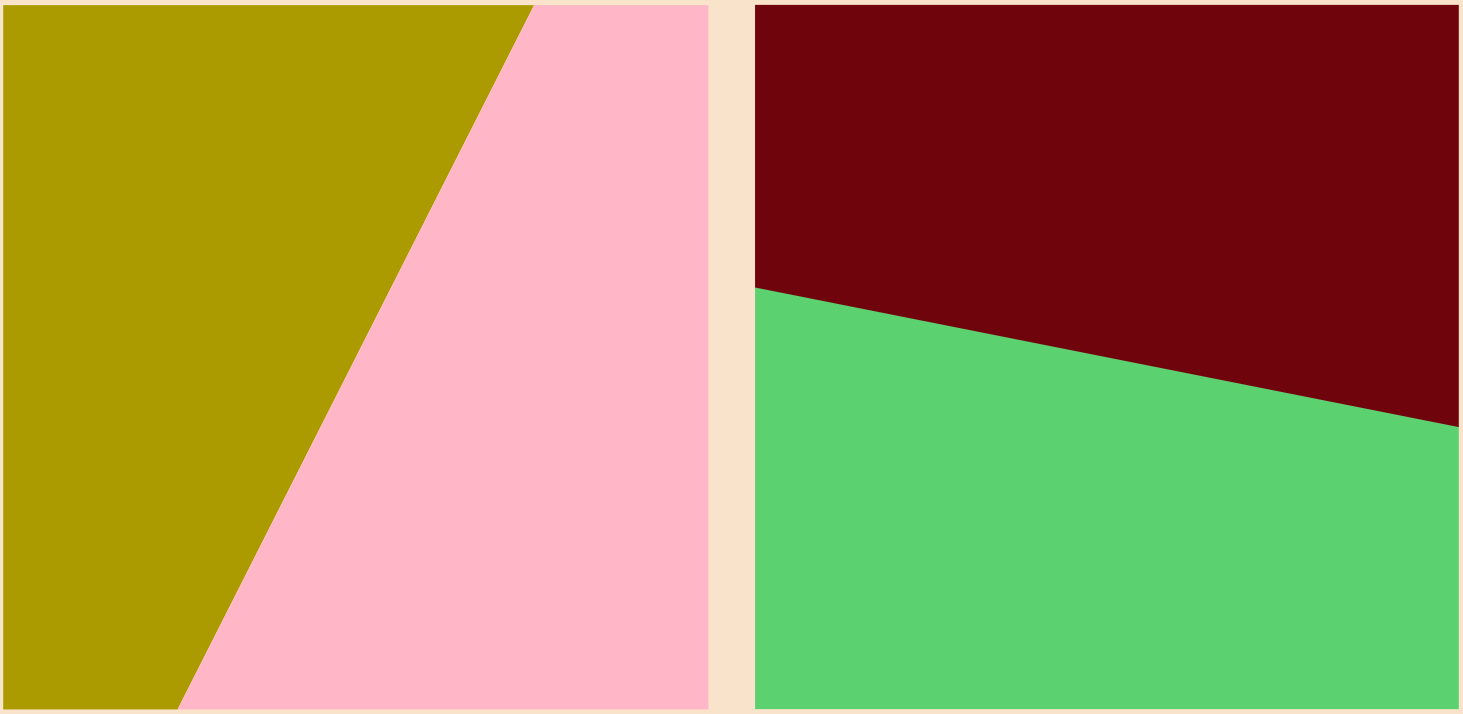

QUANTUM WALKS, RECURSION, AND TIME-SPACE TRADEOFFS FOR ST-CONNECTIVITY

Galina Pass

Quantum walks, recursion, and time-space tradeoffs for st-connectivity

Galina Pass

Quantum walks, recursion, and time-space tradeoffs for st-connectivity

UNIVERSITEIT VAN AMSTERDAM

QuSoft

Research Center for Quantum Software

The investigations were supported by the National Agenda for Quantum Technologies (NAQT), as part of the Quantum Delta NL programme.

Printed and bound by Ipskamp Printing.

ISBN: 978-94-6536-147-5

Quantum walks, recursion, and time-space tradeoffs for st-connectivity

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op woensdag 10 juni 2026, te 10.00 uur

door Galina Pass
geboren te St. Petersburg

Promotiecommissie

Promotores:	prof. dr. S.M. Jeffery	Universiteit van Amsterdam
	prof. dr. M. Walter	Ludwig-Maximilians-Universität München
Overige leden:	prof. dr. R.M. de Wolf	Universiteit van Amsterdam
	prof. dr. F. Magniez	Université Paris Cité
	prof. dr. J.A. Ellis-Monaghan	Universiteit van Amsterdam
	dr. K. Guo	Universiteit van Amsterdam
	dr. M. Ozols	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

In memory of my grandfather Matvey Pass
who inspired me to become a researcher

The results in this thesis are based on the following articles. For all articles, the authors are ordered alphabetically and co-authorship is shared equally.

1. [AJPW23] Simon Apers, Stacey Jeffery, Galina Pass, and Michael Walter. (No) Quantum Space-Time Tradeoff for USTCON. In *31st Annual European Symposium on Algorithms (ESA 2023)*, volume 274, 10:1-10:17, 2023.

2. [JP25a] 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, 54:1-54:16, 2025.

3. [JP25b] Stacey Jeffery and Galina Pass. A quantum time-space tradeoff for directed *st*-connectivity. *arXiv preprint arXiv:510.08403*, 2025.
   Presented at QIP 2026.

4. [JOP] Stacey Jeffery, Tobias J. Osborne, and Galina Pass. A quantum algorithm for the gauge problem.
   In preparation.

Contents

Chapter 1

Introduction

This thesis studies *space-bounded quantum computation*, with a focus on quantum algorithms for graph connectivity problems and the time-space tradeoffs they admit.

The first theme is *quantum time-space tradeoffs for st-connectivity*. Given a graph with distinguished vertices s and t, the st-connectivity problem asks whether there exists a path from s to t. We study this problem in undirected and directed graphs, as well as in a generalized undirected setting where edges carry additional algebraic structure. Connectivity problems are central benchmarks in space complexity theory, and they provide a natural arena for understanding which quantum speedups persist when quantum memory is severely limited.

The second theme is the development of a compositional framework for quantum algorithms based on *multidimensional quantum walks*. We introduce an object called a *subspace graph*, which formalizes the idea of treating a quantum subroutine as a "graph-like component" with a small interface. Subspace graphs can be composed recursively, allowing one to build complex quantum procedures from simpler pieces in a way that resembles classical abstraction. This framework is developed as an independent set of results, and it also serves as a unifying toolkit used throughout the thesis, most notably in our algorithms for directed st-connectivity and for a generalized st-connectivity problem on unitary-labeled graphs.

1.1 Undirected st-connectivity

For a graph $G = (V, E)$, $|V| = n, |E| = m$, with distinguished vertices $s, t \in V$, *st-connectivity* is the problem of deciding whether there exists a path from s to t. In an undirected graph, this is the problem USTCON of deciding whether s and t lie in the same connected component. Connectivity problems appear throughout graph algorithms and network problems, and they also play a central role in space complexity (see, e.g., [Wig92]).

In particular, USTCON is complete for the class SL (symmetric logspace), and Reingold's celebrated result SL = L was obtained by exhibiting a deterministic logspace algorithm for USTCON [Rei08].

A natural way to understand space-bounded computation is through *time-space tradeoffs*. Classically, the standard approach to *st*-connectivity is breadth-first search (BFS), which solves USTCON in polynomial time but uses $\widetilde{O}(n)$ space. For undirected *st*-connectivity, random-walk-based algorithms yield a dramatic space improvement, giving an $O(\log n)$-space algorithm for USTCON [AKL+79].

For USTCON in the adjacency-array model of access to the graph, a long line of work [BKRU89, BBR+90, Edm93, BF93, Fei93] culminated in a time-space tradeoff of the form

$$T = \widetilde{O}(n^2/S)$$

for any space bound $S = \Omega(\log(n))$ and $S = O(n^2/m)$, due to Kosowski [Kos13]. While there is no matching time-space lower bound, it is unlikely that this tradeoff can be significantly improved (see [Kos13, Section 5.1 of arXiv v2] for a discussion).

For comparison, in the adjacency-matrix model the randomized complexity of USTCON is $\widetilde{\Theta}(n^2)$, and in this setting there is no nontrivial time-space tradeoff.

In the quantum setting, the algorithm of Dürr, Heiligman, Høyer and Mhalla [DHHM06] yields $\widetilde{O}(n^{3/2})$ time for USTCON in the adjacency matrix model and $\widetilde{O}(n)$ time for USTCON in the adjacency array model, using $\widetilde{O}(n)$ workspace (of which only $O(\log n)$ must be quantum, and the remainder is classical memory with quantum random access (QCRAM)).

Belovs and Reichardt later achieved the same $\widetilde{O}(n^{3/2})$ time for USTCON in the adjacency matrix model using only $O(\log n)$ space [BR12]. Moreover, a quantum-walk approach solves USTCON in the adjacency array model in $O(\log n)$ space and $\widetilde{O}(\sqrt{nm})$ time [Bel13], assuming quantum access to adjacency arrays.

Results of this thesis

We develop new quantum walk algorithms for USTCON, based on a quantum analogue of the adjacency array access model (Chapter 3). We give a one-sided error quantum algorithm solving USTCON in $\widetilde{O}(n)$ time using only $O(\log n)$ space. This implies that, unlike in the classical setting, USTCON admits essentially no meaningful quantum time-space tradeoff: one algorithm achieves optimal time and space simultaneously.

Motivated by the connection between USTCON and classical logspace, we then ask when nontrivial quantum time-space tradeoffs can arise. We study the case where the input graph satisfies a promise on the random walk spectral gap (equivalently, good mixing properties). Assuming that the spectral gap is at least some $\delta > 0$ we

find a bounded-error quantum algorithm with a tradeoff of the form

$$T = \widetilde{O}\Big(\frac{S}{\delta} + \sqrt{\frac{n}{\delta S}}\Big)$$

for any space $S \in \Omega(\log n)$ (Chapter 3).

1.2 Directed *st*-connectivity

In a directed graph, *st*-connectivity is the problem DSTCON (reachability) of deciding whether there is a directed path from s to t. The problem DSTCON is NL-complete, where NL is nondeterministic logspace. Consequently, understanding the complexity of DSTCON has broad implications for space-bounded computation. For example, a classical or quantum algorithm for DSTCON using $O(\log n)$ space would imply $\mathsf{NL} \subseteq \mathsf{L}$ (or its quantum analogue), which would be a major breakthrough.

Classically, DSTCON can also be solved by the BFS algorithm in polynomial time and $\widetilde{O}(n)$ space. On the other extreme, Savitch's algorithm [Sav70] solves DSTCON in $O(\log^2 n)$ space but quasipolynomial time

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

A classical algorithm with a tradeoff was obtained by Barnes, Buss, Ruzzo and Schieber [BBRS98], that runs in time

$$T \leq 2^{\log^2(\frac{n}{S}) + O(\log n \log\log n)} \tag{1.2.1}$$

given any space $S \geq \log^2(n)$. It is based on combining BFS with a recursive subroutine.

A quantum speedup was only known in the large-space regime: one can achieve near-optimal polynomial time using large space [DHHM06]. Until recently, no quantum improvements over the best known classical tradeoff curve were known.

Moreover, directed *st*-connectivity introduces additional obstacles that are absent in the undirected case. Random walks and quantum walks are powerful tools for undirected graphs, but they do not directly generalize to the directed setting. The best known classical tradeoff algorithm for DSTCON is not straightforwardly quantizable because it relies on irreversible procedures and making it reversible by standard methods tends to increase the space complexity. Additionally, it is based on a recursive subroutine which cannot be turned into a quantum algorithm in a naive way. This motivates the search for algorithmic ideas that go beyond direct quantizations of known approaches.

Results of this thesis

We give the first nontrivial quantum time-space tradeoff for DSTCON (Chapter 6). Specifically, we design a quantum algorithm that for any space bound $S \geq \log^2 n$ decides DSTCON in time

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

This improves over the best known classical tradeoff in the regime $S = o(n^{1/2})$. Moreover, only $O(\log^2 n)$ of the space needs to be quantum (qubits); the remaining S memory can be classical, effectively trading classical space for quantum time.

The classical algorithm of Barnes et al. [BBRS98] combines a space-efficient BFS with a recursive subroutine for $\textsc{Dist}_L$ (testing whether there is a directed path between two vertices of length at most L). We design a new quantum algorithm for $\textsc{Dist}_L$ using a recursively constructed *switching network*. Switching networks are undirected and naturally reversible, and can be compiled into quantum algorithms with complexity controlled by structural network parameters [JK17, JJKP18, JP25a]. Crucially, recursion occurs at the level of switching networks, so bounded error is introduced only once at compilation time, avoiding $\log^d$ or c^d losses even for recursion depth d.

Finally, we keep the outer BFS classical, so most space remains classical: for any overall space bound S, the algorithm uses only $O(\log^2 n)$ quantum space, with the remainder classical memory.

1.3 The gauge problem

In this thesis, we also consider a generalization of graph connectivity in which edges carry genuinely quantum data. A *unitary-labeled graph* (also called a *connection graph*) consists of an undirected graph together with an assignment of a unitary matrix $U_{uv} \in \mathbb{C}^{k\times k}$ to each oriented edge (u, v), satisfying the consistency condition $U_{vu} = U_{uv}^\dagger$. Intuitively, these edge labels specify how an internal quantum state $|\psi\rangle \in \mathbb{C}^k$ is updated when it traverses the edge from u to v.

Graphs equipped with unitary-valued edge data have a long history. In physics, they arise in discrete gauge-theoretic models where edge variables encode parallel transport; this viewpoint goes back to Wilson's work and the development of lattice gauge theory [Wil74, Cre83]. More broadly, variants of unitary-labeled graphs appear across mathematics and theoretical computer science as a way of enriching classical graphs with local algebraic structure (see, e.g., [RKMC25] for a representative recent example and further references), and related constructions have also been studied from the viewpoint of quantum complexity theory (see, e.g., [BCO17]).

We study a problem suggested by this model that resembles *st*-connectivity, but with an additional layer of structure. Given two vertices s and t, the presence of unitary edge labels means that each *st*-path induces an ordered product of unitaries, and hence a transformation of the internal Hilbert space. In general this transformation may depend on the chosen path. In the promise setting studied in this thesis, we assume a *flatness* condition ensuring that the induced product depends only on the endpoints. As a result, for any vertices s, t in the same connected component there is a well-defined unitary $U_s(t)$ capturing the global transformation between them.

This allows one to define a connectivity-type promise problem that strictly extends undirected *st*-connectivity. In the special case $k = 1$ with trivial labels, $U_s(t)$ is necessarily the identity whenever s and t are connected, and the problem collapses to deciding whether s and t lie in the same connected component, i.e. to USTCON. Given boundary states $|\psi_s\rangle, |\psi_t\rangle \in \mathbb{C}^k$, the task is to decide whether $U_s(t)|\psi_s\rangle = |\psi_t\rangle$ or whether these states are orthogonal, using only local oracle access to the underlying graph and its edge labels. We refer to this promise decision problem as the *gauge problem*.

Results of this thesis

We show that this *gauge problem* can be solved by a bounded-error quantum algorithm running in $\widetilde{O}(n^{3/2})$ time using $O(\log n + \log k)$ space (Chapter 7). Our algorithm relies on a framework introduced later called *subspace graphs*. Informally, a subspace graph is a graph in which edges and vertices are associated with Hilbert subspaces. The gauge problem is naturally suited to the subspace-graph framework because it combines graph structure with a higher-dimensional internal Hilbert space attached to edges. We design our algorithm by equipping the instance with an appropriate subspace-graph structure and analyzing its complexity. This demonstrates that the subspace-graph viewpoint is not only a tool for recursion, but also a useful way to design and analyze quantum algorithms in generalized *st*-connectivity settings.

1.4 Visual quantum reasoning, composition, and recursion

As we will see in the directed *st*-connectivity setting, the main challenge is not only to obtain quantum speedups for individual steps, but to *compose* them efficiently. Standard quantum circuit reasoning tends to introduce worst-case overheads and repeated amplification costs, preventing naive recursive or modular constructions from yielding global improvements. This motivates a new compositional viewpoint,

developed later in the thesis via multidimensional quantum walks and applied to directed *st*-connectivity and to the gauge problem.

Classical algorithms are often understood through graphical models. A deterministic algorithm corresponds to a single computation *path*, while a randomized algorithm forms a computation *tree* (or branching program), where different branches represent choices based on the input and randomness. Subroutines can be treated abstractly as edges in the graph, hiding internal structure when convenient. More generally, randomized algorithms can be viewed as random walks on graphs, where multiple paths between two points represent alternative computations.

In the quantum setting, graph-based intuitions remain important, most notably via quantum walks, but incorporating subroutines compositionally is subtle in the standard circuit model. Multidimensional quantum walks [JZ23] were introduced as a way to restore a more classical "many-paths" perspective on quantum algorithms, and were used in [Jef22] to obtain improved subroutine composition bounds in several settings. Informally, this approach supports treating a quantum subroutine as a modular graph-like component with a small interface, enabling it to be composed into larger procedures while abstracting away internal structure.

A motivating setting is *divide & conquer*, where a problem is split into smaller instances, solved recursively and then combined. In the quantum setting, even if one has a quantum procedure that speeds up each subproblem, a naive recursive composition may fail to yield a global speedup. One reason is that recursion can cause constant overheads to compound exponentially with the recursion depth, eliminating any asymptotic advantage. Another is error accumulation: if subroutines succeed with constant probability, then composing them recursively typically drives the overall success probability down rapidly. Restoring correctness then requires amplification at each level of recursion, turning constant losses into logarithmic factors (or worse), which may again erase the speedup.

On the other hand, it is known that such recursive problems can sometimes still achieve optimal quantum query complexity, despite the failure of straightforward divide & conquer composition. For example, [CKKD+22] showed how to recover divide & conquer upper bounds in quantum query complexity by composing *dual adversary solutions*. The key feature enabling their results is *perfect* composition: no additional error, no amplification costs, and not even constant overhead. However, these results do not automatically yield time-efficient algorithms, since dual adversary solutions do not by themselves specify efficient implementations.

Results of this thesis

Subspace Graphs Although [JZ23, Jef22] rely on closely related ideas, they do not formally define what a *multidimensional quantum walk* is. We introduce *subspace graphs*, which abstract the common structure underlying these works and

other techniques (Chapter 4). A subspace graph is a graph whose vertices and edges are labeled with subspaces, with constraints on how these subspaces can overlap with respect to the graph structure. This provides a precise model for multidimensional quantum walks.

A key feature of subspace graphs is compositionality at different abstraction levels: one can "zoom out" and treat a complex subroutine as a single edge, or "zoom in" and reveal its internal structure. While quantum algorithms cannot be explained purely through classical intuition, subspace graphs offer a framework that blends classical structure with quantum reasoning.

We focus on one type of composition, *switch composition*, and use it to obtain a time-efficient version of one of the divide & conquer results from [CKKD+22] (Chapter 5).

This result yields up to a quadratic speedup for a broad class of divide & conquer algorithms. As an application, we obtain a quadratic speedup of Savitch's divide & conquer algorithm for directed *st*-connectivity [Sav70] (Chapter 5).

A crucial point is that we compose *subspace graphs* rather than fully specified quantum circuits: compiling each step into a concrete algorithm introduces constant-factor overheads depending on the chosen gate model. By composing at the level of subspace graphs and only converting it into a quantum circuit once at the end, these overheads appear only once, similarly to compositions via span programs [Rei09] and transducers [BJY24].

Switching Networks A switching network is a graph whose edges are labeled by Boolean variables. Given an assignment, an edge is "on" exactly when its variable equals 1. The network defines a Boolean function f that outputs 1 if and only if two special vertices are connected by a path of "on" edges. Switching networks have strong classical complexity connections: Shannon showed that series-parallel switching networks correspond to Boolean formulas [Sha38, Sha49], and Lee showed that general switching networks can model branching programs [Lee59].

Quantum algorithms for evaluating switching networks (implicitly studied as *st*-connectivity problems) were given in [JK17], using span program constructions building on [BR12], with tight analysis in [JJKP18]. With query access to edge variables, the evaluation time is governed by effective resistance in the "on" subgraph and minimum *st*-cuts formed from "off" edges. Prior work implies time

$$\sqrt{\max_{x\in f^{-1}(1)} \mathcal{R}_{s,t}(G(x)) \max_{x\in f^{-1}(0)} \sum_{e\in F_x} \mathsf{w}_e} \cdot \max_{e\in E} T_e,$$

where T_e is the cost to query edge e. We improve this to (Chapter 4)

$$\sqrt{\max_{x\in f^{-1}(1)} \mathcal{R}_{s,t}(G(x)) \max_{x\in f^{-1}(0)} \sum_{e\in F_x} \mathsf{w}_e T_e^2}.$$

This parallels variable-time improvements in [Jef22] (and includes variable-time quantum search [Amb10]). Since switching networks exactly model Boolean formulas without compilation overhead, they serve as a natural building block for our divide & conquer constructions.

1.5 Organization of the thesis

The remainder of the thesis is organized as follows. Chapter 2 collects background material and fixes notation used throughout the thesis. Chapter 3 studies quantum algorithms and tradeoffs for undirected *st*-connectivity. Chapter 4 introduces subspace graphs and develops their composition theory. Chapter 5 develops a time-efficient quantum divide & conquer construction, and shows a quadratic quantum speedup for Savitch's algorithm. Chapter 6 presents the quantum time-space tradeoff for directed *st*-connectivity, based on a switching network for $\textsc{Dist}_L$. Chapter 7 studies the gauge problem and gives a log-space quantum algorithm via a subspace-graph construction.

Chapter 2

Preliminaries

This chapter collects background material and fixes notation used throughout the thesis. Here we recall the necessary concepts from probability theory, generating quantum superpositions, Boolean formulas, graph theory, random walks, quantum walk search, and phase estimation algorithms. While much of this material is standard, we also introduce several definitions and conventions (e.g. edge-centred view on undirected graphs) that are specific to our setting and will be used in later chapters.

First, we state a few simple conventions and notational definitions. For a positive integer n, we let $[n] = \{1, \ldots, n\}$. Unless otherwise specified, logarithms are with respect to base 2. For any function f, we let $\widetilde{O}(f(n)) = f(n) \cdot \mathit{polylog}(n)$. So, for example, $\widetilde{O}(1)$ is polylog(n).

2.1 Probability theory

A *(probability) distribution* on a finite set V is a non-negative function $\sigma \colon V \to \mathbb{R}_{\geq 0}$ such that $\sum_{v \in V} \sigma(v) = 1$. Its *support* is defined as $\mathrm{supp}(\sigma) := \{v \in V : \sigma(v) > 0\}$. We will implicitly identify such σ with *row* vectors, as is customary in the random walk literature. To any distribution σ, we also associate a quantum state $|\sigma\rangle := \sum_{v \in V} \sqrt{\sigma(v)}|v\rangle$. Measuring $|\sigma\rangle$ in the standard basis returns a sample from σ.

For any distribution σ on V, and any subset $M \subseteq V$, we will let $\sigma(M) = \sum_{u \in M} \sigma(u)$. We let σ_M denote the *normalized restriction* of σ to M, defined by $\sigma_M(u) = \sigma(u)/\sigma(M)$ for all $u \in M$ and $\sigma_M(u) = 0$ elsewhere.

Finally, the *total variation distance* between two distributions σ and τ on V is defined as

$$\|\sigma - \tau\|_{\mathrm{TV}} := \frac{1}{2} \sum_{u \in V} |\sigma(u) - \tau(u)| = \max_{A \subseteq V} |\sigma(A) - \tau(A)|.$$

2.2 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.

2.2.1. Lemma ([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 $O(Tm)$.*

2.2.2. Remark (Larger alphabets). The result of Lemma 2.2.1 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 $O(Tm)$.

2.3 Function composition and Boolean formulas

A Boolean formula, φ, on N variables is a rooted tree with N leaves, where each internal vertex is labelled by either $\vee$ (OR) or $\wedge$ (AND), and each leaf is labelled by a unique variable or its negation. The depth of a formula is the length of the longest path from the root to a leaf. We assume that for each internal vertex with d children, the outgoing edges are labelled by some d-element set, without loss of generality, $[d]$, so that every vertex can be labelled by a string of the labels of its path from the root. That is: the root is labelled by the empty string, denoted $\emptyset$, and any other vertex is labelled by the label of its parent, concatenated with the label of the edge from its parent to it. We call the set of all strings labelling leaves Σ, so $|\Sigma| = N$. We let Σ_i denote the set of strings in Σ whose first letter is i. We let $\overline{\Sigma}$ denote the set of labels of all vertices, so it is the set of prefixes of strings in Σ.

We label the variable associated with a leaf $\sigma \in \Sigma$ by σ as well. A fixed setting of the variables $x \in \{0,1\}^\Sigma$ induces a value at each vertex of φ, as follows. If a leaf σ is labelled by an unnegated variable, then its value is x_σ, and otherwise its value is $\neg x_\sigma$. If a vertex is not a leaf, if it is labelled by $\vee$, then its value is the OR of all the values of its children, whereas if it is labelled by $\wedge$, its value is the AND of all the values of its children. We let $\varphi(x)$ denote the value of the root of the tree.

2.3.1. Definition. We say a formula is *symmetric* if for every internal node, the sub-trees of its children are identical aside from their leaves.

Note that symmetric formulas do not necessarily compute symmetric functions – functions that only depend on the Hamming weight of the input.

2.3.2. Definition. We say a (family of) formulas is balanced if there is a constant c such that for every node, if its subtree has N leaves, and it has d children, then the sub-tree of each child has at most cN/d leaves.

For any $f : \{0,1\}^\Sigma \to \{0,1\}$ and $\{f_\sigma : \{0,1\}^{m_\sigma} \to \{0,1\}\}_{\sigma\in\Sigma}$, we define the composed function $f \circ (f_\sigma)_{\sigma\in\Sigma} : \{0,1\}^{\sum_{\sigma\in\Sigma} m_\sigma} \to \{0,1\}$ by

$$f \circ (f_\sigma)_{\sigma\in\Sigma}(x) = f(f_\sigma(x^\sigma))_{\sigma\in\Sigma}$$

where $x = (x^\sigma)_{\sigma\in\Sigma}$ is the $\sum_\sigma m_\sigma$-bit string obtained by concatenating the strings $\{x^\sigma\}_\sigma$, and $(f_\sigma(x^\sigma))_{\sigma\in\Sigma}$ is the $|\Sigma|$-bit string whose σ-th bit is $f_\sigma(x^\sigma)$. We can also write $f \circ (f_\sigma)_{\sigma\in F}$ for some $F \subset \Sigma$, in which case, we implicitly assume that f_σ is the identity on $\{0,1\}$ for all $\sigma \in \Sigma \setminus F$. For a Boolean formula φ on Σ, we will also write $\varphi \circ (f_\sigma)_{\sigma\in\Sigma}$ as a short-hand for $f_\varphi \circ (f_\sigma)_{\sigma\in\Sigma}$, where f_φ is the function computed by the formula.

2.4 Graph theory

In this work we will use two equivalent, but conceptually different, representations of undirected graphs, depending on context. Graphs can be described either in a *vertex-centred* way, where vertices are primary objects and edges specify how they are connected, or in an *edge-centred* way, where edges are primary objects and vertex incidence specifies connectivity. Both viewpoints describe the same mathematical objects, but each is convenient for different purposes.

For random walks and basic connectivity arguments, we will mostly use the standard vertex-centred viewpoint. Later, when studying subspace graphs, switching networks, and the Hilbert spaces underlying our quantum algorithms, it will be more convenient to work in an edge-centred framework, where edges are explicitly labelled objects. We therefore introduce both descriptions here and explain how they relate.

2.4.1. Definition (Undirected graph – vertex-centred view). An *undirected graph* $G = (V(G), E(G))$ consists of a vertex set $V(G)$ and an edge set

$$E(G) \subseteq \{\{u, v\} \in V(G)^2 : u \neq v\},$$

whose elements are unordered pairs of distinct vertices.

For $u \in V(G)$ we define the neighborhood

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

and let $E(u)$ be the set of edges incident to u. The degree of u is $d_u = |N(u)| = |E(u)|$. For convenience, we assume throughout that all vertices have positive degree.

In some contexts it will be useful to fix an arbitrary orientation of each edge. This orientation is purely for technical convenience and does not affect any undirected graph property such as connectivity. If an orientation is fixed, we write $E^{\rightarrow}(u)$ for the edges oriented outwards from u and $E^{\leftarrow}(u)$ for those oriented inwards.

While the above definition is standard, it treats edges only implicitly as pairs of vertices. For our purposes, however, undirected graphs will often define linear spaces spanned by their *edges*. This motivates the following equivalent, edge-centred definition.

2.4.2. Definition (Undirected graph – edge-centred view). An *undirected graph* $G = (V(G), E(G))$ is specified by a finite set of vertex labels $V(G)$ and a finite set of edge labels $E(G)$, together with incidence sets

$$E_G(\mathsf{u}) = E_G^{\leftarrow}(\mathsf{u}) \sqcup E_G^{\rightarrow}(\mathsf{u}) \qquad \text{for each } \mathsf{u} \in V(G),$$

such that for every edge $e \in E(G)$ there exist unique distinct vertices $\mathsf{u}, \mathsf{v} \in V(G)$ with

$$e \in E_G^{\rightarrow}(\mathsf{u}) \cap E_G^{\leftarrow}(\mathsf{v}).$$

The notation u, v is used here because we will later work in a setting that additionally involves directed graphs, and we wish to ensure that vertex references remain clear.

Given such a graph, we may recover the usual vertex-pair description by mapping each edge e to the ordered pair (u, v) such that $e \in E^{\rightarrow}(\mathsf{u}) \cap E^{\leftarrow}(\mathsf{v})$. By abuse of notation, we write $(\mathsf{u}, \mathsf{v}) \in E(G)$ to mean that there exists an edge from u to v in this sense.

We say that u is *connected to* v in G if there exists a path $\mathsf{u} = \mathsf{u}_0, \ldots, \mathsf{u}_\ell = \mathsf{v}$ such that for all $i \in [\ell]$, either $(\mathsf{u}_{i-1}, \mathsf{u}_i) \in E(G)$ or $(\mathsf{u}_i, \mathsf{u}_{i-1}) \in E(G)$.

When the graph G is clear from the context, we will often omit it from the notation and simply write V and E instead of $V(G)$ and $E(G)$. Similarly, we may write $E(u)$ instead of $E_G(u)$ when no ambiguity arises.

We may associate non-negative *weights* $\{w_e\}_{e \in E}$ with the edges of a graph. Unless stated otherwise, all edge weights are assumed to be equal to 1.

2.4.3. Definition. An *st-cut-set* is a set of edges $F \subseteq E$ whose removal from G leaves s and t in different connected components.

2.4.4. Definition. Let $B \subseteq V(G)$. A *flow* on G is a real-valued function θ on $E(G)$. For $u \in V(G)$, define

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

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

2.4.5. Definition. On a weighted graph G with $s, t \in V(G)$ *effective resistance* is defined

$$\mathcal{R}_{s,t}(G) := \min_{\text{unit flows } \theta} \sum_{e \in E} \frac{\theta(e)^2}{\mathsf{w}_e}.$$

We now turn to directed graphs, where each edge carries an orientation.

2.4.6. Definition (Directed graph). A directed graph $G = (V(G), E(G))$ 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 that v is *reachable* from u if there exists a path $u = u_0, \ldots, u_\ell = v$ such that $(u_{i-1}, u_i) \in E(G)$ for all $i \in [\ell]$. When G is clear from the context, we again omit it from the notation.

Finally, we will often build a (undirected) graph G out of graphs G_1 and G_2 by "gluing" (identifying) some of the vertices in G_1 with some vertices in G_2. The following definition makes precise what we mean by this.

2.4.7. Definition. Let G_1 and G_2 be graphs, with $\mathsf{u}_{1,1}, \ldots, \mathsf{u}_{1,r} \in V(G_1)$ and $\mathsf{u}_{2,1}, \ldots, \mathsf{u}_{2,r} \in V(G_2)$. The graph G 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:

$$\begin{aligned} V(G) &= V(G_1) \sqcup (V(G_2) \setminus \{\mathsf{u}_{2,1}, \ldots, \mathsf{u}_{2,r}\}) \\ \text{and } E(G) &= E(G_1) \sqcup E(G_2) \end{aligned}$$

and incidence sets:

$$\begin{aligned} \forall \mathsf{u} \in V(G_1) \setminus \{\mathsf{u}_{1,1}, \ldots, \mathsf{u}_{1,r}\},\ E_G^{\rightarrow}(\mathsf{u}) = E_{G_1}^{\rightarrow}(\mathsf{u}) \text{ and } E_G^{\leftarrow}(\mathsf{u}) = E_{G_1}^{\leftarrow}(\mathsf{u}), \\ \forall \mathsf{u} \in V(G_2) \setminus \{\mathsf{u}_{2,1}, \ldots, \mathsf{u}_{2,r}\},\ E_G^{\rightarrow}(\mathsf{u}) = E_{G_2}^{\rightarrow}(\mathsf{u}) \text{ and } E_G^{\leftarrow}(\mathsf{u}) = E_{G_2}^{\leftarrow}(\mathsf{u}), \\ \text{and } \forall i \in [r],\ E_G^{\rightarrow}(\mathsf{u}_{1,i}) = E_{G_1}^{\rightarrow}(\mathsf{u}_{1,i}) \sqcup E_{G_2}^{\rightarrow}(\mathsf{u}_{2,i}) \\ \text{and } E_G^{\leftarrow}(\mathsf{u}_{1,i}) = E_{G_1}^{\leftarrow}(\mathsf{u}_{1,i}) \sqcup E_{G_2}^{\leftarrow}(\mathsf{u}_{2,i}). \end{aligned}$$

We made the arbitrary choice to let the glued vertices inherit their names from G_1 rather than G_2, but since in practice we will mainly work on the edges of the graph, this detail doesn't matter so much.

2.5 Random walks

Fix an undirected graph $G = (V, E)$ with $n = |V|$ and $m = |E|$. Fix edge weights given by a symmetric matrix $W \in \mathbb{R}_{\geq 0}^{V \times V}$ such that $W_{u,v} = W_{v,u}$ for all $u, v \in V$, and $W_{u,v} > 0$ if and only if $\{u, v\} \in E$. Then $G = (V, E, W)$ defines a *weighted graph*. When no W is given, the graph is *unweighted* and we let $W_{u,v} = 1$ for all $\{u, v\} \in E$. For $u \in V$, define $w_u = \sum_{v \in V} W_{u,v}$. The corresponding *(weighted) random walk* is the reversible Markov chain on V with *transition matrix* $P \in \mathbb{R}_{\geq 0}^{V \times V}$ given by

$$P_{u,v} = \begin{cases} \frac{W_{u,v}}{w_u} & \text{if } \{u, v\} \in E \\ 0 & \text{otherwise} \end{cases} \qquad \forall u, v \in V. \tag{2.5.1}$$

This means that the probability of moving from the vertex u along an edge to a neighbouring vertex v is proportional to the edge's weight. In the unweighted case, this is called the *simple random walk*; in each step it simply moves to a neighbouring vertex chosen uniformly at random.

Let $\pi \in \mathbb{R}_{>0}^{V}$ be the distribution defined by

$$\pi(u) = \frac{w_u}{\mathcal{W}(G)} \qquad \forall u \in V,$$

where $\mathcal{W}(G) = \sum_{u \in V} w_u = \sum_{u,v \in V} W_{u,v}$. In the unweighted case, π is proportional to the degree. The distribution π is a *stationary* distribution of the random walk, i.e., $\pi P = \pi$ (it is a left eigenvector of P with eigenvalue 1).

In fact, when the graph G is connected, π is also the *unique* stationary distribution of P. If in addition the graph is not bipartite, then all other eigenvalues have absolute value strictly less than one. That is, if $1 = \lambda_1 \geq \cdots \geq \lambda_n \geq -1$ are the eigenvalues of P then the *(absolute) spectral gap* $\gamma_\star = \gamma_\star(G) := \min\{1 - |\lambda_j| : j = 2, \ldots, n\} = \min\{1 - \lambda_2, 1 + \lambda_n\}$ is strictly positive. Importantly, the inverse of the spectral gap bounds the random walk's *mixing time*, that is, the time required for convergence to the stationary distribution:

2.5.1. Theorem ([LPW17, Thm. 12.4]). *Assume G is connected and not bipartite. Let $\varepsilon > 0$ and*

$$t \geq \frac{1}{\gamma_\star} \log\Big(\frac{1}{\varepsilon \pi_{\min}}\Big),$$

where $\pi_{\min} = \min_{u \in V} \pi(u)$. Then $\|\sigma P^t - \pi\|_{\mathrm{TV}} \leq \varepsilon$ for any distribution σ on V.

Conversely, it is known that $t \geq (\frac{1}{\gamma_\star} - 1) \log(\frac{1}{2\varepsilon})$ is necessary to ensure mixing from an arbitrary initial distribution [LPW17, Thm. 12.5]. In the unweighted case,

we have $\pi_{\min} \geq \frac{d_{\min}}{n\, d_{\max}} \geq \frac{1}{n^2}$, so the former shows that Theorem 2.5.1 is tight up to $\log(n)$ factors in that case.

Finally, for any $s, t \in V$ we let $\mathcal{H}_{s,t}$ denote the *hitting time* from s to t, which is the expected number of steps needed to reach t in a random walk starting from s. We let $\mathcal{C}_{s,t} = \mathcal{H}_{s,t} + \mathcal{H}_{t,s}$ denote the *commute time* between s and t – the expected number of steps needed to reach t and then return to s in a random walk starting from s. These quantities are finite if and only if s and t are in the same component of G. More generally, the commute time $\mathcal{C}_{s,M}$ from s to a subset $M \subseteq V$ is the expected number of steps needed to reach any vertex in M and then return to s in a random walk starting from s.

2.6 Quantum walk search algorithms

Quantum walk search refers to the use of quantum walks to find certain "marked" elements on a graph. We will use quantum walk search to search for a vertex connected to t in the connected component of s. Specifically, we will use the following special case of [AGJ20, Thm. 13].[1]

2.6.1. Theorem. *Let P be a random walk on a weighted graph with vertex set V, $M \subseteq V$ a subset of "marked" vertices, and $s \in V$. Let C be the (quantum) time complexity to check for a given $u \in V$ whether $u \in M$, let U be the time complexity of implementing the weighted quantum walk oracle*

$$|u\rangle|0\rangle \mapsto \sum_{v \in N(u)} \sqrt{P_{u,v}}|u\rangle|v\rangle.$$

in space $O(\log(n))$. Let $\mathcal{C}$ be a known upper bound on the commute time $\mathcal{C}_{s,M}$ in the case where s and M are connected (and in particular $M \neq \emptyset$). Then there is a quantum algorithm that, if $M \neq \emptyset$ and s is connected to M, finds an element of M with probability at least $2/3$. If $M = \emptyset$ or s is not connected to M, then the algorithm outputs a vertex not in M. The algorithm has time complexity $O(\sqrt{\log(\mathcal{C})}\log(n) + \sqrt{\mathcal{C}\log(\mathcal{C})\log(\log(\mathcal{C}))}(\mathsf{C}+\mathsf{U}))$ and space complexity $O(\log(n))$.

2.7 Phase estimation algorithms

The technique of phase estimation dates back to [Kit96], but we will use a specific application of this technique precisely defined in [JZ23], following a blueprint that had already been used many times in the quantum algorithms literature.

[1] To see that this follows from [AGJ20, Thm. 13], note that when $|\sigma\rangle = |s\rangle$, the cost to set up $|\sigma\rangle$ is $\log(n)$ and the value $C_{\sigma,M}$ from [AGJ20] is exactly the commute time from s to M [AGJ20, Thm. 4].

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

Let $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$. The algorithm defined by $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ performs phase estimation of $U_{\mathcal{AB}}$ on initial state $|\psi_0\rangle$, to sufficient precision that by measuring the phase register and checking if the output is 0, we can distinguish between a *negative case* and a *positive case*.

2.7.2. Definition (Negative Witness). A *negative witness* for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ is a vector $|w_{\mathcal{A}}\rangle \in \mathcal{A}$ such that $|w_{\mathcal{B}}\rangle := |\psi_0\rangle - |w_{\mathcal{A}}\rangle \in \mathcal{B}$.

2.7.3. Definition (Positive Witness). A *positive witness* for $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ is a vector $|w\rangle \in \mathcal{A}^\perp \cap \mathcal{B}^\perp$ such that $\langle\psi_0|w\rangle \neq 0$.

Note that a negative witness exists if and only if $|\psi_0\rangle \in \mathcal{A}+\mathcal{B}$, whereas a positive witness exists if and only if $|\psi_0\rangle$ has a non-zero component in $(\mathcal{A}+\mathcal{B})^\perp = \mathcal{A}^\perp \cap \mathcal{B}^\perp$, which is if and only if $|\psi_0\rangle \notin \mathcal{A}+\mathcal{B}$. The following theorem describes an algorithm for distinguishing these two cases.

2.7.4. Theorem ([JZ23]). *Fix $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$ as in Definition 2.7.1. Suppose we can generate the state $|\psi_0\rangle$ in cost S, and implement $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$ in cost A.*
Let $c_+ \in [1, 50]$ be some constant, and let $\mathcal{C}_- \geq 1$ be a positive real number that may scale with $|x|$, such that we are guaranteed that one of the following holds:

Positive Condition: *There is a positive witness $|w\rangle$ s.t. $\frac{|\langle w|\psi_0\rangle|^2}{\||w\rangle\|^2} \geq \frac{1}{c_+}$.*

Negative Condition: *There is a negative witness $|w_{\mathcal{A}}\rangle$ s.t. $\||w_{\mathcal{A}}\rangle\|^2 \leq \mathcal{C}_-$.*

Then there is a quantum algorithm that distinguishes these two cases with bounded error in time

$$O\left(\mathsf{S} + \sqrt{\mathcal{C}_-}\mathsf{A}\right)$$

and space $O(\log \dim H + \log \mathcal{C}_-)$.

The reason we assume that $\mathcal{A}$ and $\mathcal{B}$ are given by bases is that this extra structure allows us to implement the reflections around these spaces, which we describe shortly. For this reason, we refer to these given bases as *working bases*.

2.7.5. Definition (Working Basis Generation). We say an orthonormal basis $\Psi = \{|b_\ell\rangle\}_{\ell \in L} \subset H_G$ can be generated in time T if

1. The reflection around the subspace $\text{span}\{|\ell\rangle : \ell \in L\}$ of H_G 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.

2.7.6. Claim. *If $\Psi_{\mathcal{A}}$ and $\Psi_{\mathcal{B}}$ are working bases for $\mathcal{A}$ and $\mathcal{B}$, respectively, that can each be generated in time T, then $U = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$ can be implemented in time T.*

Proof:
Run the map $|\ell\rangle \mapsto |b_\ell\rangle$ in reverse, reflect around $\text{span}\{|\ell\rangle : \ell \in L\}$, and then run the map $|\ell\rangle \mapsto |b_\ell\rangle$. □

We end by remarking that it is enough to have a working basis for the complement of $\mathcal{A}$ or $\mathcal{B}$.

2.7.7. Corollary. *If Ψ is a working basis that can each be generated in time T, then there is a basis Ψ' for $\text{span}\{\Psi\}^\perp$ that can be generated in time T.*

Proof:
Suppose $\Psi = \{|b_\ell\rangle\}_{\ell \in L}$, and $H_G \equiv \text{span}\{|\ell\rangle : \ell \in Z\}$ where $L \subseteq Z$. Let U_Ψ be the given map that acts as $|\ell\rangle \mapsto |b_\ell\rangle$ for $\ell \in L$, in time T. Then define $\Psi' = \{|b'_\ell\rangle = U_\Psi|\ell\rangle\}_{\ell \in Z \setminus L}$. That is, Ψ' is generated by U_Ψ. Since both bases are generated by the same map, their generation has the same time complexity. □

Chapter 3

(No) Quantum space-time tradeoff for USTCON

Undirected *st*-connectivity is important both for its applications in network problems, and for its theoretical connections with logspace complexity. Classically, a long line of work led to a time-space tradeoff of $T = \widetilde{O}(n^2/S)$ for any S such that $S = \Omega(\log(n))$ and $S = O(n^2/m)$. Surprisingly, we show that quantumly there is no nontrivial time-space tradeoff: there is a quantum algorithm that achieves both optimal time $\widetilde{O}(n)$ and space $O(\log(n))$ simultaneously. This improves on previous results, which required either $O(\log(n))$ space and $\widetilde{O}(n^{1.5})$ time, or $\widetilde{O}(n)$ space and time. We additionally prove an $\Omega(n)$ lower bound, ruling out any asymptotic improvement over our algorithm beyond polylogarithmic factors. To complement this, we show that there is a nontrivial time-space tradeoff when given a lower bound on the spectral gap of a corresponding random walk.

This chapter is based on the paper "(No) Quantum Space-Time Tradeoff for USTCON" by Simon Apers, Stacey Jeffery, Galina Pass, and Michael Walter [AJPW23].

3.1 Introduction

For an undirected graph $G = (V, E)$ on $n = |V|$ vertices and $m = |E|$ edges, with $s, t \in V$, st-connectivity or USTCON is the problem of deciding whether s and t are in the same component. This problem has applications in many other graph and network problems, and is of theoretical importance for its connection with space complexity (see e.g. [Wig92]). In particular, USTCON is complete for the class *symmetric logspace*, SL, which was shown to be equal to *logspace*, L, by exhibiting a classical deterministic logspace algorithm for USTCON [Rei08]. In this chapter, we consider quantum algorithms for this problem.

There are different versions of the problem USTCON depending on how G is accessed. If G is given as an adjacency matrix, we denote the problem by $\text{USTCON}_{\text{mat}}$. If G is given as an array of arrays, one for each vertex, enumerating the neighbours, we denote the problem by $\text{USTCON}_{\text{arr}}$.[1] If one only cares about space complexity, these problems are equivalent, but the same is not true of time complexity: adjacency queries can simulate an array query, and vice versa, in logspace, but there is a non-negligible time overhead.

A classical deterministic algorithm based on breadth-first search or depth-first search can solve $\text{USTCON}_{\text{arr}}$ in $\widetilde{O}(m)$ time, using $\widetilde{O}(n)$ space. Using a random walk, the space complexity can be improved to $O(\log(n))$, at the expense of $\widetilde{O}(nm)$ time complexity [AKL+79]. A series of works [BKRU89, BBR+90, Edm93, BF93, Fei93] culminated in a space-time tradeoff for $\text{USTCON}_{\text{arr}}$ of $T = \widetilde{O}(n^2/S)$ queries for any space bound $S = \Omega(\log(n))$ and $S = O(n^2/m)$, due to Kosowski [Kos13]. While there is no matching time-space lower bound, it is unlikely that this tradeoff can be significantly improved (see [Kos13, Section 5.1 of arXiv v2] for a discussion). Kosowski's algorithm is based on using Metropolis-Hastings random walks to find connections between S sampled vertices and s, t until it is becomes possible to conclude that s and t are connected. For comparison, in the adjacency matrix model, the randomized query complexity of $\text{USTCON}_{\text{mat}}$ is $\widetilde{\Theta}(n^2)$ and there is no space-time tradeoff.

A quantum algorithm of Dürr, Heiligman, Høyer and Mhalla [DHHM06] for CONNECTIVITY can be adapted to solve $\text{USTCON}_{\text{mat}}$ in $\widetilde{O}(n^{1.5})$ time and $\text{USTCON}_{\text{arr}}$ in $\widetilde{O}(n)$ time, both of which are optimal up to polylog factors. Both of these algorithms use $\widetilde{O}(n)$ space, of which all but $O(\log(n))$ can be classical space (assuming quantum RAM access). A subsequent quantum algorithm for $\text{USTCON}_{\text{mat}}$ due to Belovs and Reichardt uses $\widetilde{O}(n^{1.5})$ time, but only $O(\log(n))$ space [BR12], which is optimal in terms of both space and time. It is also possible to solve $\text{USTCON}_{\text{arr}}$ in $O(\log(n))$ space and $\widetilde{O}(\sqrt{nm})$ time, using a quantum walk (see for

[1] There are variations on the details of this model. For now, we allow $\text{USTCON}_{\text{arr}}$ to stand in for multiple variations of the array access model, but precise details of the variations can be found in Section 3.3.

$\textsc{ustcon}_{\text{mat}}$		
	Time	TS-tradeoffs
Classical	$\widetilde{\Theta}(n^2)$	$S = O(\log(n))$, $T = \widetilde{O}(n^3/d)$ $S = \widetilde{O}(n)$, $T = \widetilde{O}(n^2)$
Quantum	$\widetilde{\Theta}(n^{1.5})$	$S = O(\log(n))$, $T = \widetilde{O}(n^{1.5})$ [BR12]
$\textsc{ustcon}_{\text{arr}}$		
	Time	TS-tradeoffs
Classical	$\widetilde{\Theta}(m)$	$T = \widetilde{O}(\max\{n^2/S, m\})$
Quantum	$\widetilde{\Theta}(n)$	$S = O(\log(n))$, $T = \widetilde{O}(n^{1.5})$ $S = T = \widetilde{O}(n)$ [DHHM06]
		This work: $S = O(\log(n))$, $T = \widetilde{O}(n)$

Table 3.1: A summary of classical (randomized) and quantum time and space complexities for USTCON in the adjacency matrix and adjacency array models. The classical results for $\textsc{ustcon}_{\text{mat}}$ follow from (1) the $\log(n)$-space result for $\textsc{ustcon}_{\text{arr}}$ with an n/d overhead for finding neighbours of the current vertex in a d-regular graph; and (2) BFS.

example [Bel13]). This quantum walk algorithm requires a quantum version of array access to the input graph, which we refer to as $\textsc{ustcon}_{\text{qw}}$ in the next section.

3.1.1 Summary of results

We describe new quantum walk algorithms for $\textsc{ustcon}_{\text{arr}}$. These algorithms consider a quantum walk version of the adjacency array model, in which the input graph is accessed by a quantum analogue of classical random walk steps. Recall that in the adjacency array model, we assume that for any vertex u, we can query, for any $i \in [d_u]$, the i-th neighbour of u, $v_i(u)$. Then a random walk step can be performed from state u by sampling a uniform $i \in [d_u]$, and then computing $v_i(u)$, which becomes the current state. In the *quantum walk access model*, we assume that for any vertex u, we can prepare a uniform superposition over the neighbours of u. While these models are not identical, they are very similar, and in Section 3.3, we formally define the models, and show that quantum walk access can be simulated in the array model with polylogarithmic overhead under reasonable additional assumptions.

Letting $\text{USTCON}_{\text{qw}}$ denote the *st*-connectivity problem in the quantum walk access model, we present a one-sided error quantum algorithm that solves $\text{USTCON}_{\text{qw}}$ in time $\widetilde{O}(n)$ and space $O(\log(n))$. Perhaps surprisingly, this means that $\text{USTCON}_{\text{qw}}$ admits *no* nontrivial tradeoff between space and time in the quantum setting – a single algorithm can solve this problem optimally in terms of both time and space (see Theorem 3.4.7 for the formal result).

3.1.1. Theorem (Informal). *There is a $O(\log(n))$-space quantum algorithm that decides $\text{USTCON}_{\text{qw}}$ with one-sided error in $\widetilde{O}(n)$ time.*

In this chapter, when we say *time*, we are counting: (1) quantum gates (unitaries that act on at most a constant number of qubits); (2) quantum walk queries to G; and (3) (quantum) random access (QCRAM) operations (QCRAM is used in our second algorithm only, see below). Inspired by [Kos13], our algorithm is based on a quantum walk search for t starting from s using a random walk that can be interpreted as a Metropolis-Hastings random walk.

Because of the close relationship between USTCON and classical logspace, we can consider what this means for logspace problems in general. It does not mean that more space does not reduce the quantum time complexity of *any* problem, but it is interesting to consider: in what settings do we get a non-trivial time-space tradeoff? We consider one such setting: when we are given a promise on the spectral gap or mixing time of the random walk on G (see Section 2.5). In that case, we prove the following theorem (see Theorem 3.5.2 for the formal result).

3.1.2. Theorem (Informal). *Suppose whenever s and t are connected, the random walk spectral gap is at least $\delta > 0$. For any $S \in \Omega(\log(n))$, there is a quantum algorithm that decides $\text{USTCON}_{\text{qw}}$ with bounded error in $O(S)$ space and $T = \widetilde{O}\left(\frac{S}{\delta} + \sqrt{\frac{n}{\delta S}}\right)$ time.*

Our algorithm adapts [Ape19] in a way that allows exploiting the spectral gap promise. The time bound decreases monotonically for $S \in \Omega(\log n)$ until $S \in O((n\delta)^{1/3})$, at which point it reaches time complexity $T = \widetilde{O}(n^{1/3}/\delta^{2/3})$. We leave it as an open problem to prove a matching lower bound (at least for some values of δ), which would prove that in certain regimes, it is not possible to achieve optimal time and space simultaneously.

In the space bound S of both algorithms in Theorems 3.1.1 and 3.1.2, only $O(\log(n))$ memory needs to be actual quantum workspace (i.e., qubits). The remaining $O(S)$ memory can be classical RAM in the first algorithm and QCRAM in the second algorithm, that is, classical RAM that is queryable at a quantum superposition of addresses. We discuss the latter in Section 3.2.

We summarize our results in Fig. 3.1. For $S = \log(n)$, the algorithm of Theorem 3.1.2 has a worse time complexity than the algorithm of Theorem 3.1.1,

whenever $\delta < \frac{1}{n}$. We leave it as an open problem to give a single algorithm that is optimal for all δ.

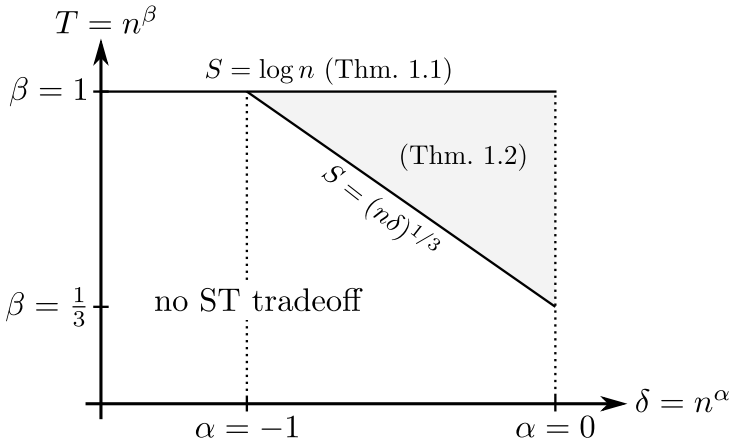

Figure 3.1: Quantum space-time tradeoffs for USTCON, with axes representing the time complexity and spectral gap promise (up to polylog-factors). The grey area represents the regime in which a non-trivial tradeoff is achieved. Theorem 3.1.1 (upper line) corresponds to the regime with space $S = O(\log n)$ and time $T = \widetilde{O}(n)$. Theorem 3.1.2 (grey area) corresponds to the regime with a promise on δ, and interpolates between $S = O(\log n)$ and $T = \widetilde{O}(n)$, and $S = O((n\delta)^{1/3})$ and $T = \widetilde{O}(n^{1/3}/\delta^{2/3})$.

Organization: The remainder of this chapter is organized as follows. We describe the problem in Section 3.2 and Section 3.3. In Section 3.4, we prove Theorem 3.1.1 by exhibiting a quantum algorithm for $\text{USTCON}_{\text{qw}}$ that is optimal in both time and space. For completeness, we also include a proof of a corresponding lower bound in Section 3.4.1. In Section 3.5, we prove Theorem 3.1.2 exhibiting a quantum time-space tradeoff when given a promise on the spectral gap.

3.2 Quantum RAM

Our algorithm will exploit the given space by saving sets of vertices which will be either connected to s or to t. For our quantum algorithm to access this space, we assume access to a so-called quantum-classical random access memory or *QCRAM*. This refers to a memory that only stores classical information, but can be queried at a superposition of addresses. More specifically, an R-bit QCRAM stores a string of bits $q \in \{0,1\}^R$ so that the following operations are supported in time $polylog(R)$:

1. *UPDATE(i, x):* store $x \in \{0,1\}$ in the i-th bit (i.e., set $q_i = x$).

2. *QUERY:* for any superposition $\sum_i \alpha_i|i\rangle|s_i\rangle$, it maps

$$\sum_i \alpha_i|i\rangle|s_i\rangle \mapsto \sum_i \alpha_i|i\rangle|s_i \oplus q_i\rangle.$$

As was first described by Kerenidis and Prakash [KP17], using such a QCRAM we can set up a data structure to generate quantum superpositions over elements in the QCRAM. We will use the following formulation based on [Ape19].

3.2.1. Lemma. *Fix integer parameters ℓ and k. Using an $O(k\ell\log(\ell))$-bit QCRAM, there is a data structure, D, that stores up to ℓ elements $x \in \{0,1\}^k$ with associated integer weights, c_x, of bounded absolute value for some poly(ℓ) bound, and supports the following operations in time $O(k \cdot \text{polylog}(k\ell))$ per operation:*

1. *insertion or deletion of a pair (x, c_x),*
2. *quantum queries of the form "Is $x \in D$?",*
3. *preparation of the quantum state $\frac{1}{\sqrt{\sum_{x\in D} c_x}} \sum_{x\in D} \sqrt{c_x}|x\rangle$.*

3.3 USTCON and the quantum walk model

In this section we define the undirected *st*-connectivity problem (USTCON). The input to this problem is an undirected graph $G = (V, E)$. Classically, there are various ways this input may be given, which may change the complexity of the problem. For example, in the *adjacency array model* (defined below), it is possible to randomly sample a neighbour of any vertex u in $O(1)$ queries to G (assuming access to the vertex degrees), facilitating a random walk on G, whereas if G is given as an *adjacency matrix*, a random walk step is not so simple.

We will work in a quantum walk analogue of the adjacency array model. We assume that G can be accessed via the *quantum walk oracle* that for every $u \in V$ outputs a uniform superposition over its neighbours:[2]

$$\mathcal{O}_W : |u\rangle|0\rangle \mapsto \frac{1}{\sqrt{d_u}} \sum_{v\in N(u)} |u\rangle|v\rangle. \tag{3.3.1}$$

Formally, we describe $\text{USTCON}_{\text{qw}}$ in terms of the input and output.

3.3.1. Problem ($\text{USTCON}_{\text{qw}}$). *Given access to an undirected graph $G = (V, E)$ via the quantum walk oracle $\mathcal{O}_W$, and two vertices $s, t \in V$, decide whether s and t are in the same connected component of G.*

[2] Note that this is exactly the quantum walk oracle defined in Theorem 2.6.1, specialized to unweighted graphs.

To compare our work with classical results on $\text{USTCON}_{\text{arr}}$, we describe an implementation of the quantum walk oracle defined above based on adjacency array access to a graph. Let $u \in V$ and $i \in [d_u]$. We assume that for each vertex u there is a fixed numbering of its neighbours from 1 to d_u. In the *adjacency array model*, two types of queries are allowed:

- Degree query $\mathcal{O}_D : |0\rangle|u\rangle \mapsto |d_u\rangle|u\rangle$
- Neighbour query $\mathcal{O}_N : |u\rangle|i\rangle|0\rangle \mapsto |u\rangle|i\rangle|v_i(u)\rangle$

In the *sorted adjacency array model* we additionally assume that for every vertex $u \in V$ its neighbours are sorted: for any $i, j \in [d_u]$, if $i < j$ then $v_i(u) < v_j(u)$. In particular, this allows us to check with $O(\log(n))$ queries whether a given pair of vertices u, v are adjacent. We define the USTCON-problem in this model as follows.

3.3.2. Problem ($\text{USTCON}_{\text{s-arr}}$). *Given access to an undirected graph $G = (V, E)$ via the sorted adjacency array model and two vertices $s, t \in V$, decide whether s and t are in the same connected component of G.*

The question that we consider is how many sorted adjacency array queries to the graph it takes to implement the quantum walk oracle $\mathcal{O}_W$.

3.3.3. Lemma.
The quantum walk oracle $\mathcal{O}_W$ for an unweighted graph G (Eq. (3.3.1)) with maximum degree $d_{\max}$ can be implemented with $O(\log(d_{\max}))$ queries in the sorted adjacency array model, and $\widetilde{O}(1)$ other elementary operations and space.

Proof:
Assume first, that there is an additional type of query allowed, namely:

$$\text{Index query: } \mathcal{O}_I : |u\rangle|v\rangle|0\rangle \mapsto |u\rangle|v\rangle|i\rangle \tag{3.3.2}$$

for $u, v \in V$ and $i \in [d_u]$ such that $v_i(u) = v$. Then $\mathcal{O}_W$ can be implemented using $\mathcal{O}_I$, $\mathcal{O}_D$, and $\mathcal{O}_N$ as follows. Let F_d denote the Fourier transform over $\mathbb{Z}_d$, and let $F = \sum_{d=1}^{n} |d\rangle\langle d| \otimes F_d$, which can be implemented (to any inverse polynomial precision) in $O(\log(n))$ gates. Then for any $u \in V$, we implement:

$$\begin{aligned}|0\rangle|u\rangle|0\rangle|0\rangle &\overset{\mathcal{O}_D}{\mapsto} |d_u\rangle|u\rangle|0\rangle|0\rangle \overset{F}{\mapsto} \frac{1}{\sqrt{d_u}}|d_u\rangle \sum_{i=1}^{d_u} |u\rangle|i\rangle|0\rangle \\ &\overset{\mathcal{O}_N}{\mapsto} \frac{1}{\sqrt{d_u}}|d_u\rangle \sum_{i=1}^{d_u} |u\rangle|i\rangle|v_i(u)\rangle \overset{\mathcal{O}_I^\dagger \mathcal{O}_D^\dagger}{\mapsto} \frac{1}{\sqrt{d_u}} \sum_{i=1}^{d_u} |u\rangle|v_i(u)\rangle = \mathcal{O}_W|u\rangle|0\rangle.\end{aligned}$$

To complete the proof, note that the index query operator $\mathcal{O}_I$ only requires $O(\log(d_u))$ sorted adjacency array queries, since the neighbours are sorted and this makes it possible to perform binary search for i such that $v_i(u) = v$. □

It follows that any quantum algorithm solving $\text{USTCON}_{\text{qw}}$ in T time and $S = \Omega(\log(n))$ space can solve $\text{USTCON}_{\text{s-arr}}$ in $\widetilde{O}(T)$ time and $O(S)$ space.

3.4 Time- and space-optimal quantum algorithm

In Section 3.4.2 and Section 3.4.3, we give an algorithm for $\text{USTCON}_{\text{qw}}$ that is optimal in both time and space. For completeness, we first give a time lower bound in Section 3.4.1.

3.4.1 Lower bound

The proof of the following lower bound follows the lines of the proof of an analogous lower bound for the strong connectivity problem described in [DHHM06]. The proof is via a reduction from PARITY.

3.4.1. PROBLEM (PARITY). *Given oracle access to a string $x \in \{0,1\}^n$ via $\mathcal{O}_x : |i\rangle|b\rangle \mapsto |i\rangle|b \oplus x_i\rangle$, return $\bigoplus_{i=0}^{n-1} x_i$.*

3.4.2. LEMMA ([BBC+01, FGGS98]). *The bounded error quantum query complexity of PARITY is $\Omega(n)$.*

We use Lemma 3.4.2 and a reduction from parity to show the following.

3.4.3. THEOREM. *The bounded error quantum query complexity of $\text{USTCON}_{\text{s-arr}}$ and $\text{USTCON}_{\text{qw}}$ is $\Omega(n)$.*

Proof:
We will reduce the PARITY problem to $\text{USTCON}_{\text{s-arr}}$. Since PARITY requires $\Omega(n)$ queries by Lemma 3.4.2, and the quantum walk oracle $\mathcal{O}_W$ can be implemented using $\widetilde{O}(1)$ sorted array queries by Lemma 3.3.3, this reduction will prove the statement of the theorem.

Let $x \in \{0,1\}^n$ be an input of PARITY. The corresponding output would be $\bigoplus_{i=0}^{n-1} x_i$. Given this, we need to build a $\text{USTCON}_{\text{s-arr}}$ input that can be queried with a constant number of queries to x. Consider an undirected graph $G = (V, E)$ defined as follows (see also Figure 3.2):

$$V = \{s = v_0, v'_0, v_1, v'_1, \ldots, v_{n-1}, v'_{n-1}, v_n, t = v'_n\}$$

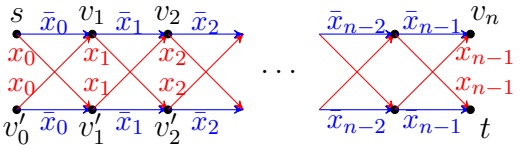

Figure 3.2: The parity graph. We include an edge labelled by "x_i" (in red) if and only if $x_i = 1$, and an edge labelled "$\bar{x}_i$" (in blue) if and only if $x_i = 0$, meaning that for each vertex we include exactly one of the two incoming edges, and exactly one of the two outgoing edges. The resulting graph has s and t connected if and only if $\textsc{parity}(x) = 1$.

$$E = \{\{v_i, v'_{i+1}\}, \{v'_i, v_{i+1}\} : i \in \{0, \dots, n-1\}, x_i = 1\}$$
$$\cup \{\{v_i, v_{i+1}\}, \{v'_i, v'_{i+1}\} : i \in \{0, \dots, n-1\}, x_i = 0\}.$$

In this setting, $\bigoplus_{i=0}^{n-1} x_i = 1$ if and only if s and t are connected in the graph G.

Next, we describe how to implement queries $\mathcal{O}_D$ and $\mathcal{O}_N$ to G as required by the $\textsc{ustcon}_{\text{s-arr}}$ problem, using queries to $\mathcal{O}_x$. Consider the following encoding of vertices of G. For $i \in \{0, \dots, n\}$, we let $v_i = (i, 0)$, and $v'_i = (i, 1)$. That is, for a vertex (i, b), $i \in \{0, \dots, n\}$ encodes the "column" and $b \in \{0, 1\}$ encodes the "row". Assume that the vertices are ordered lexicographically, i.e.

$$(i, b_i) < (j, b_j) \iff i < j \text{ or } i = j, b_i < b_j.$$

Queries to G are described according to this ordering.

1. Degree queries, $\mathcal{O}_D$, are trivial in this case as $d_{v_0} = d_{v'_0} = d_{v_n} = d_{v'_n} = 1$, and all other degrees are 2.

2. Since every vertex has degree at most 2, we explicitly describe neighbour queries, $\mathcal{O}_N$ for indices 1 and 2 such that the ordering assumption holds.

   - $\mathcal{O}_N : |i\rangle|b\rangle|1\rangle|0\rangle|0\rangle \mapsto |i\rangle|b\rangle|1\rangle|i-1\rangle|b\rangle \overset{\mathcal{O}_x}{\mapsto} |i\rangle|b\rangle|1\rangle|i-1\rangle|b \oplus x_{i-1}\rangle$, $\forall\, 0 < i \leq n$

   - $\mathcal{O}_N : |i\rangle|b\rangle|2\rangle|0\rangle|0\rangle \mapsto |i\rangle|b\rangle|2\rangle|i+1\rangle|b\rangle \overset{\mathcal{O}_x}{\mapsto} |i\rangle|b\rangle|2\rangle|i+1\rangle|b \oplus x_i\rangle$, $\forall\, 0 \leq i < n$

It can be seen from the formulas that queries to G can be implemented with a constant number of queries to the parity input x. This implies the $\Omega(n)$ lower bound in $\textsc{ustcon}_{\text{s-arr}}$.

For $\textsc{ustcon}_{\text{qw}}$, note that the graph has bounded degree, so by Theorem 3.3.3 we can simulate a query in this model using $O(1)$ queries in the sorted adjacency array model. This implies a similar $\Omega(n)$ lower bound for this model. □

3.4.2 Metropolis-Hastings walk

In this section, we consider an unweighted simple graph G. The algorithm that we propose involves a quantum walk on a modified weighted version of G that we call $G' = (V', E', W)$. We start by describing the construction of G' that was introduced in [Kos13, arXiv v2].

3.4.4. Definition (Metropolis-Hastings walk). For any graph $G = (V, E)$, the corresponding *Metropolis-Hastings walk* is the random walk on the weighted graph $G' = (V', E', W)$ defined as follows. For every $u \in V$, we include a corresponding vertex x_u in V'. In addition, for every edge $\{u, v\} \in E$, we add a new vertex $x_{u,v}$ that splits the edge into two new edges. Formally:

$$V' = \{x_u : u \in V\} \cup \{x_{u,v} : \{u, v\} \in E, u < v\}$$
$$E' = \{\{x_u, x_{u,v}\} : \{u, v\} \in E\}.$$

For every edge $\{x_u, x_{u,v}\} \in E'$, we define the edge weight $W_{x_u, x_{u,v}} = \frac{1}{d_u}$. These weights define transition probabilities for the random walk on G'.

The above has been called the *cautious walk* in [Kos13, arXiv v2], while *Metropolis-Hastings-type* walks are walks in which neighbours are sampled and accepted with some probability. Our terminology is motivated by the following observation. If we start with a vertex $u \in V$ and take two steps of the walk of Theorem 3.4.4, then we arrive at another vertex $v \in V$, which is either the same or a neighbour of u in G. The walk on G defined this way has the following alternative description: sample a uniformly random neighbour and accept it with probability $\frac{1/d_v}{1/d_u + 1/d_v} = \frac{1}{1 + d_v/d_u}$. This is precisely a random walk that falls into the Metropolis-Hastings framework, justifying our terminology. The precise choice of acceptance probabilities is sometimes called the *Glauber choice* in the literature (e.g., [LHP+20]). We note that a later version of [Kos13] uses another choice of Metropolis-Hastings walk, but of our purposes we find it convenient to stick to the walk as defined above.

While the hitting time of a random walk between two vertices in G may be as high as $O(n^3)$, in G' it is at most $O(n^2)$ [Kos13, Lemma 2 of arXiv v2]:

3.4.5. Lemma ([Kos13]). *Let $G = (V, E)$ be any unweighted graph, and G' the corresponding (weighted) Metropolis-Hastings graph as in Definition 3.4.4. For any $u, v \in V$ connected by a path, $\mathcal{H}_{u,v}(G') \leq 18n^2$.*

In order to apply Theorem 2.6.1 to G', we need to upper bound U, the cost of implementing the weighted quantum walk oracle. The oracle is defined as

$$U : |x\rangle|0\rangle \mapsto \sum_{y \in N(x)} \sqrt{P'_{xy}}|x\rangle|y\rangle, \quad \forall x \in V'$$

where P'_{xy} is the probability of walking from x to y defined by the edge weights.

3.4.6. LEMMA. *The weighted quantum walk oracle U for the Metropolis-Hastings walk G' can be implemented with $\widetilde{O}(1)$ degree queries $\mathcal{O}_D$, $\widetilde{O}(1)$ applications of the quantum walk operator $\mathcal{O}_W$ on the graph G, and $\widetilde{O}(1)$ additional gates.*

Therefore, U can be implemented with $\widetilde{O}(1)$ queries to G in the sorted adjacency array model, and $\widetilde{O}(1)$ additional gates.

Proof:
Note that the first statement implies the second one due to Lemma 3.3.3. Therefore, we will only prove the first statement. Consider the following encoding of the vertices of G'.

$$V' = \{(u,0) : u \in V\} \cup \{(u,v) : \{u,v\} \in E, u < v\} \subseteq V \times \Big(V \cup \{0\}\Big),$$

where 0 is a null symbol not contained in V. The first set of this union corresponds to original vertices of G and the second one corresponds to the added ones.

To implement U on $|x\rangle|0\rangle$, we first compute a bit in an ancilla register A that is $|0\rangle_A$ if $x = (u,0)$ for some u, and $|1\rangle_A$ otherwise. We will condition on this value.

First, conditioned on $|0\rangle_A$, our implementation proceeds as follows, for $u \in V$:

$$\begin{aligned}|x_u\rangle|0\rangle = |u,0\rangle|0\rangle &\overset{J\mathcal{O}_W}{\mapsto} \frac{1}{\sqrt{d_u}} \sum_{v \in N(u)} |u\rangle|v\rangle|u,v\rangle \\ &\overset{J'}{\mapsto} \frac{1}{\sqrt{d_u}} \sum_{v \in N(u)} |u,0\rangle|u,v\rangle = \frac{1}{\sqrt{d_u}} \sum_{v \in N(u)} |x_u\rangle|x_{u,v}\rangle = U|x_u\rangle|0\rangle\end{aligned}$$

where J is a unitary that acts as $|u,v\rangle|0\rangle \mapsto |u,v\rangle|u,v\rangle$, and J' as $|u,v\rangle|u,v\rangle \mapsto |u,0\rangle|u,v\rangle$, each of which can be implemented with $O(\log(n))$ controlled-NOT gates.

Next, conditioned on $|1\rangle_A$, our implementation proceeds as follows, for $\{u,v\} \in E$ with $u < v$, and $|0\rangle_{A'}$ a fresh ancilla:

$$\begin{aligned}|0\rangle_{A'}|x_{u,v}\rangle|0\rangle &= |0\rangle_{A'}|u,v\rangle|0\rangle \\ &\overset{1}{\mapsto} \left(\sqrt{\frac{1/d_u}{1/d_u + 1/d_v}}|0\rangle + \sqrt{\frac{1/d_v}{1/d_u + 1/d_v}}|1\rangle\right)_{A'} |u,v\rangle|0\rangle \\ &\overset{2}{\mapsto} \sqrt{\frac{1/d_u}{1/d_u + 1/d_v}}|0\rangle_{A'}|u,v\rangle|u\rangle + \sqrt{\frac{1/d_v}{1/d_u + 1/d_v}}|1\rangle_{A'}|u,v\rangle|v\rangle \\ &\overset{3}{\mapsto} |0\rangle_{A'}\left(\sqrt{\frac{1/d_u}{1/d_u + 1/d_v}}|u,v\rangle|u\rangle + \sqrt{\frac{1/d_v}{1/d_u + 1/d_v}}|u,v\rangle|v\rangle\right) \\ &= |0\rangle_{A'}\left(\sqrt{\frac{W_{x_{u,v},x_u}}{w(x_{u,v})}}|x_{u,v}\rangle|x_u\rangle + \sqrt{\frac{W_{x_{u,v},x_v}}{w(x_{u,v})}}|x_{u,v}\rangle|x_v\rangle\right)\end{aligned}$$

$$= |0\rangle_{A'} U |x_{u,v}\rangle |0\rangle,$$

where we use the following mappings:

1: Query degrees for u and v into a new ancilla register, perform the rotation controlled on the degrees (cf. [GR02]), and then uncompute the degrees ($O(1)$ degree queries to G).

2: Controlled on the first register, select one of the two vertices to copy into the last register ($O(\log(n))$ Toffoli gates).

3: Flip the bit in A' if the second vertex of $|u, v\rangle$ is the same as the one written in the third register ($O(\log(n))$ elementary gates).

To complete the proof, note that we can uncompute the bit in ancilla A, because the register containing $|x\rangle$ has not been changed. □

3.4.3 The algorithm

We can solve $\text{USTCON}_{\text{qw}}(G)$ using Algorithm 1. This leads to our main theorem of this section.

Algorithm 1 Quantum algorithm for $\text{USTCON}_{\text{qw}}$ with optimal time and space

Apply the algorithm from Theorem 2.6.1 to the Metropolis-Hastings walk P' with $M = \{t\}$, using Lemma 3.4.6 to implement the quantum walk oracle for G'. If the algorithm returns t, output "connected", and otherwise output "disconnected".

3.4.7. THEOREM. *There exists a $O(\log(n))$-space quantum algorithm that decides $\text{USTCON}_{\text{qw}}$ and $\text{USTCON}_{\text{s-arr}}$ with bounded one-sided error in $\widetilde{O}(n)$ gates and queries.*

Proof:
Let $V_s \subseteq V$ denote the connected component of s. If $t \in V_s$, the algorithm will output t with probability at least $2/3$, in which case our algorithm will output the correct answer, "connected". If $t \notin V_s$, then the algorithm will output an element of V_s with probability 1, in which case, our algorithm will output the correct answer "disconnected". This establishes correctness of Algorithm 1 with one-sided error.

To analyze the complexity, note that by Lemma 3.4.6 we have $\mathsf{U} = \widetilde{O}(1)$. For any $u \in V_s$, we can check if $u \in M$ by checking if $u = t$ in complexity $\mathsf{C} = O(\log(n)) = \widetilde{O}(1)$. To complete the analysis, we need only upper bound the commute time between s and t when $t \in V_s$. Since $\mathcal{C}_{s,t} = \mathcal{H}_{s,t} + \mathcal{H}_{t,s}$, by

Lemma 3.4.5, we have $\mathcal{C}_{s,t} \leq 36n^2 =: \mathcal{C}$. Thus, referring to Theorem 2.6.1, the complexity of our algorithm is:

$$\widetilde{O}\Big(\sqrt{\mathcal{C}\log(\mathcal{C})\log(\log(\mathcal{C}))}(\mathsf{C}+\mathsf{U})\Big) = \widetilde{O}(n).$$

□

3.5 Time-space tradeoff for bounded spectral gap

In this section we revisit the problem of undirected *st*-connectivity in the setting where one is given a lower bound on the spectral gap of the random walk. As discussed in Section 2.5, such a bound is tightly related to the mixing time of the walk. We will give a quantum algorithm that exhibits a nontrivial time-space tradeoff in this setting.

Our discussion will be general and apply to random walks on weighted graphs as defined in Eq. (2.5.1). This is useful since the spectral gaps and mixing times of random walks on G with different edge weights are in general *not* comparable. E.g., on the lollipop graph (an n-vertex clique connected to an $O(n)$-vertex path) the mixing time of the unweighted random walk is $\Theta(n^3)$ [BW90], while it is $O(n^2)$ for the Metropolis-Hastings walk.[3] On the other hand, on an n-vertex star graph the unweighted random walk has mixing time $O(1)$ while the Metropolis-Hastings walk has mixing time $\Theta(n)$. Thus, while the specific edge weights do not affect whether s and t are connected, they do impact the algorithm. Throughout this section, we assume some fixed edge weights are given, and we do not try to optimize for "good" edge weights. More specifically, we assume access to a *weighted quantum walk oracle* that for every vertex outputs a superposition of its neighbours, with squared amplitudes proportional to the edge weights:

$$\mathcal{O}_W : |u\rangle|0\rangle \mapsto \sum_{v\in N(u)} \sqrt{\frac{W_{u,v}}{w_u}}|u\rangle|v\rangle = \sum_{v\in N(u)} \sqrt{P_{u,v}}|u\rangle|v\rangle \qquad \forall u \in V$$

Moreover, we assume access to the weighted vertex degrees w_u and that these degrees are of bounded absolute value for some $\mathrm{poly}(n)$ bound. This will allow us to generate the state $|\pi_{V'}\rangle$ for any subset $V' \subseteq V$ stored in QCRAM.

3.5.1. Problem ($\mathrm{USTCON}_{\mathrm{qw},\delta}$). *Given access to an undirected weighted graph via the quantum walk oracle $\mathcal{O}_W$, two vertices $s, t \in V$, and the promise that either s*

[3] This follows from the $O(n^2)$ upper bound on the maximum hitting time of the Metropolis-Hastings walk (Theorem 3.4.5), and the fact that the maximum hitting time upper bounds the mixing time [LPW17, Lemma 10.2].

and t are disconnected or the spectral gap of the transition matrix of the walk is at least some $\delta > 0$, decide which is the case.

Our main result of this section is the following:

3.5.2. THEOREM. *Fix $\delta \geq 0$. Let $\mathcal{G}_n$ be a family of undirected weighted graphs $G = (V, E, W)$ with $n = |V|$, such that $\gamma_\star(G) \geq \delta$ whenever s and t are connected. Then for any $S = \Omega(\log(n))$, there is a quantum algorithm that decides $\textsc{ustcon}_{\mathrm{qw},\delta}$ on $\mathcal{G}_n$ with bounded error in $O(S)$ space – of which $O(\log(n/\delta))$ is quantum memory, and the remainder is QCRAM – and $T = \widetilde{O}(\frac{S}{\delta}\log(\frac{1}{\pi_{\min}}) + \sqrt{\frac{n}{\delta S}})$ queries to $\mathcal{O}_W$, elementary gates, and QCRAM queries.*

Note that we can assume $\delta \geq 1/n$. If $\delta < 1/n$ then $T \approx S/\delta$. There is no time-space tradeoff, and it is always faster to run the Metropolis-Hastings algorithm (Algorithm 1).

The algorithm is stated below as Algorithm 2. It consists of three stages. We fix some parameter p, which denotes the number of "pebbles", or vertices the algorithm will keep track of (so $S = O(p\log(n))$). First, we run $O(p)$ classical random walks starting from s, each of length $\ell = O\Big(\frac{1}{\delta}\log(\frac{n}{p\pi_{\min}})\Big)$. This allows us to sample a set L of $O(p)$ points from V_s, the connected component of s (the big-O notation suppresses a universal constant that is given in the proof). Since ℓ is at least the mixing time of G (see Theorem 2.5.1), assuming s and t are connected, each point is sampled (approximately) from π. We do the same from t to get a random subset $M \subseteq V_t$ connected to t.

Next, we use L and M to prepare (up to some error) the states $|\pi_{V_s}\rangle$ and $|\pi_{V_t}\rangle$, using *inverse quantum walk search*, which we describe in more detail in Section 3.5.2. If s and t are in the same connected component, then $|\pi_{V_s}\rangle = |\pi_{V_t}\rangle$, and otherwise, the states are orthogonal. The final step is to distinguish these two cases using a SWAP test. This roughly follows an earlier approach in [Ape19], the main difference being that we sample the sets L and M using a random walk (which allows us to exploit the gap promise), while in [Ape19] the sets are constructed using a breadth-first search.

If we specialize Theorem 3.5.2 to the unweighted graph case, we get the following corollary.

3.5.3. COROLLARY. *For any $S > 0$, there is a quantum algorithm that solves $\textsc{ustcon}_{\text{s-arr}}$ with a promise $\delta > 0$ on the random walk spectral gap using space S and time $T \in \widetilde{O}(S/\delta + \sqrt{n/(\delta S)})$.*

An analogous result holds for the Metropolis-Hastings walk described in Section 3.4.2 given a promise on its spectral gap, since we showed that the corresponding quantum walk oracle can also be efficiently implemented.

In the remainder of this section we will analyze each stage of Algorithm 2.

Algorithm 2 Quantum algorithm for $\text{USTCON}_{\text{qw}}$ with a tradeoff

`Seed set:` Run $O(p)$ classical random walks from s and $O(p)$ classical random walks from t, each for $O(\frac{1}{\delta}\log(\frac{n}{p\pi_{\min}}))$ steps. Let L and M denote the respective sets of endpoints, without duplicates. If $L \cap M \neq \emptyset$, return "connected".

`State preparation:` Run inverse quantum walk search from $|\pi_L\rangle$ and $|\pi_M\rangle$ for time $\widetilde{O}\left(\sqrt{\frac{n}{\delta p}}\right)$ to prepare $|\pi_{V_s}\rangle$ and $|\pi_{V_t}\rangle$, respectively, to precision 1/8.

`SWAP test:` Do a SWAP test on the resulting states. If the test returns "0", return "connected", otherwise return "disconnected".

3.5.1 Analysis of step 1: seed set

Recall that the first stage results in random sets $L = \{x_1, \ldots, x_{cp}\}$ and $M = \{y_1, \ldots, y_{cp}\}$, where $x_1, \ldots, y_{cp}$ are the endpoints of independent random walks starting at s or t, respectively, and $c > 0$ is some universal constant that we will choose later. Since we run those random walks for $O(\frac{1}{\delta}\log(\frac{n}{p\pi_{\min}}))$ steps, by Theorem 2.5.1 it follows that the x_j are independent samples drawn from a distribution $\tilde{\pi}$ such that

$$\|\tilde{\pi} - \pi\|_{\text{TV}} \leq \frac{p}{8n},$$

where π is the stationary distribution on V_s. If s and t are connected then $V_s = V_t$ and the samples y_j are similarly drawn from a distribution that is $p/(8n)$-close to π. Here we prove that this implies lower bounds on the stationary measure of the sets L and M.

3.5.4. Proposition. *There exists a universal constant $c > 0$ such that the following holds. Let $p \in [n]$ and assume $L \subseteq V$ is a random set obtained by sampling cp independent elements from a distribution $\tilde{\pi}$ such that $\|\tilde{\pi} - \pi\|_{\text{TV}} \leq \frac{p}{8n}$ (and removing duplicates). Then, $\mathbf{Pr}(\pi(L) \geq \frac{p}{8n}) \geq \frac{9}{10}$.*

The proof of the proposition uses the following lemma, which formalizes the intuition that adding a random element to a low-probability subset should increase the probability.

3.5.5. Lemma. *Let V be a set of cardinality n, $A \subseteq V$ an arbitrary fixed subset, and let b be drawn at random from an arbitrary distribution σ on V. Then:*

$$\mathbf{Pr}\left(\sigma(A \cup \{b\}) > \sigma(A) + \frac{1 - \sigma(A)}{2n}\right) \geq \frac{1 - \sigma(A)}{2}.$$

Proof:
Say b is *bad* if $b \in A$ or $\sigma(b) \leq \frac{1-\sigma(A)}{2n}$, and *good* otherwise. Then

$$\mathbf{Pr}\left(\sigma(A \cup \{b\}) > \sigma(A) + \frac{1-\sigma(A)}{2n}\right) = \mathbf{Pr}(b \text{ is good}) = 1 - \mathbf{Pr}(b \text{ is bad}).$$

We can compute:

$$\begin{aligned}\mathbf{Pr}(b \text{ is bad}) &= \sigma\left(A \cup \left\{x \in V : \sigma(x) \leq \frac{1-\sigma(A)}{2n}\right\}\right) \\ &\leq \sigma(A) + n \cdot \frac{1-\sigma(A)}{2n} \\ &= \frac{1+\sigma(A)}{2},\end{aligned}$$

from which the claim follows. □

Proof of Theorem 3.5.4:
Let $x_1, x_2, \ldots$ denote samples drawn independently at random from $\tilde{\pi}$. For any integer $T \geq 1$, define $L_T := \{x_1, \ldots, x_T\}$ as the set consisting of the first T samples (with duplicates removed), as well as $L_0 := \emptyset$. We say that the T-th sample is a *success* if

$$\tilde{\pi}(L_{T-1}) \geq \frac{1}{2} \quad \text{or} \quad \tilde{\pi}(L_T) \geq \tilde{\pi}(L_{T-1}) + \frac{1}{4n},$$

and a *failure* otherwise. Let T_j denote the index of the j-th success, with $T_0 := 0$. Then, clearly,

$$\tilde{\pi}(L_{T_p}) \geq \min\left\{\frac{1}{2}, \frac{p}{4n}\right\} = \frac{p}{4n} \quad \text{and hence} \quad \pi(L_{T_p}) \geq \tilde{\pi}(L_{T_p}) - \frac{p}{8n} \geq \frac{p}{8n}.$$

On the other hand, note that by Theorem 3.5.5, the T-th sample is a success with probability at least $\frac{1}{4}$, even if we condition on all prior samples. In particular, the probability that there are k failures in a row is at most $(\frac{3}{4})^k$. Therefore,

$$\mathbf{E}[T_j - T_{j-1}] = \sum_{k=1}^{\infty} k \,\mathbf{Pr}(T_j = T_{j-1} + k) \leq \sum_{k=1}^{\infty} k\left(\frac{3}{4}\right)^k = 12$$

and hence, by Markov's inequality,

$$\mathbf{Pr}(T_p > cp) \leq \frac{1}{cp}\,\mathbf{E}[T_p] = \frac{1}{cp}\sum_{j=1}^{p}\mathbf{E}[T_j - T_{j-1}] \leq \frac{12p}{cp} = \frac{1}{10}$$

provided we choose $c := 120$. Since the random set L in the statement of the lemma is defined by taking cp many samples, we obtain that

$$\mathbf{Pr}\left(\pi(L) \geq \frac{p}{8n}\right) \geq \mathbf{Pr}(T_p \leq cp) \geq 1 - \frac{1}{10} = \frac{9}{10}.$$

□

3.5.2 Analysis of step 2: state preparation

Now we turn to the analysis of the quantum walk search routine in step 2 of the algorithm. We rely on the following proposition from [Ape19], which formalizes the idea of "inverse quantum walk search".[4]

3.5.6. Proposition ([Ape19, Proposition 1]). *Consider a subset $A \subseteq V$ of a (connected) graph G, and let δ be a lower bound on the spectral gap of a random walk P on G with stationary distribution π. From $|\pi_A\rangle$, we can generate a state $|\tilde{\pi}\rangle = |\pi\rangle + |\Gamma\rangle$ with $\||\Gamma\rangle\|_2 \leq \epsilon$ using an expected number of calls to the weighted quantum walk oracle*

$$O\left(\frac{1}{\sqrt{\pi(A)\delta}} \log\left(\frac{1}{\pi(A)\epsilon}\right)\right),$$

and $O(1/\sqrt{\pi(A)\delta})$ reflections around $|\pi_A\rangle$. The algorithm uses space logarithmic in n, $1/\delta$, $1/\pi(A)$ and $1/\epsilon$.

The proposition implies the following.

3.5.7. Claim. *Step 2 of Algorithm 2 prepares $1/8$-approximations of $|\pi_{V_s}\rangle$ and $|\pi_{V_t}\rangle$ with probability at least $9/10$ in time complexity $O\left(\sqrt{\frac{n}{p\delta}} \log\left(\frac{n}{p}\right)\right)$.*

Proof:
First, step 2 prepares superpositions $|\pi_L\rangle$ and $|\pi_M\rangle$. Applying Lemma 3.2.1 with $l = n$ (upper bound on the set size) and $k = \log(n)$ (number of bits necessary to write down a vertex index) and the assumption that weighted vertex degrees w_u are of bounded absolute value for some $\mathrm{poly}(n)$ bound, shows that this superpositions preparation can be done in $\widetilde{O}(1)$ time. By Theorem 3.5.4, we have that $\pi(L), \pi(M) \geq p/(8n)$ with probability at least $9/10$. By Theorem 3.5.6 and our

[4] [Ape19, Proposition 1] only proves the proposition for simple random walks, however it trivially extends to random walks on weighted graphs.

preceding remark, we can then prepare 1/8-approximations of $|\pi_{V_s}\rangle$ and $|\pi_{V_t}\rangle$ in time

$$O\left(\sqrt{\frac{n}{p\delta}}\log\left(\frac{n}{p}\right)\right).$$

□

3.5.3 Analysis of step 3: SWAP test

In the last step of our algorithm, we wish to decide whether $|\pi_{V_s}\rangle = |\pi_{V_t}\rangle$ or whether they are orthogonal. For this we use the SWAP test.

3.5.8. Claim. *Step 3 of Algorithm 2 decides whether $|\pi_{V_s}\rangle = |\pi_{V_t}\rangle$ or whether they are orthogonal with constant probability in time $\widetilde{O}(1)$.*

Proof:
Using a single copy of two states $|\psi\rangle$ and $|\psi'\rangle$, and $O(\log(n))$ additional gates, the SWAP test returns "0" with probability $(1 + |\langle\psi|\psi'\rangle|^2)/2$ and "1" with probability $(1 - |\langle\psi|\psi'\rangle|^2)/2$ [ATS03, Claim 3.1].

By Claim 3.5.7, in step 2 we prepared states $|\tilde{\pi}_{V_s}\rangle = |\pi_{V_s}\rangle + |\Gamma_L\rangle$ and $|\tilde{\pi}_{V_t}\rangle = |\pi_{V_t}\rangle + |\Gamma_M\rangle$ such that $\||\Gamma_L\rangle\|_2, \||\Gamma_M\rangle\|_2 \leq 1/8$ with probability 9/10. By a triangle inequality, this implies that

$$\left||\langle\tilde{\pi}_{V_s}|\tilde{\pi}_{V_t}\rangle| - |\langle\pi_{V_s}|\pi_{V_t}\rangle|\right| < 1/3,$$

and so $|\langle\tilde{\pi}_{V_s}|\tilde{\pi}_{V_t}\rangle| > 2/3$ if s and t are connected, but $|\langle\tilde{\pi}_{V_s}|\tilde{\pi}_{V_t}\rangle| < 1/3$ otherwise. The SWAP test distinguishes these cases with constant probability. □

3.5.4 Proof of Theorem 3.5.2

In this section, we prove Theorem 3.5.2, which we restate here for convenience.

3.5.2. Theorem. *Fix $\delta \geq 0$. Let $\mathcal{G}_n$ be a family of undirected weighted graphs $G = (V, E, W)$ with $n = |V|$, such that $\gamma_\star(G) \geq \delta$ whenever s and t are connected. Then for any $S = \Omega(\log(n))$, there is a quantum algorithm that decides $\textsc{ustcon}_{\mathrm{qw},\delta}$ on $\mathcal{G}_n$ with bounded error in $O(S)$ space – of which $O(\log(n/\delta))$ is quantum memory, and the remainder is QCRAM – and $T = \widetilde{O}(\frac{S}{\delta}\log(\frac{1}{\pi_{\min}}) + \sqrt{\frac{n}{\delta S}})$ queries to $\mathcal{O}_W$, elementary gates, and QCRAM queries.*

Proof:
We first analyze the space complexity of Algorithm 2. Step 1 is purely classical, and uses $O(p\log(n))$ space to store the $O(p)$ vertices in L and M, with each random walk using $O(\log(n))$ space. We can implement step 2 using the algorithm referred to in Theorem 3.5.6 using $O(\log(n))$ qubits of space, but this requires that the $O(p\log(n))$ classical space used to store L and M in step 1 is QCRAM. Finally, step 3 just uses $O(\log(n))$ quantum space. Thus, the claimed space complexity follows if we set $S = p\log(n)$.

Next, we analyze the time complexity. Every random walk of step 1 adds $O\Big(\frac{1}{\delta}\log(\frac{n}{p\pi_{\min}})\Big)$ to the time complexity. The total number of walks is $O(p)$. Checking whether $L \cap M = \emptyset$ is $O(p)$ as this is the total number of samples. Hence, the overall complexity of the first step is $\widetilde{O}\Big(\frac{p}{\delta}\log(\frac{1}{\pi_{\min}})\Big)$. By Claim 3.5.7, the complexity of step 2 is $O\Big(\sqrt{\frac{n}{p\delta}}\log\Big(\frac{n}{p}\Big)\Big)$. Finally, the SWAP test in step 3 uses only $O(\log(n))$ gates, since the states being compared are $O(\log(n))$-qubit states. Hence, the total time complexity of Algorithm 2 is

$$T = \widetilde{O}\left(\frac{p}{\delta}\log\left(\frac{1}{\pi_{\min}}\right) + \sqrt{\frac{n}{\delta p}}\right) = \widetilde{O}\left(\frac{S}{\delta}\log\left(\frac{1}{\pi_{\min}}\right) + \sqrt{\frac{n}{\delta S}}\right),$$

since $S = \widetilde{O}(p)$.

Finally, for the correctness of the algorithm, by Claim 3.5.8, Algorithm 2 distinguishes between the case where $|\pi_{V_s}\rangle$ and $|\pi_{V_t}\rangle$ are equal and the case where they are orthogonal with bounded error. If $V_s = V_t$ (i.e. s and t are connected) then the states are equal, and if $V_s \cap V_t = \emptyset$ (i.e. s and t are not connected) then they are orthogonal. □

Chapter 4

Subspace graphs

We introduce an object called a *subspace graph* that formalizes the technique of multidimensional quantum walks. Composing subspace graphs allows one to seamlessly combine quantum and classical reasoning, keeping a classical structure in mind, while abstracting quantum parts into subgraphs with simple boundaries as needed. As an example, we show how to combine a *switching network* with arbitrary quantum subroutines, to compute a composed function.

This chapter is based on the paper "Multidimensional Quantum Walks, Recursion, and Quantum Divide & Conquer" by Stacey Jeffery and Galina Pass [JP25a].

4.1 Introduction

There are a number of graphical ways of reasoning about how the steps or subroutines of a classical algorithm fit together. For example, it is natural to think of a (randomized) classical algorithm as a (randomized) decision tree (or branching program), where different paths are chosen depending on the input, as well as random choices made by the algorithm. A deterministic algorithm gives rise to a computation *path*, a randomized algorithm to a computation *tree*. The edges of a path or tree, representing steps of computation, might be implemented by some subroutine that is also realized by a path (or tree) – we can abstract the subroutine's details by viewing it as an edge, or zoom in and see those details, as convenient. More generally, we often think of a classical randomized algorithm as a random walk on a (possibly directed) graph, where there may be multiple parallel paths from point a to point b, with the cost of getting from a to b being derived from the expected length of these paths.

This picture appears to break down for quantum algorithms, at least in the standard circuit model. A quantum circuit can be thought of as a path, with edges representing its steps, but it is not immediately clear how to augment this reasoning with subroutines. Consider calling subroutines with varying time complexities $\{T_i\}_i$ in superposition. Even if the subroutines are all classical deterministic, in the standard quantum circuit model, we tend to incur a cost of $\max_i T_i$ if we call a superposition of subroutines, since we must wait for the slowest subroutine to finish before we can apply the next step of the computation. This problem was addressed in [Jef22], where the technique of multidimensional quantum walks [JZ23] was used to show how to get an average in place of a max in several settings where a quantum algorithm calls subroutines in superposition: a general setting, as well as the setting of quantum walks. The intuition behind this work is that a quantum walk does keep the classical intuition of parallel paths representing a superposition of possible computations, and *any* quantum algorithm can be viewed as some sort of a quantum walk on a simple underlying graph (something like a path), but with some additional structure associated with it.

Multidimensional quantum walks, which we study more formally in this chapter as an object than has been done previously, are valuable as a way of combining quantum and classical reasoning. A quantum algorithm can be abstracted as a graph with perhaps complicated internal structure, but a simple *boundary* with an "in" and an "out" terminal, that can be seamlessly hooked into other graph-like structures, perhaps representing simple classical reasoning, such as a quantum random walk, or perhaps with their own complicated very quantum parts.

Subspace graphs While [JZ23] and [Jef22] use similar techniques, it is not formally defined what a *multidimensional quantum walk* is. We formally define

an object called a *subspace graph* (Definition 4.3.1) that abstracts the structures in [JZ23] and [Jef22], as well as some other previous algorithmic techniques. A subspace graph is simply a graph with some subspaces associated with each edge and vertex, where the structure of the graph constrains how the spaces can overlap. Defining what we mean, precisely, by a multidimensional quantum walk (i.e. subspace graph) is the first step to developing a general theory of recursive constructions of subspace graphs.

The recursive structure of subspace graphs is useful for composing quantum algorithms, but as a design tool, it is also convenient to be able to view a subspace graph in varying levels of abstraction. We can "zoom out" and view a complicated process as just a special "edge", or zoom in on that edge and understand its structure as an involved graph with additional structure.

We cannot hope to be able to understand all quantum algorithms using purely classical ideas – quantum computing is *not* classical computing. But perhaps the next best thing is a way to seamlessly combine classical and quantum ideas, extending the classical intuition to its limits, and then employing quantum reasoning when needed, but with the possibility of abstracting out from it when needed as well, using a fully quantum form of abstraction.

In this work, we consider one specific kind of composition of subspace graphs called *switch composition* – another type is implicit in [Jef22] – but we would like to emphasize the potential for more general types of recursion, which we leave for future work.

Switching networks A switching network is a graph with Boolean variables associated with the edges that can switch the edges on or off. Originally used to model certain hardware systems, including automatic telephone exchanges, and industrial control equipment [Sha38], a switching network has an associated function f that is 1 if and only if two special vertices, s and t, are connected by a path of "on" edges. Shannon [Sha38, Sha49] showed that series-parallel switching networks are equivalent to Boolean formulas, and Lee [Lee59] showed that switching networks can model branching programs. These theoretical results have given this model a place in classical complexity theory, where they can be used to study classical space complexity and circuit depth (see [Pot15]).

Quantum algorithms for evaluating switching networks[1] were given in [JK17], using a span program construction based heavily on [BR12]. Tight analysis of these span programs for any switching network was completed in [JJKP18].

From [JK17, JJKP18], we get a quantum algorithm for evaluating any switching

[1]Switching networks have never been directly referred to in prior work on quantum algorithms, as far as we are aware, but the "st-connectivity problems" referred to in [JK17] are, in fact, switching networks.

network when we have query access to the edge variables. Let $\mathcal{R}_{s,t}(G(x))$ be the effective resistance (see Definition 2.4.5) in the subgraph of "on" edges whenever $f(x) = 1$, and whenever $f(x) = 0$, let $F_x \subseteq E$ be the minimum weight st-cut-set (see Definition 2.4.3) consisting of only edges that are "off". Then the time complexity of evaluating the switching network is

$$\sqrt{\max_{x\in f^{-1}(1)} \mathcal{R}_{s,t}(G(x)) \max_{x\in f^{-1}(0)} \sum_{e\in F_x} \mathsf{w}_e},$$

assuming we can implement a certain reflection related to the particular switching network in unit time. From this it follows that if we can query the variable associated with an edge e in time T_e, we can evaluate the switching network in time

$$\sqrt{\max_{x\in f^{-1}(1)} \mathcal{R}_{s,t}(G(x)) \max_{x\in f^{-1}(0)} \sum_{e\in F_x} \mathsf{w}_e}. \max_{e\in E} T_e.$$

Here we improve this to (see Theorem 4.4.10):

$$\sqrt{\max_{x\in f^{-1}(1)} \mathcal{R}_{s,t}(G(x)) \max_{x\in f^{-1}(0)} \sum_{e\in F_x} \mathsf{w}_e T_e^2}.$$

This is analogous to results of [Jef22], which showed a similar statement, but for quantum walks rather than switching networks[2]. It is also similar in flavour to the results of [Cor23, Chapter 7.2], where *span programs* are used to evaluate the edge variables of a switching network. Because switching networks perfectly model Boolean formulas, without the constant overhead we get when we convert to a quantum algorithm, they can be used as a building block for our divide-&-conquer results.

Model of computation We work in the same model as [JZ23], where we allow not only arbitrary quantum gates, but also assume subroutines are given via access to a unitary that applies the subroutine's t-th gate controlled on the value t in some time register. This is possible, for example, with quantum random access gates. See Section 4.2 for details.

Related and future work We can compare the model of transducers [BJY24] to the framework of subspace graphs introduced here. These models are related, in that they are both good for analyzing the time complexity of composed quantum algorithms. A subspace graph gives rise to a unitary that is the product of two reflections, and this is a transducer, as discussed in [BJY24, Section 3.2]. Compared with the model of subspace graphs, transducers are more general, and cleaner, in

[2]Both this result, and [Jef22], include *variable-time quantum search* [Amb10] as a special case.

particular how they are implemented as quantum algorithms. It is not known whether subspace graphs can reproduce the composition results of transducers in terms of error scaling. However, we feel that subspace graphs, with their additional structure, are still useful for designing and reasoning about quantum algorithms, especially those in which there is some classical intuition to hold onto. It seems likely that, once designed in the framework of subspace graphs, an algorithmic construction could be converted to the transducer framework to achieve any advantage that is missing for subspace graphs. We leave concrete exploration of the connection for future work.

Aside from the directions already mentioned, the most interesting future direction is to study more general compositions of subspace graphs, and see to what extent we can employ classical reasoning to quantum algorithms, and to what extent this (hopefully gracefully) breaks down.

Applications Subspace graphs play a central role in several later developments of this thesis. In Chapter 5, we employ switching networks to develop a framework for time-efficient quantum divide & conquer, which we apply to obtain a quadratic speedup over Savitch's algorithm for directed *st*-connectivity while maintaining the same space complexity. In Chapter 6, the switching network framework is used to derive the first quantum time-space tradeoff for directed *st*-connectivity. Finally, in Chapter 7, we use generic subspace graphs to design a quantum algorithm for the gauge problem.

4.2 Model of computation

We will be talking about combining quantum subroutines, and here we describe the model of access for the subroutines that we will assume. A quantum algorithm or subroutine is a sequence of unitaries $U_1, \dots, U_T$ acting on some common space H. Given a set of subroutines $\{(U_1^i, \dots, U_{T_i}^i)\}_{i\in\mathcal{I}}$, all acting on a space H, but possibly running in different times, $T_i \leq T_{\max}$, we will assume the operator $\sum_{i\in\mathcal{I}} |i\rangle\langle i| \otimes \sum_{t=1}^{T_{\max}} |t\rangle\langle t| \otimes U_t^i$ can be implemented in "unit cost". We use "unit cost" to describe a cost we are willing to accept as a multiplicative factor on all complexities (for example, $O(1)$ or polylogarithmic in some natural variable). This assumption is only reasonable if the unitaries each have unit cost. It does not hold for strict gate complexity when $U_1^i, \dots, U_{T_{\max}}^i$ are arbitrary gates, but it does hold in the *fully quantum* QRAM model if $U_1^i, \dots, U_{T_{\max}}^i$ are stored as a list of gates in classical memory to which a quantum computer has read-only superposition access (sometimes referred to as "QRAM" in previous literature), or if they satisfy certain uniformity conditions (see [JZ23, Section 2.2]).

4.3 Subspace graphs for multidimensional quantum walks

4.3.1 Subspace graphs

Multidimensional quantum walks were introduced as such in [JZ23], although they generalize various quantum algorithms that have appeared previously, including [Sze04, RS12, Bel13, BCJ+13, IJ19]. We wish to consider very general kinds of composition of multidimensional quantum walks, and in order to be clear about the precise types of objects we are composing, we give a more formal definition than has appeared previously.

4.3.1. Definition (Subspace graph). A *subspace graph* consists of a (undirected) graph $G = (V, E)$, a boundary $B \subseteq V$, and the following subspaces of a space $H = H_G$:

Edge and boundary spaces We assume H can be decomposed into a direct sum of spaces as follows: $H = \bigoplus_{e \in E} \Xi_e \oplus \bigoplus_{u \in B} \Xi_u$.

Edge and boundary subspaces For each $e \in E \cup B$, let $\Xi_e^{\mathcal{A}}$ and $\Xi_e^{\mathcal{B}}$ be subspaces of Ξ_e. These need not be orthogonal, and they may each be $\{0\}$, all of Ξ_e, or something in between.

Vertex spaces and boundary space For each $u \in V$, let $\mathcal{V}_u \subseteq \bigoplus_{e \in E(u)} \Xi_e$ be pairwise orthogonal spaces. Let $\mathcal{V}_B \subseteq \bigoplus_{u \in B} \Xi_u$.

Then we define

$$\mathcal{A}_G = \bigoplus_{e \in E \cup B} \Xi_e^{\mathcal{A}} \text{ and } \mathcal{B}_G = \bigoplus_{u \in V} \mathcal{V}_u + \mathcal{V}_B + \bigoplus_{e \in E \cup B} \Xi_e^{\mathcal{B}}.$$

We will refer to a subspace graph as simply G, with the associated spaces implicit. Let us give some intuition for the meaning of the graph structure in a subspace graph. We will shortly see (in Section 4.3.2) that there is a quantum algorithm associated with a subspace graph, which alternatively applies reflections around $\mathcal{A}_G$ and $\mathcal{B}_G$ to some initial state $|\psi_0\rangle$ to distinguish between the cases:

Negative case: $|\psi_0\rangle \in \mathcal{A}_G + \mathcal{B}_G$

Positive case: $|\psi_0\rangle$ has a (large) component, called a *positive witness* in $(\mathcal{A}_G + \mathcal{B}_G)^\perp = \mathcal{A}_G^\perp \cap \mathcal{B}_G^\perp$

Then we can see all the vectors in $\mathcal{A}_G$ and $\mathcal{B}_G$ as linear constraints on the initial state – it must have a large component orthogonal to all of them in the positive case.

$\mathcal{A}_G$ and $\mathcal{B}_G$ are each decomposed into sums of subspaces, representing subsets of constraints. The graph structure of G restricts how the different sets of constraints are allowed to overlap – the constraints associated with vertex v only overlap constraints associated with edges that are incident to v. This graph structure can then be used to help analyse the algorithm. For example, a positive witness is related to a *flow* on G.

4.3.2. Remark. In [JZ23] and [Jef22], spaces are only defined for each vertex, and the spaces associated with a pair of vertices must be orthogonal unless u and v are adjacent in G (in the words of [Jef22], G must be an *overlap graph* of the spaces). Here we have found it convenient to assume that G has been augmented by putting a vertex in the middle of each edge, which is why we have a space for each vertex of G, and for each edge of G. This ensures, for example, that the graph is bipartite, which is required in [JZ23] and [Jef22]. However, the important intuition about subspace graphs is that the locality structure of the graph constrains how the spaces are allowed to overlap.

We fix some notation for talking about subspace graphs. For any $e \in E \cup B$, we let Π_e, $\Pi_e^{\mathcal{A}}$ and $\Pi_e^{\mathcal{B}}$ denote orthogonal projectors onto Ξ_e, $\Xi_e^{\mathcal{A}}$, and $\Xi_e^{\mathcal{B}}$, respectively. For any set $F \subseteq E \cup B$, we let $\Xi_F = \bigoplus_{e \in F} \Xi_e$, $\Xi_F^{\mathcal{A}} = \bigoplus_{e \in F} \Xi_e^{\mathcal{A}}$, and similarly for $\Xi_F^{\mathcal{B}}$. We let Π_F, $\Pi_F^{\mathcal{A}}$ and $\Pi_F^{\mathcal{B}}$ denote the orthogonal projectors onto Ξ_F, $\Xi_F^{\mathcal{A}}$ and $\Xi_F^{\mathcal{B}}$, respectively.

We will allow subspace graphs to implicitly depend on some input, and associate them with a computation. Specifically, if we fix some initial state $|\psi_0\rangle \in H_G$, then this, along with $\mathcal{A}_G$ and $\mathcal{B}_G$, define a phase estimation algorithm (Section 2.7) that decides if $|\psi_0\rangle \in \mathcal{A}_G + \mathcal{B}_G$. With this in mind, we say a subspace graph that depends on some input x, *computes a function* f (with respect to $|\psi_0\rangle$, which should be clear from context) if $f(x) = 0$ if and only if $|\psi_0\rangle \in \mathcal{A}_G + \mathcal{B}_G$. We make this formal in Section 4.3.2.

One motivating example from which we derive intuition is the special case of a quantum walk on a graph. We will see other examples in later sections, and our results will mainly be based on these, but we describe how quantum walks fit into this picture as an illustration. First, we define a special case for $\mathcal{V}_u$. While the definition of subspace graph is independent of any weights of the graph G, we can always assume G has associated edge weights from which we derive extra structure.

4.3.3. Definition (Simple vertex). Fix a subspace graph G with associated edge weights $\{\mathsf{w}_e\}_{e \in E}$. For $u \in V$, define

$$|\psi_\star(u)\rangle := \sum_{e \in E^{\rightarrow}(u)} \sqrt{\mathsf{w}_e}\,|{\rightarrow}, e\rangle + \sum_{e \in E^{\leftarrow}(u)} \sqrt{\mathsf{w}_e}\,|{\leftarrow}, e\rangle.$$

A vertex $u \in V$ is *simple* if for all $e \in E^{\rightarrow}(u)$, $|\rightarrow, e\rangle \in \Xi_e$, for all $e \in E^{\leftarrow}(u)$, $|\leftarrow, e\rangle \in \Xi_e$, and if $u \in V \setminus B$ $\mathcal{V}_u = \text{span}\{|\psi_\star(u)\rangle\}$; and if $u \in B$, either $|\rightarrow, u\rangle \in \Xi_u$ and $\mathcal{V}_u = \text{span}\{|\rightarrow, u\rangle + |\psi_\star(u)\rangle\}$ or $|\leftarrow, u\rangle \in \Xi_u$ and $\mathcal{V}_u = \text{span}\{|\leftarrow, u\rangle + |\psi_\star(u)\rangle\}$.

In our applications, we generally have two degrees of freedom associated with each edge, which we can think of as having subdivided each edge into two edges (each associated with a single degree of freedom). When we subdivide edge e by putting a vertex in the middle of it - call that vertex e - it now has two incident edges going into it: $|\rightarrow, e\rangle$ and $|\leftarrow, e\rangle$. To make the simple vertex definition agree with the orientation on the edges, we let a star state $|\psi_\star(u)\rangle$ overlap $|\rightarrow, e\rangle$ if e is an outgoing edge of u, and $|\leftarrow, e\rangle$ if e is an incoming edge of u.
To motivate this definition, consider the following example.

4.3.4. Example (Quantum walk on a graph). A subspace graph G, with associated edge weights, in which

1. all vertices are simple,
2. $\Xi^{\mathcal{B}}_{E \cup B} = \{0\}$,
3. for all $e \in E$, $\Xi^{\mathcal{A}}_e = \text{span}\{|\rightarrow, e\rangle + |\leftarrow, e\rangle\}$,
4. $B = \{s\} \sqcup M$, where for all $u \in M$, $\Xi^{\mathcal{A}}_u + \Xi^{\mathcal{B}}_u = \{0\}$,

is a *quantum walk* on the weighted graph $G = (V, E)$. To motivate this terminology, the phase estimation algorithm in the next section implements a quantum walk on G, in the sense that it decides if $\exists t \in M$ in the same component as s.[3]

Simple vertices are motivated by quantum walks, and related constructions, as follows. In a random walk on a weighted graph, in order to take a step from a vertex $u \in V$ to a neighbour, the random walker chooses an edge $e \in E(u)$ to traverse, with probability $\mathsf{w}_e / (\sum_{e' \in E(u)} \mathsf{w}_{e'})$. This is precisely the distribution obtained by measuring $|\psi_\star(u)\rangle / \||\psi_\star(u)\rangle\|$. The states $|\psi_\star(u)\rangle$ correspond to *quantum walk states* – alternating a reflection around these with a reflection around the states $\{|\rightarrow, e\rangle + |\leftarrow, e\rangle\}_{e \in E}$ implements a discrete-time quantum walk, as in [Sze04, MNRS11, Bel13].[4]

A related example is that of a *switching network*, which, unlike quantum walks, will actually be one of the building blocks of our results in this chapter. We first define what it means for an edge to be a switch.

[3] This is if we use $|\psi_0\rangle = |s\rangle$. More generally, we can let $B = S \sqcup M$ and let $|\psi_0\rangle = \sum_{s \in S} \sqrt{\sigma(s)} |s\rangle$ for *initial distribution* σ on S.

[4] In some previous works, it is assumed that the graph is bipartite, and then the walk is implemented by alternating reflections around the states $|\psi_\star(u)\rangle$ of the two parts of the bipartition. We are actually doing the same thing here, we have just ensured the graph is bipartite by inserting a vertex in the middle of each edge.

4.3.5. Definition (Switch edge). Fix a subspace graph G. We call an edge $e \in E$ a *switch* (or *switch edge*) if there is some value $\varphi(e) \in \{0,1\}$ associated with that edge (implicitly depending on the input), such that

$$\Xi_e = \mathrm{span}\{|\rightarrow, e\rangle, |\leftarrow, e\rangle\},$$
$$\Xi_e^{\mathcal{A}} = \mathrm{span}\{|\rightarrow, e\rangle - (-1)^{\varphi(e)}|\leftarrow, e\rangle\} \quad \text{and} \quad \Xi_e^{\mathcal{B}} = \mathrm{span}\{|\rightarrow, e\rangle + |\leftarrow, e\rangle\},$$

and moreover, if $e \in E^{\rightarrow}(u) \cap E^{\leftarrow}(v)$, (that is, $e = (u,v)$), then $\mathcal{V}_u \cap \Xi_e = \mathrm{span}\{|\rightarrow, e\rangle\}$ and $\mathcal{V}_v \cap \Xi_e = \mathrm{span}\{|\leftarrow, e\rangle\}$.

The idea behind a switch edge is that if $\varphi(e) = 0$, $\Xi_e^{\mathcal{A}} + \Xi_e^{\mathcal{B}} = \Xi_e$, and so $\Xi_e^{\perp} = \{0\}$. Recall that a *positive witness* is some $|w\rangle \in \mathcal{A}_G^{\perp} \cap \mathcal{B}_G^{\perp}$ (Definition 2.7.3). If $\Xi_e^{\perp} = \{0\}$, then $|w\rangle$ can have no overlap with Ξ_e, so e is essentially blocked from use, so we say the edge is *switched off*. In switching networks, defined shortly, the positive witness is an st-flow, and it is restricted to edges in the subgraph $G(x)$ of edges that are switched on (we discuss this more in Section 4.3.3 – in particular, see (4.3.4)).

Next, we define some conditions that are satisfied by switching networks, as well as other examples considered in this chapter.

4.3.6. Definition (Canonical st-boundary). We say a subspace graph G has *weak canonical st-boundary* if $B = \{s,t\}$, where:

- $\Xi_s = \mathrm{span}\{|s\rangle, |\leftarrow, s\rangle\}$, $\Xi_s^{\mathcal{A}} = \mathrm{span}\{|s\rangle + |\leftarrow, s\rangle\}$ and $\Xi_s^{\mathcal{B}} = \{0\}$, where $\mathcal{V}_s$ only overlaps $|\leftarrow, s\rangle$.
- $\Xi_t = \mathrm{span}\{|\rightarrow, t\rangle, |t\rangle\}$, $\Xi_t^{\mathcal{A}} = \mathrm{span}\{|\rightarrow, t\rangle + |t\rangle\}$ and $\Xi_t^{\mathcal{B}} = \{0\}$, where $\mathcal{V}_t$ only overlaps $|\rightarrow, t\rangle$.

We say a subspace graph G has *canonical st-boundary* if additionally holds

- $|\leftarrow, s\rangle + |\rightarrow, t\rangle \in \mathcal{B}_G$.

Note that when G has canonical st-boundary, we always have $|s\rangle + |t\rangle \in \mathcal{A}_G + \mathcal{B}_G$.

As discussed in Section 4.1, a switching network is actually a classical object, but we overload the term to describe a specific kind of subspace graph. This is justified by the fact that, evaluating such a subspace graph is equivalent to evaluating a switching network (see the discussion in Section 4.3.3).

4.3.7. Definition (Switching network). A *switching network* is a subspace graph with canonical st-boundary in which all edges are switches (Definition 4.3.5), and all vertices are simple (Definition 4.3.3).

Note that if a subspace graph is a switching network, then all its spaces are fixed by the graph structure (V, E), and the associated weights. This is not true for subspace graphs more generally, and in Section 4.3.5, we will see an example of a subspace graph with other kinds of subspaces.

We discuss more properties of switching networks, including several examples, in Section 4.3.3. The examples we will detail in this chapter will not be restricted to switching networks, but all, aside from some pedagogical examples, will have canonical boundary, and we will let some of the edges be switches.

4.3.2 Phase estimation algorithm

A subspace graph, as in Definition 4.3.1, gives rise to a phase estimation algorithm (see Section 2.7) with spaces $\mathcal{A}_G$ and $\mathcal{B}_G$ as in Definition 4.3.1 and some initial state $|\psi_0\rangle \in \Xi_B \cap \mathcal{B}_G^\perp$. Though it is possible to consider more general states $|\psi_0\rangle$, throughout this chapter, we will always assume G has canonical st-boundary, and will let

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}(|s\rangle - |t\rangle). \tag{4.3.1}$$

To fully specify a phase estimation algorithm, we need a pair of orthogonal bases for $\mathcal{A}_G$ and $\mathcal{B}_G$ that are easily generated, in order to implement their respective reflections. We therefore always assume we have such a basis pair associated with G, which we call the *working bases*. For example, if $\Xi_e^{\mathcal{B}} = \{0\}$ for all e, then it suffices to have good bases for all the pairwise orthogonal spaces $\mathcal{V}_v$; and good bases for all the pairwise orthogonal spaces $\Xi_e^{\mathcal{A}}$. However, it is in our interest to put as much of $\Xi_e^{\mathcal{A}} + \Xi_e^{\mathcal{B}}$ into $\Xi_e^{\mathcal{B}}$ as we can manage while still maintaining a good working basis for $\mathcal{B}_G$. That is because of how the negative witness complexity is defined (Definition 2.7.2, and also Definition 4.3.11 below): only the part of $|\psi_0\rangle$ in $\mathcal{A}_G$ is counted in the complexity. We will see this in action in switching networks, in Section 4.3.3.

To facilitate our later constructions, we will make the following extra assumptions on the working bases.

4.3.8. Definition (Composable bases). Fix a subspace graph with canonical st-boundary, and some $\mathsf{r} \in \mathbb{R}_{>0}$. We say a pair of working bases $\Psi_{\mathcal{A}}$ and $\Psi_{\mathcal{B}}$ (see Definition 2.7.5) for $\mathcal{A}_G$ and $\mathcal{B}_G$ are st-composable, with scaling factor r, if they satisfy the following assumptions:

1. $\Psi_{\mathcal{A}} = \bigcup_{e \in E \cup B} \Psi_{\mathcal{A}}(e)$, where $\Psi_{\mathcal{A}}(e)$ is an orthonormal basis for $\Xi_e^{\mathcal{A}}$;
2. $\Psi_{\mathcal{B}} = \Psi_{\mathcal{B}}^- \cup \bigcup_{e \in E \cup B} \Psi_{\mathcal{B}}(e)$, for some $\Psi_{\mathcal{B}}^-$, where $\Psi_{\mathcal{B}}(e)$ is an orthonormal basis for $\Xi_e^{\mathcal{B}}$;

3. $|b_0\rangle \in \Psi_{\mathcal{B}}^-$ satisfies $|b_0\rangle = \frac{1}{\sqrt{2}}(|\leftarrow, s\rangle + |\rightarrow, t\rangle)$;
4. $|b_1\rangle \in \Psi_{\mathcal{B}}^-$ satisfies $|b_1\rangle = \frac{|\leftarrow,s\rangle - |\rightarrow,t\rangle + \sqrt{\mathsf{r}}|\bar{b}_1\rangle}{\sqrt{2+\mathsf{r}}}$ for some unit vector $|\bar{b}_1\rangle$ that is orthogonal to $|\leftarrow, s\rangle$ and $|\rightarrow, t\rangle$;
5. $\Psi_{\mathcal{B}} \setminus \{|b_0\rangle, |b_1\rangle\}$ is orthogonal to $|\leftarrow, s\rangle$ and $|\rightarrow, t\rangle$.

We note that generally we will be able to choose the scaling factor r by applying a linear map to H_G that scales Ξ_E relative to Ξ_B. See also the remark after Theorem 4.3.14.

Now we can define the special form of subspace graph that will apply to all subspace graphs studied in this chapter (aside from some examples mentioned informally):

4.3.9. Definition. We say a subspace graph G is *st-composable* if:

1. G has canonical *st*-boundary (Definition 4.3.6);
2. G is equipped with composable working bases (Definition 4.3.8);
3. for each $e \in E \setminus \overline{E}$, $\Xi_e^{\mathcal{B}} = \{0\}$, where $\overline{E}$ is the set of edges that are switches.

To analyze a phase estimation algorithm, we need to exhibit positive and negative witnesses. A positive witness for a phase estimation algorithm (Definition 2.7.3) is a vector $|w\rangle \in \mathcal{A}_G^\perp \cap \mathcal{B}_G^\perp$ such that $\langle\psi_0|w\rangle \neq 0$. By scaling $|w\rangle$, we can make $\langle\psi_0|w\rangle \neq 0$ take any scalar value, without impacting the complexity c_+ from Theorem 2.7.4. This motivates the following definition.

4.3.10. Definition (Positive witness for a graph). We say $|w\rangle \in \mathcal{A}_G^\perp \cap \mathcal{B}_G^\perp$ is a positive witness for G if it can be expressed as

$$|w\rangle = |s\rangle - |\leftarrow, s\rangle + \Pi_E|w\rangle + |\rightarrow, t\rangle - |t\rangle.$$

We let $W_+(G)$ be an upper bound (over some implicit input) on the minimum $\||w\rangle\|^2$ of any positive witness for G. We call $|\hat{w}\rangle := \Pi_E|w\rangle$ the *cropped witness*, and we let $\hat{W}_+(G) = W_+(G) - 4$ be an upper bound on the minimum $\||\hat{w}\rangle\|^2$.

To parse the definition above, note that if $\langle\psi_0|w\rangle = \sqrt{2}$, then we can always express $|w\rangle$ as

$$|w\rangle = \sqrt{2}|\psi_0\rangle + (I - |\psi_0\rangle\langle\psi_0|)|w\rangle = |s\rangle - |t\rangle + (I - |\psi_0\rangle\langle\psi_0|)|w\rangle,$$

but more than that, $|w\rangle$ is orthogonal to $|s\rangle + |t\rangle \in \mathcal{A}_G + \mathcal{B}_G$, so $(I - |\psi_0\rangle\langle\psi_0|)|w\rangle$ is orthogonal to both $|s\rangle - |t\rangle$ and $|s\rangle + |t\rangle$, so we have:

$$|w\rangle = |s\rangle - |t\rangle + (I - |s\rangle\langle s| - |t\rangle\langle t|)|w\rangle = |s\rangle - |\leftarrow, s\rangle - |t\rangle + |\rightarrow, t\rangle + (I - \Pi_B)|w\rangle,$$

since $|w\rangle$ must be orthogonal to $|s\rangle + |\leftarrow, s\rangle$ and $|t\rangle + |\rightarrow, t\rangle$. We consider $|w\rangle$ with its boundary $\Pi_B|w\rangle$ removed – which we call a cropped witness – because in Section 4.4, when we compose subspace graphs, we will crop off their boundary spaces to save additive constants in the witness sizes, which become relevant when we have very deep recursive structures. Thus, we similarly define:

4.3.11. Definition (Negative witness for a graph). We say $|w_{\mathcal{A}}\rangle \in \mathcal{A}_G$ is a negative witness for G if $|w_{\mathcal{A}}\rangle - |s\rangle + |t\rangle \in \mathcal{B}_G$. We define the corresponding *cropped negative witness* by $|\hat{w}_{\mathcal{A}}\rangle = \Pi_E|w_{\mathcal{A}}\rangle$.

For the above, recall that a negative witness for a phase estimation algorithm is a vector $|w_{\mathcal{A}}\rangle \in \mathcal{A}$ such that $|w_{\mathcal{B}}\rangle := |\psi_0\rangle - |w_{\mathcal{A}}\rangle \in \mathcal{B}$. We have slightly modified the definition, by replacing $|\psi_0\rangle = \frac{1}{\sqrt{2}}(|s\rangle - |t\rangle)$ with $|s\rangle - |t\rangle$, in order to save factors of $\sqrt{2}$ everywhere, but this changes very little. We will use the following later in the chapter.

4.3.12. Lemma. *Assume G has canonical st-boundary. Then $|w_{\mathcal{A}}\rangle \in \mathcal{A}_G$ is a negative witness for G if and only if $|\hat{w}_{\mathcal{A}}\rangle + |\leftarrow, s\rangle - |\rightarrow, t\rangle \in \mathcal{B}_G$.*

Proof:
$|w_{\mathcal{A}}\rangle \in \mathcal{A}_G$ is a negative witness for G if and only if $|w_{\mathcal{A}}\rangle = |s\rangle - |t\rangle + |w_{\mathcal{B}}\rangle$ for some $|w_{\mathcal{B}}\rangle \in \mathcal{B}_G$. Since the only vectors in $\mathcal{A}_G$ that overlap the boundary Ξ_B are $\Xi_B^{\mathcal{A}} = \mathrm{span}\{|s\rangle + |\leftarrow, s\rangle, |t\rangle + |\rightarrow, t\rangle\}$, it must be the case that

$$|w_{\mathcal{A}}\rangle = |s\rangle + |\leftarrow, s\rangle - |t\rangle - |\rightarrow, t\rangle + \underbrace{\Pi_E|w_{\mathcal{A}}\rangle}_{=|\hat{w}_{\mathcal{A}}\rangle}.$$

Thus, $|w_{\mathcal{A}}\rangle$ is a negative witness if and only if

$$\begin{aligned} & |w_{\mathcal{A}}\rangle - |s\rangle + |t\rangle \in \mathcal{B}_G \\ \Leftrightarrow\ & |\hat{w}_{\mathcal{A}}\rangle + |\leftarrow, s\rangle - |\rightarrow, t\rangle \in \mathcal{B}_G. \end{aligned}$$

□

In the next section, we describe what positive and negative witnesses look like for switching networks, and compare them with the witnesses for quantum walks on a graph (see Example 4.3.4).

Finally, we can define the complexity of a subspace graph by:

$$\mathcal{C}(G) := \sqrt{W_+(G)W_-(G)}.$$

In the case of subspaces graphs with canonical boundary, where we have defined cropped witnesses, we will often use

$$\hat{\mathcal{C}}(G) := \sqrt{\hat{W}_+(G)\hat{W}_-(G)},$$

which differs from $\mathcal{C}(G)$ by additive constants. Next, we present two formulations of how a subspace graph can be converted into a quantum algorithm. We will make use of both versions in the remainder of the thesis. It follows from Theorem 2.7.4 that:

4.3.13. Theorem. *Let G be a subspace graph with weak canonical st-boundary that computes $f : \{0,1\}^n \to \{0,1\}$ with respect to $|\psi_0\rangle$, such that the unitary $(2\Pi_{\mathcal{A}_G} - I)(2\Pi_{\mathcal{B}_G} - I)$ can be implemented in time T, and such that $\hat{W}_+(G) = O(1)$. Then there exists a quantum algorithm that decides f in time complexity $O\left(T\sqrt{\hat{W}_-(G)}\right)$ and space $O(\log \dim H_G + \log \hat{W}_-(G))$.*

If we additionally have working bases and canonical st-boundary, then it follows from Theorem 2.7.4 and Claim 2.7.6 that:

4.3.14. Theorem. *Let G be a subspace graph that computes $f : \{0,1\}^n \to \{0,1\}$ with respect to $|\psi_0\rangle$, with working bases that can be generated in time T, and such that $\hat{W}_+(G) = O(1)$. Then there exists a quantum algorithm that decides f in time complexity $O\left(T\sqrt{\hat{W}_-(G)}\right)$ and space $O(\log \dim H_G + \log \hat{W}_-(G))$.*

4.3.15. Remark. Given any subspace graph G, we can apply the linear map

$$\Pi_B + \frac{1}{\hat{W}_+(G)}\Pi_E$$

to H_G to get a new subspace graph G', which will still have canonical boundary if G did (this also applies to a weak canonical boundary), and such that

$$\hat{W}_+(G') = 1 \quad \text{and} \quad \hat{W}_-(G') = \hat{W}_+(G)\hat{W}_-(G).$$

This justifies thinking of $\hat{\mathcal{C}}(G)$ as a complexity. Such a scaling preserves the basis properties in Definition 4.3.8, and simply scales r. We can see this formally in Corollary 4.4.2.

4.3.3 Example: switching networks

We now discuss switching networks (Definition 4.3.7) in more detail, and give several examples. In the classical model of switching networks, a switching network is a graph with a variable $\varphi(e)$ associated with each edge e, and two terminals $s, t \in V$. A switching network computes a function $f : \{0,1\}^E \to \{0,1\}$ if for any $x \in \{0,1\}^E$, $f(x) = 1$ if and only if s and t are connected in $G(x)$, the subgraph consisting only of edges e such that $x_e = 1$ if $\varphi(e)$ is a positive literal, and $x_e = 0$ otherwise – that is, edges labelled by literals that are true when the variables are

set according to x. Here, we let "switching network" refer to a special kind of subspace graph, but this subspace graph implements the switching network, in the sense that the algorithm referred to in Theorem 4.3.14 decides if s and t are connected in $G(x)$.

We start by discussing some properties of switching networks. A switching network always has:

$$\mathcal{A}_G = \operatorname{span}\left\{|a_s\rangle := \frac{1}{\sqrt{2}}(|s\rangle + |\leftarrow, s\rangle), |a_t\rangle := \frac{1}{\sqrt{2}}(|t\rangle + |\rightarrow, t\rangle)\right\} \cup \left\{|a_e\rangle := \frac{1}{\sqrt{2}}(|\rightarrow, e\rangle - (-1)^{\varphi(e)}|\leftarrow, e\rangle) : e \in E\right\}, \quad (4.3.2)$$

so there is a natural choice of working basis $\Psi_{\mathcal{A}}$, and we immediately get the following:

4.3.16. Lemma. *If G is a switching network and $\Psi_{\mathcal{A}}$ is the basis in (4.3.2), then $\Psi_{\mathcal{A}}$ satisfies the conditions of Definition 4.3.8, and can be generated in one query to the unitary $|e\rangle \mapsto (-1)^{\varphi(e)}|e\rangle$ and $O(1)$ additional operations.*

Turning to $\mathcal{B}_G$, a switching network always has:

$$\mathcal{B}_G = \operatorname{span}\{|\psi_\star(u)\rangle : u \in V \setminus \{s,t\}\} \cup \{|\leftarrow, s\rangle + |\psi_\star(s)\rangle, |\rightarrow, t\rangle + |\psi_\star(t)\rangle\} + \operatorname{span}\left\{\frac{1}{\sqrt{2}}(|\rightarrow, e\rangle + |\leftarrow, e\rangle) : e \in E\right\}. \quad (4.3.3)$$

The working basis for $\mathcal{B}_G$ is less obvious, since the first and second space are not orthogonal, but one choice is given by the following, which we state for intuition (the proof is a simple exercise).

4.3.17. Lemma. *If G is a switching network,*

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

is an orthonormal basis for $\mathcal{B}_G$ whenever $\Psi_{\mathcal{B}}^-$ is an orthonormal basis for

$$\mathcal{B}^- = \operatorname{span}\{|\psi_\star^-(u)\rangle : u \in V \setminus \{s,t\}\} \cup \{|\leftarrow, s\rangle + |\psi_\star^-(s)\rangle, |\rightarrow, t\rangle + |\psi_\star^-(t)\rangle\},$$

where

$$|\psi_\star^-(u)\rangle = \sum_{e \in E^{\rightarrow}(u)} \frac{\sqrt{\mathsf{w}_e}}{2}(|\rightarrow, e\rangle - |\leftarrow, e\rangle) + \sum_{e \in E^{\leftarrow}(u)} \frac{\sqrt{\mathsf{w}_e}}{2}(|\leftarrow, e\rangle - |\rightarrow, e\rangle).$$

4.3.18. Remark. The statement of Lemma 4.3.17 is presented for the canonical boundary consisting of two terminals s and t. The same argument extends directly to switching networks with an arbitrary boundary $B(G) = B_{\leftarrow} \sqcup B_{\rightarrow} \subseteq V(G)$, where the boundary may contain any number of vertices. In that setting, the space $\mathcal{B}^-$ is defined by including the corresponding boundary states for all vertices in $B_{\leftarrow}$ and $B_{\rightarrow}$.

For intuition, the space $\mathcal{B}^-$ is the *cut space* of G, which is the span of all states that are (correctly weighted) superpositions of "edges" – where an edge is represented by $|\rightarrow, e\rangle - |\leftarrow, e\rangle$ – that for a cut set (their removal leaves G disconnected).

The implementation of reflections about $\mathcal{B}_G$ will be important in later applications, and we will use different approaches depending on the structure of the switching network. In Chapter 6, for the space $\mathcal{B}$, we construct a basis for the orthogonal complement and invoke Corollary 2.7.7, which guarantees that this suffices.

The following is a special case of Theorem 4.3.14, 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.

4.3.19. Theorem. *For $f : \{0,1\}^n \to \{0,1\}$, let G be a switching network that accepts $x \in \{0,1\}^n$ if and only if $f(x) = 1$. Suppose that for all x accepted by G, there is an st-path in $G(x)$ of length at most W_+. Suppose the space $\mathcal{B}^\perp$ of G 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 $O(T_B\sqrt{W_+|E(G)|})$ and space $O(\log|E(G)|)$.*

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 give some intuition on the positive and negative witnesses of a switching network, which can also, to some extent, be used for intuition about the positive and negative witnesses for the more general subspace graphs considered in this thesis. One can show that the orthogonal complement of $\mathcal{B}_G$ is the span of all unit st-flows (Definition 2.4.5) of G, by which we mean, precisely, all states of the form

$$|w\rangle = |s\rangle - |\leftarrow, s\rangle + \underbrace{\sum_{e\in E} \frac{\theta(e)}{\sqrt{\mathsf{w}_e}}(|\rightarrow, e\rangle - |\leftarrow, e\rangle)}_{|\hat{w}\rangle} + |\rightarrow, t\rangle - |t\rangle, \tag{4.3.4}$$

where θ is a unit st-flow. For every $e \in E$ such that $\varphi(e) = 0$, we add $|\rightarrow, e\rangle - |\leftarrow, e\rangle$ to $\mathcal{A}_G + \mathcal{B}_G$, which adds the constraint on $|w\rangle \in \mathcal{A}_G^{\perp} \cap \mathcal{B}_G^{\perp}$ that $\theta(e) = 0$. Thus, θ must be a unit flow on the graph $G(x)$.

A negative witness exists if and only if there is no unit st-flow on $G(x)$, meaning s and t are not connected. In that case, there is always an st-cut-set of edges missing from $G(x)$, F. It is a simple exercise to show that

$$|w_{\mathcal{A}}\rangle = |s\rangle + |\leftarrow, s\rangle + \sum_{e \in F} \sqrt{\mathsf{w}_e}(|\rightarrow, e\rangle - |\leftarrow, e\rangle) - |\rightarrow, t\rangle - |t\rangle$$

is a cropped negative witness. However, not all witnesses have this form – superpositions of cuts are also negative witnesses. Ref. [JJKP18] gives a tight analysis of negative witness for switching networks.

We briefly remark here on the difference between switching networks and quantum walks. In a quantum walk, the positive witnesses are also of the form in (4.3.4), except that they are not restricted to a subgraph $G(x)$, but rather, can be on all of G. Negative witnesses for quantum walks are somewhat different, although they are also derived from s-M-cuts; they instead involve summing over all edges in the component containing s, which should not be connected to M in the negative case. This is because in switching networks, $\mathcal{B}_G$ contains more, since $\Xi_E^{\mathcal{B}}$ is non-trivial, which makes finding a basis for $\mathcal{B}_G$ more difficult, but makes negative witnesses smaller.

We give two examples of switching networks – for computing the OR of d bits (Section 4.3.3) and the AND of d bits (Section 4.3.3). These examples will also be important building blocks for our later results. Switching networks for Boolean formulas date back to work of Shannon [Sha38, Sha49], and their analysis as subspace graphs was already implicit in the span program constructions of [JK17]. While [JK17] also analyzes the time complexity of evaluating switching networks, the reflection around $\mathcal{B}_G$ is done via a quantum walk on G, which results in a dependence on the relaxation time of G. In our examples, we improve on this by giving an orthonormal basis for $\mathcal{B}_G$ that can be used to reflect around it directly. Such a basis that is also efficient to generate might not be available for arbitrary switching networks, but in the case of series-parallel graphs, which correspond exactly to switching networks for Boolean formulas, our work implies that a good basis exists.

Switching network for OR

In this section, we describe a switching network (Definition 4.3.7) that computes the OR of d Boolean variables. This serves as a simple example, and will also be a building block in our divide-and-conquer application in Section 5.3. Formally, we prove the following.

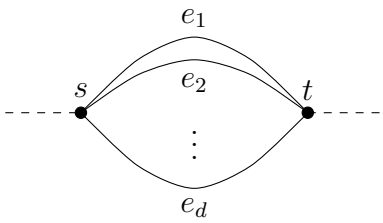

Figure 4.1: The graph G_{OR}. The dashed lines represent dangling boundary "edges".

4.3.20. Lemma. *For any $d \geq 1$, and positive weights $\{\mathsf{w}_i\}_{i\in[d]}$, there is a switching network $G_{\text{OR},d}$ that computes $\bigvee_{i=1}^d \varphi(e_i)$ with $\dim H_{G_{\text{OR},d}} = 2d + 4$ such that:*

1. *$G_{\text{OR},d}$ has st-composable working bases, with scaling factor $\mathsf{r} = 2\sum_{i=1}^d \mathsf{w}_i$, that can be generated in $O(\log d)$ time, assuming the state proportional to $\sum_{i=1}^d \sqrt{\mathsf{w}_i}|i\rangle$ can be generated in time $O(\log d)$;*
2. *if $\varphi(e_i) = 1$ for some $i \in [d]$, $G_{\text{OR},d}$ has cropped positive witness $|\hat{w}\rangle = \frac{1}{\sqrt{\mathsf{w}_i}}(|\rightarrow, i\rangle - |\leftarrow, i\rangle)$; and*
3. *if $\varphi(e_i) = 0$ for all $i \in [d]$, $G_{\text{OR},d}$ has cropped negative witness $|\hat{w}_{\mathcal{A}}\rangle = \sum_{i=1}^d \sqrt{\mathsf{w}_i}(|\rightarrow, i\rangle - |\leftarrow, i\rangle)$.*

We remark that if $\mathsf{w}_i = 1$ for all i, then Lemma 4.3.20 implies $\hat{W}_+(G_{\text{OR},d}) = 2$ and $\hat{W}_-(G_{\text{OR},d}) = 2d$, so by Theorem 4.3.14, there is a quantum algorithm for evaluating d-bit OR with time complexity $\widetilde{O}(\sqrt{d})$, which is optimal. This is a good sanity check, but we will mostly be interested in this construction as a building block, rather than in its own right.

Let $G = G_{\text{OR},d}$ be defined, as shown in Figure 4.1 by:

$$V = \{s, t\} \text{ and } E = \{e_i : i \in [d]\}$$

where each e_i has endpoints s and t, so $E(s) = E^{\rightarrow}(s) = E$ and $E(t) = E^{\leftarrow}(t) = E$. Since G is a switching network, it has boundary $B = V = \{s, t\}$. We will let the graph be weighted, with weights $\mathsf{w}_{e_i} = \mathsf{w}_i$. The only reason to let these vary in i is for later when we replace an edge with a gadget by composition, but for the sake of intuition, the reader may wish to imagine $\mathsf{w}_i = 1$ for all i. Since G is a switching network, every edge is a switch, which fixes the following spaces (we simplify notation by using i to label the edge e_i):

$$\forall i \in [d],\ \Xi_{e_i} = \operatorname{span}\{|\rightarrow, i\rangle, |\leftarrow, i\rangle\},$$
$$\Xi_{e_i}^{\mathcal{A}} = \operatorname{span}\{|\rightarrow, i\rangle - (-1)^{\varphi(e_i)}|\leftarrow, i\rangle\},\ \text{and}\ \Xi_{e_i}^{\mathcal{B}} = \operatorname{span}\{|\rightarrow, i\rangle + |\leftarrow, i\rangle\}.$$

Furthermore, as a switching network has canonical *st*-boundary, the following spaces are fixed:

$$\Xi_s = \operatorname{span}\{|s\rangle, |\leftarrow, s\rangle\}, \quad \Xi_s^{\mathcal{A}} = \operatorname{span}\{|s\rangle + |\leftarrow, s\rangle\}, \text{ and } \Xi_s^{\mathcal{B}} = \{0\}$$
$$\Xi_t = \operatorname{span}\{|\rightarrow, t\rangle, |t\rangle\}, \quad \Xi_t^{\mathcal{A}} = \operatorname{span}\{|\rightarrow, t\rangle + |t\rangle\}, \text{ and } \Xi_t^{\mathcal{B}} = \{0\}.$$

Finally, since all vertices are simple, the following spaces are fixed:

$$\mathcal{V}_s = \operatorname{span}\left\{ |\leftarrow, s\rangle + \underbrace{\sum_{i=1}^{d} \sqrt{\mathsf{w}_i} |\rightarrow, i\rangle}_{|\psi_\star(s)\rangle} \right\} \text{ and } \mathcal{V}_t = \operatorname{span}\left\{ \underbrace{\sum_{i=1}^{d} \sqrt{\mathsf{w}_i} |\leftarrow, i\rangle}_{|\psi_\star(t)\rangle} + |\rightarrow, t\rangle \right\}.$$

Then we have:

$$\begin{aligned} \mathcal{A}_G &= \bigoplus_{e \in E \cup B} \Xi_e^{\mathcal{A}} \\ &= \operatorname{span}\{|\rightarrow, i\rangle - (-1)^{\varphi(e_i)} |\leftarrow, i\rangle : i \in [d]\} \cup \{|s\rangle + |\leftarrow, s\rangle, |t\rangle + |\rightarrow, t\rangle\} \\ \text{and } \mathcal{B}_G &= \mathcal{V}_s \oplus \mathcal{V}_t + \bigoplus_{e \in E} \Xi_e^{\mathcal{B}} = \mathcal{V}_s \oplus \mathcal{V}_t + \operatorname{span}\{|\rightarrow, i\rangle + |\leftarrow, i\rangle : i \in [d]\}. \end{aligned} \tag{4.3.5}$$

By Lemma 4.3.16, since we assume we can query the values $\varphi(e)$ in unit cost, there is a working basis for $\Psi_{\mathcal{A}}$ satisfying the conditions of Definition 4.3.8 that can be generated in unit time. To complete the proof of the the first item of Lemma 4.3.20, we prove the following.

4.3.21. Lemma. *Let*

$$\Psi_{\mathcal{B}} = \left\{ |b_0\rangle = \frac{1}{\sqrt{2}}(|\leftarrow, s\rangle + |\rightarrow, t\rangle), |b_1\rangle = \frac{|\leftarrow, s\rangle - |\rightarrow, t\rangle + \sqrt{\mathsf{r}}|\bar{b}_1\rangle}{\sqrt{2+\mathsf{r}}} \right\}$$
$$\cup \left\{ |b_{e_i}\rangle = \frac{1}{\sqrt{2}}(|\rightarrow, i\rangle + |\leftarrow, i\rangle) : i \in [d] \right\},$$

where

$$\mathsf{r} = 2\sum_{i=1}^{d} \mathsf{w}_i \quad \text{and} \quad |\bar{b}_1\rangle = \frac{1}{\sqrt{\mathsf{r}}} \sum_{i=1}^{d} \sqrt{\mathsf{w}_i}(|\rightarrow, i\rangle - |\leftarrow, i\rangle).$$

Then $\Psi_{\mathcal{B}}$ is an orthonormal basis for $\mathcal{B}_G$, and it satisfies all the conditions of an st-composable basis in Definition 4.3.8. Furthermore, as long as we can generate the state proportional to $\sum_{i=1}^{d} \sqrt{\mathsf{w}_i}|i\rangle$ in time $O(\log d)$, $\Psi_{\mathcal{B}}$ can be generated in time $O(\log d)$.

Proof:
Note that:

$$\begin{aligned}&|\leftarrow,s\rangle+|\rightarrow,t\rangle\\&=\underbrace{|\leftarrow,s\rangle+\sum_{i=1}^{d}\sqrt{\mathsf{w}_i}|\rightarrow,i\rangle}_{\in\mathcal{V}_s}+\underbrace{\sum_{i=1}^{d}\sqrt{\mathsf{w}_i}|\leftarrow,i\rangle+|\rightarrow,t\rangle}_{\in\mathcal{V}_t}-\sum_{i=1}^{d}\sqrt{\mathsf{w}_i}\underbrace{(|\rightarrow,i\rangle+|\leftarrow,i\rangle)}_{\in\Xi_e^{\mathcal{B}}}\end{aligned}$$

and

$$|\leftarrow,s\rangle-|\rightarrow,t\rangle+\sqrt{\mathsf{r}}|\bar{b}_1\rangle=\underbrace{|\leftarrow,s\rangle+\sum_{i=1}^{d}\sqrt{\mathsf{w}_i}|\rightarrow,i\rangle}_{\in\mathcal{V}_s}-\underbrace{\sum_{i=1}^{d}\sqrt{\mathsf{w}_i}|\leftarrow,i\rangle+|\rightarrow,t\rangle}_{\in\mathcal{V}_t}.$$

From there, it is simple to check that $\Psi_{\mathcal{B}}\subset\mathcal{B}_G$. It is also simple to see that $\Xi_e^{\mathcal{B}}\subset\mathrm{span}\Psi_{\mathcal{B}}$ for all e. Thus, we check $\mathcal{V}_s$ and $\mathcal{V}_t$. We have:

$$\begin{aligned}|\leftarrow,s\rangle+\sum_{i=1}^{d}\sqrt{\mathsf{w}_i}|\rightarrow,i\rangle=&\frac{1}{2}\left(|\leftarrow,s\rangle+\sum_{i=1}^{d}\sqrt{\mathsf{w}_i}(|\rightarrow,i\rangle-|\leftarrow,i\rangle)-|\rightarrow,t\rangle\right)\\&+\frac{1}{2}(|\leftarrow,s\rangle+|\rightarrow,t\rangle)+\frac{1}{2}\sum_{i=1}^{d}\sqrt{\mathsf{w}_i}(|\rightarrow,i\rangle+|\leftarrow,i\rangle),\end{aligned}$$

showing that $\mathcal{V}_s\subset\mathrm{span}\Psi_{\mathcal{B}}$. A similar proof shows that $\mathcal{V}_t\subset\mathrm{span}\Psi_{\mathcal{B}}$.

It is clear by inspection that $\Psi_{\mathcal{B}}$ satisfies the properties of Definition 4.3.8, so to complete the proof, it is a simple exercise to observe that the basis can be generated in $O(\log d)$ time. □

To prove the second item of Lemma 4.3.20, we show the following.

4.3.22. Lemma. *If $\varphi(e_i)=1$ for some i, then the following is a positive witness for G (Definition 4.3.10):*

$$|w\rangle=|s\rangle-|\leftarrow,s\rangle+\underbrace{\frac{1}{\sqrt{\mathsf{w}_i}}(|\rightarrow,i\rangle-|\leftarrow,i\rangle)}_{|\hat{w}\rangle}+|\rightarrow,t\rangle-|t\rangle.$$

Proof:
One can check that $|w\rangle\in\mathcal{A}_G^{\perp}\cap\mathcal{B}_G^{\perp}$, by verifying that it is orthogonal to each of the spaces in (4.3.5). For orthogonality with

$$\Xi_{e_i}^{\mathcal{A}}=\mathrm{span}\{|\rightarrow,i\rangle-(-1)^{\varphi(i)}|\leftarrow,i\rangle\},$$

we relied on the fact that $\varphi(i) = 1$. Otherwise, we would have $\Xi^{\mathcal{A}}_{e_i} + \Xi^{\mathcal{B}}_{e_i} = \Xi_{e_i}$, and a positive witness must be orthogonal to Ξ_e, so the edge e is effectively blocked. □

Finally, to prove the third item of Lemma 4.3.20, we show the following.

4.3.23. Lemma. *If $\varphi(e_i) = 0$ for all i, then the following is a negative witness for G (Definition 4.3.11):*

$$|w_{\mathcal{A}}\rangle = |s\rangle + |\leftarrow, s\rangle + \underbrace{\sum_{i=1}^{d} \sqrt{\mathsf{w}_i}\Big(|\rightarrow, i\rangle - (-1)^{\varphi(i)}|\leftarrow, i\rangle\Big)}_{|\hat{w}_{\mathcal{A}}\rangle} - |\rightarrow, t\rangle - |t\rangle.$$

Proof:
Since $\varphi(i) = 0$ for all $i \in [d]$, we have:

$$\begin{aligned}
&|s\rangle - |t\rangle \\
&= \underbrace{|s\rangle + |\leftarrow, s\rangle}_{\in \Xi^{\mathcal{A}}_s} - \Bigg(\underbrace{|\leftarrow, s\rangle + \sum_{i=1}^{d} \sqrt{\mathsf{w}_i}|\rightarrow, i\rangle}_{\in \mathcal{V}_s \subset \mathcal{B}}\Bigg) + \sum_{i=1}^{d} \sqrt{\mathsf{w}_i}\, \underbrace{\big(|\rightarrow, i\rangle - (-1)^{\varphi(i)}|\leftarrow, i\rangle\big)}_{\in \Xi^{\mathcal{A}}_{e_i}} \\
&+ \Bigg(\underbrace{\sum_{i=1}^{d} \sqrt{\mathsf{w}_i}|\leftarrow, i\rangle + |\rightarrow, t\rangle}_{\in \mathcal{V}_i \subset \mathcal{B}}\Bigg) - \big(\underbrace{|\rightarrow, t\rangle + |t\rangle}_{\in \Xi^{\mathcal{A}}_t}\big),
\end{aligned}$$

implying that $|w_{\mathcal{A}}\rangle$ is a negative witness. Note that this crucially relies on $\varphi(e_i) = 0$ for all i, otherwise the above expression fails to hold. □

Switching network for AND

In this section, we describe a switching network (Definition 4.3.7) that computes the AND of d Boolean variables.

4.3.24. Lemma. *For any $d \geq 1$, and positive weights $\{\mathsf{w}_i\}_{i \in d}$, there is a switching network $G_{AND,d}$ that computes $\bigwedge_{i=1}^{d} \varphi(e_i)$ with $\dim H_{G_{AND,d}} = 2d + 4$ such that:*

1. *$G_{AND,d}$ has st-composable working bases, with scaling factor $\mathsf{r} = 2/\sum_{i=1}^{d}(1/\mathsf{w}_i)$, that can be generated in $O(\log d)$ time, assuming the state proportional to $\sum_{i=1}^{d} \frac{1}{\sqrt{\mathsf{w}_i}}|i\rangle$ can be generated in time $O(\log d)$;*
2. *if $\varphi(e_i) = 1$ for all $i \in [d]$, $G_{AND,d}$ has cropped positive witness $|\hat{w}\rangle = \sum_{i=1}^{d} \frac{1}{\sqrt{\mathsf{w}_i}}(|\rightarrow, i\rangle - |\leftarrow, i\rangle)$; and*

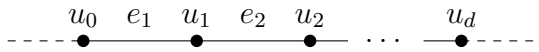

Figure 4.2: The graph G_{AND}. The dashed lines represent dangling boundary "edges".

3. *if* $\varphi(e_i) = 0$ *for some* $i \in [d]$, $G_{\text{AND},d}$ *has cropped negative witness* $|\hat{w}_{\mathcal{A}}\rangle = \sqrt{\mathsf{w}_i}(|\rightarrow, i\rangle - |\leftarrow, i\rangle)$.

As with the OR switching network in Section 4.3.3, a corollary of this lemma is that there is a quantum algorithm for evaluating AND in optimal time $\widetilde{O}(\sqrt{d})$.

Let $G = G_{\text{AND},d}$ be the graph in Figure 4.2, defined by

$$V = \{s = u_0, u_1, \ldots, u_d = t\} \text{ and } E = \{e_i = (u_{i-1}, u_i) : i \in [d]\},$$

so $E(s) = E^{\rightarrow}(s) = \{e_1\}$, $E(t) = E^{\leftarrow}(t) = \{e_d\}$, and for all $i \in [d-1]$, $E^{\leftarrow}(u_i) = \{e_i\}$ and $E^{\rightarrow}(u_i) = \{e_{i+1}\}$. Since G is a switching network, it has boundary $B = \{s, t\}$. We will let the graph be weighted, with weights $\mathsf{w}_{e_i} = \mathsf{w}_i$. Since G is a switching network: every edge is a switch, which fixes Ξ_{e_i}, $\Xi^{\mathcal{A}}_{e_i}$ and $\Xi^{\mathcal{B}}_{e_i}$ for all $i \in [d]$; G has canonical st-boundary, which fixes Ξ_s, $\Xi^{\mathcal{A}}_s$, $\Xi^{\mathcal{B}}_s$, Ξ_t, $\Xi^{\mathcal{A}}_t$, and $\Xi^{\mathcal{B}}_t$; and every vertex is simple, which fixes the spaces $\mathcal{V}_{u_i}$ for $i \in \{0, \ldots, d\}$. In particular, letting $\mathsf{w}_0 = \mathsf{w}_{d+1} = 1$, $s = 0$, $t = d+1$, and $i \in [d]$ label e_i, we must have:

$$\forall i \in \{0, \ldots, d\},\ \mathcal{V}_{u_i} = \operatorname{span}\{\sqrt{\mathsf{w}_i}|\leftarrow, i\rangle + \sqrt{\mathsf{w}_{i+1}}|\rightarrow, i+1\rangle\}.$$

This fully defines

$$\mathcal{A}_G = \bigoplus_{e \in E \cup B} \Xi^{\mathcal{A}}_e \quad \text{and} \quad \mathcal{B}_G = \bigoplus_{i=0}^{d} \mathcal{V}_{u_i} + \bigoplus_{e \in E} \Xi^{\mathcal{B}}_e.$$

By Lemma 4.3.16, since we assume we can query the values $\varphi(e)$ in unit cost, there is a working basis for $\Psi_{\mathcal{A}}$ satisfying the conditions of Definition 4.3.8 that can be generated in unit time. To complete the proof of the the first item of Lemma 4.3.24, we prove the following.

4.3.25. Lemma. *Let* $|b_2\rangle, \ldots, |b_d\rangle$ *be an orthonormal basis for*

$$\operatorname{span}\{|\rightarrow, i\rangle - |\leftarrow, i\rangle\} \cap \operatorname{span}\left\{|\bar{b}_1\rangle = \frac{\sum_{i=1}^{d} \frac{1}{\sqrt{\mathsf{w}_i}}(|\rightarrow, i\rangle - |\leftarrow, i\rangle)}{\sqrt{\sum_{i=1}^{d} \frac{2}{\mathsf{w}_i}}}\right\}^{\perp}.$$

Let

$$\Psi^{-}_{\mathcal{B}} = \left\{|b_0\rangle = \frac{1}{\sqrt{2}}(|\leftarrow, s\rangle + |\rightarrow, t\rangle), |b_1\rangle = \frac{|\leftarrow, s\rangle - |\rightarrow, t\rangle + \sqrt{\mathsf{r}}|\bar{b}_1\rangle}{\sqrt{2+\mathsf{r}}}\right\}$$

$$\cup\{|b_2\rangle, \ldots, |b_d\rangle\}.$$

Then

$$\Psi_{\mathcal{B}} = \Psi_{\mathcal{B}}^{-} \cup \left\{ |b_{e_i}\rangle = \frac{1}{\sqrt{2}}(|\rightarrow, i\rangle + |\leftarrow, i\rangle) : i \in [d] \right\}$$

is an orthonormal basis for $\mathcal{B}_G$ *when* $\mathsf{r} = 2/\sum_{i=1}^{d} \frac{1}{\mathsf{w}_i}$*, and it satisfies all the conditions of an st-composable basis in Definition 4.3.8. Furthermore, as long as we can generate the state proportional to* $\sum_{i=1}^{d} \frac{1}{\sqrt{\mathsf{w}_i}}|i\rangle$ *in time* $O(\log d)$*,* $\Psi_{\mathcal{B}}$ *can be generated in time* $O(\log d)$*.*

Proof:
First, it is easy to verify that $\Psi_{\mathcal{B}}$ is indeed orthonormal. For each $i \in [d]$, $|\rightarrow, i\rangle + |\leftarrow, i\rangle$ is orthogonal to $\Psi_{\mathcal{B}}^{-} \subset \mathrm{span}\{|\leftarrow, s\rangle, |\rightarrow, t\rangle\} \cup \{|\rightarrow, i\rangle - |\leftarrow, i\rangle : i \in [d]\}$. Finally, $\Psi_{\mathcal{B}}^{-}$ is an orthonormal set because for $i \in \{2, \ldots, d\}$, $\langle \leftarrow, s|b_i\rangle = \langle \rightarrow, t|b_i\rangle = \langle \bar{b}_1|b_i\rangle = 0$, and $|b_0\rangle$ and $|b_1\rangle$ are also orthogonal.

Next, observe

$$\begin{aligned}\mathcal{B} = \mathrm{span}\{&\sqrt{\mathsf{w}_i}|\leftarrow, i\rangle + \sqrt{\mathsf{w}_{i+1}}|\rightarrow, i+1\rangle : i \in \{0, \ldots, d\}\} \\ &\cup \{\sqrt{\mathsf{w}_i}(|\rightarrow, i\rangle + |\leftarrow, i\rangle) : i \in [d]\},\end{aligned}$$

from which it follows that $\dim \mathcal{B} \leq d + 1 + d = 2d + 1$, and in fact, it is not difficult to see that $\dim \mathcal{B} = 2d + 1$.

We next argue that $\overline{\mathcal{B}}_G = \mathcal{B}_G^{\perp}$, where

$$\overline{\mathcal{B}}_G = \mathrm{span}\{|s\rangle, |t\rangle\} \oplus \mathrm{span}\left\{ |\leftarrow, s\rangle - |\rightarrow, t\rangle - \frac{2}{\sqrt{\mathsf{r}}}|\bar{b}_1\rangle \right\}. \tag{4.3.6}$$

We clearly have $|s\rangle, |t\rangle \in \mathcal{B}_G^{\perp}$, and rewriting:

$$\begin{aligned}|\leftarrow, s\rangle - |\rightarrow, t\rangle - \frac{2}{\sqrt{\mathsf{r}}}|\bar{b}_1\rangle &= |\leftarrow, s\rangle - |\rightarrow, t\rangle - \sum_{i=1}^{d} \frac{1}{\sqrt{\mathsf{w}_i}}(|\rightarrow, i\rangle - |\leftarrow, i\rangle) \\ &= \sum_{i=0}^{d} \left(\frac{1}{\sqrt{\mathsf{w}_i}}|\leftarrow, i\rangle - \frac{1}{\sqrt{\mathsf{w}_{i+1}}}|\rightarrow, i+1\rangle \right),\end{aligned}$$

where we have used $s = 0$, $t = d + 1$, and $\mathsf{w}_0 = \mathsf{w}_{d+1} = 1$, we see that it is also in $\mathcal{B}_G^{\perp}$, so $\overline{\mathcal{B}}_G \subseteq \mathcal{B}_G^{\perp}$. Since $\dim \overline{\mathcal{B}}_G = 3$, and

$$\dim \mathcal{B}_G^{\perp} = \underbrace{\dim \Xi_s + \sum_{i=1}^{d} \dim \Xi_{e_i} + \dim \Xi_t}_{H_G} - \dim \mathcal{B} = 2 + 2d + 2 - (2d + 1) = 3,$$

(see Definition 4.3.5 and Definition 4.3.6) we see that $\overline{\mathcal{B}}_G = \mathcal{B}_G^{\perp}$.

Next, to check that $\Psi_{\mathcal{B}} \subseteq \mathcal{B}_G$, we just need to check that each element of $\Psi_{\mathcal{B}}$ is orthogonal to each of the three vectors in the definition of $\overline{\mathcal{B}}_G$. This will be sufficient, since $\Psi_{\mathcal{B}}$ is an orthonormal set with $|\Psi_{\mathcal{B}}| = 2 + d - 1 + d = 2d + 1 = \dim \mathcal{B}_G$. It is simple to see that everything in $\Psi_{\mathcal{B}}$ is orthogonal to $|s\rangle$ and $|t\rangle$. It is also simple to see that everything in $\Psi_{\mathcal{B}} \setminus \Psi_{\mathcal{B}}^-$ is orthogonal to all of $\overline{\mathcal{B}}_G$.

We turn our attention to $\Psi_{\mathcal{B}}^-$. $|b_0\rangle$ is obviously orthogonal to $\overline{\mathcal{B}}_G$. For $|b_1\rangle$, we have:

$$\begin{aligned}
&\langle b_1| \left(|\leftarrow, s\rangle - |\rightarrow, t\rangle - \frac{2}{\sqrt{\mathsf{r}}} |\bar{b}_1\rangle \right) \\
&= \frac{1}{\sqrt{2+\mathsf{r}}} \left(\langle \leftarrow, s| - \langle \rightarrow, t| + \sqrt{\mathsf{r}} \langle \bar{b}_1| \right) \left(|\leftarrow, s\rangle - |\rightarrow, t\rangle - \frac{2}{\sqrt{\mathsf{r}}} |\bar{b}_1\rangle \right) \\
&= \frac{1}{\sqrt{2+\mathsf{r}}} \left(2 - \sqrt{\mathsf{r}} \frac{2}{\sqrt{\mathsf{r}}} \langle \bar{b}_1 | \bar{b}_1 \rangle \right) = 0.
\end{aligned}$$

Finally, for any $i \in \{2, \ldots, d\}$, we have:

$$\langle b_i| \left(|\leftarrow, s\rangle - |\rightarrow, t\rangle - \frac{2}{\sqrt{\mathsf{r}}} |\bar{b}_1\rangle \right) = -\frac{2}{\sqrt{\mathsf{r}}} \langle b_i | \bar{b}_1 \rangle = 0,$$

since the $|b_i\rangle$ are all orthogonal to $|\bar{b}_1\rangle$. Thus $\Psi_{\mathcal{B}} \subseteq \mathcal{B}_G$, and so $\Psi_{\mathcal{B}}$ is an orthonormal basis for $\mathcal{B}_G$.

It is clear by inspection that $\Psi_{\mathcal{B}}$ satisfies the properties of Definition 4.3.8, so to complete the proof, we need to argue that $\Psi_{\mathcal{B}}$ can be generated in $O(\log d)$ time. It is a simple exercise to observe that the basis in (4.3.6) can be generated in $O(\log d)$ time, and then the result follows from Corollary 2.7.7. □

To prove the second item of Lemma 4.3.24, we show the following.

4.3.26. Lemma. *If $\varphi(e_i) = 1$ for all $i \in [d]$, then the following is a positive witness for G (Definition 4.3.10):*

$$|w\rangle = |s\rangle - |\leftarrow, s\rangle + \underbrace{\sum_{i=1}^{d} \frac{1}{\sqrt{\mathsf{w}_i}} (|\rightarrow, i\rangle - |\leftarrow, i\rangle)}_{|\hat{w}\rangle} + |\rightarrow, t\rangle - |t\rangle.$$

Proof:
Since $\varphi(e_i) = 1$ for all $i \in [d]$, we have $\Xi_{e_i}^{\mathcal{A}} = \Xi_{e_i}^{\mathcal{B}} = \operatorname{span}\{|\rightarrow, i\rangle + |\leftarrow, i\rangle\}$, so clearly $|w\rangle$ is orthogonal to $\Xi_{e_i}^{\mathcal{A}} + \Xi_{e_i}^{\mathcal{B}}$ for each i. One can similarly verify by inspection that it is orthogonal to $\Xi_s^{\mathcal{A}}$, as well as $\Xi_t^{\mathcal{A}}$. This leaves only the spaces $\mathcal{V}_{u_i}$ for $i \in \{0, \ldots, d\}$, which we can verify are orthogonal to $|w\rangle$ by rewriting it as:

$$|w\rangle = |s\rangle - \sum_{i=0}^{d} \left(\frac{1}{\sqrt{\mathsf{w}_i}} |\leftarrow, i\rangle - \frac{1}{\sqrt{\mathsf{w}_{i+1}}} |\rightarrow, i+1\rangle \right) - |t\rangle.$$

We used the fact that $\mathsf{w}_0 = \mathsf{w}_{d+1} = 1$, $0 = s$ and $d+1 = t$. Thus, $|w\rangle \in \mathcal{A}_G^{\perp} \cap \mathcal{B}_G^{\perp}$. □

Finally, to prove the third item of Lemma 4.3.24, we show the following.

4.3.27. Lemma. *If $\varphi(e_i) = 0$ for some $i \in [d]$, then the following is a negative witness for G (Definition 4.3.11):*

$$|w_{\mathcal{A}}\rangle = |s\rangle + |\leftarrow, s\rangle + \underbrace{\sqrt{\mathsf{w}_j}(|\rightarrow, j\rangle - |\leftarrow, j\rangle)}_{|\hat{w}\rangle} - |\rightarrow, t\rangle - |t\rangle.$$

Proof:
First note that for any $j \in [d]$, using $\mathsf{w}_0 = 1$ and $|\leftarrow, 0\rangle = |\leftarrow, s\rangle$:

$$\sum_{i=1}^{j} \underbrace{(\sqrt{\mathsf{w}_{i-1}}|\leftarrow, i-1\rangle + \sqrt{\mathsf{w}_i}|\rightarrow, i\rangle)}_{\in \mathcal{V}_{u_{i-1}} \subset \mathcal{B}} - \sum_{i=1}^{j-1} \sqrt{\mathsf{w}_i} \underbrace{(|\rightarrow, i\rangle + |\leftarrow, i\rangle)}_{\in \Xi_{e_i}^{\mathcal{B}}}$$
$$= \sqrt{\mathsf{w}_0}|\leftarrow, 0\rangle + \sqrt{\mathsf{w}_j}|\rightarrow, j\rangle = |\leftarrow, s\rangle + \sqrt{\mathsf{w}_j}|\rightarrow, j\rangle$$

is in $\mathcal{B}_G$. By a similar argument, $\sqrt{\mathsf{w}_j}|\leftarrow, j\rangle + |\rightarrow, t\rangle \in \mathcal{B}_G$.

Since $\varphi(e_i) = 0$, $|\rightarrow, i\rangle - (-1)^{\varphi(e_i)}|\leftarrow, i\rangle = |\rightarrow, i\rangle - |\leftarrow, i\rangle$ is in $\Xi_{e_i}^{\mathcal{A}}$, we have:

$$|s\rangle - |t\rangle = \underbrace{|s\rangle + |\leftarrow, s\rangle}_{\in \Xi_s^{\mathcal{A}}} - \underbrace{(|\leftarrow, s\rangle + \sqrt{\mathsf{w}_j}|\rightarrow, j\rangle)}_{\in \mathcal{B}} + \sqrt{\mathsf{w}_j} \underbrace{(|\rightarrow, j\rangle - |\leftarrow, j\rangle)}_{\in \Xi_{e_j}^{\mathcal{A}}}$$
$$+ \underbrace{\sqrt{\mathsf{w}_j}|\leftarrow, j\rangle + |\rightarrow, t\rangle}_{\in \mathcal{B}} - \underbrace{(|\rightarrow, t\rangle + |t\rangle)}_{\in \Xi_t^{\mathcal{A}}},$$

implying that $|w_{\mathcal{A}}\rangle$ is a negative witness. □

Flows, circulations, and the cut space

Several constructions and arguments in the study of switching networks naturally involve flows and cut spaces. We therefore introduce these notions in a more general graph-theoretic setting, which will be useful throughout the remainder of the thesis. In this section, we examine the structure of *cut spaces*, of undirected graphs, with respect to any *boundary* $B(G) = B = B_{\leftarrow} \sqcup B_{\rightarrow} \subseteq V(G)$:

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

Taking $B_{\leftarrow} = \{s\}$ and $B_{\rightarrow} = \{t\}$ recovers the case of interest for us, but we leave this section slightly more general. We first show the following simple fact.

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

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

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

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

$$|\psi_u\rangle = \sum_{v \in V\setminus\{u\}} \alpha_v |\psi_v\rangle.$$

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

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

since $|\langle e|\psi_v\rangle|$ 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:

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

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. □

We can naturally view any function θ on $E(G)$ as a vector on $\mathrm{span}\{|\rightarrow, e\rangle - |\leftarrow, e\rangle : e \in E(G)\}$, which we denote $|\bar{\theta}\rangle$ in Definition 4.3.29, below. Such a *cropped flow state* enables seamless combination of flows (see Definition 2.4.4) in graphs obtained from gluing other graphs together (see Definition 2.4.7). 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 $|\theta\rangle$ below.

4.3.29. Definition. For any boundary flow θ, define:

$$|\bar{\theta}\rangle := \sum_{e \in E(G)} \theta(e)(|\rightarrow, e\rangle - |\leftarrow, e\rangle), \tag{4.3.7}$$

which we call a *cropped flow state*, and

$$|\theta\rangle = \sum_{u \in B_{\leftarrow}} \theta(u)(|u\rangle - |\leftarrow, u\rangle) + |\bar{\theta}\rangle + \sum_{u \in B_{\rightarrow}} \theta(u)(|u\rangle - |\rightarrow, u\rangle). \tag{4.3.8}$$

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 vector spaces.

4.3.30. Definition. We call

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

the space of boundary flows;

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

the space of circulations;

$$\mathcal{F}^{\mathrm{OPT}}(G) = \mathrm{span}\{|\theta\rangle \in \mathcal{F}(G) : |\theta\rangle \text{ is the optimal unit } st\text{-flow} : s, t \in B\}$$

the space of optimal boundary flows.

In the following lemmas, we establish fundamental properties of the space of boundary flows and circulations 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.

4.3.31. Lemma. *Let $s, t \in B$. Then the optimal unit st-flow is unique and its corresponding quantum states (cropped and not) are orthogonal to the space of circulations $\mathcal{C}(G)$. Consequently, the space of optimal boundary flows $\mathcal{F}^{OPT}(G)$ is orthogonal to the space of circulations $\mathcal{C}(G)$.*

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

4.3.32. Lemma. $\mathcal{F}(G) = \mathcal{F}^{OPT}(G) \oplus \mathcal{C}(G)$

Proof:
By Lemma 4.3.31, $\mathcal{F}^{\mathrm{OPT}}(\mathcal{N}) \perp \mathcal{C}(\mathcal{N})$. It is immediate that $\mathcal{F}(G) \supseteq \mathcal{F}^{\mathrm{OPT}}(G) \oplus \mathcal{C}(G)$. Thus, it remains to prove the reverse inclusion $\mathcal{F}(G) \subseteq \mathcal{F}^{\mathrm{OPT}}(G) \oplus \mathcal{C}(G)$. That is, we need to show that for any $|\theta\rangle \in \mathcal{F}(G)$ there exist $|\theta'\rangle \in \mathcal{F}^{\mathrm{OPT}}(G)$ and $|\theta^0\rangle \in \mathcal{C}(G)$ such that $|\theta\rangle = |\theta'\rangle + |\theta^0\rangle$. For this, define $b(\theta) = |\{u \in B : \theta(u) \neq 0\}|$. We proceed by induction on $b(\theta)$. In the base case, assume that $b(\theta) = 2$ and θ is a uv-flow for some $u, v \in B$. This is the base case because the total flow on the boundary must sum to 0. Without loss of generality, assume θ is scaled so that $\theta(u) = -\theta(v) = 1$. Then, by Lemma 4.3.31, there exists a unique optimal unit uv-flow $|\theta'\rangle \in \mathcal{F}^{\mathrm{OPT}}$. Let $|c\rangle = |\theta\rangle - |\theta'\rangle$. It is easy to check that it is a circulation, which establishes the base case. For the induction step, assume that $b(\theta) = k > 2$ and the claim holds for any $|\hat{\theta}\rangle$ such that $b(\hat{\theta}) < k$. Let $u, v \in B$ be such that $\theta(u), \theta(v) \neq 0$, and assume without loss of generality that θ is scaled so that $\theta(u) = 1$. Then there exists a unique optimal unit uv-flow $|\theta_{uv}\rangle$ such that $\theta_{uv}(u) = 1 = \theta(u)$. Let $|\theta - \theta_{uv}\rangle = |\theta\rangle - |\theta_{uv}\rangle$. Then $b(\theta - \theta_{uv}) \leq k - 1$. Applying the induction hypothesis to $|\theta - \theta_{uv}\rangle$ proves the statement. □

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

4.3.33. Lemma. *Let G be a switching network, with $\mathcal{B}$ as in (4.3.3). Then $\mathcal{B}^{\perp} = \mathcal{F}(G) + \mathrm{span}\{|u\rangle : u \in B(G)\}$.*

Proof:
By Lemma 4.3.17 and Remark 4.3.18 we can write:

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

Note that

$$\begin{aligned}&\left(\mathrm{span}\left\{\frac{1}{\sqrt{2}}(|\rightarrow, e\rangle + |\leftarrow, e\rangle) : e \in E(G)\right\}\right)^{\perp} \\ &= \mathrm{span}\left\{\frac{1}{\sqrt{2}}(|\rightarrow, e\rangle - |\leftarrow, e\rangle) : e \in E(G)\right\} \\ &\quad \oplus \mathrm{span}\{|s\rangle, |t\rangle, |\leftarrow, s\rangle, |\rightarrow, t\rangle : s \in B_{\leftarrow}, t \in B_{\rightarrow}\}.\end{aligned}$$

Since $\mathcal{B}^{-} \subseteq \left(\mathrm{span}\left\{\frac{1}{\sqrt{2}}(|\rightarrow, e\rangle + |\leftarrow, e\rangle) : e \in E(G)\right\}\right)^{\perp}$, we can take the orthogonal complement of $\mathcal{B}^{-}$ in $\left(\mathrm{span}\left\{\frac{1}{\sqrt{2}}(|\rightarrow, e\rangle + |\leftarrow, e\rangle) : e \in E(G)\right\}\right)^{\perp}$. That is,

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

$$|\hat{\theta}\rangle = \sum_{u \in B_{\leftarrow}} \theta_u(|u\rangle - |\leftarrow, u\rangle) + \sum_{e \in E} \theta(e)|e\rangle + \sum_{u \in B_{\rightarrow}} \theta_u(|u\rangle - |\rightarrow, u\rangle),$$

for some function θ on E and values $\{\theta_u\}_{u \in B}$, and $|b\rangle \in \text{span}\{|u\rangle : u \in B\}$. By definition of $|\psi_\star^-(u)\rangle$ (see Lemma 4.3.17), it is orthogonal to $|b\rangle$ for any $u \in V$, 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* $u \in V$,

$$\begin{aligned} \langle\hat{\theta}|\psi_\star^-(u)\rangle &= \sum_{e \in E^{\rightarrow}(u)} \frac{1}{2}\theta(e)\langle e|e\rangle - \sum_{e \in E^{\leftarrow}(u)} \frac{1}{2}\theta(e)\langle e|e\rangle \\ &= \sum_{e \in E^{\rightarrow}(u)} \theta(e) - \sum_{e \in E^{\leftarrow}(u)} \theta(e) = \theta(u). \end{aligned} \tag{4.3.9}$$

This inner product vanishes for all $u \in V \setminus B$ if and only if θ is a boundary flow.

Next, for any $s \in B_{\leftarrow}$,

$$\langle\hat{\theta}|(|\leftarrow, s\rangle + |\psi_\star^-(s)\rangle) = -\theta_s + \langle\hat{\theta}|\psi_\star^-(s)\rangle = -\theta_s + \theta(s), \tag{4.3.10}$$

by (4.3.9). This inner product vanishes for all $s \in B_{\leftarrow}$ if and only if $\theta_s = \theta(s)$ for all $s \in B_{\leftarrow}$.

Finally, for any $t \in B_{\rightarrow}$,

$$\langle\hat{\theta}|(|\rightarrow, t\rangle + |\psi_\star^-(t)\rangle) = -\theta_t + \theta(t), \tag{4.3.11}$$

again by (4.3.9). This inner product vanishes for all $t \in B_{\rightarrow}$ if and only if $\theta_t = \theta(t)$ for all $t \in B_{\rightarrow}$.

Combining (4.3.9), (4.3.10) and (4.3.11), we can see that $|\psi\rangle \in (\mathcal{B}^-)^{\perp}$ if and only if $|\hat{\theta}\rangle = |\theta\rangle$ (as defined in Definition 4.3.29) for some boundary flow θ, which is if and only if $|\hat{\theta}\rangle \in \mathcal{F}(G)$. □

Finally, we compute the dimensions of the space of flows and circulations of G and its subspaces.

4.3.34. Lemma. *Let $\mathcal{F}(G)$ be the space of boundary flows of G; $\mathcal{C}(G)$ the space of circulations of G; and $\mathcal{F}^{OPT}(G)$ the span of optimal boundary flows of G (see Definition 4.3.30). Then*

$$\dim \mathcal{F}(G) = |E(G)| + |B(G)| - |V(G)|$$

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

Proof:
The dimension of $\mathcal{F}(G)$ can be computed using Lemma 4.3.33:

$$\dim(\mathcal{F}(G) + \operatorname{span}\{|u\rangle : u \in B(G)\}) = \dim H_G - \dim \mathcal{B}(G).$$

We note that

$$\dim(\mathcal{F}(G) + \operatorname{span}\{|u\rangle : u \in B(G)\}) = \dim \mathcal{F}(G) + \dim(\operatorname{span}\{|u\rangle : u \in B(G)\}),$$

since the states $|u\rangle$ for $u \in B(G)$ are pairwise orthogonal and cannot be represented as linear combinations of flows and circulations. The dimension of H_G is equal to $2|E(G)| + 2|B(G)|$ by Definition 4.3.7. By Lemma 4.3.17, Remark 4.3.18, and Lemma 4.3.28, $\dim \mathcal{B}(G) = \dim \mathcal{B}^- + |E(G)| = |V(G)| + |E(G)|$. Combining these observations with $\dim(\operatorname{span}\{|v\rangle : v \in B(G)\}) = |B(G)|$, we obtain

$$\begin{aligned}\dim \mathcal{F}(G) &= \dim H_G - \dim \mathcal{B}(G) - \dim(\operatorname{span}\{|v\rangle : v \in B(G)\}) \\ &= |E(G)| + |B(G)| - |V(G)|.\end{aligned}$$

Next, we show that $\dim \mathcal{F}^{\mathrm{OPT}}(G) = |B(G)| - 1$ by proving that, for a fixed vertex $u \in B(G)$, the set

$$\{|\theta_{uv}\rangle : v \in B(G), v \neq u, |\theta_{uv}\rangle \text{ is an optimal unit flow from } u \text{ to } v\}$$

forms a linearly independent basis for $\mathcal{F}^{\mathrm{OPT}}(G)$. To see this, consider any pair $v_1, v_2 \in B(G)$ with $v_1, v_2 \neq u$. The optimal unit flow $|\theta_{v_1 v_2}\rangle$ can be written as

$$|\theta_{v_1 v_2}\rangle = -|\theta_{u v_1}\rangle + |\theta_{u v_2}\rangle.$$

This expression defines a unit flow from v_1 to v_2, and since it is a linear combination of optimal flows (each orthogonal to circulations), it is itself orthogonal to circulations. Therefore, it must be the unique optimal unit flow from v_1 to v_2, i.e., $|\theta_{v_1 v_2}\rangle$. This shows that all optimal flows between arbitrary pairs in $B(G)$ can be expressed in terms of the flows from u. Finally, we observe that the vectors $|\theta_{uv}\rangle$ are linearly independent. Suppose, for the sake of contradiction, that there exist coefficients α_i for $i \neq j$ for a fixed j such that

$$|\theta_{u v_j}\rangle = \sum_{i \neq j} \alpha_i |\theta_{u v_i}\rangle.$$

Consider the net flow into vertex v_j on both sides of the equation. The left-hand side has unit inflow at v_j by definition of $|\theta_{u v_j}\rangle$. On the other hand, since each

$|\theta_{uv_i}\rangle$ for $i \neq j$ is a unit flow from u to v_i with no flow entering or leaving v_j, the right-hand side has zero net flow at v_j. This contradiction implies that no such linear dependence exists, and therefore the set is linearly independent. Since

$$|\{|\theta_{uv}\rangle : v \in B(G), v \neq u, |\theta_{uv}\rangle \text{ is optimal unit flow from } u \text{ to } v\}| = |B(G)| - 1$$

and we have shown that it is a basis of $\mathcal{F}^{\mathrm{OPT}}(G)$, we can conclude that

$$\dim \mathcal{F}^{\mathrm{OPT}}(G) = |B(G)| - 1.$$

The dimension of $\mathcal{C}(G)$ is straightforward to determine, as it follows from the direct sum decomposition $\mathcal{F}(G) = \mathcal{F}^{\mathrm{OPT}}(G) \oplus \mathcal{C}(G)$ from Lemma 4.3.32. □

4.3.4 Example: classical deterministic algorithms

In the next section, we will see how to turn a quantum algorithm into a subspace graph. As a warmup example, we first see how classical deterministic algorithms fit into this framework. The simplest example is a reversible classical algorithm, which we can model as an undirected random walk on a line. It is not very sensible to model a classical algorithm as a random walk on a line, because we incur a quadratic slow-down by letting the algorithm travel backwards sometimes, instead of always forwards, but when we move from random walks to quantum walks, it is a perfectly valid thing to do, since a quantum walker traverses a line in linear time, rather than quadratic.

A reversible classical deterministic algorithm can be described by the following objects.

- A finite set $\mathcal{Z}$ of the algorithm's states;
- A sequence of permutations $\pi_1, \ldots, \pi_T : \mathcal{Z} \to \mathcal{Z}$ that depend (implicitly) on the input;
- An initial state $z_1 \in \mathcal{Z}$;
- A partitioning of $\mathcal{Z}$ into accepting and rejecting states $\mathcal{Z} = \mathcal{Z}_{\mathrm{acc}} \sqcup \mathcal{Z}_{\mathrm{rej}}$.

From these, we define $G = (V, E)$ by specifying its vertices and edges:

$$V = \bigsqcup_{\ell=0}^{T} V_\ell, \text{ where } V_\ell = \{v_{\ell,z} : z \in \mathcal{Z}\},$$
$$\text{and } E = \{e_{\ell,z} = (v_{\ell,z}, v_{\ell+1,\pi_{\ell+1}(z)}) : 0 \leq \ell \leq T-1, z \in \mathcal{Z}\}.$$

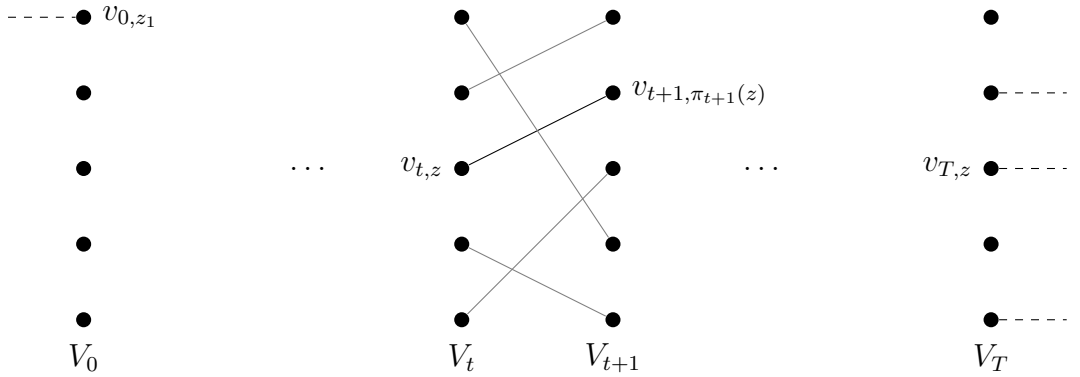

Figure 4.3: The graph G for a reversible classical deterministic algorithm with five algorithm states. Between each set V_t and V_{t+1}, there is exactly one edge connected to $s = v_{0,z_1}$. The dashed lines represent dangling boundary "edges".

Note that the graph G depends on the input because the edges are specified by the permutations. The boundary is defined $B = \{s\} \cup M$ where $s = v_{0,z_1}$ and $M = \{v_{T,z} : z \in \mathcal{Z}_{\text{acc}}\}$.

Next, we introduce a subspace graph structure on G in order to run a quantum walk following Example 4.3.4. It is easy to see that the connected component containing s is a line with one vertex in each of V_ℓ, and the algorithm accepts if and only if the vertex in V_T connected to s is in M – that is, if and only if s is connected to some vertex in M. Thus the quantum walk will decide whether $\pi_T \circ \ldots \circ \pi_1(z_1) \in \mathcal{Z}_{\text{acc}}$. See Figure 4.3 for a visualization of G. We set all edge weights to be 1, and then, following Example 4.3.4, we define boundary and edge spaces:

$$\begin{aligned}
\Xi_s &= \text{span}\{|s\rangle, |\leftarrow, s\rangle\}, \\
\Xi_s^{\mathcal{A}} &= \text{span}\{|s\rangle + |\leftarrow, s\rangle\}, \quad \text{and} \quad \Xi_s^{\mathcal{B}} = \{0\} \\
\forall z \in \mathcal{Z}_{\text{acc}},\ \Xi_{v_{T,z}} &= \text{span}\{|\rightarrow, v_{T,z}\rangle, |v_{T,z}\rangle\}, \\
\text{and} \quad \Xi_{v_{T,z}}^{\mathcal{A}} &= \Xi_{v_{T,z}}^{\mathcal{B}} = \{0\} \\
\forall \ell \in \{0, \ldots, T-1\}, z \in \mathcal{Z},\ \Xi_{e_{\ell,z}} &= \text{span}\{|\rightarrow, \ell, z\rangle, |\leftarrow, \ell, z\rangle\}, \\
\Xi_{e_{\ell,z}}^{\mathcal{A}} &= \text{span}\{|\rightarrow, \ell, z\rangle + |\leftarrow, \ell, z\rangle\}, \quad \text{and } \Xi_{e_{\ell,z}}^{\mathcal{B}} = \{0\},
\end{aligned}$$

and all vertices are simple (Definition 4.3.3), so:

$$\begin{aligned}
\forall z \in \mathcal{Z},\ \mathcal{V}_{v_{0,z}} &= \text{span}\{|\psi_\star(v_{0,z})\rangle = |\rightarrow, 0, z\rangle\}, \\
\forall t \in \{1, \ldots, T-1\}, z \in \mathcal{Z}, & \\
\mathcal{V}_{v_{\ell,z}} &= \text{span}\{|\psi_\star(v_{\ell,z})\rangle = |\leftarrow, \ell-1, \pi_\ell^{-1}(z)\rangle + |\rightarrow, \ell, z\rangle\} \\
\forall z \in \mathcal{Z},\ \mathcal{V}_{v_{T,z}} &= \text{span}\{|\psi_\star(v_{0,z})\rangle = |\leftarrow, T-1, \pi_T^{-1}(z)\rangle\}.
\end{aligned}$$

While the algorithm is represented by a path, as is the graph for AND in Section 4.3.3, in contrast to Section 4.3.3, we represent this path in terms of the local constraints $|\leftarrow, \ell-1, \pi_\ell^{-1}(z)\rangle + |\rightarrow, \ell, z\rangle$. That is because, unlike the path in Section 4.3.3, this path is defined sequentially. These local constraints can be generated by calling the relevant permutation, which is one step of the algorithm, whereas one can verify that generating a basis like that in Section 4.3.3, which includes a superposition over the whole path, or whole sequence of algorithm steps, would require running the whole algorithm.

Unlike the other examples in this chapter, to get a quantum walk algorithm, following Example 4.3.4, we need $|\psi_0\rangle = |s\rangle$. We do not have $B = \{s, t\}$ (assuming $|\mathcal{Z}_{\text{acc}}| \neq 1$), so we cannot use canonical boundary here, but it would be possible to modify this subspace graph to have canonical boundary, as follows. We can essentially take two copies of G, $G^{\rightarrow}$ and $G^{\leftarrow}$, and identify the points M in each of the two copies, so that the connected component of s is a line that first goes through $G^{\rightarrow}$ to some vertex in V_T, and then, if it has reached a point in M, goes through $G^{\leftarrow}$ back to the second copy of s, which we could call t. This is like a classical algorithm that computes a final state, copies out the answer bit, and then uncomputes, which is also what we do for quantum algorithms in the next section, except that the answer bit goes into the phase.

Another modification we could make to this subspace graph is as follows. We mentioned that a classical reversible algorithm is like a walk on a line, but G is not a line, though the connected component containing s always is. We can actually modify G to get a subspace graph G' that is a line of length T, where v_ℓ corresponds to the set V_ℓ in G, which better captures our intuition that a deterministic algorithm is a linear process. The edge spaces of G' are obtained by combining those of G: $\Xi'_{e_\ell} = \bigoplus_{z\in\mathcal{Z}} \Xi_{e_{\ell,z}}$, and similarly for $\Xi'^{\mathcal{A}}_{e_\ell}$. The vertex spaces are similarly obtained: $\mathcal{V}'_{v_\ell} = \bigoplus_{z\in\mathcal{Z}} \mathcal{V}_{v_{\ell,z}}$. This is essentially what we do for quantum algorithms in the next section.

4.3.5 Example: quantum algorithms

In this section, we describe a subspace graph from any quantum algorithm. This construction is essentially the same as the one in [Jef22], but we state it here explicitly as a subspace graph. While the construction in [Jef22] applies to *variable-time* quantum algorithms, for simplicity, we will consider algorithms that run for a fixed time T. However, Lemma 4.3.35 below, and all results that use it, also apply for variable-time algorithms, replacing T with the expected running time.

A quantum algorithm is a sequence of unitaries $U_1, \ldots, U_T$ acting on the space

$$H_{\mathcal{Y}} \otimes H_{\mathcal{Z}} = \text{span}\{|a\rangle|z\rangle : a \in \{0,1\}, z \in \mathcal{Z}\}.$$

We assume each unitary can be implemented in unit cost, and moreover, the unitary $\sum_{r=1}^{T} |r\rangle\langle r| \otimes U_r$ can be implemented in unit cost.

The algorithm may implicitly depend on some input $x \in \{0,1\}^n$, for example, by letting some of the unitaries be queries to x (or in any other way, we don't care). We say the algorithm *computes* some $f : \{0,1\}^n \to \{0,1\}$ if

$$U_T \dots U_1 |0,0\rangle \in \operatorname{span}\{|f(x), z\rangle : z \in \mathcal{Z}\}$$

on input x. We are assuming the algorithm has no error for simplicity. In some of our conclusions, we can always substitute an algorithm with error if the error is small enough that the algorithm is indistinguishable from an error-free one. In this section, we will prove the following.

4.3.35. Lemma. *From any quantum algorithm computing some $f : \{0,1\}^n \to \{0,1\}$ with no error in time T, and any positive weights $\{\alpha_r\}$ such that $\alpha_0 = 1$, we can derive an st-composable subspace graph (Definition 4.3.9) G that computes f with $\dim H_G = O(|\mathcal{Z}|T)$, $\hat{W}_+ \le 2\sum_{r=1}^{T} \frac{1}{\alpha_r}$ and $\hat{W}_- \le 2\sum_{r=1}^{T} \alpha_r$; with st-composable working bases that have scaling factor $\mathsf{r} = 2$ and can be generated in time $O(\log(T))$.*

For example, we can set the weights all to 1, and then we get $\hat{W}_+$ and $\hat{W}_-$ both at most $2T$.

In order to remain consistent with [Jef22] so that we can easily use results proven there, while still keeping the boundary of G simple, we are going to assume that $U_1 = -I$, at the expense of potentially making the algorithm one step longer. Thus, we will let $\mathsf{T} = T + 1$, and the algorithm referred to in Lemma 4.3.35 is actually $U_2, \dots, U_{\mathsf{T}}$.

We define a subspace graph $G = (V, E)$ as follows. We will assume T is even, and let:

$$\begin{aligned} V &= \{v^{\rightarrow}_{2\ell-1}, v^{\leftarrow}_{2\ell-1}\}_{\ell=1}^{\mathsf{T}/2} \\ E &= \{e^{\rightarrow}_{2\ell}, e^{\leftarrow}_{2\ell}\}_{\ell=1}^{\mathsf{T}/2-1} \cup \{e^{\leftrightarrow}_{\mathsf{T}}\}, \\ \text{and} \quad B &= \{s := v^{\rightarrow}_1, t := v^{\leftarrow}_1\} \end{aligned}$$

with $e^{\rightarrow}_i = (v^{\rightarrow}_{i-1}, v^{\rightarrow}_{i+1})$ (and similarly for $e^{\leftarrow}_i$), and $e^{\leftrightarrow}_{\mathsf{T}} = (v^{\rightarrow}_{\mathsf{T}-1}, v^{\leftarrow}_{\mathsf{T}-1})$. We define:

$$\begin{aligned} &\forall \ell \in \{1, \dots, \mathsf{T}/2 - 1\}, \\ &\Xi_{e^{\rightarrow}_{2\ell}} := \operatorname{span}\{|\rightarrow\rangle|a,z\rangle|2\ell\rangle, |\rightarrow\rangle|a,z\rangle|2\ell+1\rangle : a \in \{0,1\}, z \in \mathcal{Z}\} \\ &\forall \ell \in \{1, \dots, \mathsf{T}/2 - 1\}, \\ &\Xi_{e^{\leftarrow}_{2\ell}} := \operatorname{span}\{|\leftarrow\rangle|a,z\rangle|2\ell\rangle, |\leftarrow\rangle|a,z\rangle|2\ell+1\rangle : a \in \{0,1\}, z \in \mathcal{Z}\} \\ &\Xi_{e^{\leftrightarrow}_{\mathsf{T}}} := \operatorname{span}\{|\rightarrow\rangle|a,z\rangle|\mathsf{T}\rangle, |\leftarrow\rangle|a,z\rangle|\mathsf{T}\rangle : a \in \{0,1\}, z \in \mathcal{Z}\} \\ &\Xi_s = \Xi_{v^{\rightarrow}_1} := \operatorname{span}\{|s\rangle := |\rightarrow\rangle|0,0\rangle|0\rangle, |\leftarrow, s\rangle := |\rightarrow\rangle|0,0\rangle|1\rangle\} \\ &\Xi_t = \Xi_{v^{\leftarrow}_1} := \operatorname{span}\{|t\rangle := |\leftarrow\rangle|0,0\rangle|0\rangle, |\rightarrow, t\rangle := |\leftarrow\rangle|0,0\rangle|1\rangle\}. \end{aligned} \tag{4.3.12}$$

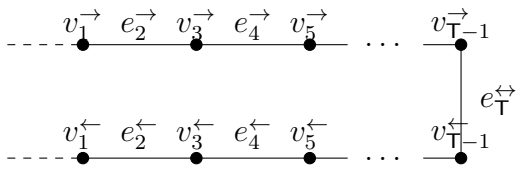

Figure 4.4: The graph G. The dashed lines represent dangling boundary "edges".

Following [Jef22, Definition 3.1], we define the following sets of vectors in $H_G = \Xi_E \oplus \Xi_B$ to use in the construction of our subspaces, for positive weights $\{\alpha_r\}_r$ such that $\alpha_0 = \alpha_1 = 1$.

4.3.36. Definition. The *forward transition states* are defined:

$$\forall r \in \{0, \dots, \mathsf{T}-1\},$$
$$\Psi_r^{\rightarrow} := \left\{ |\psi_{a,z,r}^{\rightarrow}\rangle := |\rightarrow\rangle(\sqrt{\alpha_r}|a,z\rangle|r\rangle - \sqrt{\alpha_{r+1}}U_{r+1}|a,z\rangle|r+1\rangle) : a \in \{0,1\}, z \in \mathcal{Z} \right\}.$$

The *backward transition states* are defined:

$$\forall r \in \{0, \dots, \mathsf{T}-1\},$$
$$\Psi_r^{\leftarrow} := \left\{ |\psi_{a,z,r}^{\leftarrow}\rangle := |\leftarrow\rangle(\sqrt{\alpha_r}|a,z\rangle|r\rangle - \sqrt{\alpha_{r+1}}U_{r+1}|a,z\rangle|r+1\rangle) : a \in \{0,1\}, z \in \mathcal{Z} \right\}.$$

The *reversal states* are defined:

$$\Psi_{\mathsf{T}}^{\leftrightarrow} := \left\{ |\psi_{a,z,\mathsf{T}}^{\leftrightarrow}\rangle := \sqrt{\alpha_{\mathsf{T}}}(|\rightarrow\rangle - (-1)^a|\leftarrow\rangle)|a,z\rangle|\mathsf{T}\rangle : a \in \{0,1\}, z \in \mathcal{Z} \right\}.$$

Finally, let:

$$\Psi_0 = \bigcup_{\ell=0}^{\mathsf{T}/2-1} (\Psi_{2\ell}^{\rightarrow} \cup \Psi_{2\ell}^{\leftarrow}) \cup \Psi_{\mathsf{T}}^{\leftrightarrow} \quad \text{and} \quad \Psi_1 = \bigcup_{\ell=0}^{\mathsf{T}/2} (\Psi_{2\ell-1}^{\rightarrow} \cup \Psi_{2\ell-1}^{\leftarrow}).$$

Note that since we assume $U_1 = -I$, we have:

$$|\psi_{0,0,0}^{\rightarrow}\rangle = |\rightarrow\rangle|0,0\rangle|0\rangle - |\rightarrow\rangle U_1|0,0\rangle|1\rangle = |s\rangle + |\rightarrow\rangle|0,0\rangle|1\rangle = |s\rangle + |\leftarrow, s\rangle,$$

and similarly,

$$|\psi_{0,0,0}^{\leftarrow}\rangle = |\leftarrow\rangle|0,0\rangle|0\rangle + |\leftarrow\rangle|0,0\rangle|1\rangle = |t\rangle + |\rightarrow, t\rangle.$$

Then we define:

$$\begin{aligned}
&\mathcal{V}_{v_1^{\rightarrow}} := \operatorname{span}\{|\psi^{\rightarrow}_{0,0,1}\rangle\} \quad \text{and} \quad \mathcal{V}_{v_1^{\leftarrow}} := \operatorname{span}\{|\psi^{\leftarrow}_{0,0,1}\rangle\} \\
\forall \ell \in \{2,\dots,\mathsf{T}/2\},\ &\mathcal{V}_{v^{\rightarrow}_{2\ell-1}} := \operatorname{span}\Psi^{\rightarrow}_{2\ell-1} \quad \text{and} \quad \mathcal{V}_{v^{\leftarrow}_{2\ell-1}} := \operatorname{span}\Psi^{\leftarrow}_{2\ell-1} \\
\forall \ell \in \{1,\dots,\mathsf{T}/2-1\},\ &\Xi^{\mathcal{A}}_{e^{\rightarrow}_{2\ell}} := \operatorname{span}\Psi^{\rightarrow}_{2\ell} \quad \text{and} \quad \Xi^{\mathcal{A}}_{e^{\leftarrow}_{2\ell}} := \operatorname{span}\Psi^{\leftarrow}_{2\ell} \\
&\Xi^{\mathcal{A}}_{e^{\leftrightarrow}_{\mathsf{T}}} := \operatorname{span}\Psi^{\leftrightarrow}_{\mathsf{T}} \\
&\Xi^{\mathcal{A}}_{s} := \operatorname{span}\{|\psi^{\rightarrow}_{0,0,0}\rangle\} = \operatorname{span}\{|s\rangle + |\leftarrow, s\rangle\} \\
&\Xi^{\mathcal{A}}_{t} := \operatorname{span}\{|\psi^{\leftarrow}_{0,0,0}\rangle\} = \operatorname{span}\{|t\rangle + |\rightarrow, t\rangle\} \\
&\mathcal{V}_B := \operatorname{span}\{|\leftarrow, s\rangle + |\rightarrow, t\rangle\}.
\end{aligned} \tag{4.3.13}$$

For all $e \in E \cup B$, we will have $\Xi^{\mathcal{B}}_e = \{0\}$. Note that the subspace graph we have just defined has canonical *st*-boundary (Definition 4.3.6). As per Definition 4.3.1, we have

$$\begin{aligned}
\mathcal{A}_G &= \Xi^{\mathcal{A}}_s \oplus \Xi^{\mathcal{A}}_t \oplus \bigoplus_{\ell=1}^{\mathsf{T}/2-1} (\Xi^{\mathcal{A}}_{e^{\rightarrow}_{2\ell}} \oplus \Xi^{\mathcal{A}}_{e^{\leftarrow}_{2\ell}}) \oplus \Xi^{\mathcal{A}}_{e^{\leftrightarrow}_{\mathsf{T}}} \\
\mathcal{B}_G &= \bigoplus_{\ell=1}^{\mathsf{T}/2} \left(\mathcal{V}_{v^{\rightarrow}_{2\ell-1}} \oplus \mathcal{V}_{v^{\leftarrow}_{2\ell-1}}\right) + \mathcal{V}_B.
\end{aligned} \tag{4.3.14}$$

In [Jef22], the vectors in Definition 4.3.36 form the working basis, which works perfectly fine, because since all $\Xi^{\mathcal{B}}_e$ are trivial, we have a decomposition of $\mathcal{B}_G$ into orthogonal spaces $\mathcal{V}_v$, so we can simply combine their bases. However, here, we will use a slightly different working basis for $\mathcal{B}_G$ to conform to Definition 2.7.5, as follows.

Let $\bar{\Psi}^{\rightarrow}_r$ contain normalized vectors.

$$\begin{aligned}
\Psi_{\mathcal{A}} = &\left\{ |\bar{\psi}^{\rightarrow}_{0,0,0}\rangle = \frac{1}{\sqrt{2}}(|s\rangle + |\leftarrow, s\rangle), |\bar{\psi}^{\rightarrow}_{0,0,0}\rangle = \frac{1}{\sqrt{2}}(|t\rangle + |\rightarrow, t\rangle) \right\} \\
&\cup \bigcup_{\ell=1}^{\mathsf{T}/2-1} (\bar{\Psi}^{\rightarrow}_{2\ell} \cup \bar{\Psi}^{\leftarrow}_{2\ell}) \cup \bar{\Psi}^{\leftrightarrow}_{\mathsf{T}}.
\end{aligned}$$

Note that

$$|\psi^{\rightarrow}_{0,0,1}\rangle = |\rightarrow\rangle|0,0\rangle|1\rangle - |\rightarrow\rangle U_2|0,0\rangle|2\rangle = |\leftarrow, s\rangle - |\rightarrow\rangle U_2|0,0\rangle|2\rangle$$

and similarly,

$$|\psi^{\leftarrow}_{0,0,1}\rangle = |\rightarrow, t\rangle - |\leftarrow\rangle U_2|0,0\rangle|2\rangle.$$

Define:

$$|b_0\rangle := \frac{1}{\sqrt{2}}\Big(|\psi^{\rightarrow}_{0,0,1}\rangle + |\psi^{\leftarrow}_{0,0,1}\rangle + (|\rightarrow\rangle + |\leftarrow\rangle)U_2|0,0\rangle|2\rangle)\Big) = \frac{1}{\sqrt{2}}(|\leftarrow, s\rangle + |\rightarrow, t\rangle)$$
$$|b_1\rangle := \frac{1}{2}\Big(|\psi^{\rightarrow}_{0,0,1}\rangle - |\psi^{\leftarrow}_{0,0,1}\rangle\Big) = \frac{1}{2}(|\leftarrow, s\rangle - |\rightarrow, t\rangle - (|\rightarrow\rangle - |\leftarrow\rangle)U_2|0,0\rangle|2\rangle)$$
$$|b_2\rangle := \frac{1}{\sqrt{2}}(|\rightarrow\rangle + |\leftarrow\rangle)U_2|0,0\rangle|2\rangle$$

Then we have

$$\operatorname{span}\{|b_0\rangle - |b_2\rangle, |b_1\rangle\} = \operatorname{span}\{|\psi^{\rightarrow}_{0,0,1}\rangle, |\psi^{\leftarrow}_{0,0,1}\rangle\} = \mathcal{V}_{v_1^{\rightarrow}} \oplus \mathcal{V}_{v_1^{\leftarrow}}.$$

Thus

$$\Psi_{\mathcal{B}} = \{|b_0\rangle, |b_1\rangle, |b_2\rangle\} \cup \bigcup_{\ell=2}^{\mathsf{T}/2} (\Psi^{\rightarrow}_{2\ell-1} \cup \Psi^{\leftarrow}_{2\ell-1}) \cup \Psi^{\leftrightarrow}_{\mathsf{T}}$$

is an orthonormal basis for $\mathcal{B}_G$.
The following is given by a trivial modification of [Jef22, Lemma 3.3].

4.3.37. Lemma. *Assuming unit-time access to $\sum_{\ell=1}^{\mathsf{T}} |\ell\rangle\langle\ell| \otimes U_\ell$, the working bases $\Psi_{\mathcal{A}}$ and $\Psi_{\mathcal{B}}$ can be generated in $O(\log(\mathsf{T}))$ complexity.*

We will also use the following states from [Jef22, Definition 3.4] in the construction of our witnesses:

4.3.38. Definition (Algorithm states). Define the following *algorithm states* in H_G:

$$|w^0\rangle = |0\rangle, \quad \forall t \in [\mathsf{T}],\ |w^t\rangle = U_t|w^{t-1}\rangle.$$

The *positive history state* of the algorithm is defined:

$$|w_+\rangle = (|\rightarrow\rangle + (-1)^{f(x)}|\leftarrow\rangle) \sum_{t=0}^{\mathsf{T}} \frac{1}{\sqrt{\alpha_t}}|w^t\rangle|t\rangle.$$

The *negative history state* of the algorithm is defined:

$$|w_-\rangle = (|\rightarrow\rangle - (-1)^{f(x)}|\leftarrow\rangle) \sum_{t=0}^{\mathsf{T}} \sqrt{\alpha_t}(-1)^t|w^t\rangle|t\rangle.$$

We will use the following properties from [Jef22], using the fact that the algorithm has no error:

4.3.39. Claim. *1. $\||w_-\rangle\|^2 = 2\sum_{t=0}^{\mathsf{T}} \alpha_t$ (Corollary 3.7)*

2. $\||w_+\rangle\|^2 = 2\sum_{t=0}^{\mathsf{T}} \frac{1}{\alpha_t}$ *(Corollary 3.7)*
3. $|w_+\rangle \in \mathrm{span}\Psi_0^\perp \cap \mathrm{span}\Psi_1^\perp$ *(Claim 3.8).*
4. $|w_-\rangle \in \mathrm{span}\Psi_0$ *and* $|w_-\rangle - \underbrace{(|\rightarrow\rangle - (-1)^{f(x)}|\leftarrow\rangle)|0\rangle|0\rangle}_{|s\rangle-(-1)^{f(x)}|t\rangle} \in \mathrm{span}\Psi_1$ *(Claim 3.9).*

Items (1) and (2) are rather obvious in the case of fixed time quantum algorithms that we are specializing to here in order to avoiding defining variable-time quantum algorithms. However, we remark that in [Jef22], the more general case of variable-time algorithms is considered, so T is allowed to be a random variable, and the expressions in (1) and (2) are replaced by expectations. The construction in this section, which again, is just putting the construction of [Jef22] into the language of subspace graphs, would be the same in this more general case, and thus our composition results also hold for this more general kind of algorithm.

We have the following corollaries, which, combined with Lemma 4.3.37, prove Lemma 4.3.35.

4.3.40. Corollary (Positive analysis). *When $f(x) = 1$, $|w_+\rangle$ is a positive witness for G (see Definition 4.3.10) with*

$$\||\hat{w}_+\rangle\|^2 = 2\sum_{\ell=2}^{\mathsf{T}} \frac{1}{\alpha_\ell}.$$

Proof:
When $f(x) = 1$, since $\alpha_0 = \alpha_1 = 1$ and $|w^0\rangle = |0,0\rangle$ and $|w^1\rangle = U_1|0,0\rangle$,

$$|w_+\rangle = \underbrace{|\rightarrow\rangle|0,0\rangle|0\rangle}_{|e\rangle} - \underbrace{|\leftarrow\rangle|0,0\rangle|0\rangle}_{|t\rangle} + \underbrace{|\rightarrow\rangle U_1|0,0\rangle|1\rangle}_{-|\leftarrow,s\rangle} - \underbrace{|\leftarrow\rangle U_1|0,0\rangle|1\rangle}_{-|\rightarrow,t\rangle} + \underbrace{(|\rightarrow\rangle + (-1)^{f(x)}|\leftarrow\rangle)\sum_{\ell=2}^{\mathsf{T}} \frac{1}{\sqrt{\alpha_\ell}}|w^\ell\rangle|\ell\rangle}_{|\hat{w}_+\rangle}.$$

By inspection (see (4.3.14) and Definition 4.3.36), we have $\mathcal{A}_G \subseteq \mathrm{span}\Psi_0$ and $\mathcal{B}_G \subseteq \mathrm{span}\Psi_1 \cup \{|b_0\rangle\}$. By Claim 4.3.39, $|w_+\rangle \in \mathrm{span}\Psi_0^\perp \cap \mathrm{span}\Psi_1^\perp \subset \mathcal{A}_G^\perp \cap \mathcal{B}_G^\perp$, and we can see that $|w_+\rangle$ is orthogonal to $|b_0\rangle = \frac{1}{\sqrt{2}}(|\leftarrow, s\rangle + |\rightarrow, t\rangle)$. Thus $|w_+\rangle$ is a positive witness, and

$$\||\hat{w}_+\rangle\|^2 = \|(I - \Pi_B)|w_+\rangle\|^2 = 2\sum_{\ell=2}^{\mathsf{T}} \frac{1}{\alpha_\ell}.$$

□

4.3.41. Corollary (Negative analysis). *When $f(x) = 0$, $|w_-\rangle$ is a negative witness for G (see Definition 4.3.11) with*

$$\||\hat{w}_-\rangle\|^2 = 2\sum_{\ell=2}^{\mathsf{T}} \alpha_\ell.$$

Thus $\hat{W}_-(G) \leq 2\sum_{\ell=2}^{\mathsf{T}} \alpha_\ell$.

Proof:
By Claim 4.3.39, we have $|w_+\rangle \in \mathrm{span}\Psi_0$. Though Ψ_0 is slightly bigger than $\mathcal{A}_G$, since we also have $|w_+^0\rangle = |0,0\rangle$ and $|w_+^1\rangle = -|0,0\rangle$, we can see that $|w_+\rangle \in \mathcal{A}_G$. Similarly, it is not difficult to conclude that when $f(x) = 0$, $|w_-\rangle - (|s\rangle - |t\rangle) \in \mathcal{B}_G$, since it's in Ψ_1 (Claim 4.3.39), and its projection onto $|0\rangle$ or $|1\rangle$ in the last register is in $\mathrm{span}\Psi_0^{\rightarrow} \cup \Psi_0^{\leftarrow} \cup \Psi_1^{\rightarrow} \cup \Psi_1^{\leftarrow}$. □

4.4 Recursion with subspace graphs

Subspace graphs lend themselves well to very general kinds of recursion. We can compose subspace graphs, say G_1, G_2 and G_3, as in Figure 4.5, by identifying some of the vertices on their boundaries. In full generality, this may result in a subspace graph that is difficult to analyze. For example, if we replace two parallel edges with subspace graphs derived from quantum algorithms, "flow" along these edges may have complex phases that interfere on the other side. Our nice classical intuition of flows breaks down in the fully general case, which is not surprising, since quantum algorithms are not classical. However, an important question is in which special cases, and to what extent, we can compose subspace graphs and keep enough classical intuition to analyze them.

One special case is implicit in [Jef22], and here we present another special case, which we call *switch composition*. Switch composition generalizes a very simple kind of recursion that can be done in switching networks. Since an st-path (or more generally, flow) is allowed to use an edge if and only if $\varphi(e) = 1$, we can replace it with a switching network G^e for some function f_e, which will have an st-path from one endpoint to the other if and only if $f_e(x) = 1$. We can view this as removing e from the graph, and then adding its endpoints to the boundary, to then be identified with the boundary $\{s,t\}$ of G^e. By applying this type of composition, the OR and AND switching networks in Section 4.3.3 and Section 4.3.3 can be combined to make switching networks for any Boolean formula [JK17].

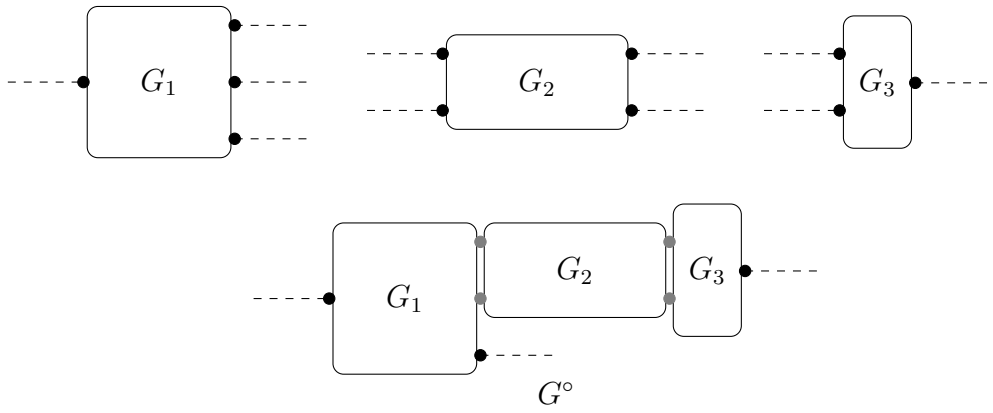

Figure 4.5: Some perhaps complicated graphs G_1, G_2 and G_3, but we need only consider their boundaries, and can abstract their internal structure. To compose them, we identify points on their boundaries, yielding a new graph, G°.

4.4.1 Switch composition

Here we investigate how to compose subspace graphs with specific properties that generalize switching networks – *st*-composable subspace graphs (Definition 4.3.9) – into the switches of a graph G. Specifically, if $\overline{E}$ is the set of edges of G that are switches, we will replace $e \in \overline{E}$ with an *st*-composable graph G^e. We can assume without loss of generality that we replace every $e \in \overline{E}$ with some G^e, since letting G^e be the graph consisting of a single edge from s to t is just like not replacing e. In particular, we will prove the following theorem.

4.4.1. THEOREM. *Let G be an st-composable subspace graph (Definition 4.3.9) with st-composable working bases that can be generated in time T, with scaling factor r. For each $e \in \overline{E}$, the set of switches of G, let G^e be an st-composable subspace graph with st-composable working bases that can be generated in time at most T', with scaling factor r^e. Then there exists an st-composable subspace graph G° with $\dim H_{G^\circ} \leq \dim H_G - 2|\overline{E}| + \sum_{e\in\overline{E}} \dim H_{G^e}$ such that:*

- *G° has st-composable working bases that can be generated in time $T+T'+O(1)$, with scaling factor r.*
- *If $|\hat{w}\rangle$ is a cropped positive witness for G (Definition 4.3.10), and for each $e \in \overline{E}$ such that $\langle \rightarrow, e|\hat{w}\rangle \neq 0$, $|\hat{w}^e\rangle$ is a cropped positive witness for G^e, then*

$$|\hat{w}^\circ\rangle = \sum_{e\in\overline{E}} \sqrt{\frac{\mathsf{r}^e}{2}} \langle \rightarrow, e|\hat{w}\rangle |\hat{w}^e\rangle + \Pi_{E\setminus\overline{E}}|\hat{w}\rangle$$

is a cropped positive witness for G°.

- *If $|\hat{w}_{\mathcal{A}}\rangle$ is a cropped negative witness for G (Definition 4.3.11), and for each $e \in \overline{E}$ such that $\langle \rightarrow, e|\hat{w}_{\mathcal{A}}\rangle \neq 0$, $|\hat{w}^e_{\mathcal{A}}\rangle$ is a cropped negative witness for G^e, then*

$$|\hat{w}^\circ_{\mathcal{A}}\rangle = \sum_{e\in\overline{E}} \sqrt{\frac{2}{\mathsf{r}^e}} \langle \rightarrow, e|\hat{w}_{\mathcal{A}}\rangle |\hat{w}^e_{\mathcal{A}}\rangle + \Pi_{E\setminus\overline{E}}|\hat{w}_{\mathcal{A}}\rangle$$

is a cropped negative witness for G°.

It follows, using the observation that $|\hat{w}\rangle$ must only overlap switches that are on, and $|\hat{w}_{\mathcal{A}}\rangle$ must only overlap switches tht are off, that if G computes f and each G^e computes f^e, the function computed by G° is the composed function $f \circ (f_e)_{e\in\overline{E}}$ (see Section 2.3).

Before we prove Theorem 4.4.1, we note the following useful corollary of Theorem 4.4.1 and Lemma 4.3.20.

4.4.2. Corollary. *Let G be an st-composable subspace graph with st-composable working bases that can be generated in time T that computes f. Then there is an st-composable subspace graph G° with st-composable bases that can be generated in time $T + O(1)$ that computes f, and such that $\dim H_{G^\circ} = \dim H_G + O(1)$, $\hat{W}_+(G^\circ) \leq 1$ and $\hat{W}_-(G^\circ) \leq \hat{W}_+(G)\hat{W}_-(G)$.*

Proof:
Let $G_{\text{OR},1}$ be the graph from Lemma 4.3.20 with $d = 1$, and $\mathsf{w}_1 = \frac{\mathsf{r}}{2}\hat{W}_+(G)$, where r is the scaling factor of the basis for G. It is *st*-composable since it is a switching network. This graph has a single edge, e_1, and we will compose G into this edge using Theorem 4.4.1, so G_{OR} plays the role of G in Theorem 4.4.1, and G plays the role of G^{e_1}. If $f(x) = 1$, then G has a cropped positive witness $|\hat{w}^1\rangle$, and using the positive witness from Lemma 4.3.20, by Theorem 4.4.1, G° has cropped positive witness

$$|\hat{w}^\circ\rangle = \sqrt{\frac{\mathsf{r}}{2}}\frac{1}{\sqrt{\mathsf{w}_1}}|\hat{w}^1\rangle = \frac{1}{\sqrt{\hat{W}_+(G)}}|\hat{w}^1\rangle,$$

so since $\||\hat{w}^1\rangle\|^2 \leq \hat{W}_+(G)$, $\hat{W}_+(G^\circ) \leq 1$.

On the other hand, if $f(x) = 0$, then G has a cropped negative witness $|\hat{w}^1_{\mathcal{A}}\rangle$, and using the negative witness for Lemma 4.3.20, by Theorem 4.4.1, G° has cropped negative witness

$$|\hat{w}^\circ_{\mathcal{A}}\rangle = \sqrt{\frac{2}{\mathsf{r}}}\sqrt{\mathsf{w}_1}|\hat{w}^1_{\mathcal{A}}\rangle = \sqrt{\hat{W}_+(G)}|\hat{w}^1_{\mathcal{A}}\rangle,$$

so $\hat{W}_-(G^\circ) \leq \hat{W}_+(G)\hat{W}_-(G)$.

To complete the proof, by Lemma 4.3.20, the basis for G_{OR} can be generated in unit time, and thus by Theorem 4.4.1, the basis for G° can be generated in time

$T + O(1)$. □

In the remainder of this section, we prove Theorem 4.4.1.

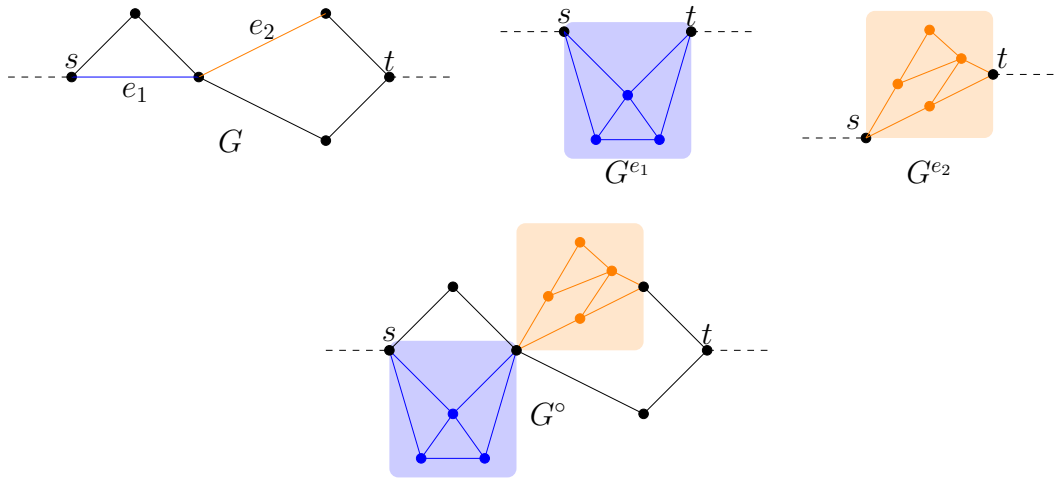

Figure 4.6: An example of replacing edges e_1 and e_2 in G with graphs G^{e_1} and G^{e_2} to obtain G°. Boundary "edges" are represented by dashed lines. Switching networks already lend themselves to this type of recursion: we can replace an edge with a 2-terminal graph, and then the "edge" is traversable if and only if the terminals are connected. The difference with the more general recursion in Theorem 4.4.1 is that the subspace graphs used to replace edges (as well as other parts of G) might have more complicated structure than just their graph structure; for example, they might encode quantum algorithms like in Section 4.3.5.

Informally, we obtain G° from G by, for each $e \in \overline{E}$, removing the edge e, and identifying its endpoints with the vertices $s = s^e$ and $t = t^e$ of the graph G^e. This is illustrated in Figure 4.6. The subspaces associated with G° are mostly inherited from G and G^e, except for those on the glued boundaries. There, intuitively, we replace $|\rightarrow, e\rangle$ with the edges incident to $s = s^e$ in G^e, and $|\leftarrow, e\rangle$ with the edges incident to $t = t^e$ in G^e. This is formalized by the map $\widetilde{\Lambda}$ defined in (4.4.5).

Definition of the subspace graph G° We now formally define G° and its associated spaces. We will shortly define a working basis $\Psi_{\mathcal{B}_{G^\circ}}$ for $\mathcal{B}_{G^\circ}$, from working bases for $\mathcal{B}_G$ and $\mathcal{B}_{G^e}$. It is actually these bases that are most important, but by defining $\mathcal{B}_{G^\circ}$ and $\mathcal{A}_{G^\circ}$ as being built up from local spaces, we show that the graph structure and intuition are preserved by the composition.

The composed graph G° has vertex and edge sets:

$$\begin{aligned} V^\circ &= V \sqcup \bigsqcup_{e\in\overline{E}} (V^e \setminus B^e) = V \sqcup \bigsqcup_{e\in\overline{E}} (V^e \setminus \{s^e, t^e\}) \\ E^\circ &= (E \setminus \overline{E}) \sqcup \bigsqcup_{e\in\overline{E}} E^e \end{aligned} \tag{4.4.1}$$

and boundary

$$B^\circ = B = \{s, t\}.$$

We identify s^e and t^e in G^e with the endpoints of e in G, so that the following edges are incident to $u \in V^\circ$:

$$E^\circ(u) = \begin{cases} E^e(u), & \text{if } u \in V^e \setminus \{s^e, t^e\}, \\ (E(u) \setminus \overline{E}) \sqcup \bigsqcup_{e\in\overline{E}\cap E^\rightarrow(u)} E^e(s^e) \\ \qquad\qquad \sqcup \bigsqcup_{e\in\overline{E}\cap E^\leftarrow(u)} E^e(t^e), & \text{if } u \in V. \end{cases} \tag{4.4.2}$$

Recall that we assume for convenience that each edge has an orientation, and $E^\rightarrow(u) \subseteq E(u)$ are the edges oriented outwards from u, and $E^\leftarrow(u) = E(u) \setminus E^\rightarrow(u)$ those oriented inwards. This is simply to decide which of the endpoints of e is identified with s^e and which with t^e.

We define the edge spaces and edge subspaces of G° from those of G and G^e as follows:

$$\Xi^\circ_{e'} = \begin{cases} \Xi_{e'} & \text{if } e' \in (E \setminus \overline{E}) \cup B \\ \Xi^e_{e'} & \text{if } e' \in E^e, \end{cases} \tag{4.4.3}$$

and similarly for $\Xi^{\circ\mathcal{A}}_e$ and $\Xi^{\circ\mathcal{B}}_e$. Note that since G has canonical st-boundary, so does G°. Furthermore, since the switches and non-switches of G° are:

$$\overline{E}^\circ = \bigsqcup_{e\in\overline{E}} \overline{E}^e \text{ and } E^\circ \setminus \overline{E}^\circ = (E \setminus \overline{E}) \sqcup \bigsqcup_{e\in\overline{E}} (E^e \setminus \overline{E}^e),$$

we inherit the property that for all $e \in E^\circ \setminus \overline{E}^\circ$, $\Xi^{\circ\mathcal{B}}_e = \{0\}$.

Next we define the vertex spaces. First, for each $e \in \overline{E}$, we are assuming that $\mathcal{V}^e_{s^e}$ and $\mathcal{V}^e_{t^e}$ can be written:

$$\begin{aligned} \mathcal{V}^e_{s^e} &= \operatorname{span}\{|\leftarrow, s^e\rangle + |\psi_\star(s^e)\rangle\} \\ \mathcal{V}^e_{t^e} &= \operatorname{span}\{|\rightarrow, t^e\rangle + |\psi_\star(t^e)\rangle\}, \end{aligned} \tag{4.4.4}$$

for some $|\psi_\star(s^e)\rangle \in \Xi^e_{E^e(s^e)}$ and $|\psi_\star(t^e)\rangle \in \Xi^e_{E^e(t^e)}$. Then define a linear map $\widetilde{\Lambda}$ on H_G by:

$$\widetilde{\Lambda} = \sum_{e\in\overline{E}} \frac{1}{\sqrt{\mathsf{r}^e}} (|\psi_\star(s^e)\rangle\langle \rightarrow, e| + |\psi_\star(t^e)\rangle\langle \leftarrow, e|) + \Pi_{E\setminus\overline{E}} + \Pi_B, \tag{4.4.5}$$

and let

$$\mathcal{V}^\circ_u = \begin{cases} \widetilde{\Lambda}(\mathcal{V}_u) & \text{if } u \in V \\ \mathcal{V}^e_u & \text{if } u \in V^e \setminus \{s^e, t^e\}. \end{cases} \tag{4.4.6}$$

The following lemma shows that G° is a well-defined subspace graph.

4.4.3. LEMMA. $\widetilde{\Lambda}(\mathcal{V}_u) \subseteq \Xi^\circ_{E^\circ(u)}$.

Proof:
We have $\mathcal{V}_u \subseteq \Xi_{E(u)}$, so

$$\widetilde{\Lambda}(\mathcal{V}_u) = \widetilde{\Lambda}\Big(\Xi_{E(u)\setminus\overline{E}} \cap \mathcal{V}_u\Big) \oplus \bigoplus_{e\in\overline{E}\cap E(u)} \widetilde{\Lambda}(\Xi_e \cap \mathcal{V}_u),$$

We have $\widetilde{\Lambda}\Big(\Xi_{E(u)\setminus\overline{E}} \cap \mathcal{V}_u\Big) = \Xi_{E(u)\setminus\overline{E}} \cap \mathcal{V}_u$, since $\widetilde{\Lambda}$ fixes $\Xi_{E\setminus\overline{E}}$. Recall from Definition 4.3.5 that for $e \in \overline{E} \cap E(u)$

$$\Xi_e \cap \mathcal{V}_u = \begin{cases} \operatorname{span}\{|\rightarrow, e\rangle\} & \text{if } e \in E^{\rightarrow}(u) \\ \operatorname{span}\{|\leftarrow, e\rangle\} & \text{if } e \in E^{\leftarrow}(u). \end{cases}$$

Thus,

$$\widetilde{\Lambda}(\Xi_e \cap \mathcal{V}_u) = \begin{cases} \operatorname{span}\{\widetilde{\Lambda}|\rightarrow, e\rangle\} = \operatorname{span}\{|\psi_\star(s^e)\rangle\} \subseteq \Xi^e_{E^e(s^e)} & \text{if } e \in E^{\rightarrow}(u) \\ \operatorname{span}\{\widetilde{\Lambda}|\leftarrow, e\rangle\} = \operatorname{span}\{|\psi_\star(t^e)\rangle\} \subseteq \Xi^e_{E^e(t^e)} & \text{if } e \in E^{\leftarrow}(u). \end{cases}$$

Thus

$$\widetilde{\Lambda}(\mathcal{V}_u) \subseteq \Xi_{E(u)\setminus\overline{E}} \oplus \bigoplus_{e\in\overline{E}\cap E^{\rightarrow}(u)} \Xi^e_{E^e(s^e)} \oplus \bigoplus_{e\in\overline{E}\cap E^{\leftarrow}(u)} \Xi^e_{E^e(t^e)} = \Xi_{E^\circ(u)}$$

by (4.4.2). □

From these definitions, we conclude:

$$\begin{aligned} \mathcal{A}_{G^\circ} &= \bigoplus_{e\in E^\circ} \Xi^{\circ\mathcal{A}}_e \oplus \bigoplus_{u\in B^\circ} \Xi^{\circ\mathcal{A}}_u = \bigoplus_{e\in E\setminus\overline{E}} \Xi^{\mathcal{A}}_e \oplus \bigoplus_{e\in\overline{E}} \bigoplus_{e'\in E^e} \Xi^{e\mathcal{A}}_{e'} \oplus \Xi^{\mathcal{A}}_s \oplus \Xi^{\mathcal{A}}_t \\ &= \bigoplus_{e\in E\setminus\overline{E}} \Xi^{\mathcal{A}}_e \oplus \bigoplus_{e\in\overline{E}} \bigoplus_{e'\in E^e} \Xi^{e\mathcal{A}}_{e'} \oplus \Xi^{\mathcal{A}}_s \end{aligned} \tag{4.4.7}$$

and similarly

$$\begin{aligned} \mathcal{B}_{G^\circ} &= \bigoplus_{u\in V^\circ} \mathcal{V}^\circ_u + \bigoplus_{e\in E\setminus\overline{E}} \Xi^{\mathcal{B}}_e \oplus \bigoplus_{e\in\overline{E}} \bigoplus_{e'\in E^e} \Xi^{e\mathcal{B}}_{e'} \oplus \Xi^{\mathcal{B}}_s \oplus \Xi^{\mathcal{B}}_t \\ &= \bigoplus_{u\in V} \widetilde{\Lambda}(\mathcal{V}_u) \oplus \bigoplus_{e\in\overline{E}} \bigoplus_{u\in V^e\setminus\{s,t\}} \mathcal{V}^e_u + \oplus \bigoplus_{e\in\overline{E}} \bigoplus_{e'\in\overline{E}^e} \Xi^{e\mathcal{B}}_{e'}. \end{aligned} \tag{4.4.8}$$

Working bases for G° By the assumptions of Definition 4.3.8, we have a working basis $\Psi_{\mathcal{A}_G}$ for $\mathcal{A}_G$ of the form:

$$\Psi_{\mathcal{A}_G} = \bigcup_{e \in E \cup B} \Psi_{\mathcal{A}_G}(e),$$

where $\Psi_{\mathcal{A}_G}(e)$ is an orthonormal basis for $\Xi_e^{\mathcal{A}}$, and similarly, for each $e \in \overline{E}$, we have a working basis $\Psi_{\mathcal{A}_{G^e}}$ for $\mathcal{A}_{G^e}$ of the form:

$$\Psi_{\mathcal{A}_{G^e}} = \bigcup_{e' \in E^e \cup B^e} \Psi_{\mathcal{A}_{G^e}}(e').$$

Since the spaces $\Xi_e^{\circ\mathcal{A}}$ are simply inherited from the corresponding spaces in G and the G^e, their bases can be as well. Specifically, we define a working basis

$$\Psi_{\mathcal{A}_{G^\circ}} = \bigcup_{e \in E^\circ \cup B^\circ} \Psi_{\mathcal{A}_{G^\circ}}(e) = \bigcup_{e \in (E \setminus \overline{E}) \cup B} \Psi_{\mathcal{A}_{G^\circ}}(e) \cup \bigcup_{e \in \overline{E}} \bigcup_{e' \in E^e} \Psi_{\mathcal{A}_{G^\circ}}(e').$$

for $\mathcal{A}_{G^\circ}$, where for all $e \in E \setminus \overline{E}$, $\Psi_{\mathcal{A}_{G^\circ}}(e) = \Psi_{\mathcal{A}_G}(e)$, and for all $e \in \overline{E}$ and $e' \in E^e$, $\Psi_{\mathcal{A}_{G^\circ}}(e') = \Psi_{\mathcal{A}_{G^e}}(e')$. Then we have the following.

4.4.4. Lemma (Working basis for $\mathcal{A}$). *$\Psi_{\mathcal{A}_{G^\circ}}$ can be generated in time $T+T'+O(1)$.*

Proof:
Assume without loss of generality that for $e \in \overline{E}$, $\Psi_{\mathcal{A}_G}(e)$, which is one-dimensional, has label set $L(e) = \{e\}$. We let

$$L^\circ_{\mathcal{A}} = (\{0\} \times (L \setminus \overline{E})) \sqcup \bigsqcup_{e \in \overline{E}} \{e\} \times L^e_{\mathcal{A}},$$

so to map $|\ell\rangle \mapsto |b^\circ_\ell\rangle$ for $\ell = (\ell_0, \ell_1) \in L^\circ$, we simply need to apply the relevant basis generation map depending on the value of $\ell_0 \in \{0\} \cup \overline{E}$. We can check if $\ell = (\ell_0, \ell_1) \in L^\circ_{\mathcal{A}}$ by membership in the relevant set, controlled on ℓ_0. For example, if $\ell_0 = 0$, we check if $\ell_1 \in L$, and $\ell_1 \notin \overline{E}$. □

By the assumptions of Definition 4.3.8, we have a working basis $\Psi_{\mathcal{B}_G}$ for $\mathcal{B}_G$ of the form

$$\Psi_{\mathcal{B}_G} = \Psi^-_{\mathcal{B}_G} \cup \left\{ \frac{1}{\sqrt{2}}(|\rightarrow, e\rangle + |\leftarrow, e\rangle) : e \in \overline{E} \right\},$$

and similarly, for each $e \in \overline{E}$, we have a working basis $\Psi_{\mathcal{B}_{G^e}}$ for $\mathcal{B}_{G^e}$ of the form

$$\Psi_{\mathcal{B}_{G^e}} = \Psi^-_{\mathcal{B}_{G^e}} \cup \left\{ \frac{1}{\sqrt{2}}(|\rightarrow, e'\rangle + |\leftarrow, e'\rangle) : e' \in \overline{E}^e \right\},$$

where $\Psi^-_{\mathcal{B}_{G^e}} \subseteq H^-_{G^e} := \Xi_{B^e \cup E^e \setminus \overline{E}^e} \oplus \mathrm{span}\{|\rightarrow, e\rangle - |\leftarrow, e\rangle : e \in \overline{E}^e\}$ contains two distinguished vectors, $|b^e_0\rangle$ and $|b^e_1\rangle$. From these, we define the following isometry on H^-_G:

$$\Lambda := \Pi_{E\setminus\overline{E}} + \Pi_B + \sum_{e\in\overline{E}} \frac{1}{\sqrt{2}} |\bar{b}^e_1\rangle(\langle \rightarrow, e| - \langle \leftarrow, e|). \tag{4.4.9}$$

Then we define:

$$\begin{aligned}\Psi_{\mathcal{B}^\circ_G} = &\underbrace{\Lambda(\Psi^-_{\mathcal{B}_G}) \cup \bigcup_{e\in\overline{E}} (\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b^e_0\rangle, |b^e_1\rangle\})}_{=:\Psi^-_{\mathcal{B}_{G^\circ}}} \\ &\cup \left\{ \underbrace{\frac{1}{\sqrt{2}}(|\rightarrow, e'\rangle + |\leftarrow, e'\rangle)}_{\Psi_{\mathcal{B}_{G^\circ}}(e'),\ \text{spans}\ \Xi^{e\mathcal{B}}_{e'}} : e' \in \overline{E}^\circ = \bigcup_{e\in\overline{E}} \overline{E}^e \right\}.\end{aligned} \tag{4.4.10}$$

In Appendix 4.A, we prove that $\Psi_{\mathcal{B}_{G^\circ}}$ is indeed a basis for $\mathcal{B}_{G^\circ}$. Again, this is actually not so important – we could also have defined $\mathcal{B}_{G^\circ}$ from the basis, in which case, we would not need the assumption, from canonical *st*-boundary, that the spaces $\mathcal{V}^e_{s^e}$ and $\mathcal{V}^e_{t^e}$ are one-dimensional. However, then it would not be obvious that any graph structure remains, which we feel is beneficial for maintaining intuition when working with complicated objects.

4.4.5. Lemma. *$\Psi_{\mathcal{A}_{G^\circ}}$ and $\Psi_{\mathcal{B}_{G^\circ}}$ are st-composable bases as in Definition 4.3.8.*

Proof:
The first two properties are clear, so we check properties 3-5. Since $\Psi_{\mathcal{B}_G}$ is composable, $\Psi^-_{\mathcal{B}_G}$ contains $|b_0\rangle = \frac{1}{\sqrt{2}}(|\leftarrow, s\rangle + |\rightarrow, t\rangle)$, so $\Psi^-_{\mathcal{B}_{G^\circ}}$ contains

$$|b^\circ_0\rangle = \Lambda(|b_0\rangle) = \frac{1}{\sqrt{2}}(|\leftarrow, s\rangle + |\rightarrow, t\rangle),$$

establishing property 3 of Definition 4.3.8. Similarly, $\Psi^-_{\mathcal{B}_G}$ contains $|b_1\rangle = \frac{1}{\sqrt{2+\mathsf{r}}}(|\leftarrow, s\rangle - |\rightarrow, t\rangle + \sqrt{\mathsf{r}}|\bar{b}_1\rangle)$, so $\Psi^-_{\mathcal{B}_{G^\circ}}$ contains

$$|b^\circ_1\rangle = \Lambda(|b_1\rangle) = \frac{1}{\sqrt{2+\mathsf{r}}}(|\leftarrow, s\rangle - |\rightarrow, t\rangle + \sqrt{\mathsf{r}}\Lambda|\bar{b}_1\rangle),$$

establishing property 4 of Definition 4.3.8. Finally, it is easy to verify that the remaining vectors of $\Psi_{\mathcal{B}_{G^\circ}}$ are orthogonal to $|\leftarrow, s\rangle$ and $|\rightarrow, t\rangle$, establishing property 5. □

4.4.6. Lemma. *If the working basis $\Psi_{\mathcal{B}_G}$ has time complexity T, and for each $e \in \overline{E}$, the working basis $\Psi_{\mathcal{B}_{G^e}}$ has time complexity T', then $\Psi_{\mathcal{B}_{G^\circ}}$ has time complexity $T + T' + O(1)$.*

Proof:
We have

$$L_\circ = \underbrace{L^- \sqcup \bigsqcup_{e \in \overline{E}} L_e^- \setminus \{0,1\}}_{L_\circ^-} \sqcup \overline{E}^\circ.$$

As always, for any label $e \in \overline{E}^\circ$, the mapping:

$$|e\rangle \mapsto |b_e^\circ\rangle = \frac{1}{\sqrt{2}}(|\rightarrow, e\rangle + |\leftarrow, e\rangle)$$

has time complexity 1 (just apply a Hadamard).

For $\ell \in L_e^- \setminus \{0,1\}$ for some e, apply the map $|\ell\rangle \mapsto |b_\ell^e\rangle$ in cost T'.

For $\ell \in L^-$, apply the map $|\ell\rangle \mapsto |b_\ell\rangle$ in cost T, and then use the direct sum of basis maps for $\bar{\Psi}_{\mathcal{B}_{G^e}}^-$ to map:

$$\frac{1}{\sqrt{2}}(|\rightarrow, e\rangle - |\leftarrow, e\rangle) \mapsto |1, e\rangle \mapsto |\bar{b}_1^e\rangle.$$

□

We complete the analysis by establishing positive and negative witnesses.

4.4.7. Lemma. *Let $|w\rangle$ be a positive witness for G, so it necessarily only overlaps switches that are on. For every switch e that is on, let $|w^e\rangle$ be a positive witness for G^e. Then*

$$|\hat{w}^\circ\rangle = \sum_{e \in \overline{E}} \sqrt{\frac{\mathsf{r}^e}{2}} \langle \rightarrow, e|w\rangle |\hat{w}^e\rangle + \Pi_{E \setminus \overline{E}} |w\rangle$$

is a positive witness for G°.

Proof:
We need to show that $|w^\circ\rangle = |s\rangle + |\leftarrow, s\rangle + |\hat{w}^\circ\rangle + |\rightarrow, t\rangle + |t\rangle$ is in $\mathcal{A}_{G^\circ}^\perp \cap \mathcal{B}_{G^\circ}^\perp$. We start by showing that $|w^\circ\rangle$ is orthogonal to everything in $\mathcal{A}_{G^\circ}$. Orthogonality with $\Xi_s^{\mathcal{A}} = \mathrm{span}\{|s\rangle - |\leftarrow, s\rangle\}$ (see Definition 4.3.6) is obvious. For $e \in E \setminus \overline{E}$,

$$\Pi_e^{\mathcal{A}} |w^\circ\rangle = \Pi_e^{\mathcal{A}} \Pi_{E \setminus \overline{E}} |w\rangle = \Pi_e^{\mathcal{A}} |w\rangle = 0$$

since $|w\rangle \in \mathcal{A}_G^\perp$ (because it is a positive witness). For all $e \in \overline{E}$ and $e' \in E^e$,

$$\Pi_{e'}^{e\mathcal{A}} |w^\circ\rangle = \sqrt{\frac{\mathsf{r}^e}{2}} \langle \rightarrow, e|w\rangle \Pi_{e'}^{e\mathcal{A}} |\hat{w}^e\rangle = \sqrt{\frac{\mathsf{r}^e}{2}} \langle \rightarrow, e|w\rangle \Pi_{e'}^{e\mathcal{A}} |w^e\rangle = 0$$

since $|\hat{w}^e\rangle = (I - \Pi_{B^e})|w^e\rangle$, and $\Pi^{e\mathcal{A}}_{e'}|w^e\rangle = 0$ because $|w^e\rangle$ is a positive witness for G^e. Thus $|w^\circ\rangle \in \mathcal{A}^\perp_{G^\circ}$.

By an identical proof, we can show that $|w^\circ\rangle$ is orthogonal to $\Xi^{\mathcal{B}}_e$ for all $e \in E\setminus\overline{E}$, and $\Xi^{e\mathcal{B}}_{e'}$ for all $e \in \overline{E}$ and $e' \in E^e$. Thus, it remains only to show orthogonality with $\Psi^-_{\mathcal{B}_{G^\circ}}$.

Note that since $|b_1\rangle \propto |\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle + \sqrt{\mathsf{r}^e}|\bar{b}^e_1\rangle \in \mathcal{B}_{G^e}$, we must have

$$\Big(\langle \leftarrow, s^e| - \langle \rightarrow, t^e| + \sqrt{\mathsf{r}^e}\langle \bar{b}^e_1|\Big)(|s^e\rangle - |\leftarrow, s^e\rangle + |\hat{w}^e\rangle + |\rightarrow, t^e\rangle - |t^e\rangle) = 0$$
$$\sqrt{\mathsf{r}^e}\langle \bar{b}^e_1|\hat{w}^e\rangle = 2.$$

Thus:

$$\begin{aligned}
\Lambda^\dagger|\hat{w}^\circ\rangle &= \sum_{e\in\overline{E}} \sqrt{\frac{\mathsf{r}^e}{2}}\langle \rightarrow, e|w\rangle \Lambda^\dagger|\hat{w}^e\rangle + \Pi_{E\setminus\overline{E}}|w\rangle \\
&= \sum_{e\in\overline{E}} \sqrt{\frac{\mathsf{r}^e}{2}}\langle \rightarrow, e|w\rangle \langle \bar{b}^e_1|\hat{w}^e\rangle \frac{1}{\sqrt{2}}(|\rightarrow, e\rangle - |\leftarrow, e\rangle) + \Pi_{E\setminus\overline{E}}|w\rangle \\
&= \sum_{e\in\overline{E}} \sqrt{\frac{\mathsf{r}^e}{2}}\langle \rightarrow, e|w\rangle \frac{2}{\sqrt{\mathsf{r}^e}}\frac{1}{\sqrt{2}}(|\rightarrow, e\rangle - |\leftarrow, e\rangle) + \Pi_{E\setminus\overline{E}}|w\rangle \\
&= \sum_{e\in\overline{E}} \langle \rightarrow, e|w\rangle (|\rightarrow, e\rangle - |\leftarrow, e\rangle) + \Pi_{E\setminus\overline{E}}|w\rangle \\
&= \Pi_{\overline{E}}|w\rangle + \Pi_{E\setminus\overline{E}}|w\rangle = |\hat{w}\rangle,
\end{aligned} \tag{4.4.11}$$

where we used the fact that $\langle \rightarrow, e|w\rangle = -\langle \leftarrow, e|w\rangle$. It follows that $\Lambda^\dagger|w^\circ\rangle = |w\rangle$, and thus we have, for any $|\psi\rangle \in \Psi^-_{\mathcal{B}_G}$,

$$\langle\psi|\Lambda^\dagger|w^\circ\rangle = \langle\psi|w\rangle = 0, \tag{4.4.12}$$

since $|w\rangle$ is a positive witness for G. To complete the proof, we show orthogonality with $\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b^e_0\rangle, |b^e_1\rangle\}$ for any $e \in \overline{E}$. $\mathcal{B}_{G^e}$ is orthogonal to $|s^e\rangle$ and $|t^e\rangle$, and furthermore, for $i > 1$, we have $\langle \leftarrow, s^e|b^e_i\rangle = \langle \rightarrow, t^e|b^e_i\rangle = 0$. We have:

$$\langle b^e_i|w^\circ\rangle = \sqrt{\frac{\mathsf{r}^e}{2}}\langle \rightarrow, e|w\rangle\langle b^e_i|\hat{w}^e\rangle.$$

Since $|\hat{w}^e\rangle$ is a cropped positive witness for G^e, we have:

$$0 = \langle b^e_i|w\rangle = \langle b^e_i|s\rangle - \langle b^e_i|\leftarrow, s\rangle + \langle b^e_i|\hat{w}^e\rangle + \langle b^e_i|\rightarrow, t^e\rangle - \langle b^e_i|t\rangle = \langle b^e_i|\hat{w}^e\rangle,$$

completing the proof. □

4.4.8. Lemma. *Let $|w_{\mathcal{A}}\rangle$ be a negative witness for G that only overlaps switches that are off (this is always true for optimal witnesses, because when a switch is on $\Xi_e^{\mathcal{B}} = \Xi_e^{\mathcal{A}}$, so might as well put any overlap with Ξ_e into $|w_{\mathcal{B}}\rangle$). For every switch $e \in \overline{E}$ that is off, let $|w_{\mathcal{A}}^e\rangle$ be a negative witness for G^e. Then:*

$$|\hat{w}_{\mathcal{A}}^{\circ}\rangle = \sum_{e\in\overline{E}} \langle \rightarrow, e|w_{\mathcal{A}}\rangle \sqrt{\frac{2}{\mathsf{r}^e}} |\hat{w}_{\mathcal{A}}^e\rangle + \Pi_{E\setminus\overline{E}}|w_{\mathcal{A}}\rangle$$

is a negative witness for G°.

Proof:
We will use the fact that if a switch is off, we always have $\Pi_e|w_{\mathcal{A}}\rangle = \langle \rightarrow, e|w_{\mathcal{A}}\rangle(|\rightarrow, e\rangle - |\leftarrow, e\rangle)$. This is true because a switch that is off always has $\Xi_e^{\mathcal{A}} = \mathrm{span}\{|\rightarrow, e\rangle - |\leftarrow, e\rangle\}$. In particular, this means that $\Pi_{\overline{E}}|\hat{w}_{\mathcal{A}}\rangle \in \Xi_{\overline{E}}^{-}$, so since $|\hat{w}_{\mathcal{A}}\rangle + |\leftarrow, s\rangle - |\rightarrow, t\rangle \in \mathcal{B}_G$, in particular, it is in $\mathcal{B}_G^-$. We have:

$$\begin{aligned}
|\hat{w}_{\mathcal{A}}\rangle + |\leftarrow, s\rangle - |\rightarrow, t\rangle &\in \mathcal{B}_G^- \\
\Pi_{E\setminus\overline{E}}|\hat{w}_{\mathcal{A}}\rangle + \sum_{e\in\overline{E}} \langle \rightarrow, e|w_{\mathcal{A}}\rangle(|\rightarrow, e\rangle - |\leftarrow, e\rangle) + |\leftarrow, s\rangle - |\rightarrow, t\rangle &\in \mathcal{B}_G^- \\
\Lambda\left(\Pi_{E\setminus\overline{E}}|\hat{w}_{\mathcal{A}}\rangle + \sum_{e\in\overline{E}} \langle \rightarrow, e|w_{\mathcal{A}}\rangle(|\rightarrow, e\rangle - |\leftarrow, e\rangle) + |\leftarrow, s\rangle - |\rightarrow, t\rangle\right) &\in \mathcal{B}_{G^{\circ}} \\
\Pi_{E\setminus\overline{E}}|\hat{w}_{\mathcal{A}}\rangle + \sum_{e\in\overline{E}} \langle \rightarrow, e|w_{\mathcal{A}}\rangle\sqrt{2}|\bar{b}_1^e\rangle + |\leftarrow, s\rangle - |\rightarrow, t\rangle &\in \mathcal{B}_{G^{\circ}} \\
|\hat{w}_{\mathcal{A}}^{\circ}\rangle + \sum_{e\in\overline{E}} \langle \rightarrow, e|w_{\mathcal{A}}\rangle\sqrt{\frac{2}{\mathsf{r}^e}}\left(\sqrt{\mathsf{r}^e}|\bar{b}_1^e\rangle - |\hat{w}_{\mathcal{A}}^e\rangle\right) + |\leftarrow, s\rangle - |\rightarrow, t\rangle &\in \mathcal{B}_{G^{\circ}}.
\end{aligned}$$

We complete the proof by noting that for all $e \in \overline{E}$, $\sqrt{\mathsf{r}^e}|\bar{b}_1^e\rangle - |\hat{w}_{\mathcal{A}}^e\rangle \in \mathcal{B}_{G^{\circ}}$. To see this, we note:

$$|\hat{w}_{\mathcal{A}}^e\rangle + |\leftarrow, s\rangle - |\rightarrow, t\rangle \in \mathcal{B}_{G^e}^- \cap \mathrm{span}\{|b_0^e\rangle\}^{\perp},$$

which follows from the fact that $|\hat{w}_{\mathcal{A}}^e\rangle$ is a cropped negative witness, and so it is orthogonal to $|\leftarrow, s^e\rangle$ and $|\rightarrow, t^e\rangle$, and in particular, $|b_0^e\rangle = \frac{1}{\sqrt{2}}(|\leftarrow, s^e\rangle + |\rightarrow, t^e\rangle)$. The only other basis vector of $\mathcal{B}_{G^e}^-$ that overlaps $|\leftarrow, s^e\rangle$ and $|\rightarrow, t^e\rangle$ is $|b_1^e\rangle \propto |\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle + \sqrt{\mathsf{r}^e}|\bar{b}_0^e\rangle$, from which it follows:

$$|\hat{w}_{\mathcal{A}}^e\rangle - \sqrt{\mathsf{r}^e}|\bar{b}_0^e\rangle \in \mathcal{B}_{G^e}^- \cap \mathrm{span}\{|b_0^e\rangle, |b_1^e\rangle\}^{\perp} \subset \mathcal{B}_{G^{\circ}}.$$

□

4.4.2 Switching networks with subroutines

A switching network can be viewed as a high-level control structure whose edge queries may themselves be implemented by nontrivial procedures. In many settings, the value associated with an edge is not obtained directly, but instead computed by a quantum subroutine. Here we show how such subroutines can be incorporated into the switching-network framework via subspace-graph composition, and we derive a corresponding performance guarantee for the resulting composed construction.

4.4.9. Lemma. *Let G be any switching network computing a Boolean function f, with working bases that can be generated in time T_{basis}. For each $e \in E$, let $\{U_r^e\}_{r=1}^{T_e}$ be a quantum algorithm computing a Boolean function f_e with bounded error. Let $T_{\max}$ be an upper bound on T_e for all e, and F_x an st-cut-set of $G(x)$ whenever $f(x) = 0$. Then there is a subspace graph G° computing $f \circ (f_e)_{e\in E}$ with complexities $\hat{W}_+(G^\circ) = O\Big(\max_{x\in f^{-1}(1)} \mathcal{R}_{s,t}(G(x)) \log T_{\max}\Big)$ and $\hat{W}_-(G^\circ) = O\Big(\max_{x\in f^{-1}(0)} \sum_{e\in F_x} \mathsf{w}_e T_e^2\Big)$, and its bases can be generated in time $T_{\text{basis}} + O(\log T_{\max})$.*

We remark that sometimes a better negative witness for G can be obtained by taking a superposition of different st-cut-sets. This also gives a better negative witness for G°, but its form is more complicated, so we omit describing this.

Proof:

For each $e \in E$, let G^e be the subspace graph from Lemma 4.3.35 derived from the algorithm $\{U_r^e\}_r$, using weights $\alpha_r = r + 1$. Thus G^e computes f_e. We will apply Theorem 4.4.1 to G, and the G^e.

Positive analysis: A positive input for $f \circ (f_e)_e$ gives rise to a positive input for f – let $|\hat{w}\rangle$ be a positive witness for G under that input. This only uses edges that are turned on, so for each e that is on (i.e. $f_e = 1$), let $|\hat{w}^e\rangle$ be a cropped positive witness for G^e. By Theorem 4.4.1, there is a cropped positive witness for G°:

$$|\hat{w}^\circ\rangle := \sum_{e\in E} \sqrt{\frac{\mathsf{r}^e}{2}} \langle \leftarrow, e|\hat{w}\rangle |\hat{w}^e\rangle,$$

since $\overline{E} = E$, where $\mathsf{r}^e = 2$ is the scaling factor of G^e's basis. Since G is a switching network, its positive witness is a unit st-flow, in the sense of (4.3.4), so $\langle \leftarrow, e|\hat{w}\rangle = \frac{\theta(e)}{\sqrt{\mathsf{w}_e}}$, so we have:

$$|\hat{w}^\circ\rangle = \sum_{e\in E} \frac{\theta(e)}{\sqrt{\mathsf{w}_e}} |\hat{w}^e\rangle,$$

and

$$\hat{W}_+(G^\circ) \le \||\hat{w}^\circ\rangle\|^2 = \sum_{e\in E} \frac{\theta(e)^2}{\mathsf{w}_e} \||\hat{w}^e\rangle\|^2 = \sum_{e\in E} \frac{\theta(e)^2}{\mathsf{w}_e} \hat{W}_+(G^e).$$

Because we have set $\alpha_r = r + 1$, by Lemma 4.3.35, we have

$$\hat{W}_+(G^e) \leq \sum_{r=1}^{T_e} \frac{1}{r+1} = O(\log T_e),$$

so

$$\hat{W}_+(G^\circ) \leq \sum_{e \in E} \frac{\theta(e)^2}{\mathsf{w}_e} \log T_{\max} = \mathcal{R}_{s,t}(G(x)) \log T_{\max}.$$

Negative analysis: Similarly, for a negative input to $f \circ (f_e)_e$, let x be its corresponding negative input to f, and let $|\hat{w}_\mathcal{A}\rangle$ be a negative witness for G on input x. For each e such that $f_e = 0$, let $|\hat{w}^e_\mathcal{A}\rangle$ be a negative witness for G^e. By Theorem 4.4.1, there is a cropped negative witness for G°:

$$|\hat{w}^\circ_\mathcal{A}\rangle = \sum_{e \in E} \langle \leftarrow, e|\hat{w}_\mathcal{A}\rangle |\hat{w}^e_\mathcal{A}\rangle.$$

Since G is a switching network, we can always choose a negative witness of the form

$$|\hat{w}_\mathcal{A}\rangle = \sum_{e \in F_x} \sqrt{\mathsf{w}_e}(|\rightarrow, e\rangle - |\leftarrow, e\rangle),$$

where F_x is an st-cut-set of $G(x)$. Then:

$$\begin{aligned}\hat{W}_-(G^\circ) &\leq \max_{x \in f^{-1}(0)} \sum_{e \in F_x} \mathsf{w}_e \||\hat{w}^e_\mathcal{A}\rangle\|^2 = \max_{x \in f^{-1}(0)} \sum_{e \in F_x} \mathsf{w}_e \hat{W}_-(G^e) \\ &\leq \max_{x \in f^{-1}(0)} \sum_{e \in F_x} \mathsf{w}_e \sum_{r=1}^{T_e} (r+1) = O\left(\max_{x \in f^{-1}(0)} \sum_{e \in F_x} \mathsf{w}_e T_e^2 \right).\end{aligned}$$

Basis generation: By Theorem 4.4.1, since the bases of G^e can be generated in time $O(\log T_e)$, the bases of G° can be generated in time $T_{\text{basis}} + O(\log T_{\max})$, completing the proof. □

Then the main theorem of this section follows directly.

4.4.10. Theorem. *Let G be any switching network computing a Boolean function f, with working bases that can be generated in time T_{basis}. For each $e \in E$, let $\{U^e_r\}_{r=1}^{T_e}$ be a quantum algorithm computing a Boolean function f_e with bounded error. Let $T_{\max}$ be an upper bound on T_e for all e, and F_x an st-cut-set of $G(x)$ whenever $f(x) = 0$. Then there is a bounded error quantum algorithm for f with complexity*

$$\widetilde{O}\left(T_{\text{basis}} \sqrt{\max_{x \in f^{-1}(1)} \mathcal{R}_{s,t}(G(x)) \max_{x \in f^{-1}(0)} \sum_{e \in F_x} \mathsf{w}_e T_e^2} \right).$$

Proof:
Let G° be the subspace graph from Lemma 4.4.9. By Corollary 4.4.2, there is a subspace graph G', also computing $f \circ (f_e)_e$, whose bases can also be generated in time $T_{\text{basis}} + O(\log T_{\max})$, and such that $\hat{W}_+(G') \leq 1$ and $\hat{W}_-(G') \leq \hat{W}_+(G^\circ)\hat{W}_-(G^\circ)$. Then we can apply Theorem 4.3.14 to G' to get a bounded error quantum algorithm for $f \circ (f_e)_e$ with complexity:

$$\begin{aligned} &O\left((T_{\text{basis}} + \log T_{\max})\sqrt{\hat{W}_+(G^\circ)\hat{W}_-(G^\circ)}\right) \\ &= O\left((T_{\text{basis}} + \log T_{\max})\sqrt{\max_{x\in f^{-1}(1)} \mathcal{R}_{s,t}(G(x)) \max_{x\in f^{-1}(0)} \sum_{e\in F_x} \mathsf{w}_e T_e^2 \log T_{\max}}\right). \end{aligned}$$

□

4.A Recovering the local structure

In this appendix, we prove that the following spaces are equal.

$$\mathcal{B}_{G^\circ} = \bigoplus_{u\in V} \tilde{\Lambda}(\mathcal{V}_u) \oplus \bigoplus_{e\in\overline{E}} \bigoplus_{u\in V^e\setminus\{s,t\}} \mathcal{V}_u^e + \bigoplus_{e\in\overline{E}} \bigoplus_{e'\in E^e} \Xi_{e'}^{e\mathcal{B}} \tag{4.A.1}$$

$$\begin{aligned} \overline{\mathcal{B}}_{G^\circ} = \operatorname{span} &\underbrace{\Lambda(\Psi^-_{\mathcal{B}_G}) \cup \bigcup_{e\in\overline{E}} (\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b_0^e\rangle, |b_1^e\rangle\})}_{=:\Psi^-_{\mathcal{B}_{G^\circ}}} \\ &\cup \underbrace{\left\{\frac{1}{\sqrt{2}}(|\rightarrow, e'\rangle + |\leftarrow, e'\rangle) : e' \in \overline{E}^\circ = \bigcup_{e\in\overline{E}} \overline{E}^e\right\}}_{\operatorname{spans}\ \Xi^{\circ\mathcal{B}}_{\overline{E}^\circ} = \bigoplus_{e\in\overline{E}} \bigoplus_{e'\in\overline{E}^e} \Xi^{e\mathcal{B}}_{e'}}. \end{aligned} \tag{4.A.2}$$

4.A.1. Lemma. *For any $e \in \overline{E}$,*

$$\operatorname{span}\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi^{e\mathcal{B}}_{\overline{E}^e} = \mathcal{B}_{G^e} \cap \{|\leftarrow, s^e\rangle, |\rightarrow, t^e\rangle\}^\perp = \bigoplus_{u\in V^e\setminus\{s,t\}} \mathcal{V}_u^e + \Xi^{e\mathcal{B}}_{\overline{E}^e}.$$

Proof:
It is clear that

$$\begin{aligned} \operatorname{span}\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi^{e\mathcal{B}}_{\overline{E}^e} \subseteq \mathcal{B}_{G^e} = \operatorname{span}\Psi^-_{\mathcal{B}_{G^e}} \oplus \Xi^{e\mathcal{B}}_{\overline{E}^e} \\ \text{and also} \quad \bigoplus_{u\in V^e\setminus\{s,t\}} \mathcal{V}_u^e + \Xi^{e\mathcal{B}}_{\overline{E}^e} \subseteq \mathcal{B}_{G^e} = \bigoplus_{u\in V^e} \mathcal{V}_u^e + \Xi^{e\mathcal{B}}_{\overline{E}^e}. \end{aligned}$$

Since we also have everything in $\mathrm{span}\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b^e_0\rangle, |b^e_1\rangle\}$, $\bigoplus_{u\in V^e\setminus\{s,t\}} \mathcal{V}^e_u$ and $\Xi^{e\mathcal{B}}_{\overline{E}^e}$ orthogonal to both $|\leftarrow, s^e\rangle$ and $|\rightarrow, t^e\rangle$, we have:

$$\mathrm{span}\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b^e_0\rangle, |b^e_1\rangle\} \oplus \Xi^{e\mathcal{B}}_{\overline{E}^e} \subseteq \mathcal{B}_{G^e} \cap \{|\leftarrow, s^e\rangle, |\rightarrow, t^e\rangle\}^\perp$$
$$\text{and} \bigoplus_{u\in V^e\setminus\{s,t\}} \mathcal{V}^e_u + \Xi^{e\mathcal{B}}_{\overline{E}^e} \subseteq \mathcal{B}_{G^e} \cap \{|\leftarrow, s^e\rangle, |\rightarrow, t^e\rangle\}^\perp.$$

For the other direction, suppose $|\phi\rangle \in \mathcal{B}_{G^e} \cap \{|\leftarrow, s^e\rangle, |\rightarrow, t^e\rangle\}^\perp$. Since it is in $\mathcal{B}_{G^e}$, for some $|\phi_-\rangle \in \mathrm{span}\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b^e_0\rangle, |b^e_1\rangle\}$ and $|\phi_+\rangle \in \Xi^{e\mathcal{B}}_{\overline{E}^e}$, and scalars a_0 and a_1, we can express it as:

$$\begin{aligned}|\phi\rangle &= a_0|b^e_0\rangle + a_1|b^e_1\rangle + |\phi_-\rangle + |\phi_+\rangle \\ &= a_0\frac{1}{\sqrt{2}}(|\leftarrow, s^e\rangle + |\rightarrow, t^e\rangle) + a_1\frac{|\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle + \sqrt{\mathsf{r}^e}|\bar{b}_1\rangle}{\sqrt{2+\mathsf{r}^e}} + |\phi_-\rangle + |\phi_+\rangle.\end{aligned}$$

Since $|\phi_-\rangle$, $|\phi_+\rangle$, and $|\bar{b}_1\rangle$ are all orthogonal to both $|\leftarrow, s^e\rangle$ and $|\rightarrow, t^e\rangle$, since $|\phi\rangle$ is as well, we must have $a_0 = a_1 = 0$. Thus $|\phi\rangle \in \mathrm{span}\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b^e_0\rangle, |b^e_1\rangle\} \oplus \Xi^{e\mathcal{B}}_{\overline{E}^e}$.

Similarly, we can express $|\phi\rangle$ as

$$|\phi\rangle = a'_0(|\leftarrow, s\rangle + |\psi_\star(s^e)\rangle) + a'_1(|\rightarrow, t\rangle + |\psi_\star(t^e)\rangle) + |\phi_V\rangle + |\phi'_+\rangle$$

for $|\phi_V\rangle \in \bigoplus_{u\in V^e\setminus\{s,t\}} \mathcal{V}^e_u$ and $|\phi'_+\rangle \in \Xi^{e\mathcal{B}}_{\overline{E}^e}$. By the same reasoning as above, we must have $a'_0 = a'_1 = 0$, so $|\phi\rangle \in \bigoplus_{u\in V^e\setminus\{s,t\}} \mathcal{V}^e_u + \Xi^{e\mathcal{B}}_{\overline{E}^e}$. □

4.A.2. Lemma. *For any $e \in \overline{E}$,*

$$\widetilde{\Lambda}(|\rightarrow, e\rangle + |\leftarrow, e\rangle) \in \mathcal{B}_{G^e} \cap \{|\leftarrow, s^e\rangle, |\rightarrow, t^e\rangle\}^\perp.$$

Proof:
Since $\mathcal{V}^e_s + \mathcal{V}^e_t \subset \mathcal{B}_{G^e}$, we have

$$|\leftarrow, s^e\rangle + |\psi_\star(s^e)\rangle, |\rightarrow, t^e\rangle + |\psi_\star(t^e)\rangle \in \mathcal{B}_{G^e}.$$

Since $|\leftarrow, s\rangle + |\rightarrow, t\rangle \in \mathcal{B}_{G^e}$ (because G^e is st-composable),

$$\begin{aligned}&\sqrt{\mathsf{r}^e}\widetilde{\Lambda}(|\rightarrow, e\rangle + |\leftarrow, e\rangle) \\ &= |\psi_\star(s^e)\rangle + |\psi_\star(t^e)\rangle \\ &= |\leftarrow, s^e\rangle + |\psi_\star(s^e)\rangle + |\rightarrow, t^e\rangle + |\psi_\star(t^e)\rangle - (|\leftarrow, s\rangle + |\rightarrow, t\rangle),\end{aligned}$$

so $\widetilde{\Lambda}(|\rightarrow, e\rangle + |\leftarrow, e\rangle) \in \mathcal{B}_{G^e}$. Clearly $\widetilde{\Lambda}(|\rightarrow, e\rangle + |\leftarrow, e\rangle)$ is also orthogonal to $|\leftarrow, s^e\rangle$ and $|\rightarrow, t^e\rangle$, concluding the proof. □

4.A.3. LEMMA. *Let $\widetilde{\Lambda}$ be as in (4.4.5), and define*

$$\widetilde{\mathcal{B}}_{G^\circ} = \operatorname{span} \underbrace{\widetilde{\Lambda}(\Psi^-_{\mathcal{B}_G}) \cup \bigcup_{e\in\overline{E}} (\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b_0^e\rangle, |b_1^e\rangle\})}_{=:\widetilde{\Psi}^-_{\mathcal{B}_{G^\circ}}} \oplus \bigoplus_{e\in\overline{E}} \bigoplus_{e'\in\overline{E}^e} \Xi^{e\mathcal{B}}_{e'}.$$

Then $\widetilde{\mathcal{B}}_{G^\circ} = \overline{\mathcal{B}}_{G^\circ}$.

Proof:

Let $|\tilde{b}^e\rangle = \frac{1}{\sqrt{\mathsf{r}^e}}(|\psi_\star(s^e)\rangle - |\psi_\star(t^e)\rangle)$. Then

$$|\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle + \sqrt{\mathsf{r}^e}|\tilde{b}^e\rangle \in \mathcal{V}_s^e + \mathcal{V}_t^e \subseteq \operatorname{span}\{|b_0^e\rangle, |b_1^e\rangle, \ldots, |b_\ell^e\rangle\} \oplus \Xi^{e\mathcal{B}}_{\overline{E}^e}.$$

Then since the only vector that overlaps $|\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle$ is $|b_1^e\rangle = \frac{1}{\sqrt{2+\mathsf{r}^e}}(|\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle) + \sqrt{\frac{\mathsf{r}^e}{2+\mathsf{r}^e}}|\bar{b}_1^e\rangle$, we must have:

$$\langle b_1^e|(|\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle + \sqrt{\mathsf{r}^e}|\tilde{b}^e\rangle) = \sqrt{2+\mathsf{r}^e},$$

from which it follows that $\langle \bar{b}_1^e|\tilde{b}^e\rangle = 1$, and we can write:

$$|\tilde{b}^e\rangle = |\bar{b}_1^e\rangle + |d^e\rangle \tag{4.A.3}$$

for some $|d^e\rangle \in \mathcal{B}_{G^e}$ that is orthogonal to both $|\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle$ and $|\leftarrow, s^e\rangle + |\rightarrow, t^e\rangle$, so by Lemma 4.A.1, $|d^e\rangle \in \operatorname{span}\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi^{e\mathcal{B}}_{\overline{E}^e}$.

Suppose $|\tilde{\psi}\rangle \in \widetilde{\mathcal{B}}_{G^\circ}$, so we can express it, for some scalars a_e, $|\psi_{E\setminus\overline{E}}\rangle \in \Xi_{E\setminus\overline{E}}$ and $|\psi'\rangle \in \operatorname{span}\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi^{e\mathcal{B}}_{\overline{E}^e}$ as:

$$\begin{aligned}
|\tilde{\psi}\rangle &= \widetilde{\Lambda}\Bigg(\underbrace{\sum_{e\in\overline{E}} a_e \frac{1}{\sqrt{2}}(|\rightarrow, e\rangle - |\leftarrow, e\rangle) + |\psi_{E\setminus\overline{E}}\rangle}_{\in \operatorname{span}\Psi^-_{\mathcal{B}_G}}\Bigg) + |\psi'\rangle \\
&= \sum_{e\in\overline{E}} a_e|\tilde{b}^e\rangle + |\psi_{E\setminus\overline{E}}\rangle + |\psi'\rangle \\
&= \sum_{e\in\overline{E}} a_e|\bar{b}_1^e\rangle + |\psi_{E\setminus\overline{E}}\rangle + \sum_{e\in\overline{E}} a_e|d^e\rangle + |\psi'\rangle \\
&= \Lambda\Bigg(\underbrace{\sum_{e\in\overline{E}} a_e \frac{1}{\sqrt{2}}(|\rightarrow, e\rangle - |\leftarrow, e\rangle) + |\psi_{E\setminus\overline{E}}\rangle}_{\in \operatorname{span}\Psi^-_{\mathcal{B}_G}}\Bigg) + \underbrace{\sum_{e\in\overline{E}} a_e|d^e\rangle + |\psi'\rangle}_{\in \operatorname{span}\Psi^-_{\mathcal{B}_{G^e}}\setminus\{|b_0^e\rangle,|b_1^e\rangle\}\oplus\Xi^{e\mathcal{B}}_{\overline{E}^e}} \quad \in \overline{\mathcal{B}}_{G^\circ}.
\end{aligned} \tag{4.A.4}$$

Thus, $\widetilde{\mathcal{B}}_{G^\circ} \subseteq \overline{\mathcal{B}}_{G^\circ}$.

For the other direction, suppose $|\psi\rangle \in \overline{\mathcal{B}}_{G^\circ}$. Then similar to above, we can express it as

$$\begin{aligned}
|\psi\rangle &= \Lambda\Bigg(\underbrace{\sum_{e\in\overline{E}} a_e \frac{1}{\sqrt{2}}(|\rightarrow,e\rangle - |\leftarrow,e\rangle) + |\psi_{E\setminus\overline{E}}\rangle}_{\in \mathrm{span}\Psi^-_{\mathcal{B}_G}}\Bigg) + |\psi'\rangle \\
&= \sum_{e\in\overline{E}} a_e |\bar{b}_1^e\rangle + |\psi_{E\setminus\overline{E}}\rangle + |\psi'\rangle \\
&= \sum_{e\in\overline{E}} a_e |\tilde{b}_1^e\rangle + |\psi_{E\setminus\overline{E}}\rangle - \sum_{e\in\overline{E}} a_e |d^e\rangle + |\psi'\rangle \\
&= \tilde{\Lambda}\Bigg(\underbrace{\sum_{e\in\overline{E}} a_e \frac{1}{\sqrt{2}}(|\rightarrow,e\rangle - |\leftarrow,e\rangle) + |\psi_{E\setminus\overline{E}}\rangle}_{\in \mathrm{span}\Psi^-_{\mathcal{B}_G}}\Bigg) - \underbrace{\sum_{e\in\overline{E}} a_e |d^e\rangle + |\psi'\rangle}_{\in \mathrm{span}\Psi^-_{\mathcal{B}_{G^e}}\setminus\{|b_0^e\rangle,|b_1^e\rangle\}\oplus\Xi^{e\mathcal{B}}_{\overline{E}^e}} \quad \in \widetilde{\mathcal{B}}_{G^\circ}.
\end{aligned} \tag{4.A.5}$$

Thus $\overline{\mathcal{B}}G^\circ \subseteq \widetilde{\mathcal{B}}_{G^\circ}$. □

4.A.4. THEOREM. *Let* $\overline{\mathcal{B}}_{G^\circ} = \mathrm{span}\Psi_{\mathcal{B}_{G^\circ}}$ *where* $\Psi_{\mathcal{B}_{G^\circ}}$ *is as in (4.4.10) (see also (4.A.2)), and* $\mathcal{B}_{G^\circ}$ *as in (4.4.8) (see also (4.A.1)). Then* $\overline{\mathcal{B}}_{G^\circ} = \mathcal{B}_{G^\circ}$.

Proof:
We will show that $\mathcal{B}_{G^\circ} = \widetilde{\mathcal{B}}_{G^\circ}$, which is sufficient, by Lemma 4.A.3. We first show that $\widetilde{\mathcal{B}}_{G^\circ} \subseteq \mathcal{B}_{G^\circ}$. First, for any $e \in \overline{E}$, it follows from Lemma 4.A.1 that

$$\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi^{e\mathcal{B}}_{\overline{E}^e} = \bigoplus_{u\in V^e\setminus\{s,t\}} \mathcal{V}_u^e + \Xi^{e\mathcal{B}}_{\overline{E}^e} \subseteq \mathcal{B}'_{G^\circ}. \tag{4.A.6}$$

Next, for any $|\psi\rangle \in \Psi^-_{\mathcal{B}_G} \subseteq \bigoplus_{u\in V} \mathcal{V}_u$, we have

$$\tilde{\Lambda}(|\psi\rangle) \in \tilde{\Lambda}\Big(\bigoplus_{u\in V} \mathcal{V}_u\Big) = \bigoplus_{u\in V} \tilde{\Lambda}(\mathcal{V}_u) \subseteq \mathcal{B}'_{G^\circ}. \tag{4.A.7}$$

From (4.A.6) and (4.A.7), it follows that $\widetilde{\mathcal{B}}_{G^\circ} \subseteq \mathcal{B}_{G^\circ}$.

We now show the other direction, $\mathcal{B}_{G^\circ} \subseteq \widetilde{\mathcal{B}}_{G^\circ}$. As in (4.A.6), for any $e \in \overline{E}$,

$$\bigoplus_{u\in V^e\setminus\{s,t\}} \mathcal{V}_u^e + \Xi^{e\mathcal{B}}_{\overline{E}^e} = \Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi^{e\mathcal{B}}_{\overline{E}^e} \subseteq \widetilde{\mathcal{B}}_{G^\circ}. \tag{4.A.8}$$

Next, for any $u \in V$ and $|\psi\rangle \in \mathcal{V}_u \subseteq \mathcal{B}_G = \mathrm{span}\Psi^-_{\mathcal{B}_G} \oplus \Xi^{\mathcal{B}}_{\overline{E}}$, we can write

$$|\psi\rangle = |\psi_-\rangle + \sum_{e \in \overline{E}} a_e(|\rightarrow, e\rangle + |\leftarrow, e\rangle)$$

for some $|\psi_-\rangle \in \mathrm{span}\Psi^-_{\mathcal{B}_G}$ and scalars a_e. Then:

$$\widetilde{\Lambda}(|\psi\rangle) = \underbrace{\widetilde{\Lambda}|\psi_-\rangle}_{\in \mathrm{span}\widetilde{\Lambda}(\Psi^-_{\mathcal{B}_G}) \subseteq \widetilde{\mathcal{B}}_{G^\circ}} + \sum_{e \in \overline{E}} a_e \widetilde{\Lambda}(|\rightarrow, e\rangle + |\leftarrow, e\rangle). \tag{4.A.9}$$

By Lemma 4.A.2, for each e, $\widetilde{\Lambda}(|\rightarrow, e\rangle + |\leftarrow, e\rangle) \in \mathcal{B}_{G^e} \cap \{|\leftarrow, s\rangle, |\rightarrow, t\rangle\}^\perp$, which is equal to $\mathrm{span}\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b^e_0\rangle, |b^e_1\rangle\} \oplus \Xi^{e\mathcal{B}}_{\overline{E}}$ by Lemma 4.A.1. Since $\mathrm{span}\Psi^-_{\mathcal{B}_{G^e}} \setminus \{|b^e_0\rangle, |b^e_1\rangle\} \oplus \Xi^{e\mathcal{B}}_{\overline{E}} \subseteq \widetilde{\mathcal{B}}_{G^\circ}$, we have $\widetilde{\Lambda}|\psi\rangle \in \widetilde{\mathcal{B}}_{G^\circ}$, from which it follows that

$$\widetilde{\Lambda}(\mathcal{V}_u) \subseteq \widetilde{\mathcal{B}}_{G^\circ}. \tag{4.A.10}$$

Combining (4.A.8) and (4.A.10) establishes $\mathcal{B}_{G^\circ} \subseteq \widetilde{\mathcal{B}}_{G^\circ}$, completing the proof. □

Chapter 5

Quantum divide & conquer and application to DSTCON

Divide & conquer methods are central in classical algorithms, yet translating these ideas to quantum algorithms is nontrivial due to error amplification and recursion overheads. In this chapter, we develop a framework for time-efficient quantum divide & conquer based on subspace graphs, focusing on compositions where subproblems are combined via Boolean formulas. As an application, we obtain a bounded-error quantum algorithm for directed *st*-connectivity running in time $2^{\frac{1}{2}\log^2 n + O(\log n)}$ while preserving $O(\log^2 n)$ space, achieving a quadratic speedup over Savitch's classical recursive algorithm.

This chapter is based on the paper "Multidimensional Quantum Walks, Recursion, and Quantum Divide & Conquer" by Stacey Jeffery and Galina Pass [JP25a].

5.1 Introduction

In Chapter 4, we introduced subspace graphs as a compositional model for quantum algorithms, together with a theorem converting a subspace graph into a time-efficient bounded-error quantum algorithm. Here we use these tools to obtain a time-efficient version of quantum divide & conquer. As a central application, we use this framework to obtain a quadratic speedup over Savitch's algorithm for directed *st*-connectivity while preserving its $O(\log^2 n)$ space complexity.

Time-efficient quantum divide & conquer A particular type of recursive algorithm is *divide & conquer*, in which a problem is broken into multiple smaller sub-problems, whose solutions, obtained by recursive calls, are combined into a solution for the original problem. As a motivating example, consider the recursively defined *nand-tree function*. Let $f_{k,d} : \{0,1\}^{d^k} \to \{0,1\}$ be defined $f_{0,d}(x) = x$, and for $k \geq 1$,

$$f_{k,d}(x) = 1 - f_{k-1,d}(x^{(1)}) \dots f_{k-1,d}(x^{(d)}),$$

where each $x^{(j)} \in \{0,1\}^{d^{k-1}}$, and $x = (x^{(1)}, \dots, x^{(d)})$. There is a natural way to break an instance x of $f_{k,d}$ into d sub-problems $x^{(1)}, \dots, x^{(d)}$ of $f_{k-1,d}$, and combine the solutions by taking the NAND (negated AND) of the d sub-problem solutions. Grover's algorithm computes this NAND in $O(\sqrt{d})$ queries, so we might hope for a speedup by recursive calls to this quantum algorithm. Unfortunately, since we recurse to depth k, the constant in front of $\sqrt{d}$ is raised to the k-th power. This kills the quantum speedup completely when d is constant (for example, the most common setting of $d = 2$), and that is not even touching on the fact that we would seem to need to amplify the success probability of the subroutine, turning those constants into log factors. On the other hand, it is known [Rei11] that $f_{k,d}$ can be evaluated in $O(\sqrt{d^k})$ quantum queries, even though our attempt to use classical divide-&-conquer reasoning combined with the basic Grover speedup failed.

More recently, [CKKD+22] showed how to employ divide-&-conquer reasoning in the study of quantum query complexity, in which one only counts the number of queries to the input. They obtained their query upper bounds by composing *dual adversary solutions*. The key to their results is that dual adversary solutions exhibit *perfect* composition: no error, no log factors, not even constant overhead. However, their results were not constructive, as dual adversary solutions do not fully specify algorithms, and in particular, the time complexity analysis of their results was unknown. In this work, we use the framework of subspace graphs to give a constructive time-complexity version of some of the query complexity results obtained in [CKKD+22]. In particular, we show (see Theorem 5.3.1 for formal statement):

5.1.1. Theorem (Informal). *Let $\{f_{\ell,n} : D_{\ell,n} \to \{0,1\}\}_{\ell,n}$ be a family of functions. Let φ be a symmetric Boolean formula on a variables, and suppose $f_{\ell,n} = \varphi(f_{\ell/b,n}, \ldots, f_{\ell/b,n}) \vee f_{aux,\ell,n}$, for some $b > 1$ and some auxiliary function $f_{aux,\ell,n}$ with quantum time complexity $T_{\text{aux}}(\ell, n)$. Then the quantum time complexity of $f_{\ell,n}$ is $\widetilde{O}(T(\ell, n))$ for $T(\ell, n)$ satisfying:*

$$T(\ell, n) \leq \sqrt{a}T(\ell/b, n) + T_{\text{aux}}(\ell, n).$$

Our framework also handles the case where $f_{\ell,n} = \varphi(f_{\ell/b,n}, \ldots, f_{\ell/b,n}, f_{\text{aux},\ell,n})$ for any formula φ on $a + 1$ variables, but then some extra, somewhat complicated looking costs need to be accounted for, and there is also a scaling in the depth, although this is not an issue if the formula has been preprocessed to be balanced [BB94].

Comparing this with the analogous classical statement, which would have a instead of $\sqrt{a}$, we get an up to quadratic speedup over a large class of classical divide-&-conquer algorithms. As an application, we show a quadratic speedup of Savitch's divide-&-conquer algorithm for directed *st*-connectivity [Sav70].

To achieve these results, it is essential that we compose subspace graphs, rather than algorithms. When we convert a subspace graph to a quantum algorithm, we get constant factors in the complexity, and these seem necessary without at least specifying what gateset we are working in. By first composing in the more abstract model of subspace graphs, and then only converting to a quantum algorithm at the end, we ensure these factors only come into the complexity once. This is similar to compositions done with other abstract models, such as span programs (of which dual adversary solutions are a special case) [Rei09], and transducers [BJY24].

Application to DSTCON Quantum algorithms for *st*-connectivity on *undirected graphs*, which are closely related to evaluating switching networks, are well studied [DHHM06, BR12, JK17, JJKP18, AJPW23], including quantum algorithms that achieve optimal time- and space-complexity simultaneously, in both the edge-list access model, and the adjacency matrix access model.[1] In contrast, quantum algorithms for *directed st*-connectivity (DSTCON) – the problem of deciding if there is a directed path from s to t in a directed graph – is less well understood. The algorithm of [DHHM06] also applies to directed graphs, deciding connectivity in $\widetilde{O}(n)$ time and space[2]. This algorithm has optimal time complexity, whereas its space complexity is far from optimal.

Directed *st*-connectivity, also called *reachability*, is a fundamental problem in classical space complexity. In particular, understanding if this problem can be

[1] We do not make a distinction between various access models, because they can simulate one another in $\text{poly}(n)$ time and $\log(n)$ space, so our result, which includes a $2^{O(\log n)}$ term, is the same in all models.

[2] In the edge-list access model.

solved in $\log(n)$ space by a quantum algorithm would resolve the relationship between quantum logspace complexity and NL, as DSTCON is NL-complete.

The best known classical (deterministic) space complexity of DSTCON is $O(\log^2(n))$, using Savitch's algorithm. We apply quantum divide & conquer (Theorem 5.1.1) to give a quadratic speedup to Savitch's algorithm, achieving $2^{\frac{1}{2}\log^2(n)+O(\log(n))}$ time, while still maintaining $O(\log^2(n))$ space (see Theorem 5.4.3).

Related and future work Time-efficient quantum divide & conquer has also been studied in [ABB+25] when either (1) φ is an OR (equivalently, an AND) or (2) φ is a minimum or maximum. OR is a Boolean formula, while minimum/maximum is not. In that sense, our results are incomparable. The framework of [ABB+25] also differs from our work in that they explicitly treat the complexity of computing sub-instances (the cost of the "create" step), whereas we assume sub-instances can simply be queried in unit cost. In one of the applications of [ABB+25], this cost is not negligible, and is even the dominating cost, so our framework, as stated, would not handle this application. This is not an inherent limitation of our techniques – it would be possible to take this cost into account in our framework as well. Ref. [ABB+25] applies their framework to problems that are distinct from our applications.

We also mention that Ref. [CKKD+22] analyzes the quantum query complexity of divide & conquer where φ is an arbitrary Boolean formula, as well as in settings where the function combining the sub-problems is more general. While our techniques do apply to composing quantum algorithms for arbitrary functions (already studied in [Jef22]), an issue is a poor scaling in the error of subroutines. If we start with a bounded-error quantum algorithm for some function, we need to amplify the success probability, as it will be called many times, incurring logarithmic factors. This becomes a serious problem if the function is called recursively to depth more than constant. We get around this in the case of Boolean formulas by using a switching network construction (from which a quantum algorithm could be derived) rather than a bounded-error quantum algorithm for evaluating the formula. Our techniques would thus also readily apply to functions for which there is an efficient quantum algorithm derived from a switching network. For more general functions, the solution might lie in a recent framework, *transducers* [BJY24], that allows for the composition of quantum algorithms without the need for success probability amplification. We leave further investigation for future work.

Outline of the chapter Section Section 5.2 develops a composition theorem for *st*-composable subspace graphs under balanced Boolean formula composition, and derives a simplified corollary for the symmetric case. Section Section 5.3 applies this composition result to obtain a time-efficient quantum divide-&-conquer

theorem, giving recurrences with a $\sqrt{a}$ dependence on the branching factor. Finally, Section Section 5.4 instantiates this framework with Savitch's recursion for directed *st*-connectivity, yielding a bounded-error quantum algorithm for DSTCON with time $2^{\frac{1}{2}\log^2 n + O(\log n)}$ and space $O(\log^2 n)$.

5.2 Boolean formula composition

In this section, we give a subspace graph computing $\varphi \circ (f_\sigma)_\sigma$ when φ is a symmetric formula, given subspace graphs for each f_σ. Subspace graphs for formulas φ can be obtained simply by composing the switching networks for OR and AND in Section 4.3.3 and Section 4.3.3 (see [JK17]). Our composition theorem, Theorem 4.4.1, generalizes this simple switching network composition to more general subspace graphs with switches.

We assume that φ is balanced (see Definition 2.3.2), as otherwise we get a factor in the depth. In the general balanced case, we must be able to generate certain superpositions efficiently, which is slightly complicated to describe (beginning with Definition 5.2.1). The special case where φ is symmetric (see Definition 2.3.1) is much simpler, as shown in Corollary 5.2.4.

5.2.1. DEFINITION. Fix a Boolean formula φ on $\{0,1\}^\Sigma$, and real values C^+_σ and C^-_σ for each $\sigma \in \Sigma$. Let $\mathsf{C}_\sigma := \sqrt{\mathsf{C}^+_\sigma \mathsf{C}^-_\sigma}$.

For $\sigma \in \overline{\Sigma} \setminus \Sigma$, let d_σ be the degree of σ, and $v(\sigma) \in \{\vee, \wedge\}$ indicate the gate labelling the node σ, so for example, if $v(\sigma) = \vee$, then the sub-formula rooted at σ is an OR of d_σ sub-formulas. Define:

$$\mathsf{C}_\sigma := \sum_{i=1}^{d} \mathsf{C}_{\sigma i},$$

$$\mathsf{C}^+_\sigma = \begin{cases} \sum_{i=1}^{d_\sigma} \mathsf{C}^+_{\sigma i} & \text{if } v(\sigma) = \vee \\ \dfrac{\mathsf{C}^2_\sigma}{\sum_{i=1}^{d_\sigma} \mathsf{C}^+_{\sigma i}} & \text{if } v(\sigma) = \wedge, \end{cases}$$

$$\text{and} \quad \mathsf{C}^-_\sigma = \begin{cases} \dfrac{\mathsf{C}^2_\sigma}{\sum_{i=1}^{d_\sigma} \mathsf{C}^-_{\sigma i}} & \text{if } v(\sigma) = \vee \\ \sum_{i=1}^{d_\sigma} \mathsf{C}^-_{\sigma i} & \text{if } v(\sigma) = \wedge. \end{cases}$$

Then note that

$$\mathsf{C}_\sigma = \sqrt{\mathsf{C}^+_\sigma \mathsf{C}^-_\sigma}, \quad \text{and also} \quad \mathsf{C} = \mathsf{C}_\emptyset = \sqrt{\sum_{\sigma \in \Sigma} \mathsf{C}^2_\sigma}.$$

We say the values $\{\mathsf{C}^+_\sigma, \mathsf{C}^-_\sigma\}_{\sigma \in \Sigma}$ incur logarithmic reflection cost if for every $\sigma \in \overline{\Sigma}$ such that $v(\sigma) = \vee$, a state proportional to $\sum_{i \in [d_\sigma]} \sqrt{\mathsf{C}^+_{\sigma i}} |i\rangle$ can be generated

in complexity $O(\log d_\sigma)$, and for all $\sigma \in \overline{\Sigma}$ such that $v(\sigma) = \wedge$, a state proportional to $\sum_{i\in[d_\sigma]} \sqrt{\mathsf{C}^-_{\sigma i}}|i\rangle$ can be generated in complexity $O(\log d_\sigma)$.

5.2.2. Lemma. *Let φ be a balanced formula (Definition 2.3.2) on $\{0,1\}^\Sigma$. Let $\{f_\sigma\}_{\sigma\in\Sigma}$ be Boolean functions, and for each $\sigma \in \Sigma$, let G_σ be an st-composable subspace graph (Definition 4.3.9) computing f_σ, with working bases that can be generated in time at most T with scaling factor r^σ, and $\log\dim H_{G_\sigma} \le S$. Suppose that for each $\sigma \in \Sigma$, C^+_σ is a known upper bound on $\mathsf{r}^\sigma \hat{W}_+(G_\sigma)$, and C^-_σ is a known upper bound on $\hat{W}_-(G_\sigma)/\mathsf{r}^\sigma$, so $\mathsf{C}_\sigma := \sqrt{\mathsf{C}^+_\sigma \mathsf{C}^-_\sigma}$ is a known upper bound on $\hat{\mathcal{C}}(G_\sigma)$. Suppose the values $\{\mathsf{C}^+_\sigma, \mathsf{C}^-_\sigma\}_{\sigma\in\Sigma}$ incur logarithmic reflection cost (see Definition 5.2.1).*

Then there is an st-composable subspace graph G° computing $\varphi \circ (f_\sigma)_{\sigma\in\Sigma}$ with $\log\dim H_{G^\circ} = S + O(\log|\Sigma|)$ and

$$\hat{\mathcal{C}}(G^\circ)^2 \le \sum_{\sigma\in\Sigma} \mathsf{C}^2_\sigma.$$

Furthermore, the working bases of G° can be generated in time $T + O(\log|\Sigma|)$.

Proof of Lemma 5.2.2:
Let c be a constant such that for each node in φ, if its subtree has N leaves, and it has d children, then the sub-tree of each child has at most cN/d leaves (this exists because φ is balanced), and let a be a constant such that the bases referred to in Lemma 4.3.20 and Lemma 4.3.24 can both be generated in complexity at most $a \log d$, and let a' be a constant such that the cost of generating the basis of G° in Theorem 4.4.1 is $T + T' + a'$. We will show by induction on the depth D of φ that there exists an *st*-composable subspace graph G° computing $\varphi \circ (f_\sigma)_\sigma$ with composable bases with scaling factor r° such that:

1. $\mathsf{r}^\circ \hat{W}_+(G^\circ) \le \mathsf{C}^+$
2. $\frac{\hat{W}_-(G^\circ)}{\mathsf{r}^\circ} \le \mathsf{C}^-$
3. $\log\dim H_{G^\circ} \le \log|\Sigma| + (2 + \log c)D + S$
4. the bases can be generated in cost at most $T + (a + a')\log(c^D|\Sigma|)$.

Since φ is balanced, $D = O(\log|\Sigma|)$, so

$$(a + a')\log(c^D|\Sigma|) = (a + a')\log|\Sigma| + (a + a')D\log c = O(\log|\Sigma|).$$

Without loss of generality, we can assume that none of the leaves of φ is negated, because if it is, we can just push this negation into f_σ. The proof will be by induction on the depth of φ.

Base case: If $D = 0$, then $\varphi \circ (f_\sigma)_{\sigma\in\Sigma} = f_\sigma$ for some σ. Then we just take $G^\circ = G_\sigma$, so $\mathsf{r}^\circ = \mathsf{r}^\sigma$. We have, by assumption, that $\mathsf{r}^\sigma \hat{W}_+(G_\sigma) \le \mathsf{C}^+_\sigma = \mathsf{C}^+$ (when the depth is 0) and $\hat{W}_-(G_\sigma)/\mathsf{r}^\sigma \le \mathsf{C}^-_\sigma = \mathsf{C}^-$. We also have, by assumption, that the working bases of G_σ can be generated in time T, and $\log\dim H_{G_\sigma} \le S$, completing the base case.

OR case: First, suppose φ has depth $D > 0$, with $\varphi = \bigvee_{i=1}^d \varphi_i$ for some d and some formulas φ_i of depth $D-1$. Let $G = G_{\mathrm{OR},d}$ be the subspace graph from Lemma 4.3.20, which is *st*-composable, because it is a switching network, and let $G_{\varphi_1}, \ldots, G_{\varphi_d}$ be the *st*-composable subspace graphs whose existence is promised by the induction hypothesis. For all $i \in [d]$, we will compose the graphs G_{φ_i} into the switch edge e_i of G using Theorem 4.4.1, to obtain G°. If $\varphi_i(x) = 1$ for some i, then G_{φ_i} must have some cropped positive witness $|\hat{w}^i\rangle$, and using the cropped positive witness $|\hat{w}\rangle$ from Lemma 4.3.20, by Theorem 4.4.1, G° has a cropped positive witness

$$|\hat{w}^\circ\rangle = \sqrt{\frac{\mathsf{r}^i}{2}}\frac{1}{\sqrt{\mathsf{w}_i}}|\hat{w}^i\rangle,$$

where r^i is the scaling factor of the basis for G_{φ_i}. Thus, if we choose

$$\mathsf{w}_i = \mathsf{C}^+_i/2, \quad \text{so} \quad \mathsf{r}^\circ = 2\sum_{i=1}^d \mathsf{w}_i = \sum_{i=1}^d \mathsf{C}^+_i$$

we get

$$\hat{W}_+(G^\circ) \le \frac{\mathsf{r}^i}{2\mathsf{w}_i}\hat{W}_+(G_{\varphi_i}) \le \frac{\mathsf{C}^+_i}{\mathsf{C}^+_i} = 1, \quad \text{and so} \quad \mathsf{r}^\circ\hat{W}_+(G^\circ) \le \sum_{i=1}^d \mathsf{C}^+_i = \mathsf{C}^+$$

using the induction hypothesis, which says that $\mathsf{r}^i\hat{W}_+(G_{\varphi_i}) \le \mathsf{C}^+_i$.

On the other hand, suppose for all $i \in [d]$, $\varphi_i(x) = 0$ and so there exists a cropped negative witness $|\hat{w}^i_{\mathcal{A}}\rangle$ for G_{φ_i}. Then using the cropped negative witness from Lemma 4.3.20, by Theorem 4.4.1, we have a cropped negative witness for G°:

$$|\hat{w}^\circ_{\mathcal{A}}\rangle = \sum_{i=1}^d \sqrt{\frac{2}{\mathsf{r}^i}}\sqrt{\mathsf{w}_i}|\hat{w}^i_{\mathcal{A}}\rangle.$$

Thus:

$$\hat{W}_-(G^\circ) \le \sum_{i=1}^d \frac{2\mathsf{w}_i}{\mathsf{r}^i}\hat{W}_-(G_{\varphi_i}) \le \sum_{i=1}^d \mathsf{C}^+_i\mathsf{C}^-_i = \sum_{i=1}^d \mathsf{C}^2_i = \mathsf{C}^2,$$

by the induction hypothesis, which says that $\frac{\hat{W}_-(G_{\varphi_i})}{\mathsf{r}^i} \le \mathsf{C}^-_i$.

By the induction hypothesis, the cost of generating the bases for any G_φ is at most:

$$T' := (a+a')\log\Big(c^{D-1}c|\Sigma|/d\Big) = (a+a')\log\Big(c^D|\Sigma|/d\Big).$$

Thus, by Theorem 4.4.1, since the bases of G can be generated in cost at most $T = a\log d$ – assuming the state proportional to $\sum_i \sqrt{\mathsf{w}_i}|i\rangle \propto \sum_i \sqrt{\mathsf{C}_i^+}|i\rangle$ can be generated in $O(\log d)$ time – we can generate the bases of G° in cost

$$T + T' + a' = (a+a')\log(c^{D-1}(c|\Sigma|/d)) + a\log d + a' \le (a+a')\log(c^D|\Sigma|),$$

as needed.

Similarly, by Theorem 4.4.1, we have

$$\begin{aligned}\dim H_{G^\circ} &\le \dim H_G - 2d + \sum_{i=1}^d \dim H_{G_{\varphi_i}} \\ &\le 2d + 4 - 2d + d2^{\log(|\Sigma_i|)+(2+\log c)(D-1)+S} \le 4d2^{\log(|\Sigma_i|)+(2+\log c)(D-1)+S}\end{aligned}$$

by the induction hypothesis and Lemma 4.3.20. Thus,

$$\log\dim H_{G^\circ} \le 2+\log d+\log\frac{c|\Sigma|}{d}+(2+\log c)(D-1)+S = \log|\Sigma|+(2+\log c)D+S.$$

AND case: Next, suppose $\varphi = \bigwedge_{i=1}^d \varphi_i$ for some formulas of depth $D-1$. Let $G = G_{\mathrm{AND},d}$ be the graph from Lemma 4.3.24, which is a switching network, and hence *st*-composable. Let $G_{\varphi_1},\dots,G_{\varphi_d}$ be the *st*-composable subspace graphs whose existence is promised by the induction hypothesis. We will compose the graphs G_{φ_i} into the switch edge e_i of G to obtain G°. If $\varphi_i(x) = 1$ for all i, then there exist cropped positive witnesses $|\hat{w}^i\rangle$ for each G_{φ_i}, so using the cropped positive witness for G from Lemma 4.3.24, by Theorem 4.4.1, G° has cropped positive witness

$$|\hat{w}^\circ\rangle = \sum_{i=1}^d \sqrt{\frac{\mathsf{r}^i}{2}}\frac{1}{\sqrt{\mathsf{w}_i}}|\hat{w}^i\rangle.$$

Thus, setting

$$\mathsf{w}_i = \frac{1}{2\mathsf{C}_i^-}, \quad \text{so} \quad \frac{1}{\mathsf{r}^\circ} = \sum_{i=1}^d \frac{1}{2\mathsf{w}_i} = \sum_{i=1}^d \mathsf{C}_i^-,$$

we have:

$$\hat{W}_+(G^\circ) \le \sum_{i=1}^d \frac{\mathsf{r}^i}{2\mathsf{w}_i}\hat{W}_+(G_{\varphi_i}) \le \sum_{i=1}^d \mathsf{C}_i^-\mathsf{C}_i^+ = \sum_{i=1}^d \mathsf{C}_i^2 = \mathsf{C}^2,$$

by the induction hypothesis, which says that $\mathsf{r}^i\hat{W}_+(G_{\varphi_i}) \le \mathsf{C}_i^+$.

On the other hand, suppose $\varphi_i(x) = 0$ for some $i \in [d]$, so there exists a cropped negative witness $|\hat{w}^i_{\mathcal{A}}\rangle$ for G_{φ_i}. Then using the cropped negative witness from Lemma 4.3.24, by Theorem 4.4.1, we have a cropped negative witness for G°:

$$|\hat{w}^\circ_{\mathcal{A}}\rangle = \sqrt{\frac{2}{\mathsf{r}^i}}\sqrt{\mathsf{w}_i}|\hat{w}^i_{\mathcal{A}}\rangle.$$

Thus, we have:

$$\hat{W}_-(G^\circ) \le \frac{2\mathsf{w}_i}{\mathsf{r}^i}\hat{W}_-(G_{\varphi_i}) \le \frac{\mathsf{C}^-_i}{\mathsf{C}^-_i} = 1, \quad \text{and so} \quad \frac{\hat{W}_-(G^\circ)}{\mathsf{r}^\circ} \le \sum_{i=1}^{d} \mathsf{C}^-_i = \mathsf{C}^-,$$

using the induction hypothesis, which says that $\frac{\hat{W}_-(G_{\varphi_i})}{\mathsf{r}^i} \le \mathsf{C}^-_i$.

A similar argument about the basis efficiency and dimension of H_{G° as in the OR case completes the proof. □

We will prove a more easily-applied corollary for the case where φ is symmetric. We will use the following two lemmas.

5.2.3. Lemma. *Let φ be a symmetric formula (see Definition 2.3.1) of depth D, and for $D' \in [D]$, let $d_{D'}$ be the out-degree of the nodes whose subtrees have depth D'. Assume without loss of generality that the nodes whose children are leaves are $\wedge$ gates. Suppose for all $\sigma \in \Sigma$, $\mathsf{C}^-_\sigma = \mathsf{L}^-$ is a known upper bound on $\hat{W}_-(G_\sigma)/r^\sigma$, independent of σ; and C^+_σ is a known upper bound on $\hat{W}_+(G_\sigma)\hat{W}_-(G_\sigma)$. Let $\{\mathsf{C}^+_\sigma, \mathsf{C}^-_\sigma, \mathsf{C}_\sigma\}_{\sigma\in\overline{\Sigma}}$ be as in Definition 5.2.1. Then for all $\sigma \in \overline{\Sigma}\setminus\Sigma$ whose subtrees have depth D',*

$$\mathsf{C}^+_\sigma = \frac{\mathsf{C}^2_\sigma}{\mathsf{L}^- \prod_{j=1}^{\lceil D'/2\rceil} d_{2j-1}} \quad \textit{and} \quad \mathsf{C}^-_\sigma = \mathsf{L}^- \prod_{j=1}^{\lceil D'/2\rceil} d_{2j-1}$$

Proof:
First note that we can assume without loss of generality that the nodes with D' odd are labelled by $\wedge$, and D' even are labelled by $\vee$.

We will prove the statement by induction. Since $\mathsf{C}_\sigma = \sqrt{\mathsf{C}^+_\sigma \mathsf{C}^-_\sigma}$, it is sufficient to prove the statement about C^-_σ.

Base case: Let $\sigma \in \Sigma$, so $D' = 0$. Then, indeed, $\mathsf{C}^-_\sigma = \mathsf{L}^- = \mathsf{L}^- \prod_{j=1}^{0} d_{2j-1}$.

Odd case: Let $D' > 0$ be odd. By the induction hypothesis, and since $D' - 1$ is even, for all $i \in [d_{D'}]$

$$\mathsf{C}^-_{\sigma i} = \mathsf{L}^- \prod_{j=1}^{(D'-1)/2} d_{2j-1}.$$

Since D' is odd, $v(\sigma) = \wedge$, and so

$$\mathsf{C}^-_{\sigma} = \sum_{i=1}^{d_{D'}} \mathsf{C}^-_{\sigma i} = d_{D'} \mathsf{L}^- \prod_{j=1}^{(D'-1)/2} d_{2j-1} = \mathsf{L}^- \prod_{j=1}^{\lceil D'/2 \rceil} d_{2j-1}.$$

Even case: Let $D' > 0$ be even. By the induction hypothesis, for all $i \in [d_{D'}]$

$$\mathsf{C}^+_{\sigma i} = \frac{\mathsf{C}^2_{\sigma i}}{\mathsf{C}^-_{\sigma i}} = \frac{\mathsf{C}^2_{\sigma i}}{\mathsf{L}^- \prod_{j=1}^{\lceil (D'-1)/2 \rceil} d_{2j-1}} = \frac{\mathsf{C}^2_{\sigma i}}{\mathsf{L}^- \prod_{j=1}^{\lceil D'/2 \rceil} d_{2j-1}}.$$

Since D' is even, $v(\sigma) = \vee$, and so

$$\mathsf{C}^+_{\sigma} = \sum_{i=1}^{d_{D'}} \mathsf{C}^+_{\sigma i} = \frac{\sum_{i=1}^{d_{D'}} \mathsf{C}^2_{\sigma,i}}{\mathsf{L}^- \prod_{j=1}^{\lceil D'/2 \rceil} d_{2j-1}} = \frac{\mathsf{C}^2_{\sigma}}{\mathsf{L}^- \prod_{j=1}^{\lceil D'/2 \rceil} d_{2j-1}}.$$

Thus

$$\mathsf{C}^-_{\sigma} = \frac{\mathsf{C}^2_{\sigma}}{\mathsf{C}^+_{\sigma}} = \mathsf{L}^- \prod_{j=1}^{\lceil D'/2 \rceil} d_{2j-1}.$$

□

5.2.4. Corollary. *Let φ be a symmetric formula (Definition 2.3.1) on $\{0,1\}^\Sigma$. Let $\{f_\sigma\}_{\sigma \in \Sigma}$ be Boolean functions, and for each $\sigma \in \Sigma$, let G_σ be an st-composable subspace graph (Definition 4.3.9) computing f_σ, with working bases that can be generated in time at most T with scaling factor r^σ, and $\log \dim H_{G_\sigma} \leq S$. Suppose that for each $\sigma \in \Sigma$, L^- is a known upper bound on $\hat{W}_-(G_\sigma)/\mathsf{r}^\sigma$, and C^+_σ is a known upper bound on $\mathsf{r}^\sigma \hat{W}_+(G_\sigma)$. Let $\mathsf{C}_\sigma = \sqrt{\mathsf{C}^+_\sigma \mathsf{C}^-}$, which is then a known upper bound on $\hat{\mathcal{C}}(G_\sigma)$. For any $\sigma \in \overline{\Sigma} \setminus \Sigma$ (i.e. proper prefixes of the strings in Σ), if the children of σ in φ are labelled by $[d]$, define*

$$\mathsf{C}_\sigma = \sqrt{\sum_{i \in [d]} \mathsf{C}^2_{\sigma i}},$$

and suppose a state proportional to $\sum_{i \in [d]} \mathsf{C}_{\sigma i} |i\rangle$ can be generating in cost $O(\log d)$. Then there is an st-composable subspace graph G° computing $\varphi \circ (f_\sigma)_{\sigma \in \Sigma}$ with $\log \dim H_{G^\circ} = S + O(\log |\Sigma|)$ and

$$\mathsf{r}^\circ \hat{W}_+(G^\circ) \leq \frac{\sum_{\sigma \in \Sigma} \mathsf{C}^2_\sigma}{\mathsf{L}^- \prod_{j=1}^{\lceil D/2 \rceil} d_{2j-1}} \quad \text{and} \quad \hat{W}_-(G^\circ)/\mathsf{r}^\circ \leq \mathsf{L}^- \prod_{j=1}^{\lceil D/2 \rceil} d_{2j-1}$$

so

$$\hat{\mathcal{C}}(G^\circ)^2 \leq \sum_{\sigma\in\Sigma} \mathsf{C}_\sigma^2.$$

Furthermore, the working bases of G° can be generated in time $T + O(\log|\Sigma|)$.

Proof:
By Lemma 5.2.3, since we can generate $\sum_{i\in[d]} \mathsf{C}_{\sigma i}|i\rangle$ in $\log d$ complexity, our upper bounds $\mathsf{C}_\sigma^\pm$ incur logarithmic reflection cost, so we can apply Lemma 5.2.2. □

5.3 Quantum divide & conquer

In this section, we state our time-efficient quantum divide-&-conquer results. For an instance x of a function $f_{\ell,n}$, we say that a sub-instance x' of $f_{\ell',n'}$ for some $\ell' \leq \ell$ and $n' \leq n$ is *unit-time computable* if any bit x'_i of x' can be computed as a function of x and i in unit time. The following is a time-efficient version of Strategy 1 in [CKKD+22]. Note that it is fairly natural to assume $\varphi = \varphi'(z_1,\ldots,z_a) \vee z_{a+1}$ for some formula φ', since often $z_{a+1} = f_{\text{aux}}(x)$ simply handles the base case of the recursion. We prove the statement for symmetric formulas, however, using Lemma 5.2.2 instead of Corollary 5.2.4, a similar statement, possibly with extra time overhead, holds for any balanced formula φ, as long as we can account for the possibly more complicated reflection costs.

5.3.1. Theorem. *Fix a function family $f_{\ell,n} : D_{\ell,n} \to \{0,1\}$. Fix unit-time-computable functions $\lambda_1,\lambda_2 : \mathbb{N}\to\mathbb{N}$, and a formula φ on $\{0,1\}^{a+1}$ such that $\varphi = \varphi'(z_1,\ldots,z_a)\vee z_{a+1}$ for some symmetric formula φ'. Suppose $\{\mathcal{P}_{\text{aux},\ell,n}\}_{\ell,n\in\mathbb{N}}$ is a family of quantum algorithms such that:*

- *$\mathcal{P}_{\text{aux},\ell,n}$ decides some $f_{\text{aux},\ell,n} : D_{\ell,n}\to\{0,1\}$ with time and space complexities $T_{\text{aux}}(\ell,n)$ and $S_{\text{aux}}(\ell,n)$;*
- *if $\ell \leq \ell_0$, $f_{\ell,n}(x) = f_{\text{aux},\ell,n}(x)$;*
- *if $\ell > \ell_0$, $f_{\ell,n} = \varphi\circ(f_i)_{i\in[a+1]}$ where each f_i for $i\in[a]$ is such that $f_i(x) = f_{\lambda_1(\ell),\lambda_2(n)}(x^i)$ for some unit-time-computable instance x^i of $f_{\lambda_1(\ell),\lambda_2(n)}$, and $f_{a+1}(x) = f_{\text{aux},\ell,n}(x^{a+1})$ for some unit-time computable instance x^{a+1} of $f_{\text{aux},\ell,n}$.*

Then there is a bounded-error quantum algorithm that compute $f_{\ell,n}$ with time complexity $\widetilde{O}(T(\ell,n))$, and space complexity $O(S_{\text{aux}}(\ell,n)+\log T(\ell,n))$, where for all $\ell > \ell_0$:

$$T(\ell,n) := \sqrt{aT(\lambda_1(\ell),\lambda_2(n))^2 + 4T_{\text{aux}}(\ell,n)^2} \leq \sqrt{a}T(\lambda_1(\ell),\lambda_2(n)) + 2T_{\text{aux}}(\ell,n)$$

and for $\ell \le \ell_0$, $T(\ell, n) = 2T_{\text{aux}}(\ell, n)$.

We remark that, while we assume the subroutine $\mathcal{P}_{\text{aux},\ell,n}$ has no error, if it has sufficiently small error inversely proportional to the number of times it is called, the algorithm must still work, as this cannot be distinguished from having no error. Amplifying a bounded-error quantum algorithm to have such small error incurs factors logarithmic in the number of times it is called, so $2T_{\text{aux}}(\ell, n)$ becomes $O(T_{\text{aux}}(\ell, n) \log T(\ell, n))$. We stress that these log factors are only acceptable because we do not *recursively* call $\mathcal{P}_{\text{aux},\ell,n}$.

Proof:
We will use the shorthand $\lambda(\ell, n) = (\lambda_1(\ell), \lambda_2(n))$. Let $D_{\varphi'}$ be the depth of φ', and for each $j \in [D_{\varphi'}]$, let d_j be the number of children of any node at distance j from the leaves. Define $\bar{d} = \prod_{j=1}^{\lceil D_\varphi/2 \rceil} d_j$ for $D_{\varphi'}$. Let $T_{\text{aux},0}$ be an upper bound on $T_{\text{aux}}(\ell, n)$ when $\ell \le \ell_0$. For all (ℓ, n), let $\mathsf{C}^-(\ell, n)$ be defined recursively, as follows.

$$\mathsf{C}^-(\ell, n) := \begin{cases} T_{\text{aux},0} & \text{if } \ell \le \ell_0 \\ \frac{T(\ell,n)^2}{\frac{aT(\lambda(\ell,n))^2}{\bar{d}\mathsf{C}^-(\lambda(\ell,n))} + 4T_{\text{aux}}(\ell,n)} & \text{else.} \end{cases}$$

The proof will be by induction. Specifically, we will show that there is an *st*-composable subspace graph $G_{\ell,n}$ that computes $f_{\ell,n}$ with

1. $\dim H_G \le 2^{S(\ell,n)}$;
2. $\hat{W}_-(G_{\ell,n})/r_{\ell,n} \le \mathsf{C}^-(\ell, n)$, and $\hat{\mathcal{C}}(G_{\ell,n}) \le T(\ell, n) =: \mathsf{C}(\ell, n)$;
3. $\log \dim H_{G_{\ell,n}} = S_{\ell,n} \le cD \log(2a) + cS_{\text{aux}}(\ell, n) + c \log T_{\text{aux}}(\ell, n)$;
4. and basis generation cost $L_{\ell,n} := Dc' \log(2a) + c \log T_{\text{aux}}(\ell, n)$,

where D is the depth of recursion, and c and c' are sufficiently large constants.

Base case: For the base case, suppose $\ell \le \ell_0$. Then we let $G_{\ell,n}$ be the subspace graph from Lemma 4.3.35 derived from the algorithm $\mathcal{P}_{\text{aux},\ell,n}$ and using $\alpha_r = 1$ for all r. Then $G_{\ell,n}$ computes $f_{\text{aux},\ell,n}$ with $\dim H_{G_{\ell,n}} \le c(S_{\text{aux}}(\ell, n) + \log T_{\text{aux}}(\ell, n))$, and basis generation cost at most $c \log T_{\text{aux}}(\ell, n)$, for sufficiently large constant c. By Lemma 4.3.35, we also have

$$\hat{W}_-(G_{\ell,n})/\mathsf{r}_{\ell,n} \le 2T_{\text{aux}}(\ell, n)/2 \le T_{\text{aux},0}, \quad \text{and} \quad \hat{\mathcal{C}}(G_{\ell,n}) \le 2T_{\text{aux}}(\ell, n).$$

Induction case: For the induction step, let $G_{\lambda(\ell,n)}$ be the subspace graph whose existence is guaranteed by the induction hypothesis. First, we will use Corollary 5.2.4, where $G_1, \ldots, G_a$ are each copies of $G_{\lambda(\ell,n)}$ – so $\mathsf{C}_i := \mathsf{C}(\lambda(\ell,n)) = T(\lambda(\ell,n))$ and $\mathsf{L}^- := \mathsf{C}^-(\lambda(\ell,n))$ are known upper bounds on $\hat{\mathcal{C}}(G_i)$ and $\hat{W}_-(G_i)/\mathsf{r}^i$, by the induction hypothesis – to get a subspace graph $G^{\varphi'}_{\ell,n}$. In order to apply Corollary 5.2.4, we have used the fact that L^- does not depend on i. By Corollary 5.2.4, we have

$$\hat{W}_-(G^{\varphi'}_{\ell,n})/\mathsf{r} \leq \mathsf{L}^-\bar{d} = \mathsf{C}^-(\lambda(\ell,n))\bar{d} =: \mathsf{C}_0^-(\ell,n)$$
$$\text{and} \quad \hat{\mathcal{C}}(G^{\varphi'}_{\ell,n})^2 \leq \sum_{i=1}^{a} \mathsf{C}_i^2 = aT(\lambda(\ell,n))^2 =: \mathsf{C}_0(\ell,n)^2.$$

Next, we will use Lemma 5.2.2, with the formula $x_0 \vee x_1$, and subspace graphs $G'_0 = G^{\varphi'}_{\ell,n}$ and G'_1 a subspace graph computing $f_{\mathrm{aux},\ell,n}$ from Lemma 4.3.35, using the known upper bounds $\mathsf{C}_0(\ell,n)$ and $\mathsf{C}_0^-(\ell,n)$ defined above, and $\mathsf{C}_1^-(\ell,n) := T_{\mathrm{aux}}(\ell,n) \geq \hat{W}_-(G'_1)\mathsf{r}'_1$ and $\mathsf{C}_1(\ell,n) := 2T_{\mathrm{aux}}(\ell,n) \geq \hat{\mathcal{C}}(G'_1)$ from Lemma 4.3.35. This gives a subspace graph $G_{\ell,n}$ computing $f_{\ell,n}$ with

$$\hat{\mathcal{C}}(G_{\ell,n})^2 \leq \mathsf{C}_0(\ell,n)^2 + \mathsf{C}_1(\ell,n)^2 \leq aT(\lambda(\ell,n))^2 + 4T_{\mathrm{aux}}(\ell,n)^2 = T(\ell,n)^2$$
$$\hat{W}_-(G_{\ell,n})/\mathsf{r}_{\ell,n} \leq \frac{\mathsf{C}_0(\ell,n)^2 + \mathsf{C}_1(\ell,n)^2}{\frac{\mathsf{C}_0(\ell,n)^2}{\mathsf{C}_0^-(\ell,n)} + \frac{\mathsf{C}_1(\ell,n)^2}{\mathsf{C}_1^-(\ell,n)}} \leq \frac{T(\ell,n)^2}{\frac{aT(\lambda(\ell,n))}{d\mathsf{C}^-(\lambda(\ell,n))} + \frac{4T_{\mathrm{aux}}(\ell,n)^2}{T_{\mathrm{aux}}(\ell,n)}} = \mathsf{C}^-(\ell,n).$$

In order to apply Lemma 5.2.2, we must be able to generate a state proportional to

$$\sqrt{\frac{\mathsf{C}_0(\ell,n)^2}{\mathsf{C}_0^-(\ell,n)}}|0\rangle + \sqrt{\frac{\mathsf{C}_1(\ell,n)^2}{\mathsf{C}_1^-(\ell,n)}}|1\rangle$$

in constant complexity. Since the amplitudes are fixed, input-independent values, we can do this with a single 2-local unitary.

The cost of generating the bases for each of the graphs $G_1, \ldots, G_a$ is at most

$$L_{\lambda(\ell,n)} = (D-1)c' \log(2a) + c \log T_{\mathrm{aux}}(\ell,n)$$

by the induction hypothesis, and so by Corollary 5.2.4, the cost to generate the bases for $G^{\varphi'}_{\ell,n}$ is at most

$$c' \log(a) + L_{\lambda(\ell,n)}.$$

The cost to generate the basis for G'_1 is at most $c \log T_{\text{aux}}(\ell, n) \leq c' \log(a) + L_{\lambda(\ell,n)}$, by Lemma 4.3.35. Then by Lemma 5.2.2 the cost to generate the basis for $G_{\ell,n}$ is at most:

$$\begin{aligned} c' \log(2) + c' \log(a) + L_{\lambda(\ell,n)} &\leq c' \log(2a) + (D-1)c' \log(2a) + c \log T_{\text{aux}}(\ell, n) \\ &= Dc' \log(2a) + c \log T_{\text{aux}}(\ell, n) = L_{\ell,n}, \end{aligned}$$

as needed. Finally, we note that for all $i \in [a]$, by the induction hypothesis

$$\begin{aligned} \log \dim H_{G_i} &\leq c(D-1) \log(2a) + cS_{\text{aux}}(\lambda(\ell, n)) + c \log T_{\text{aux}}(\lambda(\ell, n)) \\ &\leq c(D-1) \log(2a) + cS_{\text{aux}}(\ell, n) + c \log T_{\text{aux}}(\ell, n) \end{aligned}$$

and so by Corollary 5.2.4,

$$\log \dim H_{G^{\varphi'}_{\ell,n}} \leq c(D-1) \log(2a) + cS_{\text{aux}}(\ell, n) + c \log T_{\text{aux}}(\ell, n) + c \log(a).$$

By Lemma 4.3.35,

$$\log \dim H_{G'_1} \leq cS_{\text{aux}}(\ell, n) + c \log T_{\text{aux}}(\ell, n).$$

Thus, by Lemma 5.2.2, if c is a sufficiently large constant,

$$\begin{aligned} \log \dim H_{G_{\ell,n}} &\leq c(D-1) \log(2a) + cS_{\text{aux}}(\ell, n) + c \log T_{\text{aux}}(\ell, n) + c \log(a) + c \log(2) \\ &= cD \log(2a) + cS_{\text{aux}}(\ell, n) + c \log T_{\text{aux}}(\ell, n) = S_{\ell,n}. \end{aligned}$$

This completes the induction. To complete the proof of the theorem, we first apply Corollary 4.4.2 to get a subspace graph $G'_{\ell,n}$ with $\hat{W}_+ \leq 1$, and then apply Theorem 4.3.14, which turns $G'_{\ell,n}$ into a bounded-error quantum algorithm for $f_{\ell,n}$ with time complexity

$$O\left(L_{\ell,n} \sqrt{T(\ell, n)}\right),$$

and space complexity $O(S_{\ell,n} + \log T(\ell, n))$. We complete the proof by noting that

$$\log T(\ell, n) \geq \log \sqrt{a}^D = \frac{D}{2} \log a \geq \frac{D}{4} \log(2a),$$

so $L_{\ell,n} = O(\log(T(\ell, n)))$, and $S_{\ell,n} = O(S_{\text{aux}}(\ell, n) + \log T(\ell, n))$. □

5.4 Application to DSTCON

In this section we consider the *directed st-connectivity* problem. First, we specify the graph access model. We will work in the adjacency matrix model, where we

assume that for a directed graph $G = (V, E)$ the input is given as an oracle $\mathcal{O}_G$ that can be queried in unit cost, where for any $u, v \in V$

$$\mathcal{O}_G : |u\rangle|v\rangle|0\rangle \mapsto \begin{cases} |u\rangle|v\rangle|1\rangle & \text{if } (u,v) \in E \\ |u\rangle|v\rangle|0\rangle & \text{otherwise.} \end{cases}$$

Without loss of generality, we assume $(u, u) \in E$ for any $u \in V$. This doesn't change the presence or absence of paths in G.
However, we note that the time complexity $2^{\frac{1}{2}\log^2 n + O(\log n)}$ in Theorem 5.4.3 also holds for the edge list model, in which the algorithm can query the i-th out- or in-neighbour of a vertex. That is because given such access, we can implement a query to $\mathcal{O}_G$ in $\mathrm{poly}(n)$ time and $O(\log n)$ space, and $\mathrm{poly}(n)$ factors are suppressed in $2^{O(\log n)}$.

5.4.1. Problem (DSTCON). *Given access to a directed graph $G = (V, E)$ via the oracle $\mathcal{O}_G$, and two vertices $s, t \in V$, decide whether there is a directed path from s to t in G.*

There is a classical recursive algorithm for DSTCON that operates in the low-space regime due to Savitch [Sav70]. We first describe the subroutine $\textsc{path}_\ell(\mathcal{O}_G, u, v)$ that will be called recursively in the algorithm. This subroutine decides whether there is a path of length at most ℓ from u to v in G. For $\ell \geq 2$, the subroutine searches over all vertices for some w such that there are paths of length at most $\ell/2$ from s to w and w to t:

Subroutine 1: $\textsc{path}_\ell(\mathcal{O}_G, u, v)$ for $\ell \geq 2$

Input: Oracle $\mathcal{O}_G$, $u, v \in V$
Output: 1 if there is a path of length $\leq \ell$ from u to v in G; 0 otherwise.
for $w \in V$ **do**
  $b_w \coloneqq \textsc{path}(\ell/2, u, w) \wedge \textsc{path}(\ell/2, w, v)$;
return $\bigvee_{w \in V} b_w$;

Next, we describe the base case. When $\ell = 1$, the subroutine simply performs a single query $\mathcal{O}_G(u, v)$ and outputs:

$$\textsc{path}_1(\mathcal{O}_G, u, v) = \begin{cases} 1 & \text{if } (u,v) \in E \\ 0 & \text{otherwise.} \end{cases}$$

Then Savitch's algorithm simply outputs $\textsc{path}_n(\mathcal{O}_G, s, t)$, where $n = |V|$, to decide if there is a path from s to t.
The following result of Savitch is easy to verify.

Algorithm 3 Savitch's algorithm for DSTCON

Input: Oracle $\mathcal{O}_G$ for $G = (V, E)$ with $|V| = n$, $s, t \in V$
Output: 1 if there is a path from s to t in G; 0 otherwise.
return $\text{PATH}_n(\mathcal{O}_G, s, t)$;

5.4.2. THEOREM. *Algorithm 3 decides DSTCON in time* $O((2n)^{\log n} = 2^{\log^2 + O(\log n)})$ *and space* $O(\log^2 n)$.

Next, we show a quantum speedup for Savitch's algorithm via application of Theorem 5.3.1.

5.4.3. THEOREM. *Let* $G = (V, E)$ *be a directed graph,* $|V| = n$. *Then there exists a recursive quantum algorithm that decides DSTCON on* G *with bounded error in time* $\widetilde{O}((\sqrt{2n})^{\log n}) = 2^{\frac{1}{2}\log^2 n + O(\log n)}$ *and space* $O(\log^2 n)$.

Proof:
To show existence of such a quantum algorithm, we will rephrase algorithm Algorithm 3 in terms of the condition of Theorem 5.3.1 and analyze its complexity. We start with defining a function corresponding to DSTCON.

$$f_{\ell,n} : \{0,1\}^{n^2} \times \{0,1\}^{\log n} \times \{0,1\}^{\log n} \to \{0,1\}$$

$$(G, u, v) \mapsto \begin{cases} 1 & \text{if there is a path from } u \text{ to } v \text{ in } G \text{ of length} \leq \ell \\ 0 & \text{otherwise,} \end{cases}$$

where G encodes a directed graph on n vertices and u,v encode two vertices in G.

Next, we define $\lambda_1(\ell) = \ell/2$ and $\lambda_2(n) = n$. This specifies the recursion. Finally, we set $\ell_0 = 1$ and define

$$f_{aux,1,n}(G, u, v) = \begin{cases} 1 & \text{if } (u, v) \in E \\ 0 & \text{otherwise.} \end{cases}$$

and $f_{aux,\ell,n} = 0$ for all $\ell > 1$. Then $\mathcal{P}_{\text{aux},\ell,n}$ is a single edge query when $\ell = 1$ and does nothing otherwise. This implies $T_{\text{aux}}(\ell, n) = O(1)$ and $S_{\text{aux}}(\ell, n) = O(\log n)$ as this is the space needed to write down a vertex name. We have everything to write down $f_{\ell,n}$ recursively, in terms of a symmetric formula φ' in $a = 2n$ variables:

$$f_{\ell,n}(G, u, v) = \underbrace{\bigvee_{w \in V} \left(f_{\ell/2,n}(G, u, w) \wedge f_{\ell/2,n}(G, w, v)\right)}_{\varphi'} \vee f_{\text{aux},\ell,n}(G, u, v).$$

Now we apply Theorem 5.3.1 to everything described above and conclude that there is a recursive bounded error quantum algorithm that computes $f_{\ell,n}$ with time complexity $\widetilde{O}(T(\ell,n))$ and space complexity $O(S_{\text{aux}}(\ell,n)+\log T(\ell,n))$, such that we have for all $\ell > 1$:

$$T(\ell,n) = \sqrt{2n}T(\ell/2,n) + O(1),$$

and $T(1,n) = O(1)$. By solving the recursion, we obtain

$$T(n,n) = O((\sqrt{2n})^{\log n}) = O\Big(2^{\frac{1}{2}(\log^2 n+\log n)}\Big).$$

The space complexity of the algorithm for $f_{n,n}$ is thus

$$O(S_{\text{aux}}(n,n)+\log T(n,n)) = O(\log^2 n).$$

Since $f_{n,n}$ precisely describes the problem DSTCON, this concludes the proof. □

Chapter 6

A quantum time-space tradeoff for directed st-connectivity

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.

This chapter is based on the paper "A quantum time-space tradeoff for directed st-connectivity" by Stacey Jeffery and Galina Pass [JP25b].

6.1 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 $S = \tilde{O}(n)$ 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

$$T \leq 2^{\log^2(\frac{n}{S})+O(\log n \log\log n)} \tag{6.1.1}$$

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, Chapter 5 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 (6.1.1)) 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 6.2.6.

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

[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.

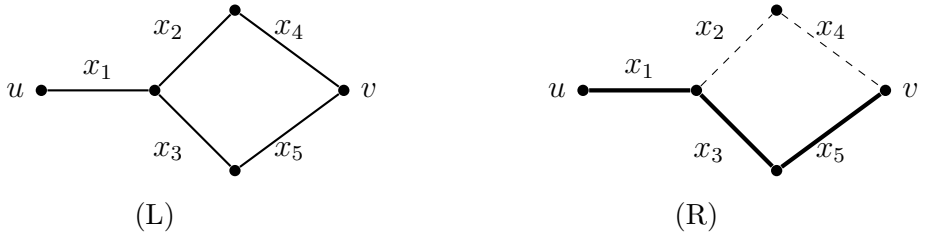

Figure 6.1: (L) An example of a switching network. (R) The same switching network, with edges switched on (thick) or off (dashed) by the assignment $x = 10101$. In this example, u and v are connected by a path of edges labelled by variables that are true under x, and so the switching network accepts x.

from u to v of length at most L. We call this problem $\textsc{Dist}_L$. 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 $\textsc{Dist}_L$ 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.

To address this, we design a new quantum algorithm for $\textsc{Dist}_L$ that is based on a recursively constructed *switching network*. A switching network (Definition 4.3.7) 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 6.1.

Switching networks are a natural and well-motivated model in classical com-

puting, 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 Chapter 4, 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 $\textsc{Dist}_L$, the remainder of the S space is classical.

Our algorithm for $\textsc{Dist}_L$ 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 $\textsc{Dist}_L$ 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 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 chapter is organized as follows. In Section 6.2, 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 $\textsc{Dist}_L$, which we describe in Section 6.3.

6.2 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 chapter, which we prove in Section 6.3. Specifically, we present a quantum algorithm for deciding directed st-connectivity under an additional constraint on the path length (i.e. solving $\textsc{Path}_L$). While this subroutine is quantum, the algorithm we decide in the remainder of this section is otherwise classical.

6.2.1. THEOREM. *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 $i, j \in [n]$, $b \in \{0, 1\}$*

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

Let $L \leq n$ be a power of 2. Then there is a bounded-error quantum algorithm, $\mathtt{D}_L(G, u, v)$, 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 6.3, we prove the statement for L a power of 2, using a recursive structure, but a simple corollary extends this result to any L.

6.2.2. COROLLARY. *Let G be as in Theorem 6.2.1, 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 $O(\log(L)\log(n))$.

Proof:
To prove the statement, we exhibit a quantum algorithm, Algorithm 4 that uses one call to the subroutine $\mathtt{D}$ from Theorem 6.2.1, 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 6.2). Since G' can be queried using at most one query to G, the result follows. □

Algorithm 4 $\mathtt{Dist}_L(G,u,v)$

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

6.2.1 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 5, from [BBRS98], that makes calls to a subroutine for the problem $\textsc{Dist}_L(G,u,v)$, of deciding whether there is a path of

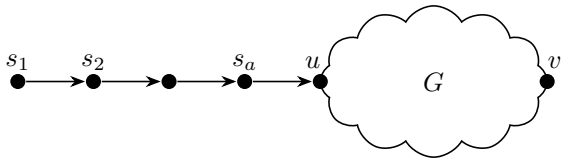

Figure 6.2: The graph G' constructed from G. It is clear that there is a uv-path of length at most L in G if and only if there is a s_1v-path of length at most $L + a$ in G'.

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 $\texttt{Dist}_L$ from Corollary 6.2.2.
Ref. [BBRS98] show that this algorithm correctly decides DSTCON whenever $\texttt{Dist}_L$ decides $\textsc{Path}_L$. From [BBRS98], or by inspecting Algorithm 5, we get the following.

6.2.3. Lemma. *Let $\texttt{Dist}_L(G, u, v)$ be a bounded-error algorithm for $\textsc{Dist}_L$, with time complexity $D_T(n, L)$, and the space complexity is $D_S(n, L)$. Then the time complexity of Algorithm 5 is*

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

and its space complexity is

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

If $\texttt{Dist}_L$ 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 6.2.2 yields the following theorem, which combines our quantum subroutine with the classical BFS algorithm. Although the algorithm in Corollary 6.2.2 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.

6.2.4. Theorem. *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 $i, j \in [n]$, $b \in \{0, 1\}$,*

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

Algorithm 5 $\texttt{DSTCON}_L(G, s, t)$ [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

for $j = 0, \ldots, L-1$ **do**
  $S \leftarrow \{s\}$
  for all vertices $v \in V$ **do**
    if $\texttt{Dist}_j(s, v) = 1 \wedge \texttt{Dist}_{j-1}(s, v) = 0$ **then**
      if $|S| > n/L$ **then** try next j
      else add v to S
      end if
    end if
  end for
  for $i = 1, \ldots, \lfloor n/L \rfloor$ **do**
    $S' = \varnothing$
    for all vertices $v \in V$ **do**
      if $\exists u \in S : \texttt{Dist}_L(u, v) = 1 \wedge \forall u \in S : \texttt{Dist}_{L-1}(u, v) = 0$ **then**
        if $|S| + |S'| > n/L$ **then** try next j
        else add v to S'
        end if
      end if
    end for
    $S = S \cup S'$
  end for
  if t within distance L of a vertex in S **then return** (CONNECTED)
  else return (NOT CONNECTED)
  end if
end for

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

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

and total space

$$\widetilde{O}\bigg(\Big(\frac{n}{L} + \log L\Big)\log n\bigg)$$

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

6.2.2 Complexity comparison: classical vs. quantum

6.2.5. THEOREM ([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.

6.2.6. THEOREM. *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 $i, j \in [n]$, $b \in \{0, 1\}$,*

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

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 6.2.4. 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

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

□

6.2.7. Remark. For $S = o(n^{1/2})$, the quantum algorithm of Theorem 6.2.6 achieves a better time-space tradeoff than the classical time-space tradeoff stated in Theorem 6.2.5.

6.3 Quantum short path subroutine

In this section, we prove Theorem 6.2.1 by describing and analyzing a quantum algorithm for $\textsc{Dist}_L(G, s, t)$, 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 4.3.7), and then applying Theorem 4.3.19.

In Section 6.3.1, we describe the switching network, through a recursive construction. In order to apply Theorem 4.3.19, 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 6.3.2; and describe a basis for the space $\mathcal{B}^{\perp}$, and a procedure for generating it, which we do in Section 6.3.3. Finally, in Section 6.3.4, we put it all together to prove Theorem 6.2.1.

6.3.1 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 2.4.7), 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 4.3.7) 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 6.3.1.

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 6.3.1. The final construction $\mathcal{N}_L(s)$ has source $\mathtt{s} = [s]$ and a single sink $\mathtt{t} = [s, t]$. We write $\mathcal{N}_L(s, t)$ when we want to emphasize this.

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:

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

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 6.3). 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

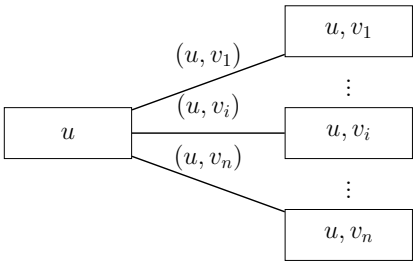

Figure 6.3: Graph construction for the switching network $\mathcal{N}_1(u)$ that decides whether each vertex of a graph $G = (V, E)$ is reachable from a vertex $u \in V$ by a path of length 1. Each edge $([u], [u, v_i])$ of $\mathcal{N}_1(u)$ has query label (u, v_i) and is "on" in $\mathcal{N}_1(u)(G)$ 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 $\mathcal{N}_1(u)(G)$.

(see Definition 2.4.2 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).

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 $\mathcal{N}^v_{2^{\ell-1}}(u)$ 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:

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

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 2.4.7) the i-th sink of $\mathcal{N}^0_{2^\ell}$ – which encodes $[u, v_i]$ – to the source of $\mathcal{N}^{(1,i)}_{2^\ell}$ – 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 6.4. 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 $\mathcal{N}_{2^\ell}(u)$.

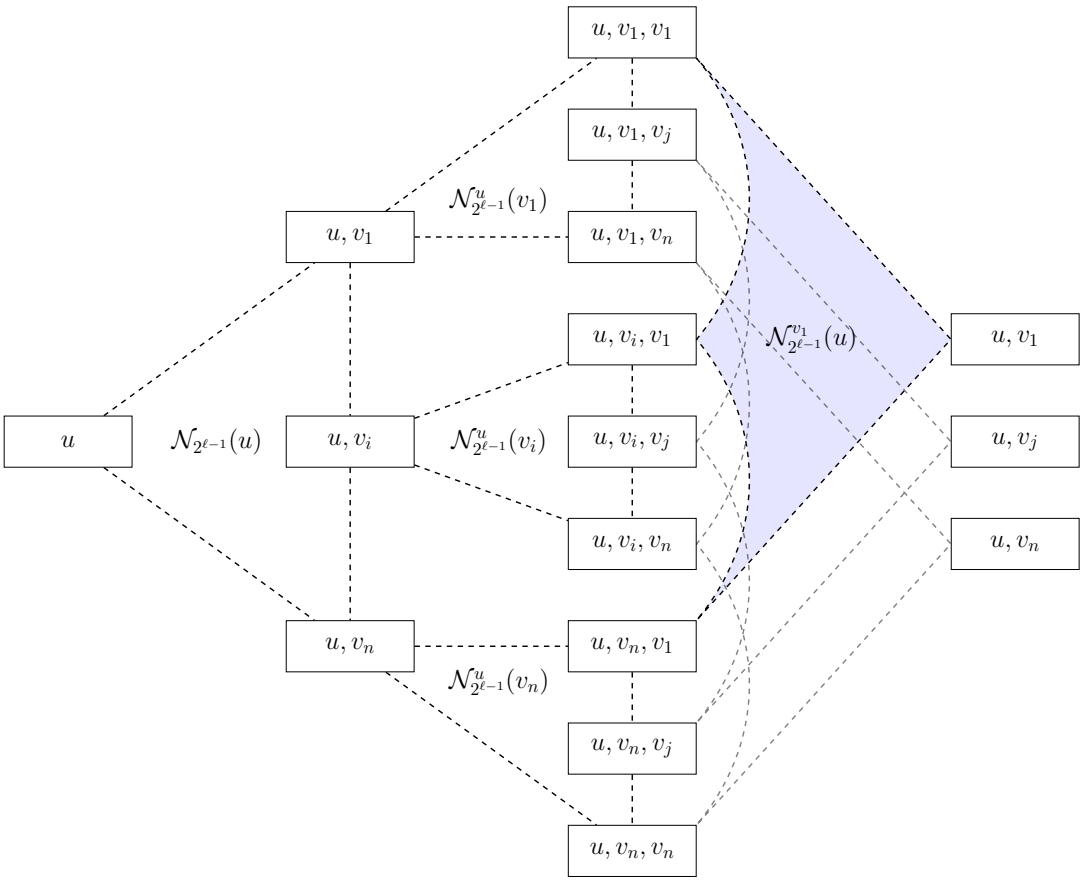

Figure 6.4: Graph construction for the switching network $\mathcal{N}_{2^\ell}(u)$ that decides whether each vertex of a graph $G = (V, E)$ is reachable from a vertex $u \in V$ by a path of length 2^ℓ.

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:

$$E_{2^\ell} = \bigsqcup_{\sigma \in \Sigma} E(\mathcal{N}_{2^{\ell-1}}^{\sigma}) = \Sigma \times E_{2^{\ell-1}} = \Sigma^\ell \times E_1. \tag{6.3.1}$$

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

$$(-1)^{|\sigma|_2} |e_i^\sigma\rangle := |\sigma, e_i\rangle, \tag{6.3.2}$$

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*[2] (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

$$|\psi^\sigma\rangle = (-1)^{|\sigma|_2}|\sigma\rangle|\psi\rangle, \tag{6.3.3}$$

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 $\mathcal{N}^\sigma_{2^\ell}$) 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 $\mathcal{N}^\sigma_{2^{\ell-1}}$, 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 $\mathcal{N}^\sigma_{2^{\ell-k}}$ 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.

6.3.1. Lemma. *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 $\mathcal{N}^{v_j}_{2^{\ell-1}}(u)$ 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,

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

$f_1(\sigma) = f_1(\tilde{\sigma})$, 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 $\mathcal{N}^{(1,i)}_{2^{\ell-1}}$ for $i \in [n]$. These are the switching networks $\mathcal{N}^{u}_{2^{\ell-1}}(v_i)$ 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 $\mathcal{N}_{2^{\ell+1}}$.

6.3.2. Lemma. *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 $\mathcal{N}_{2^{\ell+1}}$. □

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.

6.3.3. Lemma. *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 6.3.3, 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.

6.3.4. Lemma ([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 $D(\ell) \leq 2^{\ell-1} - 1$.*

6.3.5. Lemma. *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 6.3.4. 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 $D(\ell) + 1$. □

6.3.6. Lemma. *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 $M_L = 3M_{L/2} = 3^{\log L} = L^{\log 3}$. □

6.3.7. Lemma. *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.

□

6.3.8. Lemma. *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 6.3.6, 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 6.3.7, v is reachable from u in G. □

6.3.9. Lemma. *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. □

Proof of Lemma 6.3.3:

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

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 6.3.9, there is a sequence of pebbling game moves on G that transforms $[u]$ into $[u, v_i]$. Therefore, by Lemma 6.3.8, 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 6.3.5, the path in G is of length at most 2^ℓ. □

6.3.2 Complexity analysis of the switching network

6.3.10. Lemma. *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 6.3.3. □

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

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

6.3.3 Basis of the space of st-flows

In order to apply Theorem 4.3.19, we need to describe a basis for $\mathcal{B}^\perp$, and give an efficient procedure for generating it. By Lemma 4.3.33, it is enough to describe a basis for the space $\mathcal{F}(\mathcal{N}_L) + \text{span}\{|[u]\rangle, |[u, v_i]\rangle\}$ (see Definition 4.3.30), and by Lemma 4.3.32, we can further decompose this task into finding a basis for the circulation space $\mathcal{C}(\mathcal{N}_L)$ (see Definition 4.3.30) and an optimal st-flow. We first define the basis, and then describe a procedure for generating it.

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 $\mathcal{N}_L(s, t)$.

6.3.12. THEOREM. *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 (4.3.7). For $i, j \in \{0,1\}^{\log n}$, let*

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

where the superscript encodes the copy of $\mathcal{N}_{2^{\ell-1}}$ in $\mathcal{N}_{2^\ell}$ (see Figure 6.5), 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

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

Then

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

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

This result follows directly from Lemma 6.3.13 (below), which recursively constructs an orthogonal basis for the circulation space, $\mathcal{C}(\mathcal{N}_{2^\ell})$, of $\mathcal{N}_{2^\ell}$ (see Definition 4.3.30); Lemma 6.3.15 (below), which recursively constructs an optimal flow state, which is a basis for the space of optimal flows; and Lemma 6.3.16 (below), which states that $\mathcal{B}^\perp$ decomposes as the direct sum of the circulation space and a three-dimensional space that contains the optimal flow.

6.3.13. Lemma. *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 (4.3.7). 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 6.3.12. Let* $\{|b_1\rangle,\ldots,|b_D\rangle\}$ *be an orthogonal basis for* $\mathcal{C}(\mathcal{N}_{2^{\ell-1}})$. *Then*

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

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 $\sigma\in\Sigma$, $\mathrm{span}\{|\sigma\rangle|b_d\rangle : d\in[D]\}$ 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 $\mathrm{span}\{|\sigma\rangle|b_d\rangle : d\in[D]\}$, 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 4.3.31, and don't overlap other blocks.

Next, we compute the dimension of $\mathcal{C}(\mathcal{N}_{2^\ell})$. By Lemma 4.3.34, $\dim\mathcal{C}(\mathcal{N}_{2^\ell}) = |E(\mathcal{N}_{2^\ell})|-|V(\mathcal{N}_{2^\ell})|+1$. 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,

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

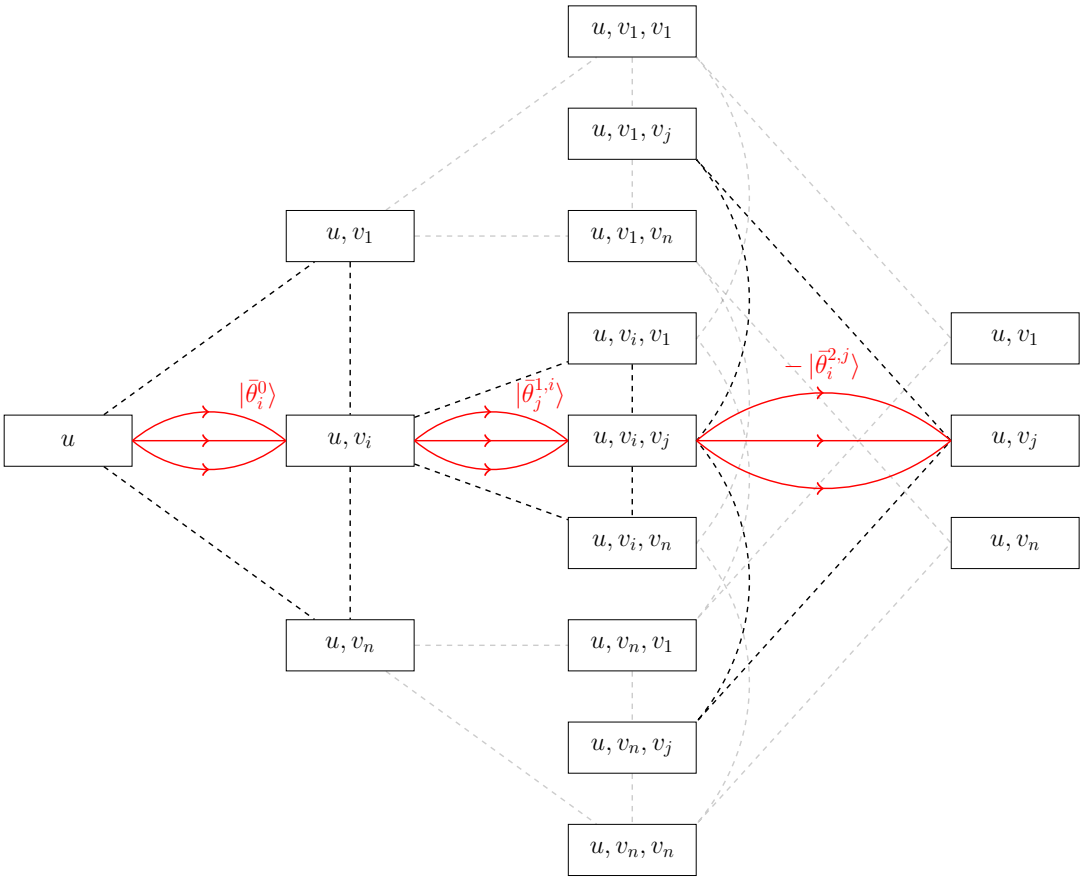

Figure 6.5: Visualization of the state $|p_{ij}\rangle = |\bar{\theta}_i^0\rangle + |\bar{\theta}_j^{1i}\rangle - |\bar{\theta}_i^{2j}\rangle = |0,\bar{0}\rangle|\bar{\theta}_i\rangle + |1,i\rangle|\bar{\theta}_j\rangle + |2,j\rangle|\bar{\theta}_i\rangle$ in $\mathcal{N}_{2^\ell}$, where $j \in \{0,1\}^{\log n}$ 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 (4.3.7). The superscript encodes the copy of $\mathcal{N}_{2^{\ell-1}}$ in $\mathcal{N}_{2^\ell}$.

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: $\psi_{z,x}(\mathsf{u}) = \sum_{e\in E^{\rightarrow}(\mathsf{u})} \psi_{z,x}(e) - \sum_{e\in E^{\leftarrow}(\mathsf{u})} \psi_{z,x}(e)$.

Note that each p_{ij} is a $[u],[u,v_j]$-flow in $\mathcal{N}_{2^\ell}$, as shown in Figure 6.5, 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

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

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.

6.3.14. Claim. *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,

$$\begin{aligned}
\langle p_{ij}|p_{i'j'}\rangle &= \langle \bar{\theta}^0_i | \bar{\theta}^0_{i'} \rangle + \langle \bar{\theta}^{1i}_j | \bar{\theta}^{1i'}_{j'} \rangle + \langle \bar{\theta}^{2j}_i | \bar{\theta}^{2j'}_{i'} \rangle \\
&= \begin{cases} c_{00} := c_0 + 0 + 0, & \text{if } i \neq i', j \neq j' \\ c_{01} := c_0 + 0 + c_0, & \text{if } i \neq i', j = j' \\ c_{10} := c_1 + c_0 + 0, & \text{if } i = i', j \neq j' \\ c_{11} := c_1 + c_1 + c_1, & \text{if } i = i', j = j'. \end{cases}
\end{aligned} \tag{6.3.4}$$

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:

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

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

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

$$\begin{aligned}
\langle\psi_{z',x'}|\psi_{z,x}\rangle &= \sum_{i,j\in\{0,1\}^{\log n}} (-1)^{x'\cdot j+z'\cdot i}\langle p_{ij}|\psi_{z,x}\rangle \\
&= \sum_{i,j\in\{0,1\}^{\log n}} (-1)^{x'\cdot j+z'\cdot i}(-1)^{z\cdot i}\Big(c(-1)^{x\cdot j} + c'A_{x,j}\Big) \\
&\quad \text{(by Claim 6.3.14)} \\
&= \sum_{i\in\{0,1\}^{\log n}} (-1)^{(z'+z)\cdot i} \cdot \sum_{j\in\{0,1\}^{\log n}} \Big(c(-1)^{(x'+x)\cdot j} + c'(-1)^{x'\cdot j}A_{x,j}\Big).
\end{aligned}$$

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

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

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. □

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 $[u, v_j]$.

6.3.15. Lemma. *For $i, j \in \{0,1\}^{\log n}$, let $|p_{ij}\rangle = |p_{ij}(2^\ell)\rangle$ be as in Theorem 6.3.12. Then for any j, the optimal unit $[u], [u, v_j]$-flow in $\mathcal{N}_{2^\ell}(u, v_j)$ is*

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

Proof:
Note that each p_{ij} is a $[u], [u, v_j]$-flow in $\mathcal{N}_{2^\ell}$, as shown in Figure 6.5, 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 $|\tilde{\theta}_j(2^\ell)\rangle$ is orthogonal to the circulation subspace $\mathcal{C}(\mathcal{N}_{2^\ell})$. By Lemma 4.3.31 and Lemma 4.3.32, this orthogonality implies that $|\tilde{\theta}_j(2^\ell)\rangle$ is the optimal unit flow.

By Lemma 6.3.13, 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 6.3.12, and $\{|b_1\rangle, \ldots, |b_D\rangle\}$ is an orthogonal basis for the circulation space $\mathcal{C}(\mathcal{N}_{2^{\ell-1}})$. All $|\tilde{\theta}_j(2^\ell)\rangle$ 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 $\langle \tilde{\theta}_j(2^\ell) | \psi_{z,x} \rangle = 0$ for every $j, z, x \in \{0,1\}^{\log n}$ such that $z \neq \bar{0}$. We apply Claim 6.3.14 to get:

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

□

6.3.16. Lemma. *The space $\mathcal{B}^\perp$ admits the following direct-sum decomposition:*

$$\mathcal{B}^\perp = \mathcal{C}(\mathcal{N}_L) \oplus \operatorname{span}\left\{|\tilde{\theta}_j(L)\rangle, |[u]\rangle, |[u, v_j]\rangle\right\}.$$

Proof:
By Lemma 4.3.33 and Lemma 4.3.32, we can write

$$\begin{aligned}\mathcal{B}^{\perp} &= \mathcal{F}(\mathcal{N}_L) + \mathrm{span}\{|[u]\rangle, |[u,v_i]\rangle\} \\ &= \mathcal{C}(\mathcal{N}_L) \oplus \mathcal{F}^{\mathrm{OPT}}(\mathcal{N}_L) + \mathrm{span}\{|[u]\rangle, |[u,v_i]\rangle\}.\end{aligned}$$

Note that circulations cannot overlap any boundary states. Hence,

$$\mathcal{B}^{\perp} = \mathcal{C}(\mathcal{N}_L) \oplus \Big(\mathcal{F}^{\mathrm{OPT}}(\mathcal{N}_L) + \mathrm{span}\{|[u]\rangle, |[u,v_i]\rangle\}\Big).$$

The space $\mathcal{F}^{\mathrm{OPT}}(\mathcal{N}_L)$ is one-dimensional and is spanned by the optimal unit flow state

$$|\theta_j(L)\rangle = -|[u]\rangle + |\tilde{\theta}_j(L)\rangle + |[u,v_i]\rangle.$$

Hence, we can write

$$\mathcal{B}^{\perp} = \mathcal{C}(\mathcal{N}_L) \oplus \mathrm{span}\{|\theta_j(L)\rangle, |[u]\rangle, |[u,v_i]\rangle\}.$$

Finally, note that $|\theta_j(L)\rangle \in \mathrm{span}\{|\tilde{\theta}_j(L)\rangle, |[u]\rangle, |[u,v_j]\rangle\}$ and conversely $|\tilde{\theta}_j(L)\rangle \in \mathrm{span}\{|\theta_j(L)\rangle, |[u]\rangle, |[u,v_j]\rangle\}$. Therefore,

$$\mathrm{span}\{|\theta_j(L)\rangle, |[u]\rangle, |[u,v_j]\rangle\} = \mathrm{span}\{|\tilde{\theta}_j(L)\rangle, |[u]\rangle, |[u,v_j]\rangle\}.$$

This concludes the proof. □

Basis generation

We show in this section how to prepare the basis from Theorem 6.3.12 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 (6.3.1) 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 $\tau \in \{0,1,2\}^\ell$:

$$E_\tau := \{(\sigma, i) \in E_{2^\ell} : \bar{\sigma} = \tau\} = \{(\sigma, i) \in \Sigma^\ell \times \{0,1\}^{\log n} : \bar{\sigma} = \tau\}. \tag{6.3.5}$$

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.

6.3.17. LEMMA. *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 (6.3.5), we have, for any $\tau \in \{0,1,2\}^{\ell-1}$:

$$\begin{aligned} E_{0\tau} &= \{((0,\bar{0})\sigma, i) \in E_{2^\ell} : \bar{\sigma} = \tau\} \\ &= \{((0,\bar{0}), e) : e \in E_\tau\} \\ \text{and for } a \in \{1,2\},\ E_{a\tau} &= \{((a,j)\sigma, i) \in E_{2^\ell} : \bar{\sigma} = \tau\} \\ &= \{((a,j), e) : e \in E_\tau, j \in \{0,1\}^{\log n}\}. \end{aligned} \tag{6.3.6}$$

From this, and the induction hypothesis, we have:

$$\begin{aligned} |E_{0\tau}| = |E_\tau| = n^{1+|\tau|-|\tau|_0} = n^{1+|\tau|+1-(|\tau|_0+1)} = n^{1+|0\tau|-|0\tau|_0} \\ \text{and for } a \in \{1,2\},\ |E_{a\tau}| = n|E_\tau| = n \cdot n^{1+|\tau|-|\tau|_0} = n^{1+(1+|\tau|)-|\tau|_0} = n^{1+|a\tau|-|a\tau|_0}. \end{aligned}$$

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

6.3.18. LEMMA. *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 6.3.17 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. □

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

6.3.19. Lemma. *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

$$\sum_{j\in\{0,1\}^{\log n}} |\bar{\theta}_j(2^\ell)\rangle = n \cdot \sum_{\tau\in\{0,1,2\}^\ell} \frac{1}{|E_\tau|} \sum_{e\in E_\tau} |e\rangle. \tag{6.3.7}$$

Using the definition in the statement of Theorem 6.3.12, 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 6.3.12, we have

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

by (6.3.3), 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

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

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 (6.3.7) holds for $\mathcal{N}_{2^{\ell-1}}$. We have

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

where we again used (6.3.3). We can use the induction hypothesis on each of the three terms to get:

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

From (6.3.6), and Lemma 6.3.17, it follows that

$$\begin{aligned}\sum_{j\in\{0,1\}^{\log n}} |\bar{\theta}_j(2^\ell)\rangle &= \sum_{\tau\in\{0,1,2\}^{\ell-1}} \frac{1}{n^{|\tau|-|\tau|_0}} \sum_{e\in E_\tau} |(0,\bar{0}),e\rangle \\ &+ \sum_{a\in\{1,2\}} \sum_{\tau\in\{0,1,2\}^{\ell-1}} \frac{1}{n^{1+|\tau|-|\tau|_0}} \sum_{i\in\{0,1\}^{\log n}, e\in E_\tau} |(a,i),e\rangle \\ &= \sum_{\tau\in\{0,1,2\}^{\ell-1}} \frac{1}{n^{|0\tau|-|0\tau|_0}} \sum_{e'\in E_{0\tau}} |e'\rangle \\ &+ \sum_{a\in\{1,2\}} \sum_{\tau\in\{0,1,2\}^{\ell-1}} \frac{1}{n^{|a\tau|-|a\tau|_0}} \sum_{e'\in E_{a\tau}} |e'\rangle \\ &= n \sum_{\tau'\in\{0,1,2\}^{\ell}} \frac{1}{|E_{\tau'}|} \sum_{e'\in E_{\tau'}} |e'\rangle,\end{aligned}$$

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 2.2.2, 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 6.3.17 and Lemma 6.3.18 to compute the amplitudes:

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

Finally, we compute the partial sums, again using Lemma 6.3.17 and Lemma 6.3.18. 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

$$\sum_{\tau \in \{0,1,2\}^\ell} \frac{1}{\sqrt{|E_\tau|}} \sum_{e \in E_\tau} \frac{1}{\sqrt{|E_\tau|}} |e\rangle = \sum_{\tau \in \{0,1,2\}^\ell} \frac{1}{|E_\tau|} \sum_{e \in E_\tau} |e\rangle, \tag{6.3.8}$$

which is proportional to $\sum_{j \in \{0,1\}^{\log n}} |\bar{\theta}_j(2^\ell)\rangle$ by (6.3.7). □

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 6.3.12.

6.3.20. Lemma. *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 6.3.19, 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.

$$\begin{aligned}
\sum_{j\in\{0,1\}^{\log n}} (-1)^{x\cdot j}|\bar{\theta}_j(2)\rangle &= \sum_{j\in\{0,1\}^{\log n}} (-1)^{x\cdot j} \sum_{i\in\{0,1\}^{\log n}} \left(|e_i^0\rangle + |e_j^{1i}\rangle - |e_i^{2j}\rangle\right) \\
&= \underbrace{\sum_{j\in\{0,1\}^{\log n}} (-1)^{x\cdot j}}_{=0} \sum_{i\in\{0,1\}^{\log n}} |e_i^0\rangle \\
&\quad + \sum_{i,j\in\{0,1\}^{\log n}} (-1)^{x\cdot j}|e_j^{1i}\rangle \\
&\quad - \sum_{i,j\in\{0,1\}^{\log n}} (-1)^{x\cdot j}|e_i^{2j}\rangle \\
&= \sum_{i,j\in\{0,1\}^{\log n}} (-1)^{x\cdot j}|1,i\rangle|j\rangle + \sum_{i,j\in\{0,1\}^{\log n}} (-1)^{x\cdot j}|2,j\rangle|i\rangle \\
&\propto |1\rangle H^{\otimes \log n}|\bar{0}\rangle H^{\otimes \log n}|x\rangle + |2\rangle H^{\otimes \log n}|x\rangle H^{\otimes \log n}|\bar{0}\rangle.
\end{aligned} \tag{6.3.9}$$

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 (6.3.2)). 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 $\{0,1\}^{\log n}$).

$$\begin{aligned}\sum_{j\in\{0,1\}^{\log n}}(-1)^{x\cdot j}|\bar{\theta}_j(2^\ell)\rangle &= \sum_j(-1)^{x\cdot j}\sum_i\Big(|\bar{\theta}_i^0(2^{\ell-1})\rangle + |\bar{\theta}_j^{1i}(2^{\ell-1})\rangle - |\bar{\theta}_i^{2j}(2^{\ell-1})\rangle\Big)\\ &= \underbrace{\sum_j(-1)^{x\cdot j}}_{=0}\sum_i|\bar{\theta}_i^0(2^{\ell-1})\rangle\\ &+\sum_j\sum_i(-1)^{x\cdot j}|\bar{\theta}_j^{1i}(2^{\ell-1})\rangle\\ &-\sum_j\sum_i(-1)^{x\cdot j}|\bar{\theta}_i^{2j}(2^{\ell-1})\rangle\\ &=\sum_j\sum_i(-1)^{x\cdot j}|1,i\rangle|\bar{\theta}_j(2^{\ell-1})\rangle\\ &+\sum_j\sum_i(-1)^{x\cdot j}|2,j\rangle|\bar{\theta}_i(2^{\ell-1})\rangle.\end{aligned}$$

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 (6.3.2). Continuing, we use the induction hypothesis to compute (again, all sums are over $\{0,1\}^{\log n}$):

$$\begin{aligned}&\sum_{j\in\{0,1\}^{\log n}}^{n}(-1)^{x\cdot j}|\bar{\theta}_j(2^\ell)\rangle\\ &=\sum_i|1,i\rangle\otimes\sum_j(-1)^{x\cdot j}|\bar{\theta}_j(2^{\ell-1})\rangle+\sum_j(-1)^{x\cdot j}|2,j\rangle\otimes\sum_i|\bar{\theta}_i(2^{\ell-1})\rangle\\ &=|1\rangle\otimes\sqrt{n}H^{\otimes\log n}|\bar{0}\rangle\otimes\left\|\sum_{j\in\{0,1\}^{\log n}}(-1)^{x\cdot j}|\bar{\theta}_j(2^{\ell-1})\rangle\right\|C_{2^{\ell-1}}|x\rangle\\ &\qquad+|2\rangle\otimes\sqrt{n}H^{\otimes\log n}|x\rangle\otimes\left\|\sum_{j\in\{0,1\}^{\log n}}|\bar{\theta}_j(2^{\ell-1})\rangle\right\|C_{2^{\ell-1}}|\bar{0}\rangle.\end{aligned}\tag{6.3.10}$$

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

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

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 2.2.1, 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 (6.3.7) and Lemma 6.3.18, we can conclude that for any $\ell > 0$:

$$N_{\bar{0}}(2^\ell) := \left\| \sum_{i \in \{0,1\}^{\log n}} |\bar{\theta}_i(2^\ell)\rangle \right\|^2 = n^2 \sum_{\tau \in \{0,1,2\}^\ell} \frac{1}{|E_\tau|} = n^2 \frac{(n+2)^\ell}{n^{\ell+1}}. \tag{6.3.11}$$

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 $\ell \geq 1$:

$$N_x(2^\ell) = n^{\ell+1} + \frac{n^3}{(n-2)(n+1)} \left(n^\ell - \frac{(n+2)^\ell}{n^\ell} \right). \tag{6.3.12}$$

For the base step, we can observe from (6.3.9) that $N_x(2) = 2n^2$, which is equal to the right-hand-side of (6.3.12) when $\ell = 1$. For the induction step, we can use (6.3.10) 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 (6.3.12). □

We now describe how to generate the states $|\psi_{z,x}(2^\ell)\rangle$, which form most of the basis described in Theorem 6.3.12.

6.3.21. Lemma. *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 6.3.12) and split it into three orthogonal terms. (All sums below are over $\{0,1\}^{\log n}$).

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

By (6.3.3), 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:

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

where $C_{2^{\ell-1}}$ is the circuit from Lemma 6.3.20, and $N_z(2^{\ell-1})$ is as in (6.3.12). Note by (6.3.12) 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 (6.3.11) and (6.3.12), 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. □

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 $|\tilde{\theta}_j(L)\rangle$, which is the final state needed for the basis in Theorem 6.3.12.

6.3.22. Lemma. *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.

$$\begin{aligned} |\bar{\theta}_j(2)\rangle &= \frac{1}{n} \sum_{i\in\{0,1\}^{\log n}} \left(|e_i^0\rangle + |e_j^{1i}\rangle - |e_i^{2j}\rangle\right) \\ &= \frac{1}{n} \sum_{i\in\{0,1\}^{\log n}} |(0,\bar{0}),i\rangle + \frac{1}{n} \sum_{i\in\{0,1\}^{\log n}} |(1,i),j\rangle + \frac{1}{n} \sum_{i\in\{0,1\}^{\log n}} |(2,j),i\rangle. \end{aligned} \tag{6.3.13}$$

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

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

for any $j \in \{0,1\}^{\log n}$. From Lemma 6.3.15 (see also (4.3.7)) and the definition of $|p_{ij}\rangle$ in Theorem 6.3.12, we have:

$$\begin{aligned} n|\bar{\theta}_j(2^\ell)\rangle &= \sum_{i\in\{0,1\}^{\log n}} |p_{ij}(2^\ell)\rangle \\ &= |0,\bar{0}\rangle \sum_{i\in\{0,1\}^{\log n}} |\bar{\theta}_i(2^{\ell-1})\rangle + \sum_{i\in\{0,1\}^{\log n}} |1,i\rangle|\bar{\theta}_j(2^{\ell-1})\rangle + |2,j\rangle \sum_{i\in\{0,1\}^{\log n}} |\bar{\theta}_i(2^{\ell-1})\rangle \\ &= |0,\bar{0}\rangle \otimes \sqrt{N_{\bar{0}}(2^{\ell-1})}C_{2^{\ell-1}}|\bar{0}\rangle \\ &\quad + \sqrt{n}|1\rangle H^{\otimes \log n}|\bar{0}\rangle \otimes \left\||\bar{\theta}_j(2^{\ell-1})\rangle\right\| C'_{2^{\ell-1}}|j\rangle \\ &\quad + |2,j\rangle \otimes \sqrt{N_{\bar{0}}(2^{\ell-1})}C_{2^{\ell-1}}|\bar{0}\rangle \end{aligned} \tag{6.3.14}$$

where $C_{2^{\ell-1}}$ is the circuit from Lemma 6.3.20, and $N_{\bar{0}}(2^{\ell-1})$ is as in (6.3.11).

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 2.2.1, 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 (6.3.11)), it suffices to obtain a closed-form expression for $\left\||\bar{\theta}_j(2^{\ell-1})\rangle\right\|$. For $\ell > 0$, define $F_j(2^\ell) \coloneqq \left\||\bar{\theta}_j(2^\ell)\rangle\right\|^2$. We will prove by induction that

$$F_j(2^\ell) = \frac{1}{n^\ell}\cdot\frac{2(n+2)^\ell + n - 1}{n+1}. \tag{6.3.15}$$

For the base step, let $\ell = 1$. By (6.3.13), we have $F_j(2) = 3/n$, which is equal to the right-hand side of (6.3.15) for the setting $\ell = 1$. Next, we show the induction step.

$$\begin{aligned}
F_j(2^\ell) &= \frac{1}{n^2}\Big(2N_{\bar{0}}(2^{\ell-1}) + nF_j(2^{\ell-1})\Big) && \text{by (6.3.14)}\\
&= \frac{2}{n}\left(1+\frac{2}{n}\right)^{\ell-1} + \frac{1}{n}F_j(2^{\ell-1}) && \text{by (6.3.11)}\\
&= \frac{2}{n^\ell}(n+2)^{\ell-1} + \frac{1}{n^\ell}\cdot\frac{2(n+2)^{\ell-1}+n-1}{n+1} && \text{by i.h.}\\
&= \frac{1}{n^\ell}\cdot\frac{2(n+1)(n+2)^{\ell-1} + 2(n+2)^{\ell-1} + n - 1}{n+1}\\
&= \frac{1}{n^\ell}\cdot\frac{2(n+2)(n+2)^{\ell-1} + n - 1}{n+1} = \frac{1}{n^\ell}\cdot\frac{2(n+2)^\ell + n - 1}{n+1}.
\end{aligned}$$

□

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

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

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 6.3.12 can be efficiently generated.

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

6.3.4 Complexity of the subroutine

We conclude by combining the switching network and corresponding basis generation we have given in the previous sections with Theorem 4.3.19 to prove Theorem 6.2.1, which states the existence of an algorithm for $\textsc{Dist}_L$.

Proof of Theorem 6.2.1:
From Lemma 6.3.3, 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 4.3.19 to $\mathcal{N}_L(s,t)$. By Lemma 6.3.10, 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 6.3.11, $|E(\mathcal{N}_L(s,t))| \leq (2n+1)^{\log L} n$. By Lemma 6.3.24 and Lemma 4.3.17, 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 4.3.19, 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

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

and space

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

□

Chapter 7

A quantum algorithm for the gauge problem

We study the *gauge problem* on an undirected graph whose edges are labeled by $k \times k$ unitaries U_{uv} (with $U_{vu} = U_{uv}^\dagger$) under a *flatness* promise ensuring that the ordered product of edge labels along any path depends only on its endpoints, thereby defining a well-defined transport unitary $U_s(t)$. Given boundary states $|\psi_s\rangle, |\psi_t\rangle \in \mathbb{C}^k$ and oracle access that reveals neighbors while coherently applying the corresponding edge unitary, the task is to decide (under promise) whether $U_s(t)|\psi_s\rangle = |\psi_t\rangle$ or $\langle\psi_t|U_s(t)|\psi_s\rangle = 0$, generalizing undirected *st*-connectivity to settings with nontrivial internal quantum degrees of freedom. Using the *subspace graph* framework, we construct a bounded-error quantum algorithm for deciding a global unitary transport constraint from purely local access to the labeled graph with time complexity $\widetilde{O}(n^{3/2})$ and space complexity $O(\log n + \log k)$, requiring only logarithmic workspace in the graph size and local dimension.

This chapter is based on the paper "A quantum algorithm for the gauge problem" by Stacey Jeffery, Tobias J. Osborne, and Galina Pass which is currently in preparation.

7.1 Introduction

A *unitary labeled graph*, also referred to as a *connection graph*, is a graph in which each edge is equipped with a unitary operator acting on a fixed internal Hilbert space. Such graphs naturally encode how quantum states transform when transported between adjacent vertices. The original idea of decorating the edges of a graph with unitaries is due to Wilson [Wil74]. This led to creation of lattice gauge theory [Cre83], a framework that enables quantitative predictions in fundamental physics.

Since its introduction, the idea of associating unitary operators with the edges of a graph has been further developed in a variety of settings. Such unitary labeled graphs appear in work across physics, mathematics, and computer science as a way of augmenting classical graphs with additional local structure; see, e.g., [RKMC25] for a representative recent example and further references. From a modeling perspective, unitary labeled graphs can be viewed as a natural extension of classical graphs, in which edges carry additional algebraic structure. Related structures have also appeared in the quantum complexity literature, notably in [BCO17].

In this chapter, we study a promise problem on a d-regular undirected graph in which each edge $\{u, v\}$ is equipped with a unitary $U_{uv} \in \mathbb{C}^{k\times k}$ satisfying $U_{vu} = U_{uv}^\dagger$. Regularity is assumed only for notational and expository convenience, and the results generalize naturally to non-regular graphs. Under a *flatness* promise, meaning that the ordered product of edge unitaries along any path depends only on its endpoints, any two vertices in the same connected component $s, t \in V(G)$ define a unitary $U_s(t)$. Given boundary states $|\psi_s\rangle, |\psi_t\rangle \in \mathbb{C}^k$, the task is to decide whether $U_s(t)|\psi_s\rangle = |\psi_t\rangle$ or whether these two states are orthogonal. We refer to this decision problem as the *gauge problem* (see Problem 7.2.1 for a formal definition).

This problem can be viewed as a generalization of the well-studied undirected *st*-connectivity problem, in which the task is to decide whether there exists a path between two vertices in a graph. When the internal space is one-dimensional and all edge labels are trivial, the problem reduces to standard *st*-connectivity in an undirected graph. If we instead consider directed graphs, the problem reduces to directed *st*-connectivity. In the general setting considered here, the flatness promise guarantees that any path from s to t yields the same global transformation. The algorithmic challenge is therefore to access the action of the induced map $U_s(t)$ on $|\psi_s\rangle$ using only local oracle access to the graph and the edge unitaries.

Optimal quantum algorithms achieving simultaneously optimal time and space bounds have been found for undirected *st*-connectivity, both in the adjacency array and adjacency matrix models [BR12, AJPW23]. In both models, these algorithms use $O(\log n)$ space; the running time is $\widetilde{O}(n)$ in the adjacency array model and $\widetilde{O}(n^{3/2})$ in the adjacency matrix model, as discussed in Chapter 3. In

the directed setting, a quantum time-space tradeoff has been found which provides an up to quadratic improvement over the best classical algorithms in certain space regimes and trades classical space for quantum time [JP25b], as discussed in Chapter 5. These works illustrate how quantum resources can be leveraged to improve upon classical connectivity algorithms under different models of time and space constraints.

Our contribution. We show that the gauge problem can be solved by a bounded-error quantum algorithm running in $\widetilde{O}(n^{3/2})$ time and using $O(\log n + \log k)$ space on graphs with n vertices, where k is the dimension of the edge unitaries. This shows that adding a k-dimensional internal degree of freedom under a flatness promise still admits a polynomial-time quantum algorithm using logarithmic space in the graph size and local dimension.

Throughout this chapter, when we refer to the running time of an algorithm, we count the following resources: (1) elementary quantum gates, i.e., unitary operations acting on at most a constant number of qubits; (2) queries to the graph oracle O_G, which on input $(u, i, |\psi\rangle)$ outputs the i-th neighbor v of u together with the application of the associated edge unitary U_{uv} to $|\psi\rangle$; and (3) the state-preparation unitaries U_s and U_t that prepare the boundary states $|\psi_s\rangle$ and $|\psi_t\rangle$ from $|0\rangle$.

Our techniques. We design our quantum algorithm by equipping the input graph with a subspace graph structure introduced in Chapter 4. The gauge problem is well suited to the subspace graph framework because it naturally combines a graph structure with additional higher-dimensional structure attached to each edge. Our construction is inspired by the approach of [Jef22] that is also described in Section 4.3.5, but differs in that we work with a different oracle model and extend the framework from line graphs to general graphs.

Importantly, our oracle access model is closely analogous to the quantum adjacency array (or quantum walk) model: given a vertex and an index register in superposition, the oracle coherently returns the corresponding neighbor together with the action of the associated edge unitary on the internal state. This places our setting in direct analogy with the model used by optimal quantum algorithms for undirected *st*-connectivity discussed in Chapter 3, which achieve $\widetilde{O}(n)$ time. A natural open question is whether those techniques can be extended to the richer setting of unitary labeled graphs under the flatness promise, potentially improving our $\widetilde{O}(n^{3/2})$ running time to $\widetilde{O}(n)$. We leave this as a direction for future work.

Our subspace graph distinguishes between the YES case, in which there exists an *st*-path and $U_s(t)|\psi_s\rangle = |\psi_t\rangle$, and the NO case, in which either no *st*-path exists or there exists an *st*-path but $(U_s(t)|\psi_s\rangle)^\dagger|\psi_t\rangle = 0$. We analyze the algorithm by bounding the witness complexity of the construction and the cost of implementing

the associated reflections.

Our framework also suggests several natural extensions. First, instead of distinguishing perfect equality from orthogonality, one could consider a promise version in which $U_s(t)|\psi_s\rangle$ is either close to $|\psi_t\rangle$ or far from it. Because our algorithm ultimately estimates an overlap between two states, we expect that adapting the analysis to this approximate setting would be relatively straightforward, at the cost of a precision-dependent overhead. Second, it would be interesting to extend the model beyond unitary edge labels. Allowing more general linear operators, or even quantum channels, would substantially broaden the applicability of the framework, but would require new ideas to handle the loss of reversibility that underlies the current construction. We leave these directions for future work.

Organization The remainder of this chapter is organized as follows. In Section 7.2 we show our quantum algorithm for the gauge problem, starting with formally defining the gauge problem and specifying the oracle access model. In Section 7.2.1 we construct a subspace graph corresponding to a given gauge instance. Section 7.2.2 analyzes the witness complexity of this construction in both the positive and negative cases. In Section 7.2.3 we show how to implement the required reflections efficiently. Finally, in Section 7.2.4 we combine these ingredients to obtain our main quantum algorithm and prove its time and space complexity guarantees.

7.2 A quantum algorithm for the gauge problem

We start with formally stating the problem. Let $G = (V, E)$ be an undirected d-regular graph. Throughout this chapter, we fix the notations n and d to always denote the number of vertices of the graph and the degree of the vertices correspondingly. For each vertex $u \in V$, fix a bijection

$$f_u : \{0, \ldots, d-1\} \to N(u),$$

where $N(u)$ denotes the set of neighbors of u. We say that $f_u(i)$ is the *i-th neighbor of u*.

For every edge $\{u, v\} \in E$, there exist unique indices $i_{u,v}, i_{v,u} \in \{0, \ldots, d-1\}$ such that

$$f_u(i_{u,v}) = v \qquad \text{and} \qquad f_v(i_{v,u}) = u.$$

7.2.1. Problem. *Let $G = (V, E)$ be an undirected d-regular graph (Definition 2.4.1) with two distinguished vertices $s, t \in V$. For each edge $\{u, v\} \in E$, let there be an*

associated unitary operator $U_{uv} \in \mathbb{C}^{k\times k}$ *satisfying* $U_{vu} = U_{uv}^\dagger$. *For any* $u, v \in V$ *and any two paths*

$$u = u_0, u_1, \ldots, u_L = v \quad \textit{and} \quad u = u'_0, u'_1, \ldots, u'_{L'} = v$$

assume that

$$U_{u_{L-1}u_L} \cdots U_{u_0u_1} = U_{u'_{L'-1}u'_{L'}} \cdots U_{u'_0u'_1} =: U_u(v).$$

Let $|\psi_s\rangle, |\psi_t\rangle \in \mathbb{C}^k$ *be states associated with* s *and* t, *respectively. Given access to* G *via an oracle* O_G *acting as*

$$|u\rangle|i\rangle|\psi\rangle \mapsto |v\rangle|j\rangle\, U_{uv}|\psi\rangle,$$

where $v := f_u(i)$, *and* $f_v(j) = u$, *and acting as the identity on all other basis states, and the unitaries* U_s *and* U_t *that map* $|0\rangle$ *to* $|\psi_s\rangle$ *and* $|\psi_t\rangle$ *correspondingly, the task is to decide whether*

$$U_s(t)|\psi_s\rangle = |\psi_t\rangle \ \textit{or}\ (U_s(t)|\psi_s\rangle)^\dagger|\psi_t\rangle = 0.$$

In our cost model, each call to the oracle O_G and to the unitaries U_s and U_t is counted as unit time.

7.2.1 Subspace graph construction

We equip the input graph G with the subspace graph structure of Definition 4.3.1. The boundary is $B = \{s, t\}$, and the associated spaces are defined as follows.

Edge and boundary spaces For each edge $\{u, v\} \in E$, let $i_{u,v}, i_{v,u} \in \{0, \ldots, d-1\}$ be the unique indices such that $f_u(i_{u,v}) = v$ and $f_v(i_{v,u}) = u$, and define

$$\Xi_{uv} = \operatorname{span}\{|u, i_{u,v}\rangle|z\rangle, |v, i_{v,u}\rangle|z\rangle : z \in [k]\}.$$

For the boundary vertices,

$$\Xi_s = \operatorname{span}\{|s\rangle|d+1\rangle|0\rangle, |s\rangle|d\rangle|\psi_s\rangle\}, \quad \Xi_t = \operatorname{span}\{|t\rangle|d+1\rangle|0\rangle, |t\rangle|d\rangle|\psi_t\rangle\}.$$

Edge and boundary subspaces For each edge $\{u, v\} \in E$, define

$$\Xi_{uv}^{\mathcal{A}} = \operatorname{span}\{|u, i_{u,v}\rangle|z\rangle + |v, i_{v,u}\rangle U_{uv}|z\rangle : z \in [k]\}, \qquad \Xi_{uv}^{\mathcal{B}} = \{0\}.$$

For boundary vertices, define

$$\Xi_s^{\mathcal{A}} = \operatorname{span}\{|s\rangle|d+1\rangle|0\rangle + |s\rangle|d\rangle|\psi_s\rangle\}, \quad \Xi_s^{\mathcal{B}} = \{0\},$$
$$\Xi_t^{\mathcal{A}} = \operatorname{span}\{|t\rangle|d+1\rangle|0\rangle + |t\rangle|d\rangle|\psi_t\rangle\}, \quad \Xi_t^{\mathcal{B}} = \{0\}.$$

Vertex spaces and boundary vertex space For each vertex $u \in V \setminus \{s,t\}$, let

$$\mathcal{V}_u = \mathrm{span}\left\{\sum_{i=0}^{d-1} |u,i\rangle|z\rangle : z \in [k]\right\}.$$

For a boundary vertex $b \in \{s,t\}$ we define

$$\mathcal{V}_b = \mathrm{span}\left\{|b\rangle|d\rangle|\psi_b\rangle + \sum_{i=0}^{d-1} |b,i\rangle|\psi_b\rangle\right\} \oplus \mathrm{span}\left\{\sum_{i=0}^{d-1} |b,i\rangle|\psi\rangle : \langle\psi_b|\psi\rangle = 0\right\}.$$

We set the boundary subspace to be

$$\mathcal{V}_B = \{0\}.$$

Then, as in Definition 4.3.1, the global subspaces are

$$\mathcal{A}_G = \bigoplus_{e\in E\cup B} \Xi_e^{\mathcal{A}}, \qquad \mathcal{B}_G = \left(\bigoplus_{u\in V} \mathcal{V}_u\right) \oplus \mathcal{V}_B \oplus \bigoplus_{e\in E\cup B} \Xi_e^{\mathcal{B}} = \bigoplus_{u\in V} \mathcal{V}_u.$$

Note that the defined subspace graph has a weak canonical st-boundary (see Definition 4.3.6). Finally, we set the starting state to be the same as in Chapter 4:

$$|\psi_0\rangle = \frac{1}{\sqrt{2}}(|s\rangle|d+1\rangle|0\rangle - |t\rangle|d+1\rangle|0\rangle).$$

7.2.2 Witness analysis

In this section, we analyze the witness complexity of the subspace graph defined above and show that it indeed decides Problem 7.2.1. We start with studying the positive case.

7.2.2. Lemma. *If there exists an st-path in G and $U_s(t)|\psi_s\rangle = |\psi_t\rangle$ then there exists a positive witness (see Definition 4.3.10) of size $O(n)$.*

Proof:
Let $u_0 = s, u_1, \ldots, u_L = t$ be an st-path such that, for every $\ell \le L-1$, $f_{u_\ell}(i_\ell) = u_{\ell+1}$ and, for every $\ell \ge 1$, $f_{u_\ell}(j_\ell) = u_{\ell-1}$. We show that

$$\begin{aligned} |w_+\rangle = &-|s\rangle|d+1\rangle|0\rangle + |s\rangle|d\rangle|\psi_s\rangle - |u_0,i_0\rangle|\psi_s\rangle \\ &+ \sum_{\ell=1}^{L-1}(|u_\ell,j_\ell\rangle - |u_\ell,i_\ell\rangle)U_s(u_\ell)|\psi_s\rangle \\ &+ |u_L,j_L\rangle|\psi_t\rangle - |t\rangle|d\rangle|\psi_t\rangle + |t\rangle|d+1\rangle|0\rangle. \end{aligned} \tag{7.2.1}$$

is a positive witness. That is, we show that $|w_+\rangle \in (\mathcal{A}+\mathcal{B})^\perp$.

In some parts of the proof, it will be convenient to rewrite $|w_+\rangle$ as follows.

$$\begin{aligned}|w_+\rangle &= -|s\rangle|d+1\rangle|0\rangle + |s\rangle|d\rangle|\psi_s\rangle \\ &\quad - \sum_{\ell=0}^{L-1}(|u_\ell, i_\ell\rangle U_s(u_\ell)|\psi_s\rangle - |u_{\ell+1}, j_{\ell+1}\rangle U_s(u_{\ell+1})|\psi_s\rangle) \\ &\quad - |t\rangle|d\rangle|\psi_t\rangle + |t\rangle|d+1\rangle|0\rangle.\end{aligned} \tag{7.2.2}$$

This is true because $|\psi_t\rangle = U_s(t)|\psi_s\rangle = U_s(u_L)|\psi_S\rangle$.

Orthogonality to $\mathcal{A}$. If an edge is not on the chosen path, then $|w_+\rangle$ is clearly orthogonal to the corresponding subspace because of no overlap. For any edge $\{u_\ell, u_{\ell+1}\}$ with $0 \le \ell \le L-1$,

$$\Xi^{\mathcal{A}}_{u_\ell u_{\ell+1}} = \operatorname{span}\Big\{|u_\ell, i_\ell\rangle|z\rangle + |u_{\ell+1}, j_{\ell+1}\rangle\, U_{u_\ell u_{\ell+1}}|z\rangle : z \in [k]\Big\}.$$

Then for all z and $0 \le \ell \le L-1$, using (7.2.2),

$$\begin{aligned}&\Big(|u_\ell, i_\ell\rangle|z\rangle + |u_{\ell+1}, j_{\ell+1}\rangle\, U_{u_\ell u_{\ell+1}}|z\rangle\Big)^\dagger |w_+\rangle \\ &= -\langle u_\ell, i_\ell|u_\ell, i_\ell\rangle\langle z|U_s(u_\ell)|\psi_s\rangle + \langle u_{\ell+1}, j_{\ell+1}|u_{\ell+1}, j_{\ell+1}\rangle\langle z|U^\dagger_{u_\ell u_{\ell+1}} U_s(u_{\ell+1})|\psi_s\rangle \\ &= -\langle z|U_s(u_\ell)|\psi_s\rangle + \langle z|U_s(u_\ell)|\psi_s\rangle \\ &= 0,\end{aligned}$$

where we used that

$$U_s(u_{\ell+1}) = U_{u_\ell u_{\ell+1}} U_s(u_\ell).$$

Finally, we observe that $|w_+\rangle$ is orthogonal to $|s\rangle|d+1\rangle|0\rangle + |s\rangle|d\rangle|\psi_s\rangle$ and $|t\rangle|d+1\rangle|0\rangle + |t\rangle|d\rangle|\psi_t\rangle$.

Orthogonality to $\mathcal{B}$. If u is not on the path, $|w_+\rangle$ has no overlap with its space. For an internal vertex u_ℓ with $1 \le \ell \le L-1$,

$$\mathcal{V}_{u_\ell} = \operatorname{span}\Big\{\sum_{i=0}^{d-1} |u_\ell, i\rangle|z\rangle : z \in [k]\Big\}.$$

Only two terms overlap $|w_+\rangle$: where $i = j_\ell$ and $i = i_\ell$. Hence, for all z, using (7.2.1),

$$\left(\sum_{i=0}^{d-1} |u_\ell, i\rangle|z\rangle\right)^\dagger |w_+\rangle = -\langle u_\ell, i_\ell|u_\ell, i_\ell\rangle\langle z|U_s(u_\ell)|\psi_s\rangle + \langle u_\ell, j_\ell|u_\ell, j_\ell\rangle\langle z|U_s(u_\ell)|\psi_s\rangle = 0.$$

For a boundary vertex $b \in \{s,t\}$,

$$\mathcal{V}_b = \operatorname{span}\left\{ |b\rangle|d\rangle|\psi_b\rangle + \sum_{i=0}^{d-1} |b,i\rangle|\psi_b\rangle \right\} \oplus \operatorname{span}\left\{ \sum_{i=0}^{d-1} |b,i\rangle|\psi\rangle : \langle\psi_b|\psi\rangle = 0 \right\}.$$

Since $|w_+\rangle$ only contains $|\psi_b\rangle$ and not states orthogonal to it, it is orthogonal to the second summand. For the first summand,

$$\left(|s\rangle|d\rangle|\psi_s\rangle + \sum_i |s,i\rangle|\psi_s\rangle\right)^\dagger |w_+\rangle = \langle s|s\rangle\langle d|d\rangle\langle\psi_s|\psi_s\rangle - \langle s,i_0|s,i_0\rangle\langle\psi_s|\psi_s\rangle = 0,$$
$$\left(|t\rangle|d\rangle|\psi_t\rangle + \sum_i |t,i\rangle|\psi_t\rangle\right)^\dagger |w_+\rangle = -\langle t|t\rangle\langle d|d\rangle\langle\psi_t|\psi_t\rangle + \langle t,j_L|t,j_L\rangle\langle\psi_t|\psi_t\rangle = 0.$$

Witness size. All the terms are orthonormal, and the number of nonzero terms is $2L+4 = O(L)$. Hence, $\||w_+\rangle\|^2 = O(L) \le O(n)$. □

7.2.3. Remark. The positive witness constructed above is supported on a single st-path, which leads to a witness size proportional to the path length. More generally, one could consider witnesses that coherently combine contributions from multiple st-paths. In particular, choosing amplitudes according to an optimal unit flow between s and t yields a witness whose size is governed by the effective resistance $\mathcal{R}_{s,t}(G)$, which can be strictly smaller than the length of any individual path. We present the path-based construction purely for simplicity, the more general formulation follows by the same reasoning.

Next, we consider two possible negative cases: first, when there is no st-path in G, and second, when such a path exists but the states $U_s(t)|\psi_s\rangle$ and $|\psi_t\rangle$ are orthogonal. We begin by analyzing the first case.

7.2.4. Lemma. *If there is no st-path in G, then there exists a negative witness (see Definition 4.3.11) of size $O(|E|)$.*

Proof:
We will show that, under the assumptions of the lemma, there exist vectors $|w_{\mathcal{A}}\rangle \in \mathcal{A}$ and $|w_{\mathcal{B}}\rangle \in \mathcal{B}$ such that $|\psi_0\rangle = |w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle$, so that $|w_{\mathcal{A}}\rangle$ serves as a negative witness. For $b \in \{s,t\}$, let $V_b \subseteq V$ denote the set of vertices of G connected to b (including b itself), and let $E_b \subseteq E$ denote the set of edges of G connected to b. Define

$$
\begin{aligned}
|w_{\mathcal{A}}\rangle = & \\
& \frac{1}{\sqrt{2}}\left(|s\rangle|d+1\rangle|0\rangle + |s\rangle|d\rangle|\psi_s\rangle + \sum_{\{u,v\}\in E_s}(|u,i_{u,v}\rangle U_s(u)|\psi_s\rangle + |v,i_{v,u}\rangle U_s(v)|\psi_s\rangle)\right) \\
& - \frac{1}{\sqrt{2}}\left(\sum_{\{u,v\}\in E_t}(|u,i_{u,v}\rangle U_t(u)|\psi_t\rangle + |v,i_{v,u}\rangle U_t(v)|\psi_t\rangle) + |t\rangle|d\rangle|\psi_t\rangle + |t\rangle|d+1\rangle|0\rangle\right) \\
|w_{\mathcal{B}}\rangle = & -\frac{1}{\sqrt{2}}|s\rangle|d\rangle|\psi_s\rangle - \frac{1}{\sqrt{2}}\sum_{u\in V_s}\left(\sum_{i=0}^{d-1}|u,i\rangle U_s(u)|\psi_s\rangle\right) \\
& + \frac{1}{\sqrt{2}}\sum_{u\in V_t}\left(\sum_{i=0}^{d-1}|u,i\rangle U_t(u)|\psi_t\rangle\right) + \frac{1}{\sqrt{2}}|t\rangle|d\rangle|\psi_t\rangle.
\end{aligned}
$$

Clearly, $|w_{\mathcal{A}}\rangle \in \mathcal{A}$ and $|w_{\mathcal{B}}\rangle \in \mathcal{B}$. For $b \in \{s,t\}$, and any vertex $u \in V_b$, and $i \in \{0,\ldots,d-1\}$, each term $\frac{1}{\sqrt{2}}|u,i\rangle\, U_b(u)|\psi_b\rangle$ appears in $|w_{\mathcal{A}}\rangle$ and in $|w_{\mathcal{B}}\rangle$ exactly once, with opposite signs. An analogous cancellation holds for the boundary contributions $|s\rangle|d\rangle|\psi_s\rangle$ and $|t\rangle|d\rangle|\psi_t\rangle$. Consequently, all such terms cancel in the sum $|w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle$, and therefore

$$|w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle = \frac{1}{\sqrt{2}}(|s\rangle|d+1\rangle|0\rangle - |t\rangle|d+1\rangle|0\rangle) = |\psi_0\rangle.$$

Finally, each edge $\{u,v\} \in E_s \cup E_t$ contributes exactly two orthogonal terms to $|w_{\mathcal{A}}\rangle$, each of norm $1/\sqrt{2}$. Hence,

$$\||w_{\mathcal{A}}\rangle\|^2 = \frac{1}{2}\cdot(2|E_s \cup E_t| + 4) \leq |E| + 2.$$

Therefore, $\||w_{\mathcal{A}}\rangle\|^2 = O(|E|)$, so the negative witness has size $O(|E|)$. □

Finally, we turn to the second negative case.

7.2.5. Lemma. *If there exists an st-path in G and $(U_s(t)|\psi_s\rangle)^\dagger|\psi_t\rangle = 0$ then there exists a negative witness of size $O(|E|)$.*

Proof:
As in the previous lemma, we will show that, under the assumptions of the lemma, there exist vectors $|w_{\mathcal{A}}\rangle \in \mathcal{A}$ and $|w_{\mathcal{B}}\rangle \in \mathcal{B}$ such that $|\psi_0\rangle = |w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle$, so that $|w_{\mathcal{A}}\rangle$ serves as a negative witness. For $b \in \{s,t\}$, let $V_b \subseteq V$ denote the set of vertices of G connected to b (including b itself), and let $E_b \subseteq E$ denote the set of edges of G connected to b. In this case, $E_s = E_t$ and $V_s = V_t$. As before, define:

$$
\begin{aligned}
|w_{\mathcal{A}}\rangle = & \\
& \frac{1}{\sqrt{2}}\left(|s\rangle|d+1\rangle|0\rangle + |s\rangle|d\rangle|\psi_s\rangle + \sum_{\{u,v\}\in E_s}(|u,i_{u,v}\rangle U_s(u)|\psi_s\rangle + |v,i_{v,u}\rangle U_s(v)|\psi_s\rangle)\right) \\
& - \frac{1}{\sqrt{2}}\left(\sum_{\{u,v\}\in E_t}(|u,i_{u,v}\rangle U_t(u)|\psi_t\rangle + |v,i_{v,u}\rangle U_t(v)|\psi_t\rangle) + |t\rangle|d\rangle|\psi_t\rangle + |t\rangle|d+1\rangle|0\rangle\right) \\
|w_{\mathcal{B}}\rangle = & -\frac{1}{\sqrt{2}}|s\rangle|d\rangle|\psi_s\rangle - \frac{1}{\sqrt{2}}\sum_{u\in V_s}\left(\sum_{i=0}^{d-1}|u,i\rangle U_s(u)|\psi_s\rangle\right) \\
& + \frac{1}{\sqrt{2}}\sum_{u\in V_t}\left(\sum_{i=0}^{d-1}|u,i\rangle U_t(u)|\psi_t\rangle\right) + \frac{1}{\sqrt{2}}|t\rangle|d\rangle|\psi_t\rangle.
\end{aligned}
$$

Unlike in the previous lemma, we now have $t \in V_s = V_t$ and $s \in V_t = V_s$. Since we assumed $(U_s(t)|\psi_s\rangle)^\dagger|\psi_t\rangle = 0$, the state $\sum_{i=0}^{d-1}|t,i\rangle U_s(t)|\psi_s\rangle$ is in $\mathcal{V}_t$, and the state $\sum_{i=0}^{d-1}|s,i\rangle U_t(s)|\psi_t\rangle$ is in $\mathcal{V}_s$ since

$$0 = (U_s(t)|\psi_s\rangle)^\dagger|\psi_t\rangle = \langle\psi_s|U_s(t)^\dagger|\psi_t\rangle = \langle\psi_s|U_t(s)|\psi_t\rangle.$$

Hence, $|w_{\mathcal{A}}\rangle \in \mathcal{A}$ and $|w_{\mathcal{B}}\rangle \in \mathcal{B}$. The same as in the previous lemma, for $b \in \{s,t\}$, and any vertex $u \in V_b$, and any $i \in \{0,\ldots,d-1\}$, the term $\frac{1}{\sqrt{2}}|u,i\rangle\, U_b(u)|\psi_b\rangle$ appears in $|w_{\mathcal{A}}\rangle$ and in $|w_{\mathcal{B}}\rangle$ exactly once, with opposite signs. Same as before, an analogous cancellation holds for the boundary contributions $|s\rangle|d\rangle|\psi_s\rangle$ and $|t\rangle|d\rangle|\psi_t\rangle$. Consequently, all such terms cancel in the sum $|w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle$, and therefore

$$|w_{\mathcal{A}}\rangle + |w_{\mathcal{B}}\rangle = \frac{1}{\sqrt{2}}(|s\rangle|d+1\rangle|0\rangle - |t\rangle|d+1\rangle|0\rangle) = |\psi_0\rangle.$$

Finally, each edge $\{u,v\} \in E_s \cup E_t = E_s = E_t$ contributes at most four orthogonal terms to $|w_{\mathcal{A}}\rangle$, each of norm $1/\sqrt{2}$. Hence,

$$\||w_{\mathcal{A}}\rangle\|^2 \leq \frac{1}{2}\cdot(4|E_s|+4) \leq 2|E|+2.$$

Therefore, $\||w_{\mathcal{A}}\rangle\|^2 = O(|E|)$, so the negative witness has size $O(|E|)$. □

7.2.3 Implementation of reflections

Here we show that the unitary

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

associated with the subspace graph can be implemented in $\widetilde{O}(1)$ time. We do this by arguing that the reflections through $\mathcal{A}$ and $\mathcal{B}$ can each be implemented in $\widetilde{O}(1)$ time. We begin with the reflection through $\mathcal{A}$.

7.2.6. Lemma. *The reflection $2\Pi_{\mathcal{A}} - I$ can be implemented in $\widetilde{O}(1)$ time.*

Proof:
Recall that the global Hilbert space decomposes as

$$H_G = \bigoplus_{\{u,v\}\in E} \Xi_{uv} \oplus \bigoplus_{b\in\{s,t\}} \Xi_b.$$

and

$$\mathcal{A} = \bigoplus_{\{u,v\}\in E} \Xi^{\mathcal{A}}_{uv} \oplus \bigoplus_{b\in\{s,t\}} \Xi^{\mathcal{A}}_b,$$

where

$$\Xi^{\mathcal{A}}_{uv} = \mathrm{span}\{|u, i_{u,v}\rangle|z\rangle + |v, i_{v,u}\rangle U_{uv}|z\rangle : z \in [k]\} \tag{7.2.3}$$

$$\Xi^{\mathcal{A}}_b = |b\rangle|d+1\rangle|0\rangle + |b\rangle|d\rangle|\psi_b\rangle. \tag{7.2.4}$$

We note that it is enough to implement reflections through $\bigoplus_{\{u,v\}\in E} \Xi^{\mathcal{A}}_{uv}$ and $\bigoplus_{b\in\{s,t\}} \Xi^{\mathcal{A}}_b$. Once these are available, the reflection through $\mathcal{A}$ can be obtained by consecutively applying these two smaller reflections and applying $-I$. Indeed,

$$\begin{aligned}
&(2\Pi_{\bigoplus_{\{u,v\}\in E} \Xi^{\mathcal{A}}_{uv}} - I)(2\Pi_{\bigoplus_{b\in\{s,t\}} \Xi^{\mathcal{A}}_b} - I)(-I) \\
&= (-2\Pi_{\bigoplus_{\{u,v\}\in E} \Xi^{\mathcal{A}}_{uv}} - 2\Pi_{\bigoplus_{b\in\{s,t\}} \Xi^{\mathcal{A}}_b} + I)(-I) \\
&= 2(\Pi_{\bigoplus_{\{u,v\}\in E} \Xi^{\mathcal{A}}_{uv}} + \Pi_{\bigoplus_{b\in\{s,t\}} \Xi^{\mathcal{A}}_b}) - I \\
&= 2\Pi_{\mathcal{A}} - I.
\end{aligned}$$

Observe that on the subspace $\bigoplus_{\{u,v\}\in E} \Xi_{uv}$ the oracle O_G itself acts as a reflection through $\bigoplus_{\{u,v\}\in E} \Xi^{\mathcal{A}}_{uv}$. Indeed, for any $\{u,v\} \in E$ and $z \in [k]$, define

$$|\phi^+_{uv}(z)\rangle = |u, i_{u,v}\rangle|z\rangle + |v, i_{v,u}\rangle U_{uv}|z\rangle, \qquad |\phi^-_{uv}(z)\rangle = |u, i_{u,v}\rangle|z\rangle - |v, i_{v,u}\rangle U_{uv}|z\rangle.$$

A direct calculation shows

$$\begin{aligned} O_G|\phi^+_{uv}(z)\rangle &= |v, i_{v,u}\rangle U_{uv}|z\rangle + |u, i_{u,v}\rangle U_{vu}U_{uv}|z\rangle = |u, i_{u,v}\rangle|z\rangle + |v, i_{v,u}\rangle U_{uv}|z\rangle \\ &= |\phi^+_{uv}(z)\rangle, \\ O_G|\phi^-_{uv}(z)\rangle &= |v, i_{v,u}\rangle U_{uv}|z\rangle - |u, i_{u,v}\rangle U_{vu}U_{uv}|z\rangle = -\Big(|u, i_{u,v}\rangle|z\rangle - |v, i_{v,u}\rangle U_{uv}|z\rangle\Big) \\ &= -|\phi^-_{uv}(z)\rangle. \end{aligned}$$

Thus $\Xi^{\mathcal{A}}_{uv} = \mathrm{span}\{|\phi^+_{uv}(z)\rangle : z \in [k]\}$ is the $(+1)$-eigenspace of O_G on Ξ_{uv}, and its orthogonal complement in Ξ_{uv} is the (-1)-eigenspace. Indeed,

$$\begin{aligned} \Xi_{uv} &= \mathrm{span}\{|u, i_{u,v}\rangle|z\rangle, |v, i_{v,u}\rangle|z\rangle : z \in [k]\} \\ &= \mathrm{span}\{|u, i_{u,v}\rangle|z\rangle : z \in [k]\} \oplus \mathrm{span}\{|v, i_{v,u}\rangle|z\rangle : z \in [k]\} \\ &= \mathrm{span}\{|u, i_{u,v}\rangle|z\rangle : z \in [k]\} \oplus \mathrm{span}\{|v, i_{v,u}\rangle U_{uv}|z\rangle : z \in [k]\} \\ &= \mathrm{span}\{|u, i_{u,v}\rangle|z\rangle, |v, i_{v,u}\rangle U_{uv}|z\rangle : z \in [k]\} \\ &= \mathrm{span}\{|u, i_{u,v}\rangle|z\rangle + |v, i_{v,u}\rangle U_{uv}|z\rangle, |u, i_{u,v}\rangle|z\rangle - |v, i_{v,u}\rangle U_{uv}|z\rangle : z \in [k]\} \\ &= \Xi^{\mathcal{A}}_{uv} \oplus \mathrm{span}\{|u, i_{u,v}\rangle|z\rangle - |v, i_{v,u}\rangle U_{uv}|z\rangle : z \in [k]\}. \end{aligned}$$

Since the subspaces Ξ_{uv} are pairwise orthogonal, the global $(+1)$-eigenspace of O_G in $\bigoplus_{\{u,v\}\in E} \Xi_{uv}$ is precisely

$$\bigoplus_{\{u,v\}\in E} \Xi^{\mathcal{A}}_{uv},$$

and the (-1)-eigenspace is $\left(\bigoplus_{\{u,v\}\in E} \Xi^{\mathcal{A}}_{uv}\right)^{\perp}$ within $\bigoplus_{\{u,v\}\in E} \Xi_{uv}$.

To complete O_G to a reflection through $\bigoplus_{\{u,v\}\in E} \Xi^{\mathcal{A}}_{uv}$ on the entire space, we must flip the phase of all states in $\bigoplus_{b\in\{s,t\}} \Xi_b$. We will do this by first implementing a reflection through this subspace and then applying a global phase flip. Recall that

$$\begin{aligned} \bigoplus_{b\in\{s,t\}} \Xi_b = \mathrm{span}\{&|s\rangle|d+1\rangle|0\rangle, |s\rangle|d\rangle|\psi_s\rangle, \\ &|t\rangle|d+1\rangle|0\rangle, |t\rangle|d\rangle|\psi_t\rangle\}. \end{aligned}$$

To reflect through this span, it is sufficient to implement a reflection through

$$\mathrm{span}\{|s\rangle|d+1\rangle|0\rangle, |s\rangle|d\rangle|0\rangle, |t\rangle|d+1\rangle|0\rangle, |t\rangle|d\rangle|0\rangle\},$$

which is a controlled phase flip and requires only $\widetilde{O}(1)$ time, and, for each $b \in \{s, t\}$, to implement the mapping

$$|b\rangle|d\rangle|0\rangle \mapsto |b\rangle|d\rangle|\psi_b\rangle,$$

which uses a single controlled application of U_b, and therefore also takes $\widetilde{O}(1)$ time. Hence, the overall time complexity of reflecting through $\bigoplus_{\{u,v\}\in E} \Xi_{uv}^{\mathcal{A}}$ is $\widetilde{O}(1)$.

Next, we show how to implement a reflection through $\bigoplus_{b\in\{s,t\}} \Xi_b^{\mathcal{A}}$. Recall that

$$\bigoplus_{b\in\{s,t\}} \Xi_b^{\mathcal{A}} = \operatorname{span}\{|s\rangle|d+1\rangle|0\rangle + |s\rangle|d\rangle|\psi_s\rangle, \\ |t\rangle|d+1\rangle|0\rangle + |t\rangle|d\rangle|\psi_t\rangle\}.$$

In a manner analogous to the argument above, it is sufficient to implement a reflection through

$$\operatorname{span}\{|s\rangle|0\rangle|0\rangle, |t\rangle|0\rangle|0\rangle\},$$

which is a controlled phase flip and requires only $\widetilde{O}(1)$ time, and, for each $b \in \{s,t\}$, to implement the mapping

$$\begin{aligned} |b\rangle|0\rangle|0\rangle &\mapsto \frac{1}{\sqrt{2}}(|b\rangle|d+1\rangle|0\rangle + |b\rangle|d\rangle|0\rangle) \\ &\mapsto \frac{1}{\sqrt{2}}(|b\rangle|d+1\rangle|0\rangle + |b\rangle|d\rangle|\psi_b\rangle). \end{aligned}$$

which uses a single controlled application of U_b and $\widetilde{O}(1)$ other gates, and therefore also takes $\widetilde{O}(1)$ time. Hence, the overall time complexity of reflecting through $\bigoplus_{b\in\{s,t\}} \Xi_b^{\mathcal{A}}$ is $\widetilde{O}(1)$. Thus, reflection through $\mathcal{A}$ can be implemented in $\widetilde{O}(1)$ time. □

Next, we show how to implement the reflection through $\mathcal{B}$.

7.2.7. Lemma. *The reflection $2\Pi_{\mathcal{B}} - I$ can be implemented in $\widetilde{O}(1)$ time.*

Proof:
For reflection through $\mathcal{B}$, we show how to prepare a working basis of $\mathcal{B}$ (see Definition 2.7.5). By definition,

$$\mathcal{B} = \bigoplus_{u\in V} \mathcal{V}_u,$$

where for internal vertices $u \in V \setminus \{s,t\}$,

$$\mathcal{V}_u = \operatorname{span}\left\{\sum_{i=0}^{d-1} |u,i\rangle|z\rangle : z \in [k]\right\},$$

for boundary vertices $b \in \{s,t\}$ we have

$$\mathcal{V}_b = \operatorname{span}\left\{|b\rangle|d\rangle|\psi_b\rangle + \sum_{i=0}^{d-1} |b,i\rangle|\psi_b\rangle\right\} \oplus \operatorname{span}\left\{\sum_{i=0}^{d-1} |b,i\rangle|\psi\rangle : \langle\psi_b|\psi\rangle = 0\right\}.$$

We take the following set of vectors as an orthonormal basis of $\mathcal{B}$.

$$\left(\bigcup_{u\in V\setminus\{s,t\}}\left\{\frac{1}{\sqrt{d}}\sum_{i=0}^{d-1}|u,i\rangle|z\rangle : z\in[k]\right\}\right)$$
$$\cup\left\{\frac{1}{\sqrt{d+1}}\left(|s\rangle|d\rangle|\psi_s\rangle+\sum_{i=0}^{d-1}|s,i\rangle|\psi_s\rangle\right)\right\}\cup\left\{\frac{1}{\sqrt{d}}\sum_{i=0}^{d-1}|s,i\rangle|\psi_s^z\rangle : z\in[k]\setminus\{0\}\right\}$$
$$\cup\left\{\frac{1}{\sqrt{d+1}}\left(|t\rangle|d\rangle|\psi_t\rangle+\sum_{i=0}^{d-1}|t,i\rangle|\psi_t\rangle\right)\right\}\cup\left\{\frac{1}{\sqrt{d}}\sum_{i=0}^{d-1}|t,i\rangle|\psi_t^z\rangle : z\in[k]\setminus\{0\}\right\},$$

where $|\psi_b^z\rangle = U_b|z\rangle$ for $b\in\{s,t\}, z\in[k]\setminus\{0\}$, with U_b provided by the black-box access to the input described in Problem 7.2.1. It is a basis of $\mathrm{span}\{|\psi_b\rangle\}^\perp$ in $\mathbb{C}^k$ since U_b maps $|0\rangle$ to $|\psi_b\rangle$ and is a unitary.

For an internal vertex $u\in V\setminus\{s,t\}$, we index the basis states by $z\in[k]$. Then, the corresponding basis state preparation

$$|u\rangle|0\rangle|z\rangle\mapsto\frac{1}{\sqrt{d}}\sum_{i=0}^{d-1}|u,i\rangle|z\rangle$$

can be done in $\widetilde{O}(1)$ time. For $b\in\{s,t\}$, we index the basis states by $z\in[k]$. For $z=0$ the corresponding basis state preparation

$$|b\rangle|0\rangle|0\rangle\mapsto\frac{1}{\sqrt{d+1}}\left(|b\rangle|d\rangle|\psi_b\rangle+\sum_{i=0}^{d-1}|b,i\rangle|\psi_b\rangle\right)$$

can be done by preparing a uniform superposition and one application of U_b which requires $\widetilde{O}(1)$ time. For $z\in[k]\setminus\{0\}$, the corresponding basis state preparation

$$|b\rangle|0\rangle|z\rangle\mapsto\frac{1}{\sqrt{d}}\sum_{i=0}^{d-1}|b,i\rangle|\psi_b^z\rangle$$

can be done, again, by preparing a uniform superposition and one application of U_b which requires $\widetilde{O}(1)$ time. For every $u\in V, z\in[k]$, we showed how to map a state of the form $|u\rangle|0\rangle|z\rangle$ to the corresponding basis state. Hence, it remains to implement reflection through

$$\mathrm{span}\{|u\rangle|0\rangle|z\rangle : u\in V, z\in[k]\}.$$

This reflection is just a phase flip controlled on the state being zero in the second register concatenated with a global phase flip. This takes $\widetilde{O}(1)$ time. Hence, reflection through $\mathcal{B}$ takes $\widetilde{O}(1)$ time. □

We summarize the results of this section in the following corollary.

7.2.8. Corollary. *The unitary $U_{\mathcal{AB}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$ can be implemented in $\widetilde{O}(1)$ time.*

7.2.4 Main theorem

In this section, we state the main theorem of the chapter, which combines the results established above and shows the existence of an efficient quantum algorithm for the gauge problem.

7.2.9. THEOREM. *There exists a bounded-error quantum algorithm that solves the gauge (Problem 7.2.1) in time complexity*

$$\widetilde{O}(n^{3/2})$$

and space

$$O(\log n + \log k).$$

Proof:
We apply Theorem 4.3.13 to the subspace graph constructed above for the gauge instance. As observed above, this subspace graph has a weak canonical st-boundary. In the YES case of Problem 7.2.1, i.e. when there exists an st-path and $U_s(t)|\psi_s\rangle = |\psi_t\rangle$, Lemma 7.2.2 constructs a positive witness with squared norm $O(n)$. This implies $\hat{W}_+(G) = O(n)$ for our choice of $|\psi_0\rangle$.

In the NO case of Problem 7.2.1, we are under the promise that either there is no st-path, or there exists an st-path and $(U_s(t)|\psi_s\rangle)^\dagger|\psi_t\rangle = 0$. Lemma 7.2.4 handles the first possibility and Lemma 7.2.5 handles the second, and in either case they provide a negative witness of size $O(|E|)$. Since $|E| \leq \binom{n}{2} = O(n^2)$, we have $\hat{W}_-(G) = O(n^2)$.

By Corollary 7.2.8 the unitary

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

can be implemented in time $T = \widetilde{O}(1)$.

Hence, by Theorem 4.3.13 and Remark 4.3.15, there exists a bounded-error quantum algorithm deciding the problem in time

$$O\Big(T\sqrt{\hat{W}_-(G)\hat{W}_+(G)}\Big) = \widetilde{O}\big(n^{3/2}\big).$$

Here $\dim H_G$ is polynomial in n and k, so $\log\dim H_G = O(\log n + \log k)$, and $\log \hat{W}_-(G) = O(\log|E|) = O(\log n)$. Thus the total space is $O(\log n + \log k)$. This concludes the proof. □

Bibliography

[ABB+25] Jonathan Allcock, Jinge Bao, Aleksandrs Belovs, Troy Lee, and Miklos Santha. On the Quantum Time Complexity of Divide and Conquer. In *52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025)*, volume 334, pages 9:1–9:20, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 98

[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. 118

[AGJ20] Simon Apers, András Gilyén, and Stacey Jeffery. A unified framework of quantum walk search. In *Proceedings of the 38th Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 6:1–6:13, 2020. 15

[AHV95] Serge Abiteboul, Richard Hull, and Victor Vianu. *Foundations of Databases*. Addison-Wesley, 1995. 114

[AJPW23] Simon Apers, Stacey Jeffery, Galina Pass, and Michael Walter. (No) Quantum Space-Time Tradeoff for USTCON. In *31st Annual European Symposium on Algorithms (ESA 2023)*, volume 274, pages 10:1–10:17, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 19, 97, 115, 118, 152

[AKL+79] Romas Aleliunas, Richard M. Karp, Richard J. Lipton, Laszlo Lovasz, and Charles Rackoff. Random walks, universal traversal sequences,

and the complexity of maze problems. In *Proceedings of the 20th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 218–223, 1979. 2, 20

[ALSU06] Alfred V. Aho, Monica S. Lam, Ravi Sethi, and Jeffrey D. Ullman. *Compilers: Principles, Techniques, and Tools*. Pearson, 2nd edition, 2006. 114

[Amb10] Andris Ambainis. Quantum search with variable times. *Theory of Computing Systems*, 47(3):786–807, 2010. 8, 42

[Ape19] Simon Apers. Quantum walk sampling by growing seed sets. In *Proceedings of the 27th Annual European Symposium on Algorithms (ESA)*, volume 144, pages 9:1–9:12. Springer, 2019. 22, 24, 32, 35

[ATS03] Dorit Aharonov and Amnon Ta-Shma. Adiabatic quantum state generation and statistical zero knowledge. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 20–29, 2003. 36

[BB94] M. L. Bonet and S. R. Buss. Size-depth tradeoffs for boolean formulae. *Information Processing Letters*, 49:151–155, 1994. 97

[BBC+01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. Earlier version in FOCS'98. arXiv: `quant-ph/9802049` 26

[BBR+90] Paul Beame, Allan Borodin, Prabhakar Raghavan, Walter L Ruzzo, and Martin Tompa. Time-space tradeoffs for undirected graph traversal. In *Proceedings of the 1990 IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 429–438. IEEE, 1990. 2, 20

[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. 3, 4, 114, 115, 117, 119, 120, 121, 122

[BCJ+13] Aleksandrs Belovs, Andrew M. Childs, Stacey Jeffery, Robin Kothari, and Frédéric Magniez. Time-efficient quantum walks for 3-distinctness. In *Proceedings of the 40th International Colloquium on Automata, Languages, and Programming (ICALP)*, pages 105–122, 2013. 44

[BCO17] Johannes Bausch, Toby Cubitt, and Maris Ozols. The complexity of translationally invariant spin chains with low local dimension. *Annales Henri Poincaré*, 18(11):3449–3513, Nov 2017. 4, 152

[Bel13] Aleksandrs Belovs. Quantum walks and electric networks. arXiv: 1302.3143, 2013. 2, 21, 44, 46

[BF93] Greg Barnes and Uriel Feige. Short random walks on graphs. In *Proceedings of the 1993 ACM Symposium on the Theory of Computing (STOC)*, pages 728–737, 1993. 2, 20

[BJ25] Aleksandrs Belovs and Stacey Jeffery. Space-efficient quantum error reduction without log factors, 2025. arXiv: 2502.09249 116

[BJY24] Aleksandrs Belovs, Stacey Jeffery, and Duyal Yolcu. Taming quantum time complexity. *Quantum*, 8(1444), 2024. 7, 42, 97, 98, 116

[BK08] Christel Baier and Joost-Pieter Katoen. *Principles of Model Checking*. MIT Press, 2008. 114

[BKRU89] Andrei Z Broder, Anna R Karlin, Prabhakar Raghavan, and Eli Upfal. Trading space for time in undirected st-connectivity. In *Proceedings of the 1989 ACM Symposium on the Theory of Computing (STOC)*, pages 543–549, 1989. 2, 20

[BR12] Aleksandrs Blovs 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, 7, 20, 21, 41, 97, 115, 117, 118, 152

[BW90] Graham Brightwell and Peter Winkler. Maximum hitting time for random walks on graphs. *Random Structures & Algorithms*, 1(3):263–276, 1990. 31

[CGP99] Edmund M. Clarke, Orna Grumberg, and Doron A. Peled. *Model Checking*. MIT Press, 1999. 114

[CKKD+22] Andrew M. Childs, Robin Kothari, Matt Kovacs-Deak, Aarthi Sundaram, and Daochen Wang. Quantum divide and conquer. arXiv: 2210.06419, 2022. 6, 7, 96, 98, 105

[Cor23] Arjan Cornelissen. *Quantum multivariate estimation and span program algorithms*. PhD thesis, University of Amsterdam, 2023. 42, 64

[Cre83] Michael Creutz. *Quarks, gluons and lattices*. Cambridge University Press, 1983. 4, 152

[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, 3, 20, 21, 26, 97, 114

[Edm93] Jeff Edmonds. Time-space trade-offs for undirected st-connectivity on a JAG. In *Proceedings of the 1993 ACM Symposium on the Theory of Computing (STOC)*, pages 718–727, 1993. 2, 20

[Fei93] Uriel Feige. A randomized time-space tradeoff of $\tilde{O}(m\hat{R})$ for USTCON. In *Proceedings of the 1993 IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 238–246. IEEE, 1993. 2, 20

[FGGS98] Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Limit on the speed of quantum computation in determining parity. *Physical Review Letters*, 81(24):5442, 1998. 26

[GR02] Lov Grover and Terry Rudolph. Creating superpositions that correspond to efficiently integrable probability distributions. *arXiv preprint quant-ph/0208112*, 2002. 10, 30

[IJ19] Tsuyoshi Ito and Stacey Jeffery. Approximate span programs. *Algorithmica*, 79:2158–2195, 2019. 44

[Jef22] Stacey Jeffery. Quantum subroutine composition. arXiv: 2209.14146, 2022. 6, 8, 40, 41, 42, 45, 70, 71, 72, 73, 74, 75, 76, 98, 153

[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. 4, 7, 41, 53, 54, 97, 117

[JK17] Stacey Jeffery and Shelby Kimmel. Quantum algorithms for graph connectivity and formula evaluation. *Quantum*, 1:26, 2017. 4, 7, 41, 53, 54, 76, 97, 99, 117

[JP25a] 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, pages 54:1–54:16, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 4, 39, 95

[JP25b] Stacey Jeffery and Galina Pass. A quantum time-space tradeoff for directed *st*-connectivity. *arXiv preprint arXiv:2510.08403*, 2025. 113, 153

[JZ23] Stacey Jeffery and Sebastian Zur. Multidimensional quantum walks and application to k-distinctness. In *Proceedings of the 55th ACM Symposium on the Theory of Computing (STOC)*, pages 1125–1130, 2023. 6, 15, 16, 40, 41, 42, 43, 44, 45

[Kit96] Alexei Y. Kitaev. Quantum measurements and the Abelian stabilizer problem. *ECCC*, TR96-003, 1996. 15

[Kos13] Adrian Kosowski. Faster walks in graphs: A $\tilde{O}(n^2)$ time-space trade-off for undirected s-t connectivity. In *Proceedings of the 2013 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1873–1883, 2013. arXiv: 1204.1136v2 2, 20, 22, 28

[KP17] Iordanis Kerenidis and Anupam Prakash. Quantum recommendation systems. In *Proceedings of the 8th Innovations in Theoretical Computer Science Conference (ITCS)*, pages 49:1–49:21. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2017. 24

[Lee59] C. Y. Lee. Representation of switching functions by binary decision programs. *Bell Systems Technical Journal*, 38(4):985–999, 1959. 7, 41

[LHP+20] Jessica Lemieux, Bettina Heim, David Poulin, Krysta Svore, and Matthias Troyer. Efficient quantum walk circuits for metropolis-hastings algorithm. *Quantum*, 4:287, 2020. 28

[LPW17] David A Levin, Yuval Peres, and Elizabeth L Wilmer. *Markov chains and mixing times*, volume 107. American Mathematical Society, 2017. second edition. 14, 31

[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. 129

[MNRS11] Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. Search via quantum walk. *SIAM Journal on Computing*, 40(1):142–164, 2011. Earlier version in STOC'07. arXiv: quant-ph/0608026 46

[NNH99] Flemming Nielson, Hanne R. Nielson, and Chris Hankin. *Principles of Program Analysis*. Springer, 1999. 114

[Pot14] Aaron Potechin. Bounds on monotone switching networks for directed connectivity. *Journal of the ACM*, 9(4), 2014. 124

[Pot15] Aaron H Potechin. *Analyzing monotone space complexity via the switching network model*. PhD thesis, Massachusetts Institute of Technology, 2015. 41, 117, 128

[Rei08] Omer Reingold. Undirected connectivity in log-space. *Journal of the ACM*, 55(4), 2008. 2, 20

[Rei09] Ben W. Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every Boolean function. In *Proceedings of the 50th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 544–551, 2009. 7, 97

[Rei11] Ben W. Reichardt. Faster quantum algorithm for evaluating game trees. In *Proceedings of the 22nd ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 546–559, 2011. 96

[RKMC25] Sawyer Jack Robertson, Dhruv Kohli, Gal Mishne, and Alexander Cloninger. On a generalization of wasserstein distance and the beckmann problem to connection graphs. *SIAM Journal on Scientific Computing*, 47(5):A2774–A2800, 2025. 4, 152

[RS12] Ben W. Reichardt and Robert Spalek. Span-program-based quantum algorithm for evaluating formulas. *Theory of Computing*, 8(13):291–319, 2012. 44

[Sav70] Walter J Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of computer and system sciences*, 4(2):177–192, 1970. 3, 7, 97, 109

[Sha38] Claude E. Shannon. A symbolic analysis of relay and switching networks. *Transactions of the American Institute of Electrical Engineers*, 57(12):713–723, 1938. 7, 41, 54

[Sha49] Claude E. Shannon. The synthesis of two-terminal switching circuits. *Bell System Technical Journal*, 28(1):59–98, 1949. 7, 41, 54

[Sze04] Mario Szegedy. Quantum speed-up of Markov chain based algorithms. In *Proceedings of the 45th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 32–41, 2004. 44, 46

[Ull89] Jeffrey D. Ullman. *Principles of Database and Knowledge-Base Systems, Volumes 1–2*. Computer Science Press, 1988–1989. 114

[Wig92] Avi Wigderson. The complexity of graph connectivity. In *International Symposium on Mathematical Foundations of Computer Science*, pages 112–132. Springer, 1992. 1, 20

[Wil74] Kenneth G Wilson. Confinement of quarks. *Physical review D*, 10(8):2445, 1974. 4, 152

Abstract

This thesis studies space-bounded quantum computation, focusing on quantum algorithms in models where available memory is limited. The central goal is to understand which quantum advantages persist in such settings and how quantum resources influence classical time–space tradeoffs. This thesis investigates these questions through graph *st*-connectivity problems, a core class of problems in space complexity theory that has been widely studied.

The thesis develops quantum algorithms and tradeoff results for both undirected and directed *st*-connectivity. Connectivity problems are particularly well suited for studying space-bounded computation, as they are complete for fundamental space-bounded complexity classes and naturally model computation under memory constraints. In the undirected setting, we show that quantum algorithms can simultaneously achieve near-optimal time and logarithmic space, revealing a contrast with classical computation, where nontrivial time–space tradeoffs arise. These results demonstrate that space-bounded quantum and classical computation can exhibit fundamentally different behavior.

For directed *st*-connectivity, we establish the first nontrivial quantum time–space tradeoff. Directed reachability presents challenges absent from the undirected case, as techniques based on quantum walks and reversibility do not naturally extend. We show that quantum computation can improve upon known classical tradeoff bounds even when only a small quantum workspace is available. This provides new evidence that quantum effects can yield meaningful advantages in space-constrained regimes, beyond the large-memory settings where quantum speedups are more commonly studied.

A unifying theme of the thesis is the development of a compositional framework for quantum algorithms based on multidimensional quantum walks. We introduce subspace graphs, an abstract model that supports modular reasoning, abstraction, and time-efficient recursion. Subspace graphs provide a structured way to describe

quantum processes that incorporate subroutines while avoiding the overheads that typically obstruct recursive quantum constructions. This perspective enables composition at a higher level of abstraction, allowing complex procedures to be analyzed and combined without repeatedly incurring implementation costs. We apply this framework to obtain a time-efficient quantum divide & conquer construction, derive the quantum time–space tradeoff for directed *st*-connectivity, and obtain further results for generalized *st*-connectivity and other problems.

Finally, the thesis extends connectivity beyond classical graph models by studying unitary-labeled graphs, where edges carry quantum transformations. In this setting, paths induce transformations of an internal Hilbert space, giving rise to a generalized *st*-connectivity-type promise problem referred to as the gauge problem. We present an efficient quantum algorithm for this problem, showing that this generalization of undirected *st*-connectivity still admits a time- and space-efficient solution. This result highlights the broader applicability of the subspace-graph framework to computational problems involving genuinely quantum data.

Together, these results contribute to the understanding of space-bounded quantum computation, quantum time–space tradeoffs, and compositional methods for quantum algorithm design, while highlighting the role of connectivity problems as a bridge between quantum algorithms and space complexity theory.

Samenvatting

Dit proefschrift bestudeert geheugenbegrensde quantumcomputatie, met de nadruk op quantumalgoritmen in modellen waarin het beschikbare geheugen beperkt is. Het centrale doel is te begrijpen welke quantumvoordelen in dergelijke omgevingen behouden blijven en hoe quantumresources klassieke tijd–geheugen-tradeoffs beïnvloeden. Dit proefschrift onderzoekt deze vragen aan de hand van graaf-*st*-connectiviteitsproblemen, een fundamentele klasse van problemen in de geheugencomplexiteitstheorie die uitgebreid is bestudeerd.

Het proefschrift ontwikkelt quantumalgoritmen en tijd–geheugen-tradeoffs voor zowel ongerichte als gerichte *st*-connectiviteit. Connectiviteitsproblemen zijn bijzonder geschikt voor het bestuderen van geheugenbegrensde computatie, omdat zij compleet zijn voor fundamentele geheugencomplexiteitsklassen en op natuurlijke wijze berekening onder geheugenbeperkingen modelleren. In het ongerichte geval tonen wij aan dat quantumalgoritmen gelijktijdig een bijna optimale looptijd en logaritmisch geheugengebruik kunnen bereiken. Dit staat in contrast met klassieke computatie, waar niet-triviale tijd–geheugen-tradeoffs optreden. Deze resultaten laten zien dat geheugenbegrensde quantum- en klassieke computatie fundamenteel verschillend gedrag kunnen vertonen.

Voor gerichte *st*-connectiviteit stellen wij de eerste niet-triviale quantum tijd–geheugen-tradeoff vast. Gerichte bereikbaarheid brengt uitdagingen met zich mee die afwezig zijn in het ongerichte geval, aangezien technieken gebaseerd op quantumwandelingen en omkeerbaarheid zich niet op natuurlijke wijze laten uitbreiden. Wij tonen aan dat quantumcomputatie bekende klassieke tradeoff-grenzen kan verbeteren, zelfs wanneer slechts een kleine quantumwerkruimte beschikbaar is. Dit levert nieuw bewijs dat quantumeffecten betekenisvolle voordelen kunnen bieden in regimes met beperkt geheugen, voorbij de situaties met groot geheugen waarin quantumverbeteringen doorgaans worden bestudeerd.

Een centraal thema in dit proefschrift is de ontwikkeling van een raamwerk

voor het samenstellen van quantumalgoritmen, gebaseerd op multidimensionale quantumwandelingen. Wij introduceren subruimtegrafen, een abstract model dat modulaire analyse, abstractie en tijdefficiënte recursie ondersteunt. Subruimtegrafen bieden een gestructureerde manier om quantumprocessen te beschrijven die subroutines gebruiken, terwijl de overhead wordt vermeden die recursieve quantumconstructies doorgaans belemmert. Dit perspectief maakt het mogelijk quantumalgoritmen op een hoger abstractieniveau samen te stellen, waardoor complexe procedures kunnen worden geanalyseerd en gecombineerd zonder herhaaldelijk implementatiekosten te maken. Wij passen dit raamwerk toe om een tijdefficiënte quantum divide-and-conquer-constructie te verkrijgen, de quantum tijd–geheugen-tradeoff voor gerichte *st*-connectiviteit af te leiden, en verdere resultaten te verkrijgen voor gegeneraliseerde *st*-connectiviteit en verwante problemen.

Ten slotte breidt dit proefschrift het connectiviteitsbegrip uit voorbij klassieke graafmodellen door unitaire-gelabelde grafen te bestuderen, waarbij randen quantumtransformaties representeren. In deze setting induceren paden transformaties op een interne Hilbertruimte, wat leidt tot een gegeneraliseerd *st*-connectiviteitsachtig belofteprobleem, het zogeheten gauge-probleem. Wij presenteren een efficiënt quantumalgoritme voor dit probleem en tonen aan dat deze generalisatie van ongerichte *st*-connectiviteit nog steeds een tijd- en geheugenefficiënte oplossing toelaat. Dit resultaat onderstreept de bredere toepasbaarheid van het subruimtegraafraamwerk op berekeningsproblemen waarin intrinsiek quantummechanische data een rol spelen.

Gezamenlijk dragen deze resultaten bij aan een beter begrip van geheugenbegrensde quantumcomputatie, quantum tijd–geheugen-tradeoffs en technieken voor het samenstellen van quantumalgoritmen. Daarnaast benadrukken zij de rol van connectiviteitsproblemen als brug tussen quantumalgoritmen en de geheugencomplexiteitstheorie.

Acknowledgments

First and foremost, I would like to thank my supervisors, Stacey and Michael.

Stacey, I feel very lucky to have had you as my supervisor. This is not something one can take for granted as a PhD student, but I always felt that you had my best interests in mind. During a time that can be quite uncertain, wondering whether things will work out and whether there will be enough results or papers, it meant a lot to know that you had my back. It made the whole journey feel much less intimidating. I also want to thank you for teaching me so much. Doing a PhD is not always easy, but it has definitely been very interesting and exciting for me, and this is entirely thanks to you. Beyond the academic side, I'm very grateful for your support as a person. These years haven't been easy, and having you there meant a great deal to me. You helped me more than I could have expected, and I truly appreciate that.

Michael, I would like to thank you for your guidance and support during my PhD. Although we did not work as closely together over time, I am very grateful for your input and for the role you played along the way. I especially want to thank you for guiding me through the beginning of my PhD and for teaching me how to write my first paper, which was such an important step in my PhD journey. I have greatly appreciated your perspective, your thoughtful feedback, and our discussions.

I would like to thank my committee, Ronald, Frédéric, Jo, Krystal, and Māris for taking the time to read my thesis and for the valuable feedback.

I would like to thank my coauthors Simon and Tobias.

Simon, we worked a lot together throughout my PhD, and it has always been a great experience. I find it very special how you created a supportive environment where I felt encouraged and confident, which is so important for a student. I have also always found your work very interesting, and I am very grateful that you shared it with me and taught me so much. I would also like to thank you for taking

on the role of my supervisor during my visits to Paris. More generally, I think you are a great person, and I have really enjoyed working with you.

Tobias, thank you for the collaboration and the interesting discussions we had. I am also grateful to you for hosting me in Hannover. I really appreciated working with you and learning from you.

I would also like to thank Frédéric and Arjan, who played important roles throughout my PhD journey.

Frédéric, my visits to IRIF are among the best memories of my PhD, and I am very grateful to you for welcoming me. I always felt very much part of the group during that time. I also want to thank you for supervising me during my visit to Paris and for introducing me to beautiful problems.

Arjan, thank you for explaining so much to me about quantum algorithms and for being such a great collaborator. It has always been fun working with you. I also really appreciated our chats and having you as a research buddy.

I would like to thank everyone at QuSoft for creating such a stimulating and friendly environment. I feel very lucky to have been part of this group and am truly grateful to everyone who contributes to making it such a great place to do research.

I would like to thank everyone at KdVI. I was very happy to be a part of the institute, and I am especially grateful to the administrative and support staff who took care of the many practical aspects of the PhD process and ensured that everything ran smoothly.

I would like to thank everyone at IRIF for making my visits so enjoyable. I had a really great time there and met many amazing people.

I would also like to thank my friends outside academia for being there in their own way and for all the moments we shared during these years.

I would like to thank my family, who have been there from the very beginning of this journey. You have always supported me, cared for me during the lowest moments, and shared in the happiness of the successes. None of this would have been possible without you, and I draw so much strength from having you behind me. I am very aware of how lucky I am to be loved simply for who I am, and that has given me the strength to keep going and to achieve what I have. Thank you for taking my struggles seriously and for always helping as much as I needed. Everything I have achieved is built on your support.

Philip, thank you for everything you have been to me throughout this journey. I won't even try to put into words what that has meant to me.