqrt{2}}|1\rangle|\bar{0}\rangle|x\rangle + \frac{1}{\sqrt{2}}|2\rangle|x\rangle|\bar{0}\rangle.$$

We can complete the computation by applying Hadamards to all but the first register. This can be done in $O(\log n)$ gates.

Assume, for the induction step, that there is a circuit $C_{2^{\ell-1}}$ with $\widetilde{O}(1)$ gates that implements

$$|x\rangle \mapsto \frac{\sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |\bar{\theta}_j(2^{\ell-1})\rangle}{\left\| \sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |\bar{\theta}_j(2^{\ell-1})\rangle \right\|}$$

for any x. We rewrite the desired state analogously to the base case (all sums below are over $\{0, 1\}^{\log n}$).

$$\sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |\bar{\theta}_j(2^\ell)\rangle = \sum_j (-1)^{x \cdot j} \sum_i \left(|\bar{\theta}_i^0(2^{\ell-1})\rangle + |\bar{\theta}_j^{1i}(2^{\ell-1})\rangle - |\bar{\theta}_i^{2j}(2^{\ell-1})\rangle \right)$$

$$= \underbrace{\sum_j (-1)^{x \cdot j} \sum_i |\bar{\theta}_i^0(2^{\ell-1})\rangle}_{=0} + \sum_j \sum_i (-1)^{x \cdot j} |\bar{\theta}_j^{1i}(2^{\ell-1})\rangle - \sum_j \sum_i (-1)^{x \cdot j} |\bar{\theta}_i^{2j}(2^{\ell-1})\rangle$$

$$= \sum_j \sum_i (-1)^{x \cdot j} |1, i\rangle |\bar{\theta}_j(2^{\ell-1})\rangle + \sum_j \sum_i (-1)^{x \cdot j} |2, j\rangle |\bar{\theta}_i(2^{\ell-1})\rangle.$$

We used $|\bar{\theta}_j^{1i}(2^{\ell-1})\rangle = |1, i\rangle|\bar{\theta}_j(2^{\ell-1})\rangle$ and $|\bar{\theta}_i^{2j}(2^{\ell-1})\rangle = |2, j\rangle|\bar{\theta}_i(2^{\ell-1})\rangle$, by (9). Continuing, we use the induction hypothesis to compute (again, all sums are over $\{0, 1\}^{\log n}$):

$$\sum_{j \in \{0,1\}^{\log n}}^n (-1)^{x \cdot j} |\bar{\theta}_j(2^\ell)\rangle = \sum_i |1, i\rangle \otimes \sum_j (-1)^{x \cdot j} |\bar{\theta}_j(2^{\ell-1})\rangle + \sum_j (-1)^{x \cdot j} |2, j\rangle \otimes \sum_i |\bar{\theta}_i(2^{\ell-1})\rangle$$

$$= |1\rangle \otimes \sqrt{n} H^{\otimes \log n} |\bar{0}\rangle \otimes \left\| \sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |\bar{\theta}_j(2^{\ell-1})\rangle \right\| C_{2^{\ell-1}} |x\rangle \qquad (17)$$

$$+ |2\rangle \otimes \sqrt{n} H^{\otimes \log n} |x\rangle \otimes \left\| \sum_{j \in \{0,1\}^{\log n}} |\bar{\theta}_j(2^{\ell-1})\rangle \right\| C_{2^{\ell-1}} |\bar{0}\rangle.$$

Thus, as in the base case, we first put appropriate weight on $|1\rangle$ and $|2\rangle$, using a simple rotation:

$$|0, \bar{0}\rangle|x\rangle \mapsto \left(\sqrt{\alpha}|1\rangle + \sqrt{1 - \alpha}|2\rangle \right) |\bar{0}\rangle|x\rangle,$$

where

$$\alpha = \frac{\left\| \sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |\bar{\theta}_j(2^{\ell-1})\rangle \right\|^2}{\left\| \sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |\bar{\theta}_j(2^{\ell-1})\rangle \right\|^2 + \left\| \sum_{i \in \{0,1\}^{\log n}} |\bar{\theta}_i(2^{\ell-1})\rangle \right\|^2}.$$

We will describe how to do this first rotation shortly. As in the base step, we next swap the second and third registered controlled on the first, and then complete the computation by applying $H^{\otimes \log n}$ to the second register and $C_{2^{\ell-1}}$ to the third register.

We complete the proof by describing how to do the first rotation in terms of α. By Lemma 2.1, such a superposition can be prepared in $\widetilde{O}(1)$ time if α can be computed in $\widetilde{O}(1)$ time. Hence, it suffices to compute closed-form expressions for the two norms in the definition of α. From (14) and Lemma 4.17, we can conclude that for any $\ell > 0$:

$$N_{\bar{0}}(2^\ell) := \left\| \sum_{i \in \{0,1\}^{\log n}} |\bar{\theta}_i(2^\ell)\rangle \right\|^2 = n^2 \sum_{\tau \in \{0,1,2\}^\ell} \frac{1}{|E_\tau|} = n^2 \frac{(n+2)^\ell}{n^{\ell+1}}. \qquad (18)$$

Letting $N_x(2^\ell) := \left\| \sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |\bar\theta_j(2^\ell)\rangle \right\|^2$, we show by induction that for $\ell \geq 1$:

$$N_x(2^\ell) = n^{\ell+1} + \frac{n^3}{(n-2)(n+1)}\left(n^\ell - \frac{(n+2)^\ell}{n^\ell}\right). \tag{19}$$

For the base step, we can observe from (16) that $N_x(2) = 2n^2$, which is equal to the right-hand-side of (19) when $\ell = 1$. For the induction step, we can use (17) to show that for $\ell > 1$:

$$
\begin{aligned}
N_x(2^\ell) &= \left\| \sqrt{n} H^{\otimes \log n} |\bar 0\rangle \otimes \sqrt{N_x(2^{\ell-1})} C_{2^{\ell-1}}|x\rangle \right\|^2 + \left\| \sqrt{n} H^{\otimes \log n} |x\rangle \otimes \sqrt{N_{\bar 0}(2^{\ell-1})} C_{2^{\ell-1}}|\bar 0\rangle \right\|^2 \\
&= n N_x(2^{\ell-1}) + n N_{\bar 0}(2^{\ell-1}) \\
&= n\left(n^\ell + \frac{n^3}{(n-2)(n+1)}\left(n^{\ell-1} - \frac{(n+2)^{\ell-1}}{n^{\ell-1}}\right)\right) + n^2 \frac{(n+2)^{\ell-1}}{n^{\ell-1}},
\end{aligned}
$$

by induction. This is easily shown to be equal to the right-hand-side of (19). $\qquad\square$

We now describe how to generate the states $|\psi_{z,x}(2^\ell)\rangle$, which form most of the basis described in Theorem 4.12.

Lemma 4.20. *For all $\ell \in [\log(L)]$, a map that acts, for all $x, z \in [n]$ such that $z \neq \bar 0$, as*

$$|x, z\rangle \mapsto \propto |\psi_{z,x}(2^\ell)\rangle$$

can be implemented in $\widetilde{O}(1)$ steps.

Proof. For a fixed $z, x \in [n], z \neq \bar 0$, we write down the state $|\psi_{z,x}\rangle$ by definition its definition (see Theorem 4.12) and split it into three orthogonal terms. (All sums below are over $\{0,1\}^{\log n}$.)

$$
\begin{aligned}
|\psi_{z,x}(2^\ell)\rangle &= \sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} \sum_{i \in \{0,1\}^{\log n}} (-1)^{z \cdot i} |p_{ij}(2^\ell)\rangle \\
&= \sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} \sum_{i \in \{0,1\}^{\log n}} (-1)^{z \cdot i}\left(|\bar\theta_i^0(2^{\ell-1})\rangle + |\bar\theta_j^{1i}(2^{\ell-1})\rangle - |\bar\theta_i^{2j}(2^{\ell-1})\rangle\right) \\
&= \sum_j (-1)^{x \cdot j} \sum_i (-1)^{z \cdot i} |\bar\theta_i^0(2^{\ell-1})\rangle + \sum_{i,j} (-1)^{z \cdot i + x \cdot j} |\bar\theta_j^{1i}(2^{\ell-1})\rangle - \sum_{i,j} (-1)^{z \cdot i + x \cdot j} |\bar\theta_i^{2j}(2^{\ell-1})\rangle.
\end{aligned}
$$

By (10), we can write $|\bar\theta_i^0\rangle = |0, \bar 0\rangle|\bar\theta_i\rangle$, $|\bar\theta_j^{1i}\rangle = |1, i\rangle|\bar\theta_j\rangle$ and $|\bar\theta_i^{2j}\rangle = -|2, j\rangle|\bar\theta_i\rangle$. If $x \neq \bar 0$, the first sum vanishes, otherwise it adds up to n, giving:

$$
\begin{aligned}
|\psi_{z,x}(2^\ell)\rangle &= \delta_{x,\bar 0} n \cdot |0, \bar 0\rangle \sum_{i \in \{0,1\}^{\log n}} (-1)^{z \cdot i} |\bar\theta_i(2^{\ell-1})\rangle + \sum_{i \in \{0,1\}^{\log n}} (-1)^{z \cdot i} |1, i\rangle \sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |\bar\theta_j(2^{\ell-1})\rangle \\
&\quad + \sum_{j \in \{0,1\}^{\log n}} (-1)^{x \cdot j} |2, j\rangle \sum_{i \in \{0,1\}^{\log n}} (-1)^{z \cdot i} |\bar\theta_i(2^{\ell-1})\rangle \\
&= \delta_{x,\bar 0} n |0, \bar 0\rangle \otimes \sqrt{N_z(2^{\ell-1})} C_{2^{\ell-1}}|z\rangle \\
&\quad + |1\rangle H^{\otimes \log n}|z\rangle \otimes \sqrt{N_x(2^{\ell-1})} C_{2^{\ell-1}}|x\rangle + |2\rangle H^{\otimes \log n}|x\rangle \otimes \sqrt{N_z(2^{\ell-1})} C_{2^{\ell-1}}|z\rangle
\end{aligned}
$$

where $C_{2^{\ell-1}}$ is the circuit from Lemma 4.19, and $N_z(2^{\ell-1})$ is as in (19). Note by (19) that, for non-zero z, this is independent of z, so in that case, we simply write $N(2^{\ell-1})$.

To prepare the superposition of two or three terms, we begin by creating a superposition of $|0\rangle$, $|1\rangle$ and $|2\rangle$ with appropriate amplitudes:

$$
|0\rangle|\bar 0\rangle|x\rangle|z\rangle \mapsto
\begin{cases}
\frac{n\sqrt{N(2^{\ell-1})}|0\rangle + \sqrt{N_{\bar 0}(2^{\ell-1})}|1\rangle + \sqrt{N(2^{\ell-1})}|2\rangle}{\sqrt{(n^2+1)N(2^{\ell-1}) + N_{\bar 0}(2^{\ell-1})}} |\bar 0\rangle|x\rangle|z\rangle & \text{if } x = \bar 0 \\
\left(\frac{1}{\sqrt 2}|1\rangle + \frac{1}{\sqrt 2}|2\rangle\right)|\bar 0\rangle|x\rangle|z\rangle & \text{else.}
\end{cases}
$$

We have used the fact that when $x \neq \bar{0}$, $N_x(2^{\ell-1}) = N_z(2^{\ell-1})$, so the amplitudes on $|1\rangle$ and $|2\rangle$ are equal. Using the closed-form expressions in (18) and (19), we can compute these amplitudes in $\widetilde{O}(1)$ steps, so we can implement this rotation in this number of steps as well.

Next, conditioned on $|1\rangle$ in the first register, we swap the second and third registers. Then we apply a Hadamard to the second register conditioned on $|1\rangle$ or $|2\rangle$ in the first register, and apply $C_{2^{\ell-1}}$ to the last register. $\qquad \square$

Finally, if we take $\ell = \log L$, and $v_j = t$, in the following lemma, this gives a procedure for preparing a state proportional to the optimal unit flow $|\theta_j(L)\rangle$, which is the final state needed for the basis in Theorem 4.12.

Lemma 4.21. *For all $\ell \in [\log(L)]$, a map that acts, for all $j \in \{0,1\}^{\log n}$, as*

$$|j\rangle \mapsto \propto |\bar{\theta}_j(2^\ell)\rangle = \frac{1}{n} \sum_{i \in \{0,1\}^{\log n}} |p_{ij}(2^\ell)\rangle$$

can be implemented in $\widetilde{O}(1)$ steps.

Proof. We prove the statement by induction. Let $\ell = 1$ for the base case.

$$
\begin{aligned}
|\bar{\theta}_j(2)\rangle &= \frac{1}{n} \sum_{i \in \{0,1\}^{\log n}} \left(|e_i^0\rangle + |e_j^{1i}\rangle - |e_i^{2j}\rangle \right) \\
&= \frac{1}{n} \sum_{i \in \{0,1\}^{\log n}} |(0,\bar{0}),i\rangle + \frac{1}{n} \sum_{i \in \{0,1\}^{\log n}} |(1,i),j\rangle + \frac{1}{n} \sum_{i \in \{0,1\}^{\log n}} |(2,j),i\rangle.
\end{aligned}
\tag{20}
$$

Clearly, a state proportional to this can be prepared in $\widetilde{O}(1)$ time.

For the induction step, assume there is an efficient circuit $C'_{2^{\ell-1}}$ that implements the map

$$|j\rangle \mapsto \frac{|\bar{\theta}_j(2^{\ell-1})\rangle}{\||\bar{\theta}_j(2^{\ell-1})\rangle\|}$$

for any $j \in \{0,1\}^{\log n}$. From Lemma 4.15 (see also (3)) and the definition of $|p_{ij}\rangle$ in Theorem 4.12, we have:

$$
\begin{aligned}
n|\bar{\theta}_j(2^\ell)\rangle &= \sum_{i \in \{0,1\}^{\log n}} |p_{ij}(2^\ell)\rangle \\
&= |0,\bar{0}\rangle \sum_{i \in \{0,1\}^{\log n}} |\bar{\theta}_i(2^{\ell-1})\rangle + \sum_{i \in \{0,1\}^{\log n}} |1,i\rangle |\bar{\theta}_j(2^{\ell-1})\rangle + |2,j\rangle \sum_{i \in \{0,1\}^{\log n}} |\bar{\theta}_i(2^{\ell-1})\rangle \\
&= |0,\bar{0}\rangle \otimes \sqrt{N_{\bar{0}}(2^{\ell-1})} C_{2^{\ell-1}} |\bar{0}\rangle + \sqrt{n}|1\rangle H^{\otimes \log n} |\bar{0}\rangle \otimes \left\||\bar{\theta}_j(2^{\ell-1})\rangle\right\| C'_{2^{\ell-1}} |j\rangle + |2,j\rangle \otimes \sqrt{N_{\bar{0}}(2^{\ell-1})} C_{2^{\ell-1}} |\bar{0}\rangle
\end{aligned}
\tag{21}
$$

where $C_{2^{\ell-1}}$ is the circuit from Lemma 4.19, and $N_{\bar{0}}(2^{\ell-1})$ is as in (18).

To prepare a state proportional to this, we first generate the superposition

$$|j\rangle \mapsto \frac{\sqrt{N_{\bar{0}}(2^{\ell-1})}|0\rangle + \sqrt{n} \cdot \left\||\bar{\theta}_j(2^{\ell-1})\rangle\right\| |1\rangle + \sqrt{N_{\bar{0}}(2^{\ell-1})}|2\rangle}{\sqrt{n \cdot \left\||\bar{\theta}_j(2^{\ell-1})\rangle\right\|^2 + 2 \cdot N_{\bar{0}}(2^{\ell-1})}} |j\rangle,$$

before mapping $|j\rangle$ to the correct state, controlled on $|0\rangle$, $|1\rangle$, or $|2\rangle$, using swap, $C_{2^{\ell-1}}$, and $H^{\otimes \log n}$. By Lemma 2.1, such a superposition can be prepared in $\widetilde{O}(1)$ time, provided that the amplitudes are computable in $\widetilde{O}(1)$ time. As we already have an efficiently computable closed form expression for $N_{\bar{0}}$ (see (18)), it suffices to obtain a closed-form expression for $\left\||\bar{\theta}_j(2^{\ell-1})\rangle\right\|$. For $\ell > 0$, define $F_j(2^\ell) := \left\||\bar{\theta}_j(2^\ell)\rangle\right\|^2$. We will prove by induction that

$$F_j(2^\ell) = \frac{1}{n^\ell} \cdot \frac{2(n+2)^\ell + n - 1}{n+1}.$$
\tag{22}

For the base step, let $\ell = 1$. By (20), we have $F_j(2) = 3/n$, which is equal to the right-hand side of (22) for the setting $\ell = 1$. Next, we show the induction step.

$$
\begin{aligned}
F_j(2^\ell) &= \frac{1}{n^2}\left(2N_{\bar{0}}(2^{\ell-1}) + nF_j(2^{\ell-1})\right) && \text{by (21)} \\
&= \frac{2}{n}\left(1 + \frac{2}{n}\right)^{\ell-1} + \frac{1}{n}F_j(2^{\ell-1}) && \text{by (18)} \\
&= \frac{2}{n^\ell}(n+2)^{\ell-1} + \frac{1}{n^\ell}\cdot\frac{2(n+2)^{\ell-1}+n-1}{n+1} && \text{by i.h.} \\
&= \frac{1}{n^\ell}\cdot\frac{2(n+1)(n+2)^{\ell-1}+2(n+2)^{\ell-1}+n-1}{n+1} \\
&= \frac{1}{n^\ell}\cdot\frac{2(n+2)(n+2)^{\ell-1}+n-1}{n+1} = \frac{1}{n^\ell}\cdot\frac{2(n+2)^{\ell}+n-1}{n+1}. && \square
\end{aligned}
$$

Corollary 4.22. *For all $\ell \in [\log(L)]$, the map that prepares a state proportional to*

$$
|\theta_j(2^\ell)\rangle = -|\leftarrow, [u]\rangle + \frac{1}{n}\sum_{i\in\{0,1\}^{\log n}}|p_{ij}(2^\ell)\rangle + |\rightarrow, [u, v_j]\rangle
$$

can be implemented in $\widetilde{O}(1)$ steps.

We summarize the results of the two sections in the following lemma, which states that the orthonormal basis of $\mathcal{B}^\perp$ from Theorem 4.12 can be efficiently generated.

Lemma 4.23. *There exists an orthonormal basis of $\mathcal{B}^\perp(\mathcal{N}_L(s,t))$ that can be generated in time $\widetilde{O}(1)$.*

4.4 Complexity of the subroutine

We conclude by combining the switching network and corresponding basis generation we have given in the previous sections with Theorem 2.9 to prove Theorem 3.1, which states the existence of an algorithm for DIST_L.

Proof of Theorem 3.1. From Lemma 4.3, it follows that the switching network $\mathcal{N}_L(s,t)$ accepts G if and only if there is a path of length at most L from s to t in G. We apply Theorem 2.9 to $\mathcal{N}_L(s,t)$. By Lemma 4.10, whenever there is a st-path of length $\leq L$ in G, there is a path from the source $[s]$ to the sink $[s,t]$ in $\mathcal{N}_L(s,t)(G)$ of length at most $W_+ := L^{\log 3}$. By Lemma 4.11, $|E(\mathcal{N}_L(s,t))| \leq (2n+1)^{\log L}n$. By Lemma 4.23 and Lemma 2.10, there exists an orthonormal basis of $\mathcal{B}^\perp(\mathcal{N}_L(s,t))$ that can be generated in time $T_B = \widetilde{O}(1)$. Hence, by Theorem 2.9, there exists a quantum algorithm that decides whether there is a path of length at most L from s to t in G in time

$$
O(T_B\sqrt{|E(\mathcal{N}_L(u,v))|W_+}) = \widetilde{O}\left(\left(L^{\log 3}(2n+1)^{\log L}n\right)^{1/2}\right)
$$

and space

$$
O(\log|E(\mathcal{N}_L(u,v))|) = O(\log L \log n). \qquad\qquad \square
$$

Acknowledgments

References

[AE25]   Simon Apers and Roman Edenhofer. Directed st-Connectivity with Few Paths Is in Quantum Logspace. In Srikanth Srinivasan, editor, *40th Computational Complexity Conference (CCC 2025)*, volume 339 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 18:1–18:15, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 4

[AHV95]   Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases.* Addison-Wesley, 1995. 1

[AJPW23]   Simon Apers, Stacey Jeffery, Galina Pass, and Michael Walter. (No) Quantum Space-Time Tradeoff for USTCON. In Inge Li Gørtz, Martin Farach-Colton, Simon J. Puglisi, and Grzegorz Herman, editors, *31st Annual European Symposium on Algorithms (ESA 2023)*, volume 274 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:17, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 2, 3

[ALSU06]   Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools.* Pearson, 2nd edition, 2006. 1

[BBRS98]   Greg Barnes, Jonathan F Buss, Walter L Ruzzo, and Baruch Schieber. A sublinear space, polynomial time algorithm for directed st connectivity. *SIAM Journal on Computing*, 27(5):1273–1282, 1998. 1, 2, 3, 11, 12, 13

[BJ25]   Aleksandrs Belovs and Stacey Jeffery. Space-efficient quantum error reduction without log factors, 2025. arXiv: 2502.09249   2

[BJY24]   Aleksandrs Belovs, Stacey Jeffery, and Duyal Yolcu. Taming quantum time complexity. *Quantum*, 8(1444), 2024. arXiv: 2311.15873 2

[BK08]   Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking.* MIT Press, 2008. 1

[BR12]   Aleksandrs Belovs and Ben W. Reichardt. Span programs and quantum algorithms for *st*-connectivity and claw detection. In *Proceedings of the 20th Annual European Symposium on Algorithms (ESA)*, pages 193–204, 2012. 2, 3

[CGP99]   Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking.* MIT Press, 1999. 1

[Cor23]   Arjan Cornelissen. *Quantum multivariate estimation and span program algorithms.* PhD thesis, University of Amsterdam, 2023. 8

[DHHM06]   Christoph Dürr, Mark Heiligman, Peter Høyer, and Mehdi Mhalla. Quantum query complexity of some graph problems. *SIAM Journal on Computing*, 35(6):1310–1328, 2006. Earlier version in ICALP'04. arXiv: quant-ph/0401091 2

[GR02]   Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002. 4

[JJKP18]   Michael Jarret, Stacey Jeffery, Shelby Kimmel, and Alvaro Piedrafita. Quantum algorithms for connectivity and related problems. In *Proceedings of the 26th Annual European Symposium on Algorithms (ESA)*, pages 49:1–49:13, 2018. 3, 7

[JK17]   Stacey Jeffery and Shelby Kimmel. Quantum algorithms for graph connectivity and formula evaluation. *Quantum*, (26), 2017. 3, 5, 6

[JP25]   Stacey Jeffery and Galina Pass. Multidimensional Quantum Walks, Recursion, and Quantum Divide & Conquer. In *42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025)*, volume 327 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 54:1–54:16, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 2, 3, 5, 6, 7

[LV96]   Ming Li and Paul Vitányi. Reversibility and adiabatic computation: trading time and space for energy. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, 452(1947):769–789, 1996. 17

[NNH99]   Flemming Nielson, Hanne R. Nielson, and Chris Hankin. *Principles of Program Analysis*. Springer, 1999. 1

[Pot14]   Aaron Potechin. Bounds on monotone switching networks for directed connectivity. *Journal of the ACM*, 9(4), 2014. 14

[Pot15]   Aaron H Potechin. *Analyzing monotone space complexity via the switching network model*. PhD thesis, Massachusetts Institute of Technology, 2015. 3, 5, 17

[Ull89]   Jeffrey D. Ullman. *Principles of Database and Knowledge-Base Systems, Volumes 1–2*. Computer Science Press, 1988–1989. 1