

# Multidimensional Quantum Walks, Recursion, and Quantum Divide & Conquer

Stacey Jeffery<sup>1,2</sup> and Galina Pass<sup>2</sup>

<sup>1</sup>CWI, Amsterdam

<sup>2</sup>QuSoft & University of Amsterdam

May 9, 2024

## Abstract

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. As another application, we give a time-efficient implementation of quantum Divide & Conquer when the sub-problems are combined via a Boolean formula. We use this to quadratically speed up Savitch’s algorithm for directed *st*-connectivity.

## 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 paper 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 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.

**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  $f_{k,d} : \{0, 1\}^{d^k} \rightarrow \{0, 1\}$  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<sup>+</sup>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<sup>+</sup>22]. In particular, we show (see Theorem 4.13 for formal statement):

**Theorem 1.1** (Informal). *Let  $\{f_{\ell,n} : D_{\ell,n} \rightarrow \{0, 1\}\}_{\ell,n}$  be a family of functions. Let  $\varphi$  be a symmetric Boolean formula on  $a$  variables, and suppose  $f_{\ell,n} = \varphi(f_{\ell/b,n}, \dots, f_{\ell/b,n}) \vee f_{aux,\ell,n}$ , for some  $b > 1$  and*

some auxiliary function  $f_{\text{aux},\ell,n}$  with quantum time complexity  $T_{\text{aux}}(\ell, n)$ . Then the quantum time complexity of  $f_{\ell,n}$  is  $\tilde{O}(T(\ell, n))$  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  $f_{\ell,n} = \varphi(f_{\ell/b,n}, \dots, f_{\ell/b,n}, f_{\text{aux},\ell,n})$  for any formula  $\varphi$  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 [BJY23].

**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<sup>1</sup> 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 network when we have query access to the edge variables. Let  $\mathcal{R}_{s,t}(G(x))$  be the effective resistance (see Definition 2.1) 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 2.2) consisting of only edges that are “off”. Then the time complexity of evaluating the switching network is

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

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

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

Here we improve this to (see Theorem 4.15):

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

---

<sup>1</sup>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.

This is analogous to results of [Jef22], which showed a similar statement, but for quantum walks rather than switching networks<sup>2</sup>. 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.

**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, AJPW22], including quantum algorithms that achieve optimal time- and space-complexity simultaneously, in both the edge-list access model, and the adjacency matrix access model.<sup>3</sup> 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  $\tilde{O}(n)$  time and space<sup>4</sup>. 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 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 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 5.3).

**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 2.1 for details.

**Related and Future Work** While writing this manuscript, we became aware of an independent work that also achieves time-efficient quantum divide & conquer [ABB<sup>+</sup>23] when either (1)  $\varphi$  is an OR (equivalently, an AND) or (2)  $\varphi$  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<sup>+</sup>23] 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<sup>+</sup>23], 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<sup>+</sup>23] applies their framework to problems that are distinct from our applications.

We also mention that Ref. [CKKD<sup>+</sup>22] analyzes the quantum query complexity of divide & conquer where  $\varphi$  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

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

<sup>3</sup>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.

<sup>4</sup>In the edge-list access model.

framework, *transducers* [BJY23], that allows for the composition of quantum algorithms without the need for success probability amplification. We leave further investigation for future work.

We can compare the model of transducers [BJY23] 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 [BJY23, Section 3.2]. Compared with the model of subspace graphs, transducers are more general, and cleaner, in 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.

## 2 Preliminaries

### 2.1 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]).

### 2.2 Graph Theory

In this paper, a *graph* will be an undirected graph, and we will only consider directed graphs in Section 5. A graph  $G = (V, E)$  has vertex set  $V = V(G)$  and edge set  $E = E(G)$ . For  $u \in V$ , we let  $E(u)$  be the subset of edges of  $E$  that are incident to  $u$ . 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. Then for any  $e \in E$ , there is a unique  $u \in V$  such that  $e \in E^{\rightarrow}(u)$ , and a unique  $v \in V$  such that  $e \in E^{\leftarrow}(v)$ , and we can write  $e = (u, v)$ . We can associate non-negative *weights*  $\{w_e\}_{e \in E}$  to the edges of  $G$ .

**Definition 2.1.** A flow on a weighted graph  $G$  is a real-valued function  $\theta$  on  $E$ . Define

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

We call  $\theta$  a unit *st-flow* if  $\theta(s) = -\theta(t) = 1$ , and for all  $u \in V \setminus \{s, t\}$ ,  $\theta(u) = 0$ . The effective resistance is defined

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

**Definition 2.2.** An *st-cut-set* is a set  $F \subseteq E$  whose removal from  $G$  would leave  $s$  and  $t$  in different connected components.



## 2.3 Function Composition and Boolean Formulas

A Boolean formula,  $\varphi$ , 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  $\Sigma$ , so  $|\Sigma| = N$ . We let  $\Sigma_i$  denote the set of strings in  $\Sigma$  whose first letter is  $i$ . We let  $\bar{\Sigma}$  denote the set of labels of all vertices, so it is the set of prefixes of strings in  $\Sigma$ .

We label the variable associated with a leaf  $\sigma \in \Sigma$  by  $\sigma$  as well. A fixed setting of the variables  $x \in \{0, 1\}^\Sigma$  induces a value at each vertex of  $\varphi$ , as follows. If a leaf  $\sigma$  is labelled by an unnegated variable, then its value is  $x_\sigma$ , 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.

**Definition 2.3.** *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.

**Definition 2.4.** *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  $f : \{0, 1\}^\Sigma \rightarrow \{0, 1\}$  and  $\{f_\sigma : \{0, 1\}^{m_\sigma} \rightarrow \{0, 1\}\}_{\sigma \in \Sigma}$ , we define the composed function  $f \circ (f_\sigma)_{\sigma \in \Sigma} : \{0, 1\}^{\sum_{\sigma \in \Sigma} m_\sigma} \rightarrow \{0, 1\}$  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  $\sigma$ -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  $f_\sigma$  is the identity on  $\{0, 1\}$  for all  $\sigma \in \Sigma \setminus F$ . For a Boolean formula  $\varphi$  on  $\Sigma$ , 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  $f_\varphi$  is the function computed by the formula.

## 2.4 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.

**Definition 2.5** (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  $\Psi^A, \Psi^B \subset H$ . We further assume that  $|\psi_0\rangle$  is orthogonal to every vector in  $\Psi^B$ . Let  $\Pi_A$  be the orthogonal projector onto  $\mathcal{A} = \text{span}\{\Psi^A\}$ , and similarly for  $\Pi_B$ .*

Let  $U_{AB} = (2\Pi_A - I)(2\Pi_B - I)$ . The algorithm defined by  $(H, |\psi_0\rangle, \Psi^A, \Psi^B)$  performs phase estimation of  $U_{AB}$  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*.

**Definition 2.6** (Negative Witness). *A negative witness for  $(H, |\psi_0\rangle, \Psi^A, \Psi^B)$  is a vector  $|w_A\rangle \in \mathcal{A}$  such that  $|w_B\rangle := |\psi_0\rangle - |w_A\rangle \in \mathcal{B}$ .*

**Definition 2.7** (Positive Witness). *A positive witness for  $(H, |\psi_0\rangle, \Psi^A, \Psi^B)$  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.

**Theorem 2.8** ([JZ23]). *Fix  $(H, |\psi_0\rangle, \Psi^{\mathcal{A}}, \Psi^{\mathcal{B}})$  as in Definition 2.5. Suppose we can generate the state  $|\psi_0\rangle$  in cost  $S$ , and implement  $U_{\mathcal{A}\mathcal{B}} = (2\Pi_{\mathcal{A}} - I)(2\Pi_{\mathcal{B}} - I)$  in cost  $A$ .*

*Let  $c_+ \in [1, 50]$  be some constant, and let  $C_- \geq 1$  be a positive real number that may scale with  $|x|$ , such that we are guaranteed that one of the following holds:*

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

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

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

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

*and space  $O(\log \dim H + \log C_-)$ .*

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*.

**Definition 2.9** (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  $T$ .*

**Claim 2.10.** *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  $|\ell\rangle \mapsto |b_\ell\rangle$ .  $\square$

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

**Corollary 2.11.** *If  $\Psi$  is a working basis that can each be generated in time  $T$ , then there is a basis  $\Psi'$  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  $U_\Psi$  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,  $\Psi'$  is generated by  $U_\Psi$ . Since both bases are generated by the same map, their generation has the same time complexity.  $\square$

### 3 Subspace Graphs for Multidimensional Quantum Walks

#### 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, RŠ12, 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.

**Definition 3.1** (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  $\Xi_e^A$  and  $\Xi_e^B$  be subspaces of  $\Xi_e$ . These need not be orthogonal, and they may each be  $\{0\}$ , all of  $\Xi_e$ , 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

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

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 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  $G$ .

**Remark 3.2.** 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  $\Pi_e$ ,  $\Pi_e^A$  and  $\Pi_e^B$  denote orthogonal projectors onto  $\Xi_e$ ,  $\Xi_e^A$ , and  $\Xi_e^B$ , respectively. For any set  $F \subseteq E \cup B$ , we let  $\Xi_F = \bigoplus_{e \in F} \Xi_e$ ,  $\Xi_F^A = \bigoplus_{e \in F} \Xi_e^A$ , and similarly for  $\Xi_F^B$ . We let  $\Pi_F$ ,  $\Pi_F^A$  and  $\Pi_F^B$  denote the orthogonal projectors onto  $\Xi_F$ ,  $\Xi_F^A$  and  $\Xi_F^B$ , 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 2.4](#)) 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 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.

**Definition 3.3** (Simple Vertex). Fix a subspace graph  $G$  with associated edge weights  $\{w_e\}_{e \in E}$ . For  $u \in V$ , define

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



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.

**Example 3.4** (Quantum Walk on a Graph). A subspace graph  $G$ , with associated edge weights, in which

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

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  $s$ <sup>5</sup>.

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  $w_e / (\sum_{e' \in E(u)} w_{e'})$ . 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].<sup>6</sup>

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 paper. We first define what it means for an edge to be a switch.

**Definition 3.5** (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

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

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

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). If  $\Xi_e^\perp = \{0\}$ , then  $|w\rangle$  can have no overlap with  $\Xi_e$ , 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 3.3 - in particular, see (4)).

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

<sup>5</sup>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*  $\sigma$  on  $S$ .

<sup>6</sup>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.

**Definition 3.6** (Canonical  $st$ -boundary). *We say a subspace graph  $G$  has canonical  $st$ -boundary if  $B = \{s, t\}$ , where:*

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

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 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 3.3](#)).

**Definition 3.7** (Switching Network). *A switching network is a subspace graph with canonical  $st$ -boundary in which all edges are switches ([Definition 3.5](#)), and all vertices are simple ([Definition 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 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 3.3](#). The examples we will detail in this paper 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.

### 3.2 Phase Estimation Algorithm

A subspace graph, as in [Definition 3.1](#), gives rise to a phase estimation algorithm (see [Section 2.4](#)) with spaces  $\mathcal{A}_G$  and  $\mathcal{B}_G$  as in [Definition 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 paper, we will always assume  $G$  has canonical  $st$ -boundary, and will let

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

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  $\Xi_e^B = \{0\}$  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  $\Xi_e^A$ . However, it is in our interest to put as much of  $\Xi_e^A + \Xi_e^B$  into  $\Xi_e^B$  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 2.6](#), and also [Definition 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 3.3](#).

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

**Definition 3.8** (Composable Bases). *Fix a subspace graph with canonical  $st$ -boundary, and some  $r \in \mathbb{R}_{>0}$ . We say a pair of working bases  $\Psi_A$  and  $\Psi_B$  (see [Definition 2.9](#)) for  $\mathcal{A}_G$  and  $\mathcal{B}_G$  are  $st$ -composable, with scaling factor  $r$ , if they satisfy the following assumptions:*

1.  $\Psi_A = \bigcup_{e \in E \cup B} \Psi_A(e)$ , where  $\Psi_A(e)$  is an orthonormal basis for  $\Xi_e^A$ ;
2.  $\Psi_B = \Psi_B^- \cup \bigcup_{e \in E \cup B} \Psi_B(e)$ , for some  $\Psi_B^-$ , where  $\Psi_B(e)$  is an orthonormal basis for  $\Xi_e^B$ ;
3.  $|b_0\rangle \in \Psi_B^-$  satisfies  $|b_0\rangle = \frac{1}{\sqrt{2}}(|\leftarrow, s\rangle + |\rightarrow, t\rangle)$ ;
4.  $|b_1\rangle \in \Psi_B^-$  satisfies  $|b_1\rangle = \frac{|\leftarrow, s\rangle - |\rightarrow, t\rangle + \sqrt{r}|\bar{b}_1\rangle}{\sqrt{2+r}}$  for some unit vector  $|\bar{b}_1\rangle$  that is orthogonal to  $|\leftarrow, s\rangle$  and  $|\rightarrow, t\rangle$ ;

5.  $\Psi_B \setminus \{|b_0\rangle, |b_1\rangle\}$  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  $\Xi_E$  relative to  $\Xi_B$ . See also the remark after [Theorem 3.13](#).

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

**Definition 3.9.** We say a subspace graph  $G$  is *st-composable* if:

1.  $G$  has canonical *st-boundary* ([Definition 3.6](#));
2.  $G$  is equipped with *composable working bases* ([Definition 3.8](#));
3. for each  $e \in E \setminus \bar{E}$ ,  $\Xi_e^B = \{0\}$ , where  $\bar{E}$  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](#)) 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 2.8](#). This motivates the following definition.

**Definition 3.10** (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](#), 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:

**Definition 3.11** (Negative Witness for a Graph). We say  $|w_A\rangle \in \mathcal{A}_G$  is a *negative witness* for  $G$  if  $|w_A\rangle - |s\rangle + |t\rangle \in \mathcal{B}_G$ . We define the corresponding *cropped negative witness* by  $|\hat{w}_A\rangle = \Pi_E |w_A\rangle$ .

For the above, recall that a negative witness for a phase estimation algorithm is a vector  $|w_A\rangle \in \mathcal{A}$  such that  $|w_B\rangle := |\psi_0\rangle - |w_A\rangle \in \mathcal{B}$ . 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 paper.

**Lemma 3.12.** Assume  $G$  has canonical *st-boundary*. Then  $|w_A\rangle \in \mathcal{A}_G$  is a *negative witness* for  $G$  if and only if  $|\hat{w}_A\rangle + |\leftarrow, s\rangle - |\rightarrow, t\rangle \in \mathcal{B}_G$ .

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

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

Thus,  $|w_A\rangle$  is a negative witness if and only if

$$\begin{aligned} & |w_A\rangle - |s\rangle + |t\rangle \in \mathcal{B}_G \\ \Leftrightarrow & |\hat{w}_A\rangle + |\leftarrow, s\rangle - |\rightarrow, t\rangle \in \mathcal{B}_G. \end{aligned} \quad \square$$

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 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 subspace 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. Then it follows from [Theorem 2.8](#) and [Claim 2.10](#) that:

**Theorem 3.13.** *Let  $G$  be a subspace graph that computes  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  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  $O(\log \dim H_G + \log \hat{W}_-(G))$ .*

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, 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 3.8](#), and simply scales  $r$ . We can see this formally in [Corollary 4.2](#).

### 3.3 Example: Switching Networks

We now discuss switching networks ([Definition 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  $f : \{0, 1\}^E \rightarrow \{0, 1\}$  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 3.13](#) decides if  $s$  and  $t$  are connected in  $G(x)$ .

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

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

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

**Lemma 3.14.** *If  $G$  is a switching network and  $\Psi_{\mathcal{A}}$  is the basis in (2), then  $\Psi_{\mathcal{A}}$  satisfies the conditions of [Definition 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:

$$\begin{aligned} \mathcal{B}_G = \text{span}\{|\psi_\star(u)\rangle : u \in V \setminus \{s, t\}\} \cup \{|\leftarrow, s\rangle + |\psi_\star(s)\rangle, |\rightarrow, t\rangle + |\psi_\star(t)\rangle\} \\ + \text{span} \left\{ \frac{1}{\sqrt{2}}(|\rightarrow, e\rangle + |\leftarrow, e\rangle) : e \in E \right\}. \quad (3) \end{aligned}$$

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).

**Lemma 3.15.** *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*

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

*where*

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

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).

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 paper. One can show that the orthogonal complement of  $\mathcal{B}_G$  is the span of all unit  $st$ -flows (Definition 2.1) of  $G$ , by which we mean, precisely, all states of the form

$$|w\rangle = |s\rangle - |\leftarrow, s\rangle + \underbrace{\sum_{e \in E} \frac{\theta(e)}{\sqrt{w_e}}(|\rightarrow, e\rangle - |\leftarrow, e\rangle)}_{|\tilde{w}\rangle} + |\rightarrow, t\rangle - |t\rangle, \quad (4)$$

where  $\theta$  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,  $\theta$  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

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

is a cropped negative witness. However, not all witnesses have this form – superpositions of cuts are also negative witnesses. Ref. [JKP18] 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), 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.

Finally, we give two examples of switching networks – for computing the OR of  $d$  bits (Section 3.3.1) and the AND of  $d$  bits (Section 3.3.2). 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.



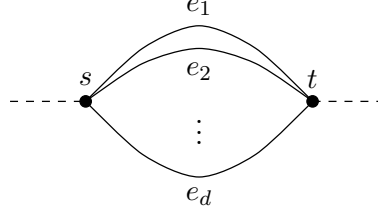


Figure 1: The graph  $G_{\text{OR}}$ . The dashed lines represent dangling boundary “edges”.

### 3.3.1 Switching Network for OR

In this section, we describe a switching network (Definition 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 4.3. Formally, we prove the following.

**Lemma 3.16.** *For any  $d \geq 1$ , and positive weights  $\{w_i\}_{i \in [d]}$ , 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  $\mathbf{r} = 2 \sum_{i=1}^d w_i$ , that can be generated in  $O(\log d)$  time, assuming the state proportional to  $\sum_{i=1}^d \sqrt{w_i} |i\rangle$  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  $|\hat{w}\rangle = \frac{1}{\sqrt{w_i}}(|\rightarrow, i\rangle - |\leftarrow, i\rangle)$ ; and
3. if  $\varphi(e_i) = 0$  for all  $i \in [d]$ ,  $G_{\text{OR},d}$  has cropped negative witness  $|\hat{w}_{\mathcal{A}}\rangle = \sum_{i=1}^d \sqrt{w_i}(|\rightarrow, i\rangle - |\leftarrow, i\rangle)$ .

We remark that if  $w_i = 1$  for all  $i$ , then Lemma 3.16 implies  $\hat{W}_+(G_{\text{OR},d}) = 2$  and  $\hat{W}_-(G_{\text{OR},d}) = 2d$ , so by Theorem 3.13, there is a quantum algorithm for evaluating  $d$ -bit OR with time complexity  $\tilde{O}(\sqrt{d})$ , 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 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  $w_{e_i} = w_i$ . 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  $w_i = 1$  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  $e_i$ ):

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

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

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

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

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

Then we have:

$$\begin{aligned} \mathcal{A}_G &= \bigoplus_{e \in E \cup B} \Xi_e^{\mathcal{A}} = \text{span}\{|\rightarrow, i\rangle - (-1)^{\varphi(e_i)}|\leftarrow, i\rangle : i \in [d]\} \cup \{ |s\rangle + |\leftarrow, s\rangle, |t\rangle + |\rightarrow, t\rangle \} \\ \text{and } \mathcal{B}_G &= \mathcal{V}_s \oplus \mathcal{V}_t + \bigoplus_{e \in E} \Xi_e^{\mathcal{B}} = \mathcal{V}_s \oplus \mathcal{V}_t + \text{span}\{|\rightarrow, i\rangle + |\leftarrow, i\rangle : i \in [d]\}. \end{aligned} \quad (5)$$

By [Lemma 3.14](#), 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 3.8](#) that can be generated in unit time. To complete the proof of the first item of [Lemma 3.16](#), we prove the following.

**Lemma 3.17.** *Let*

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

where

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

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 3.8](#). Furthermore, as long as we can generate the state proportional to  $\sum_{i=1}^d \sqrt{w_i}|i\rangle$  in time  $O(\log d)$ ,  $\Psi_{\mathcal{B}}$  can be generated in time  $O(\log d)$ .

*Proof.* Note that:

$$|\leftarrow, s\rangle + |\rightarrow, t\rangle = \underbrace{|\leftarrow, s\rangle + \sum_{i=1}^d \sqrt{w_i}|\rightarrow, i\rangle}_{\in \mathcal{V}_s} + \underbrace{\sum_{i=1}^d \sqrt{w_i}|\leftarrow, i\rangle + |\rightarrow, t\rangle}_{\in \mathcal{V}_t} - \sum_{i=1}^d \sqrt{w_i} \underbrace{(|\rightarrow, i\rangle + |\leftarrow, i\rangle)}_{\in \Xi_e^{\mathcal{B}}}$$

and

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

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

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

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

It is clear by inspection that  $\Psi_{\mathcal{B}}$  satisfies the properties of [Definition 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.  $\square$

To prove the second item of [Lemma 3.16](#), we show the following.

**Lemma 3.18.** *If  $\varphi(e_i) = 1$  for some  $i$ , then the following is a positive witness for  $G$  ([Definition 3.10](#)):*

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

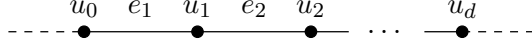


Figure 2: The graph  $G_{\text{AND}}$ . The dashed lines represent dangling boundary “edges”.

*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 (5). For orthogonality with

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

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

Finally, to prove the third item of Lemma 3.16, we show the following.

**Lemma 3.19.** *If  $\varphi(e_i) = 0$  for all  $i$ , then the following is a negative witness for  $G$  (Definition 3.11):*

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

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

$$\begin{aligned} |s\rangle - |t\rangle &= \underbrace{|s\rangle + |\leftarrow, s\rangle}_{\in \Xi_s^{\mathcal{A}}} - \underbrace{\left( |\leftarrow, s\rangle + \sum_{i=1}^d \sqrt{w_i} |\rightarrow, i\rangle \right)}_{\in \mathcal{V}_s \subset \mathcal{B}} + \sum_{i=1}^d \sqrt{w_i} \underbrace{\left( |\rightarrow, i\rangle - (-1)^{\varphi(i)} |\leftarrow, i\rangle \right)}_{\in \Xi_{e_i}^{\mathcal{A}}} \\ &\quad + \underbrace{\left( \sum_{i=1}^d \sqrt{w_i} |\leftarrow, i\rangle + |\rightarrow, t\rangle \right)}_{\in \mathcal{V}_t \subset \mathcal{B}} - \underbrace{\left( |\rightarrow, t\rangle + |t\rangle \right)}_{\in \Xi_t^{\mathcal{A}}}, \end{aligned}$$

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.  $\square$

### 3.3.2 Switching Network for AND

In this section, we describe a switching network (Definition 3.7) that computes the AND of  $d$  Boolean variables.

**Lemma 3.20.** *For any  $d \geq 1$ , and positive weights  $\{w_i\}_{i \in [d]}$ , there is a switching network  $G_{\text{AND},d}$  that computes  $\bigwedge_{i=1}^d \varphi(e_i)$  with  $\dim H_{G_{\text{AND},d}} = 2d + 4$  such that:*

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

As with the OR switching network in Section 3.3.1, a corollary of this lemma is that there is a quantum algorithm for evaluating AND in optimal time  $\tilde{O}(\sqrt{d})$ .

Let  $G = G_{\text{AND},d}$  be the graph in Figure 2, defined by

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

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  $w_{e_i} = w_i$ . Since  $G$  is a switching network: every edge is a switch, which fixes  $\Xi_{e_i}$ ,  $\Xi_{e_i}^A$  and  $\Xi_{e_i}^B$  for all  $i \in [d]$ ;  $G$  has canonical  $st$ -boundary, which fixes  $\Xi_s$ ,  $\Xi_s^A$ ,  $\Xi_s^B$ ,  $\Xi_t$ ,  $\Xi_t^A$ , and  $\Xi_t^B$ ; and every vertex is simple, which fixes the spaces  $\mathcal{V}_{u_i}$  for  $i \in \{0, \dots, d\}$ . In particular, letting  $w_0 = w_{d+1} = 1$ ,  $s = 0$ ,  $t = d+1$ , and  $i \in [d]$  label  $e_i$ , we must have:

$$\forall i \in \{0, \dots, d\}, \mathcal{V}_{u_i} = \text{span} \{ \sqrt{w_i} | \leftarrow, i \rangle + \sqrt{w_{i+1}} | \rightarrow, i+1 \rangle \}.$$

This fully defines

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

By [Lemma 3.14](#), 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 3.8](#) that can be generated in unit time. To complete the proof of the the first item of [Lemma 3.20](#), we prove the following.

**Lemma 3.21.** *Let  $|b_2\rangle, \dots, |b_d\rangle$  be an orthonormal basis for*

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

Let

$$\Psi_{\bar{\mathcal{B}}} = \left\{ |b_0\rangle = \frac{1}{\sqrt{2}} (| \leftarrow, s \rangle + | \rightarrow, t \rangle), |b_1\rangle = \frac{| \leftarrow, s \rangle - | \rightarrow, t \rangle + \sqrt{r} |\bar{b}_1\rangle}{\sqrt{2+r}} \right\} \cup \{|b_2\rangle, \dots, |b_d\rangle\}.$$

Then

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

is an orthonormal basis for  $\mathcal{B}_G$  when  $r = 2/\sum_{i=1}^d \frac{1}{w_i}$ , and it satisfies all the conditions of an  $st$ -composable basis in [Definition 3.8](#). Furthermore, as long as we can generate the state proportional to  $\sum_{i=1}^d \frac{1}{\sqrt{w_i}} |i\rangle$  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  $\Psi_{\bar{\mathcal{B}}} \subset \text{span}\{ | \leftarrow, s \rangle, | \rightarrow, t \rangle \} \cup \{ | \rightarrow, i \rangle - | \leftarrow, i \rangle : i \in [d] \}$ . Finally,  $\Psi_{\bar{\mathcal{B}}}$  is an orthonormal set because for  $i \in \{2, \dots, d\}$ ,  $\langle \leftarrow, s | b_i \rangle = \langle \rightarrow, t | b_i \rangle = \langle \bar{b}_1 | b_i \rangle = 0$ , and  $|b_0\rangle$  and  $|b_1\rangle$  are also orthogonal.

Next, observe

$$\mathcal{B} = \text{span} \{ \sqrt{w_i} | \leftarrow, i \rangle + \sqrt{w_{i+1}} | \rightarrow, i+1 \rangle : i \in \{0, \dots, d\} \} \cup \{ \sqrt{w_i} (| \rightarrow, i \rangle + | \leftarrow, i \rangle) : i \in [d] \},$$

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  $\bar{\mathcal{B}}_G = \mathcal{B}_G^{\perp}$ , where

$$\bar{\mathcal{B}}_G = \text{span}\{ |s\rangle, |t\rangle \} \oplus \text{span} \left\{ | \leftarrow, s \rangle - | \rightarrow, t \rangle - \frac{2}{\sqrt{r}} |\bar{b}_1\rangle \right\}. \quad (6)$$

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

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

where we have used  $s = 0$ ,  $t = d + 1$ , and  $w_0 = w_{d+1} = 1$ , 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

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

(see [Definition 3.5](#) and [Definition 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  $\Psi_{\mathcal{B}} \setminus \Psi_{\overline{\mathcal{B}}}$  is orthogonal to all of  $\overline{\mathcal{B}}_G$ .

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

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

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

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

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 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 [\(6\)](#) can be generated in  $O(\log d)$  time, and then the result follows from [Corollary 2.11](#).  $\square$

To prove the second item of [Lemma 3.20](#), we show the following.

**Lemma 3.22.** *If  $\varphi(e_i) = 1$  for all  $i \in [d]$ , then the following is a positive witness for  $G$  ([Definition 3.10](#)):*

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

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

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

We used the fact that  $w_0 = w_{d+1} = 1$ ,  $0 = s$  and  $d + 1 = t$ . Thus,  $|w\rangle \in \mathcal{A}_G^\perp \cap \mathcal{B}_G^\perp$ .  $\square$

Finally, to prove the third item of [Lemma 3.20](#), we show the following.

**Lemma 3.23.** *If  $\varphi(e_i) = 0$  for some  $i \in [d]$ , then the following is a negative witness for  $G$  ([Definition 3.11](#)):*

$$|w_A\rangle = |s\rangle + |\leftarrow, s\rangle + \underbrace{\sqrt{w_j} (|\rightarrow, j\rangle - |\leftarrow, j\rangle)}_{|\hat{w}\rangle} - |\rightarrow, t\rangle - |t\rangle.$$



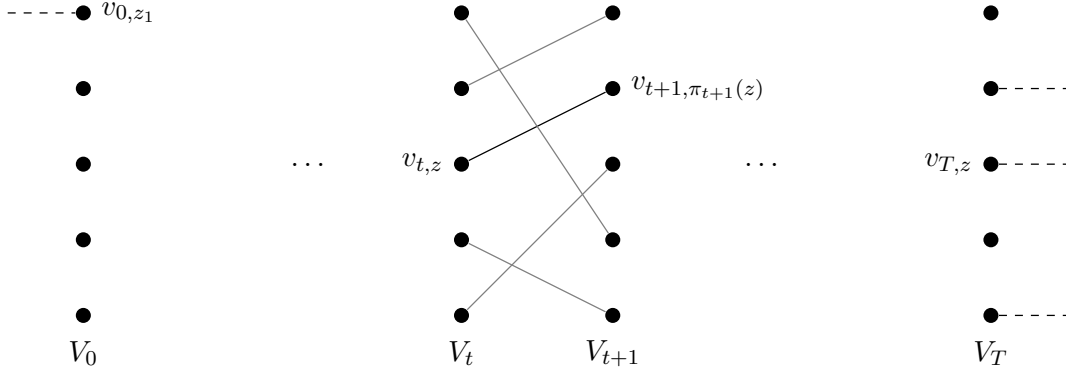


Figure 3: The graph  $G$  for a reversible classical deterministic algorithm with five algorithm states. Between each set  $V_t$  and  $V_{t+1}$ , there is exactly one edge connected to  $s = v_{0,z_1}$ . The dashed lines represent dangling boundary “edges”.

*Proof.* First note that for any  $j \in [d]$ , using  $w_0 = 1$  and  $|\leftarrow, 0\rangle = |\leftarrow, s\rangle$ :

$$\begin{aligned} & \sum_{i=1}^j \underbrace{(\sqrt{w_{i-1}}|\leftarrow, i-1\rangle + \sqrt{w_i}|\rightarrow, i\rangle)}_{\in \mathcal{V}_{u_{i-1} \subset \mathcal{B}}} - \sum_{i=1}^{j-1} \sqrt{w_i} \underbrace{(|\rightarrow, i\rangle + |\leftarrow, i\rangle)}_{\in \Xi_{e_i}^{\mathcal{B}}} \\ &= \sqrt{w_0}|\leftarrow, 0\rangle + \sqrt{w_j}|\rightarrow, j\rangle = |\leftarrow, s\rangle + \sqrt{w_j}|\rightarrow, j\rangle \end{aligned}$$

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

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:

$$\begin{aligned} |s\rangle - |t\rangle &= \underbrace{|s\rangle + |\leftarrow, s\rangle}_{\in \Xi_s^{\mathcal{A}}} - \underbrace{(|\leftarrow, s\rangle + \sqrt{w_j}|\rightarrow, j\rangle)}_{\in \mathcal{B}} + \sqrt{w_j} \underbrace{(|\rightarrow, j\rangle - |\leftarrow, j\rangle)}_{\in \Xi_{e_j}^{\mathcal{A}}} \\ &\quad + \underbrace{\sqrt{w_j}|\leftarrow, j\rangle + |\rightarrow, t\rangle}_{\in \mathcal{B}} - \underbrace{(|\rightarrow, t\rangle + |t\rangle)}_{\in \Xi_t^{\mathcal{A}}}, \end{aligned}$$

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

### 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  $\pi_1, \dots, \pi_T : \mathcal{Z} \rightarrow \mathcal{Z}$  that depend (implicitly) on the input;
- An initial state  $z_1 \in \mathcal{Z}$ ;
- A partitioning of  $\mathcal{Z}$  into accepting and rejecting states  $\mathcal{Z} = \mathcal{Z}_{\text{acc}} \sqcup \mathcal{Z}_{\text{rej}}$ .

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

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

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 3.4](#). It is easy to see that the connected component containing  $s$  is a line with one vertex in each of  $V_\ell$ , 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  $\pi_T \circ \dots \circ \pi_1(z_1) \in \mathcal{Z}_{\text{acc}}$ . See [Figure 3](#) for a visualization of  $G$ . We set all edge weights to be 1, and then, following [Example 3.4](#), we define boundary and edge spaces:

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

and all vertices are simple ([Definition 3.3](#)), so:

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

While the algorithm is represented by a path, as is the graph for AND in [Section 3.3.2](#), in contrast to [Section 3.3.2](#), 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 3.3.2](#), 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 3.3.2](#), 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 paper, to get a quantum walk algorithm, following [Example 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  $v_\ell$  corresponds to the set  $V_\ell$  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  $\Xi'^A_{e_\ell}$ . 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.

### 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 3.24](#) 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  $U_1, \dots, U_T$  acting on the space

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

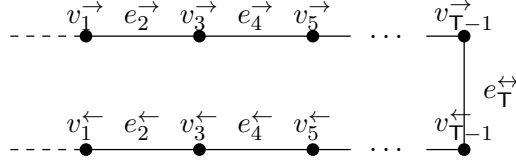


Figure 4: The graph  $G$ . The dashed lines represent dangling boundary “edges”.

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  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  if

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

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.

**Lemma 3.24.** *From any quantum algorithm computing some  $f : \{0, 1\}^n \rightarrow \{0, 1\}$  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 3.9)  $G$  that computes  $f$  with  $\dim H_G = O(|\mathcal{Z}|T)$ ,  $\hat{W}_+ \leq 2 \sum_{r=1}^T \frac{1}{\alpha_r}$  and  $\hat{W}_- \leq 2 \sum_{r=1}^T \alpha_r$ ; with *st-composable* working bases that have scaling factor  $r = 2$  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  $T = T + 1$ , and the algorithm referred to in Lemma 3.24 is actually  $U_2, \dots, U_T$ .

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

$$V = \{v_{2\ell-1}^{\rightarrow}, v_{2\ell-1}^{\leftarrow}\}_{\ell=1}^{T/2}, \quad E = \{e_{2\ell}^{\rightarrow}, e_{2\ell}^{\leftarrow}\}_{\ell=1}^{T/2-1} \cup \{e_T^{\leftrightarrow}\}, \quad \text{and} \quad B = \{s := v_1^{\rightarrow}, t := v_1^{\leftarrow}\}$$

with  $e_i^{\rightarrow} = (v_{i-1}^{\rightarrow}, v_{i+1}^{\rightarrow})$  (and similarly for  $e_i^{\leftarrow}$ ), and  $e_T^{\leftrightarrow} = (v_{T-1}^{\rightarrow}, v_{T-1}^{\leftarrow})$ . We define:

$$\begin{aligned} \forall \ell \in \{1, \dots, T/2 - 1\}, \quad \Xi_{e_{2\ell}^{\rightarrow}} &:= \text{span}\{|\rightarrow\rangle|a, z\rangle|2\ell\rangle, |\rightarrow\rangle|a, z\rangle|2\ell + 1\rangle : a \in \{0, 1\}, z \in \mathcal{Z}\} \\ \forall \ell \in \{1, \dots, T/2 - 1\}, \quad \Xi_{e_{2\ell}^{\leftarrow}} &:= \text{span}\{|\leftarrow\rangle|a, z\rangle|2\ell\rangle, |\leftarrow\rangle|a, z\rangle|2\ell + 1\rangle : a \in \{0, 1\}, z \in \mathcal{Z}\} \\ \Xi_{e_T^{\leftrightarrow}} &:= \text{span}\{|\rightarrow\rangle|a, z\rangle|T\rangle, |\leftarrow\rangle|a, z\rangle|T\rangle : a \in \{0, 1\}, z \in \mathcal{Z}\} \\ \Xi_s &= \Xi_{v_1^{\rightarrow}} := \text{span}\{|s\rangle := |\rightarrow\rangle|0, 0\rangle|0\rangle, |\leftarrow, s\rangle := |\rightarrow\rangle|0, 0\rangle|1\rangle\} \\ \Xi_t &= \Xi_{v_1^{\leftarrow}} := \text{span}\{|t\rangle := |\leftarrow\rangle|0, 0\rangle|0\rangle, |\rightarrow, t\rangle := |\leftarrow\rangle|0, 0\rangle|1\rangle\}. \end{aligned} \tag{7}$$

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  $\alpha_0 = \alpha_1 = 1$ .

**Definition 3.25.** *The forward transition states are defined:*

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

*The backward transition states are defined:*

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

*The reversal states are defined:*

$$\Psi_T^{\leftrightarrow} := \{|\psi_{a,z,T}^{\leftrightarrow}\rangle := \sqrt{\alpha_T} (|\rightarrow\rangle - (-1)^a |\leftarrow\rangle) |a, z\rangle|T\rangle : a \in \{0, 1\}, z \in \mathcal{Z}\}.$$

Finally, let:

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

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:

$$\begin{aligned} \mathcal{V}_{v_{\rightarrow}} &:= \text{span}\{|\psi_{0,0,1}^{\rightarrow}\rangle\} & \text{and} & & \mathcal{V}_{v_{\leftarrow}} &:= \text{span}\{|\psi_{0,0,1}^{\leftarrow}\rangle\} \\ \forall \ell \in \{2, \dots, \mathbb{T}/2\}, \mathcal{V}_{v_{2\ell-1}^{\rightarrow}} &:= \text{span}\Psi_{2\ell-1}^{\rightarrow} & \text{and} & & \mathcal{V}_{v_{2\ell-1}^{\leftarrow}} &:= \text{span}\Psi_{2\ell-1}^{\leftarrow} \\ \forall \ell \in \{1, \dots, \mathbb{T}/2 - 1\}, \Xi_{e_{2\ell}^{\rightarrow}}^{\mathcal{A}} &:= \text{span}\Psi_{2\ell}^{\rightarrow} & \text{and} & & \Xi_{e_{2\ell}^{\leftarrow}}^{\mathcal{A}} &:= \text{span}\Psi_{2\ell}^{\leftarrow} \\ \Xi_{e_{\mathbb{T}}^{\leftrightarrow}}^{\mathcal{A}} &:= \text{span}\Psi_{\mathbb{T}}^{\leftrightarrow} \\ \Xi_s^{\mathcal{A}} &:= \text{span}\{|\psi_{0,0,0}^{\rightarrow}\rangle\} = \text{span}\{|s\rangle + |\leftarrow, s\rangle\} \\ \Xi_t^{\mathcal{A}} &:= \text{span}\{|\psi_{0,0,0}^{\leftarrow}\rangle\} = \text{span}\{|t\rangle + |\rightarrow, t\rangle\} \\ \mathcal{V}_B &:= \text{span}\{|\leftarrow, s\rangle + |\rightarrow, t\rangle\}. \end{aligned} \tag{8}$$

For all  $e \in E \cup B$ , we will have  $\Xi_e^{\mathcal{B}} = \{0\}$ . Note that the subspace graph we have just defined has canonical  $st$ -boundary (Definition 3.6). As per Definition 3.1, we have

$$\begin{aligned} \mathcal{A}_G &= \Xi_s^{\mathcal{A}} \oplus \Xi_t^{\mathcal{A}} \oplus \bigoplus_{\ell=1}^{\mathbb{T}/2-1} (\Xi_{e_{2\ell}^{\rightarrow}}^{\mathcal{A}} \oplus \Xi_{e_{2\ell}^{\leftarrow}}^{\mathcal{A}}) \oplus \Xi_{e_{\mathbb{T}}^{\leftrightarrow}}^{\mathcal{A}} \\ \mathcal{B}_G &= \bigoplus_{\ell=1}^{\mathbb{T}/2} (\mathcal{V}_{v_{2\ell-1}^{\rightarrow}} \oplus \mathcal{V}_{v_{2\ell-1}^{\leftarrow}}) + \mathcal{V}_B. \end{aligned} \tag{9}$$

In [Jef22], the vectors in Definition 3.25 form the working basis, which works perfectly fine, because since all  $\Xi_e^{\mathcal{B}}$  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 2.9, as follows.

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

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

Note that

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

and similarly,

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

Define:

$$\begin{aligned} |b_0\rangle &:= \frac{1}{\sqrt{2}} (|\psi_{0,0,1}^{\rightarrow}\rangle + |\psi_{0,0,1}^{\leftarrow}\rangle + (|\rightarrow\rangle + |\leftarrow\rangle)U_2|0,0\rangle|2\rangle) = \frac{1}{\sqrt{2}}(|\leftarrow, s\rangle + |\rightarrow, t\rangle) \\ |b_1\rangle &:= \frac{1}{2} (|\psi_{0,0,1}^{\rightarrow}\rangle - |\psi_{0,0,1}^{\leftarrow}\rangle) = \frac{1}{2} (|\leftarrow, s\rangle - |\rightarrow, t\rangle - (|\rightarrow\rangle - |\leftarrow\rangle)U_2|0,0\rangle|2\rangle) \\ |b_2\rangle &:= \frac{1}{\sqrt{2}}(|\rightarrow\rangle + |\leftarrow\rangle)U_2|0,0\rangle|2\rangle \end{aligned}$$

Then we have

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

Thus

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

is an orthonormal basis for  $\mathcal{B}_G$ .

The following is given by a trivial modification of [Jef22, Lemma 3.3].

**Lemma 3.26.** *Assuming unit-time access to  $\sum_{\ell=1}^{\mathbb{T}} |\ell\rangle\langle\ell| \otimes U_{\ell}$ , the working bases  $\Psi_{\mathcal{A}}$  and  $\Psi_{\mathcal{B}}$  can be generated in  $O(\log(\mathbb{T}))$  complexity.*

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

**Definition 3.27** (Algorithm States). *Define the following algorithm states in  $H_G$ :*

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

The positive history state of the algorithm is defined:

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

The negative history state of the algorithm is defined:

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

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

- Claim 3.28.**
1.  $\| |w_-\rangle \|^2 = 2 \sum_{t=0}^{\mathbb{T}} \alpha_t$  (Corollary 3.7)
  2.  $\| |w_+\rangle \|^2 = 2 \sum_{t=0}^{\mathbb{T}} \frac{1}{\alpha_t}$  (Corollary 3.7)
  3.  $|w_+\rangle \in \text{span} \Psi_0^{\perp} \cap \text{span} \Psi_1^{\perp}$  (Claim 3.8).
  4.  $|w_-\rangle \in \text{span} \Psi_0$  and  $|w_-\rangle - \underbrace{(|\rightarrow\rangle - (-1)^{f(x)} |\leftarrow\rangle)}_{|s\rangle - (-1)^{f(x)} |t\rangle} |0\rangle |0\rangle \in \text{span} \Psi_1$  (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 avoid defining variable-time quantum algorithms. However, we remark that in [Jef22], the more general case of variable-time algorithms is considered, so  $\mathbb{T}$  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 3.26, prove Lemma 3.24.

**Corollary 3.29** (Positive analysis). *When  $f(x) = 1$ ,  $|w_+\rangle$  is a positive witness for  $G$  (see Definition 3.10) with*

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



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

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

By inspection (see (9) and Definition 3.25), we have  $\mathcal{A}_G \subseteq \text{span}\Psi_0$  and  $\mathcal{B}_G \subseteq \text{span}\Psi_1 \cup \{|b_0\rangle\}$ . By Claim 3.28,  $|w_+\rangle \in \text{span}\Psi_0^\perp \cap \text{span}\Psi_1^\perp \subset \mathcal{A}_G^\perp \cap \mathcal{B}_G^\perp$ , 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

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

□

**Corollary 3.30** (Negative analysis). *When  $f(x) = 0$ ,  $|w_-\rangle$  is a negative witness for  $G$  (see Definition 3.11) with*

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

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

*Proof.* By Claim 3.28, we have  $|w_+\rangle \in \text{span}\Psi_0$ . Though  $\Psi_0$  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  $\Psi_1$  (Claim 3.28), and its projection onto  $|0\rangle$  or  $|1\rangle$  in the last register is in  $\text{span}\Psi_0^\rightarrow \cup \Psi_0^\leftarrow \cup \Psi_1^\rightarrow \cup \Psi_1^\leftarrow$ . □

## 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 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 3.3.1 and Section 3.3.2 can be combined to make switching networks for any Boolean formula [JK17].

### 4.1 Switch Composition

Here we investigate how to compose subspace graphs with specific properties that generalize switching networks –  $st$ -composable subspace graphs (Definition 3.9) – into the switches of a graph  $G$ . Specifically, if  $\bar{E}$  is the set of edges of  $G$  that are switches, we will replace  $e \in \bar{E}$  with an  $st$ -composable

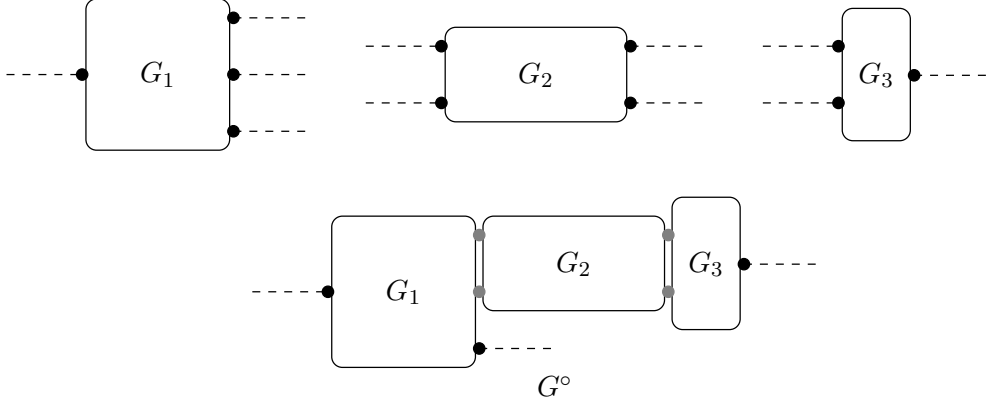


Figure 5: Some perhaps complicated graphs  $G_1$ ,  $G_2$  and  $G_3$ , but we need only consider their boundaries, and can abstract their internal structure. To compose them, we identify points on their boundaries, yielding a new graph,  $G^\circ$ .

graph  $G^e$ . We can assume without loss of generality that we replace every  $e \in \bar{E}$  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.

**Theorem 4.1.** *Let  $G$  be an  $st$ -composable subspace graph (Definition 3.9) with  $st$ -composable working bases that can be generated in time  $T$ , with scaling factor  $r$ . For each  $e \in \bar{E}$ , 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  $G^\circ$  with  $\dim H_{G^\circ} \leq \dim H_G - 2|\bar{E}| + \sum_{e \in \bar{E}} \dim H_{G^e}$  such that:*

- $G^\circ$  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 3.10), and for each  $e \in \bar{E}$  such that  $\langle \rightarrow, e | \hat{w} \rangle \neq 0$ ,  $|\hat{w}^e\rangle$  is a cropped positive witness for  $G^e$ , then

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

is a cropped positive witness for  $G^\circ$ .

- If  $|\hat{w}_A\rangle$  is a cropped negative witness for  $G$  (Definition 3.11), and for each  $e \in \bar{E}$  such that  $\langle \rightarrow, e | \hat{w}_A \rangle \neq 0$ ,  $|\hat{w}_A^e\rangle$  is a cropped negative witness for  $G^e$ , then

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

is a cropped negative witness for  $G^\circ$ .

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

Before we prove Theorem 4.1, we note the following useful corollary of Theorem 4.1 and Lemma 3.16.

**Corollary 4.2.** *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  $G^\circ$  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  $\hat{W}_-(G^\circ) \leq \hat{W}_+(G)\hat{W}_-(G)$ .*

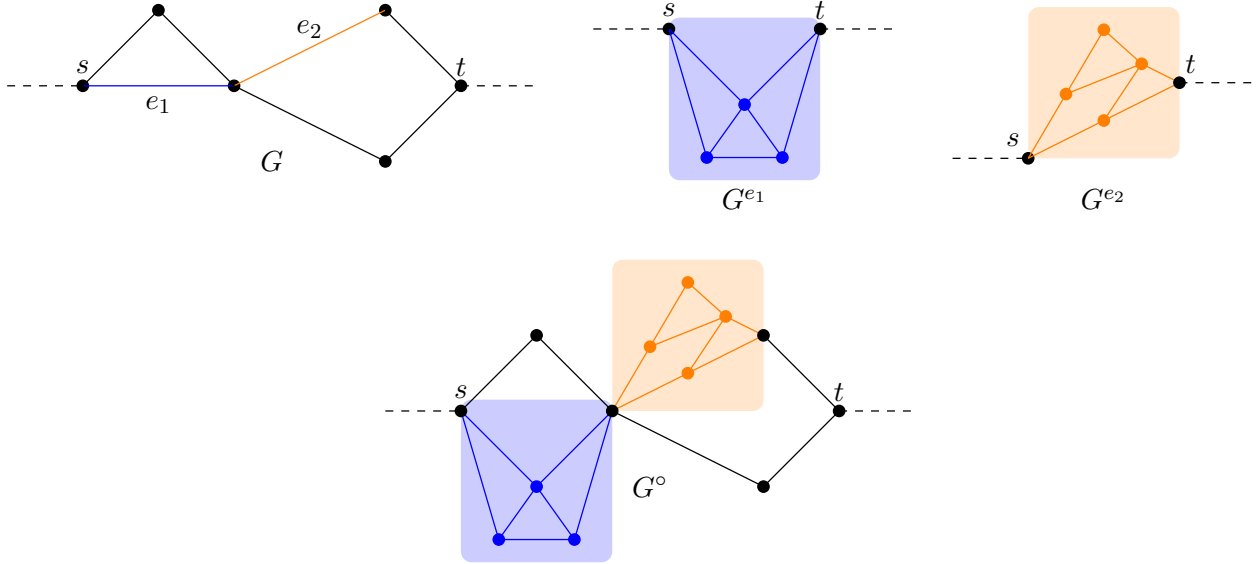


Figure 6: An example of replacing edges  $e_1$  and  $e_2$  in  $G$  with graphs  $G^{e_1}$  and  $G^{e_2}$  to obtain  $G^\circ$ . Boundary “edges” are represented by dashed lines. Switching networks already lend themselves to this type of recursion: we can replace an edge with a 2-terminal graph, and then the “edge” is traversable if and only if the terminals are connected. The difference with the more general recursion in [Theorem 4.1](#) 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 3.5](#).

*Proof.* Let  $G_{\text{OR},1}$  be the graph from [Lemma 3.16](#) with  $d = 1$ , and  $w_1 = \frac{r}{2} \hat{W}_+(G)$ , 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 4.1](#), so  $G_{\text{OR}}$  plays the role of  $G$  in [Theorem 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 3.16](#), by [Theorem 4.1](#),  $G^\circ$  has cropped positive witness

$$|\hat{w}^\circ\rangle = \sqrt{\frac{r}{2}} \frac{1}{\sqrt{w_1}} |\hat{w}^1\rangle = \frac{1}{\sqrt{\hat{W}_+(G)}} |\hat{w}^1\rangle,$$

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  $|\hat{w}_A^1\rangle$ , and using the negative witness for [Lemma 3.16](#), by [Theorem 4.1](#),  $G^\circ$  has cropped negative witness

$$|\hat{w}_A^\circ\rangle = \sqrt{\frac{2}{r}} \sqrt{w_1} |\hat{w}_A^1\rangle = \sqrt{\hat{W}_+(G)} |\hat{w}_A^1\rangle,$$

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

To complete the proof, by [Lemma 3.16](#), the basis for  $G_{\text{OR}}$  can be generated in unit time, and thus by [Theorem 4.1](#), the basis for  $G^\circ$  can be generated in time  $T + O(1)$ .  $\square$

In the remainder of this section, we prove [Theorem 4.1](#).

Informally, we obtain  $G^\circ$  from  $G$  by, for each  $e \in \bar{E}$ , 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 6](#). The subspaces associated with  $G^\circ$  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  $\tilde{\Lambda}$  defined in [\(14\)](#).

**Definition of the Subspace Graph  $G^\circ$**  We now formally define  $G^\circ$  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  $G^\circ$  has vertex and edge sets:

$$\begin{aligned} V^\circ &= V \sqcup \bigsqcup_{e \in \bar{E}} (V^e \setminus B^e) = V \sqcup \bigsqcup_{e \in \bar{E}} (V^e \setminus \{s^e, t^e\}) \\ E^\circ &= (E \setminus \bar{E}) \sqcup \bigsqcup_{e \in \bar{E}} E^e \end{aligned} \quad (10)$$

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  $u \in V^\circ$ :

$$E^\circ(u) = \begin{cases} E^e(u) & \text{if } u \in V^e \setminus \{s^e, t^e\} \\ (E(u) \setminus \bar{E}) \sqcup \bigsqcup_{e \in \bar{E} \cap E^{\rightarrow}(u)} E^e(s^e) \sqcup \bigsqcup_{e \in \bar{E} \cap E^{\leftarrow}(u)} E^e(t^e) & \text{if } u \in V. \end{cases} \quad (11)$$

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  $G^\circ$  from those of  $G$  and  $G^e$  as follows:

$$\Xi_{e'}^\circ = \begin{cases} \Xi_{e'} & \text{if } e' \in (E \setminus \bar{E}) \cup B \\ \Xi_{e'}^e & \text{if } e' \in E^e, \end{cases} \quad (12)$$

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

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

we inherit the property that for all  $e \in E^\circ \setminus \bar{E}^\circ$ ,  $\Xi_e^{\circ B} = \{0\}$ .

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

$$\begin{aligned} \mathcal{V}_{s^e}^e &= \text{span}\{|\leftarrow, s^e\rangle + |\psi_\star(s^e)\rangle\} \\ \mathcal{V}_{t^e}^e &= \text{span}\{|\rightarrow, t^e\rangle + |\psi_\star(t^e)\rangle\}, \end{aligned} \quad (13)$$

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

$$\tilde{\Lambda} = \sum_{e \in \bar{E}} \frac{1}{\sqrt{r^e}} (|\psi_\star(s^e)\rangle \langle \rightarrow, e| + |\psi_\star(t^e)\rangle \langle \leftarrow, e|) + \Pi_{E \setminus \bar{E}} + \Pi_B, \quad (14)$$

and let

$$\mathcal{V}_u^\circ = \begin{cases} \tilde{\Lambda}(\mathcal{V}_u) & \text{if } u \in V \\ \mathcal{V}_u^e & \text{if } u \in V^e \setminus \{s^e, t^e\}. \end{cases} \quad (15)$$

The following lemma shows that  $G^\circ$  is a well-defined subspace graph.

**Lemma 4.3.**  $\tilde{\Lambda}(\mathcal{V}_u) \subseteq \Xi_{E^\circ(u)}^\circ$ .

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

$$\tilde{\Lambda}(\mathcal{V}_u) = \tilde{\Lambda} \left( \Xi_{E(u) \setminus \bar{E}} \cap \mathcal{V}_u \right) \oplus \bigoplus_{e \in \bar{E} \cap E(u)} \tilde{\Lambda}(\Xi_e \cap \mathcal{V}_u),$$

We have  $\tilde{\Lambda}(\Xi_{E(u)\setminus\bar{E}} \cap \mathcal{V}_u) = \Xi_{E(u)\setminus\bar{E}} \cap \mathcal{V}_u$ , since  $\tilde{\Lambda}$  fixes  $\Xi_{E\setminus\bar{E}}$ . Recall from [Definition 3.5](#) that for  $e \in \bar{E} \cap E(u)$

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

Thus,

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

Thus

$$\tilde{\Lambda}(\mathcal{V}_u) \subseteq \Xi_{E(u)\setminus\bar{E}} \oplus \bigoplus_{e \in \bar{E} \cap E^{\rightarrow}(u)} \Xi_{E^e}^e \oplus \bigoplus_{e \in \bar{E} \cap E^{\leftarrow}(u)} \Xi_{E^e}^e = \Xi_{E^{\circ}(u)}$$

by [\(11\)](#). □

From these definitions, we conclude:

$$\begin{aligned} \mathcal{A}_{G^{\circ}} &= \bigoplus_{e \in E^{\circ}} \Xi_e^{\circ\mathcal{A}} \oplus \bigoplus_{u \in B^{\circ}} \Xi_u^{\circ\mathcal{A}} = \bigoplus_{e \in E \setminus \bar{E}} \Xi_e^{\mathcal{A}} \oplus \bigoplus_{e \in \bar{E}} \bigoplus_{e' \in E^e} \Xi_{e'}^{e\mathcal{A}} \oplus \Xi_s^{\mathcal{A}} \oplus \Xi_t^{\mathcal{A}} \\ &= \bigoplus_{e \in E \setminus \bar{E}} \Xi_e^{\mathcal{A}} \oplus \bigoplus_{e \in \bar{E}} \bigoplus_{e' \in E^e} \Xi_{e'}^{e\mathcal{A}} \oplus \Xi_s^{\mathcal{A}} \end{aligned} \quad (16)$$

and similarly

$$\begin{aligned} \mathcal{B}_{G^{\circ}} &= \bigoplus_{u \in V^{\circ}} \mathcal{V}_u^{\circ} + \bigoplus_{e \in E \setminus \bar{E}} \Xi_e^{\mathcal{B}} \oplus \bigoplus_{e \in \bar{E}} \bigoplus_{e' \in E^e} \Xi_{e'}^{e\mathcal{B}} \oplus \Xi_s^{\mathcal{B}} \oplus \Xi_t^{\mathcal{B}} \\ &= \bigoplus_{u \in V} \tilde{\Lambda}(\mathcal{V}_u) \oplus \bigoplus_{e \in \bar{E}} \bigoplus_{u \in V^e \setminus \{s, t\}} \mathcal{V}_u^e + \bigoplus_{e \in \bar{E}} \bigoplus_{e' \in E^e} \Xi_{e'}^{e\mathcal{B}}. \end{aligned} \quad (17)$$

**Working Bases for  $G^{\circ}$**  By the assumptions of [Definition 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  $e \in \bar{E}$ , 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

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

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

**Lemma 4.4** (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  $e \in \bar{E}$ ,  $\Psi_{\mathcal{A}_G}(e)$ , which is one-dimensional, has label set  $L(e) = \{e\}$ . We let

$$L_{\mathcal{A}}^{\circ} = (\{0\} \times (L \setminus \bar{E})) \sqcup \bigsqcup_{e \in \bar{E}} \{e\} \times L_{\mathcal{A}}^e,$$

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

By the assumptions of [Definition 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  $\Psi_{\mathcal{B}_{G^e}}^- \subseteq H_{G^e}^- := \Xi_{B^e \cup E^e \setminus \overline{E}^e} \oplus \text{span}\{|\rightarrow, e\rangle - |\leftarrow, e\rangle : e \in \overline{E}^e\}$  contains two distinguished vectors,  $|b_0^e\rangle$  and  $|b_1^e\rangle$ . From these, we define the following isometry on  $H_G^-$ :

$$\Lambda := \Pi_{E \setminus \overline{E}} + \Pi_B + \sum_{e \in \overline{E}} \frac{1}{\sqrt{2}} |\bar{b}_1^e\rangle (\langle \rightarrow, e| - \langle \leftarrow, e|). \quad (18)$$

Then we define:

$$\Psi_{\mathcal{B}_{G^\circ}} = \underbrace{\Lambda(\Psi_{\mathcal{B}_G}^-) \cup \bigcup_{e \in \overline{E}} (\Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\})}_{=:\Psi_{\mathcal{B}_{G^\circ}}^-} \cup \underbrace{\left\{ \frac{1}{\sqrt{2}}(|\rightarrow, e'\rangle + |\leftarrow, e'\rangle) : e' \in \overline{E}^\circ = \bigcup_{e \in \overline{E}} \overline{E}^e \right\}}_{\Psi_{\mathcal{B}_{G^\circ}}(e'), \text{ spans } \Xi_{e'}^{\mathcal{B}}} \quad (19)$$

In [Appendix 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  $\mathcal{V}_{s^e}^e$  and  $\mathcal{V}_{t^e}^e$  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.

**Lemma 4.5.**  $\Psi_{\mathcal{A}_{G^\circ}}$  and  $\Psi_{\mathcal{B}_{G^\circ}}$  are  $st$ -composable bases as in [Definition 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

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

establishing property 3 of [Definition 3.8](#). Similarly,  $\Psi_{\mathcal{B}_G}^-$  contains  $|b_1\rangle = \frac{1}{\sqrt{2+r}}(|\leftarrow, s\rangle - |\rightarrow, t\rangle + \sqrt{r}|\bar{b}_1\rangle)$ , so  $\Psi_{\mathcal{B}_{G^\circ}}^-$  contains

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

establishing property 4 of [Definition 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.  $\square$

**Lemma 4.6.** 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

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

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:

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

We complete the analysis by establishing positive and negative witnesses.

**Lemma 4.7.** *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*

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

is a positive witness for  $G^\circ$ .

*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  $\Xi_s^{\mathcal{A}} = \text{span}\{|s\rangle - |\leftarrow, s\rangle\}$  (see Definition 3.6) is obvious. For  $e \in E \setminus \bar{E}$ ,

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

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

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

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

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

Note that since  $|b_1\rangle \propto |\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle + \sqrt{r^e}|\bar{b}_1^e\rangle \in \mathcal{B}_{G^e}$ , we must have

$$\begin{aligned} \left( \langle \leftarrow, s^e| - \langle \rightarrow, t^e| + \sqrt{r^e}\langle \bar{b}_1^e| \right) \left( |s^e\rangle - |\leftarrow, s^e\rangle + |\hat{w}^e\rangle + |\rightarrow, t^e\rangle - |t^e\rangle \right) &= 0 \\ \sqrt{r^e}\langle \bar{b}_1^e|\hat{w}^e\rangle &= 2. \end{aligned}$$

Thus:

$$\begin{aligned} \Lambda^\dagger|\hat{w}^\circ\rangle &= \sum_{e \in \bar{E}} \sqrt{\frac{r^e}{2}} \langle \rightarrow, e|w\rangle \Lambda^\dagger|\hat{w}^e\rangle + \Pi_{E \setminus \bar{E}}|w\rangle \\ &= \sum_{e \in \bar{E}} \sqrt{\frac{r^e}{2}} \langle \rightarrow, e|w\rangle \langle \bar{b}_1^e|\hat{w}^e\rangle \frac{1}{\sqrt{2}} (|\rightarrow, e\rangle - |\leftarrow, e\rangle) + \Pi_{E \setminus \bar{E}}|w\rangle \\ &= \sum_{e \in \bar{E}} \sqrt{\frac{r^e}{2}} \langle \rightarrow, e|w\rangle \frac{2}{\sqrt{r^e}} \frac{1}{\sqrt{2}} (|\rightarrow, e\rangle - |\leftarrow, e\rangle) + \Pi_{E \setminus \bar{E}}|w\rangle \\ &= \sum_{e \in \bar{E}} \langle \rightarrow, e|w\rangle (|\rightarrow, e\rangle - |\leftarrow, e\rangle) + \Pi_{E \setminus \bar{E}}|w\rangle \\ &= \Pi_{\bar{E}}|w\rangle + \Pi_{E \setminus \bar{E}}|w\rangle = |\hat{w}\rangle, \end{aligned} \tag{20}$$

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  $|\psi\rangle \in \bar{\Psi}_{\mathcal{B}_G}^-$ ,

$$\langle \psi|\Lambda^\dagger|w^\circ\rangle = \langle \psi|w\rangle = 0, \tag{21}$$

since  $|w\rangle$  is a positive witness for  $G$ . To complete the proof, we show orthogonality with  $\Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\}$  for any  $e \in \bar{E}$ .  $\mathcal{B}_{G^e}$  is orthogonal to  $|s^e\rangle$  and  $|t^e\rangle$ , and furthermore, for  $i > 1$ , we have  $\langle \leftarrow, s^e | b_i^e \rangle = \langle \rightarrow, t^e | b_i^e \rangle = 0$ . We have:

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

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

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

completing the proof.  $\square$

**Lemma 4.8.** *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  $\Xi_e$  into  $|w_{\mathcal{B}}\rangle$ ). For every switch  $e \in \bar{E}$  that is off, let  $|w_{\mathcal{A}}^e\rangle$  be a negative witness for  $G^e$ . Then:*

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

is a negative witness for  $G^\circ$ .

*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  $\Xi_e^{\mathcal{A}} = \text{span}\{|\rightarrow, e\rangle - |\leftarrow, e\rangle\}$ . In particular, this means that  $\Pi_{\bar{E}} |\hat{w}_{\mathcal{A}}\rangle \in \Xi_{\bar{E}}^-$ , 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:

$$\begin{aligned} & |\hat{w}_{\mathcal{A}}\rangle + |\leftarrow, s\rangle - |\rightarrow, t\rangle \in \mathcal{B}_G^- \\ \Pi_{E \setminus \bar{E}} |\hat{w}_{\mathcal{A}}\rangle + \sum_{e \in \bar{E}} \langle \rightarrow, e | w_{\mathcal{A}} \rangle (|\rightarrow, e\rangle - |\leftarrow, e\rangle) + |\leftarrow, s\rangle - |\rightarrow, t\rangle & \in \mathcal{B}_G^- \\ \Lambda \left( \Pi_{E \setminus \bar{E}} |\hat{w}_{\mathcal{A}}\rangle + \sum_{e \in \bar{E}} \langle \rightarrow, e | w_{\mathcal{A}} \rangle (|\rightarrow, e\rangle - |\leftarrow, e\rangle) + |\leftarrow, s\rangle - |\rightarrow, t\rangle \right) & \in \mathcal{B}_{G^\circ} \\ \Pi_{E \setminus \bar{E}} |\hat{w}_{\mathcal{A}}\rangle + \sum_{e \in \bar{E}} \langle \rightarrow, e | w_{\mathcal{A}} \rangle \sqrt{2} |b_1^e\rangle + |\leftarrow, s\rangle - |\rightarrow, t\rangle & \in \mathcal{B}_{G^\circ} \\ |\hat{w}_{\mathcal{A}}^\circ\rangle + \sum_{e \in \bar{E}} \langle \rightarrow, e | w_{\mathcal{A}} \rangle \sqrt{\frac{2}{r^e}} \left( \sqrt{r^e} |b_1^e\rangle - |\hat{w}_{\mathcal{A}}^e\rangle \right) + |\leftarrow, s\rangle - |\rightarrow, t\rangle & \in \mathcal{B}_{G^\circ}. \end{aligned}$$

We complete the proof by noting that for all  $e \in \bar{E}$ ,  $\sqrt{r^e} |b_1^e\rangle - |\hat{w}_{\mathcal{A}}^e\rangle \in \mathcal{B}_{G^\circ}$ . To see this, we note:

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

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  $|b_1^e\rangle \propto |\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle + \sqrt{r^e} |b_0^e\rangle$ , from which it follows:

$$|\hat{w}_{\mathcal{A}}^e\rangle - \sqrt{r^e} |b_0^e\rangle \in \mathcal{B}_{G^e}^- \cap \text{span}\{|b_0^e\rangle, |b_1^e\rangle\}^\perp \subset \mathcal{B}_{G^\circ}. \quad \square$$

## 4.2 Boolean Formula Composition

In this section, we give a subspace graph computing  $\varphi \circ (f_\sigma)_\sigma$  when  $\varphi$  is a symmetric formula, given subspace graphs for each  $f_\sigma$ . Subspace graphs for formulas  $\varphi$  can be obtained simply by composing the switching networks for OR and AND in [Section 3.3.1](#) and [Section 3.3.2](#) (see [\[JK17\]](#)). Our composition theorem, [Theorem 4.1](#), generalizes this simple switching network composition to more general subspace graphs with switches.

We assume that  $\varphi$  is balanced (see [Definition 2.4](#)), 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 4.9](#)). The special case where  $\varphi$  is symmetric (see [Definition 2.3](#)) is much simpler, as shown in [Corollary 4.12](#).

**Definition 4.9.** Fix a Boolean formula  $\varphi$  on  $\{0, 1\}^\Sigma$ , and real values  $C_\sigma^+$  and  $C_\sigma^-$  for each  $\sigma \in \Sigma$ . Let  $C_\sigma := \sqrt{C_\sigma^+ C_\sigma^-}$ .

For  $\sigma \in \bar{\Sigma} \setminus \Sigma$ , let  $d_\sigma$  be the degree of  $\sigma$ , and  $v(\sigma) \in \{\vee, \wedge\}$  indicate the gate labelling the node  $\sigma$ , so for example, if  $v(\sigma) = \vee$ , then the sub-formula rooted at  $\sigma$  is an OR of  $d_\sigma$  sub-formulas. Define:

$$C_\sigma := \sum_{i=1}^{d_\sigma} C_{\sigma_i}, \quad C_\sigma^+ = \begin{cases} \sum_{i=1}^{d_\sigma} C_{\sigma_i}^+ & \text{if } v(\sigma) = \vee \\ \frac{C_\sigma^2}{\sum_{i=1}^{d_\sigma} C_{\sigma_i}^+} & \text{if } v(\sigma) = \wedge, \end{cases} \quad \text{and} \quad C_\sigma^- = \begin{cases} \frac{C_\sigma^2}{\sum_{i=1}^{d_\sigma} C_{\sigma_i}^-} & \text{if } v(\sigma) = \vee \\ \sum_{i=1}^{d_\sigma} C_{\sigma_i}^- & \text{if } v(\sigma) = \wedge. \end{cases}$$

Then note that

$$C_\sigma = \sqrt{C_\sigma^+ C_\sigma^-}, \quad \text{and also} \quad C = C_\emptyset = \sqrt{\sum_{\sigma \in \Sigma} C_\sigma^2}.$$

We say the values  $\{C_\sigma^+, C_\sigma^-\}_{\sigma \in \Sigma}$  incur logarithmic reflection cost if for every  $\sigma \in \bar{\Sigma}$  such that  $v(\sigma) = \vee$ , a state proportional to  $\sum_{i \in [d_\sigma]} \sqrt{C_{\sigma_i}^+} |i\rangle$  can be generated in complexity  $O(\log d_\sigma)$ , and for all  $\sigma \in \bar{\Sigma}$  such that  $v(\sigma) = \wedge$ , a state proportional to  $\sum_{i \in [d_\sigma]} \sqrt{C_{\sigma_i}^-} |i\rangle$  can be generated in complexity  $O(\log d_\sigma)$ .

**Lemma 4.10.** Let  $\varphi$  be a balanced formula (Definition 2.4) on  $\{0, 1\}^\Sigma$ . Let  $\{f_\sigma\}_{\sigma \in \Sigma}$  be Boolean functions, and for each  $\sigma \in \Sigma$ , let  $G_\sigma$  be an  $st$ -composable subspace graph (Definition 3.9) computing  $f_\sigma$ , with working bases that can be generated in time at most  $T$  with scaling factor  $r^\sigma$ , and  $\log \dim H_{G_\sigma} \leq S$ . Suppose that for each  $\sigma \in \Sigma$ ,  $C_\sigma^+$  is a known upper bound on  $r^\sigma \hat{W}_+(G_\sigma)$ , and  $C_\sigma^-$  is a known upper bound on  $\hat{W}_-(G_\sigma)/r^\sigma$ , so  $C_\sigma := \sqrt{C_\sigma^+ C_\sigma^-}$  is a known upper bound on  $\hat{C}(G_\sigma)$ . Suppose the values  $\{C_\sigma^+, C_\sigma^-\}_{\sigma \in \Sigma}$  incur logarithmic reflection cost (see Definition 4.9).

Then there is an  $st$ -composable subspace graph  $G^\circ$  computing  $\varphi \circ (f_\sigma)_{\sigma \in \Sigma}$  with  $\log \dim H_{G^\circ} = S + O(\log |\Sigma|)$  and

$$\hat{C}(G^\circ)^2 \leq \sum_{\sigma \in \Sigma} C_\sigma^2.$$

Furthermore, the working bases of  $G^\circ$  can be generated in time  $T + O(\log |\Sigma|)$ .

*Proof of Lemma 4.10.* Let  $c$  be a constant such that for each node in  $\varphi$ , 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  $\varphi$  is balanced), and let  $a$  be a constant such that the bases referred to in Lemma 3.16 and Lemma 3.20 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  $G^\circ$  in Theorem 4.1 is  $T + T' + a'$ . We will show by induction on the depth  $D$  of  $\varphi$  that there exists an  $st$ -composable subspace graph  $G^\circ$  computing  $\varphi \circ (f_\sigma)_\sigma$  with composable bases with scaling factor  $r^\circ$  such that:

1.  $r^\circ \hat{W}_+(G^\circ) \leq C^+$
2.  $\frac{\hat{W}_-(G^\circ)}{r^\circ} \leq C^-$
3.  $\log \dim H_{G^\circ} \leq \log |\Sigma| + (2 + \log c)D + S$
4. the bases can be generated in cost at most  $T + (a + a') \log(c^D |\Sigma|)$ .

Since  $\varphi$  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  $\varphi$  is negated, because if it is, we can just push this negation into  $f_\sigma$ . The proof will be by induction on the depth of  $\varphi$ .

**Base Case:** If  $D = 0$ , then  $\varphi \circ (f_\sigma)_{\sigma \in \Sigma} = f_\sigma$  for some  $\sigma$ . Then we just take  $G^\circ = G_\sigma$ , so  $r^\circ = r^\sigma$ . We have, by assumption, that  $r^\sigma \hat{W}_+(G_\sigma) \leq C_\sigma^+ = C^+$  (when the depth is 0) and  $\hat{W}_-(G_\sigma)/r^\sigma \leq C_\sigma^- = C^-$ . We also have, by assumption, that the working bases of  $G_\sigma$  can be generated in time  $T$ , and  $\log \dim H_{G_\sigma} \leq S$ , completing the base case.

**OR Case:** First, suppose  $\varphi$  has depth  $D > 0$ , with  $\varphi = \bigvee_{i=1}^d \varphi_i$  for some  $d$  and some formulas  $\varphi_i$  of depth  $D - 1$ . Let  $G = G_{\text{OR},d}$  be the subspace graph from [Lemma 3.16](#), which is *st*-composable, because it is a switching network, and let  $G_{\varphi_1}, \dots, G_{\varphi_d}$  be the *st*-composable subspace graphs whose existence is promised by the induction hypothesis. For all  $i \in [d]$ , we will compose the graphs  $G_{\varphi_i}$  into the switch edge  $e_i$  of  $G$  using [Theorem 4.1](#), to obtain  $G^\circ$ . If  $\varphi_i(x) = 1$  for some  $i$ , then  $G_{\varphi_i}$  must have some cropped positive witness  $|\hat{w}^i\rangle$ , and using the cropped positive witness  $|\hat{w}\rangle$  from [Lemma 3.16](#), by [Theorem 4.1](#),  $G^\circ$  has a cropped positive witness

$$|\hat{w}^\circ\rangle = \sqrt{\frac{r^i}{2}} \frac{1}{\sqrt{w_i}} |\hat{w}^i\rangle,$$

where  $r^i$  is the scaling factor of the basis for  $G_{\varphi_i}$ . Thus, if we choose

$$w_i = C_i^+ / 2, \quad \text{so} \quad r^\circ = 2 \sum_{i=1}^d w_i = \sum_{i=1}^d C_i^+$$

we get

$$\hat{W}_+(G^\circ) \leq \frac{r^i}{2w_i} \hat{W}_+(G_{\varphi_i}) \leq \frac{C_i^+}{C_i^+} = 1, \quad \text{and so} \quad r^\circ \hat{W}_+(G^\circ) \leq \sum_{i=1}^d C_i^+ = C^+$$

using the induction hypothesis, which says that  $r^i \hat{W}_+(G_{\varphi_i}) \leq C_i^+$ .

On the other hand, suppose for all  $i \in [d]$ ,  $\varphi_i(x) = 0$  and so there exists a cropped negative witness  $|\hat{w}_{\mathcal{A}}^i\rangle$  for  $G_{\varphi_i}$ . Then using the cropped negative witness from [Lemma 3.16](#), by [Theorem 4.1](#), we have a cropped negative witness for  $G^\circ$ :

$$|\hat{w}_{\mathcal{A}}^\circ\rangle = \sum_{i=1}^d \sqrt{\frac{2}{r^i}} \sqrt{w_i} |\hat{w}_{\mathcal{A}}^i\rangle.$$

Thus:

$$\hat{W}_-(G^\circ) \leq \sum_{i=1}^d \frac{2w_i}{r^i} \hat{W}_-(G_{\varphi_i}) \leq \sum_{i=1}^d C_i^+ C_i^- = \sum_{i=1}^d C_i^2 = C^2,$$

by the induction hypothesis, which says that  $\frac{\hat{W}_-(G_{\varphi_i})}{r^i} \leq C_i^-$ .

By the induction hypothesis, the cost of generating the bases for any  $G_\varphi$  is at most:

$$T' := (a + a') \log(c^{D-1} c |\Sigma| / d) = (a + a') \log(c^D |\Sigma| / d).$$

Thus, by [Theorem 4.1](#), since the bases of  $G$  can be generated in cost at most  $T = a \log d$  – assuming the state proportional to  $\sum_i \sqrt{w_i} |i\rangle \propto \sum_i \sqrt{C_i^+} |i\rangle$  can be generated in  $O(\log d)$  time – we can generate the bases of  $G^\circ$  in cost

$$T + T' + a' = (a + a') \log(c^{D-1} (c |\Sigma| / d)) + a \log d + a' \leq (a + a') \log(c^D |\Sigma|),$$

as needed.

Similarly, by [Theorem 4.1](#), we have

$$\begin{aligned} \dim H_{G^\circ} &\leq \dim H_G - 2d + \sum_{i=1}^d \dim H_{G_{\varphi_i}} \\ &\leq 2d + 4 - 2d + d 2^{\log(|\Sigma_i|) + (2 + \log c)(D-1) + S} \leq 4d 2^{\log(|\Sigma_i|) + (2 + \log c)(D-1) + S} \end{aligned}$$

by the induction hypothesis and [Lemma 3.16](#). Thus,

$$\log \dim H_{G^\circ} \leq 2 + \log d + \log \frac{c |\Sigma|}{d} + (2 + \log c)(D - 1) + S = \log |\Sigma| + (2 + \log c)D + S.$$

**AND Case:** Next, suppose  $\varphi = \bigwedge_{i=1}^d \varphi_i$  for some formulas of depth  $D - 1$ . Let  $G = G_{\text{AND},d}$  be the graph from Lemma 3.20, 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  $G_{\varphi_i}$  into the switch edge  $e_i$  of  $G$  to obtain  $G^\circ$ . If  $\varphi_i(x) = 1$  for all  $i$ , then there exist cropped positive witnesses  $|\hat{w}^i\rangle$  for each  $G_{\varphi_i}$ , so using the cropped positive witness for  $G$  from Lemma 3.20, by Theorem 4.1,  $G^\circ$  has cropped positive witness

$$|\hat{w}^\circ\rangle = \sum_{i=1}^d \sqrt{\frac{r^i}{2}} \frac{1}{\sqrt{w_i}} |\hat{w}^i\rangle.$$

Thus, setting

$$w_i = \frac{1}{2C_i^-}, \quad \text{so} \quad \frac{1}{r^\circ} = \sum_{i=1}^d \frac{1}{2w_i} = \sum_{i=1}^d C_i^-,$$

we have:

$$\hat{W}_+(G^\circ) \leq \sum_{i=1}^d \frac{r^i}{2w_i} \hat{W}_+(G_{\varphi_i}) \leq \sum_{i=1}^d C_i^- C_i^+ = \sum_{i=1}^d C_i^2 = C^2,$$

by the induction hypothesis, which says that  $r^i \hat{W}_+(G_{\varphi_i}) \leq C_i^+$ .

On the other hand, suppose  $\varphi_i(x) = 0$  for some  $i \in [d]$ , so there exists a cropped negative witness  $|\hat{w}_{\mathcal{A}}^i\rangle$  for  $G_{\varphi_i}$ . Then using the cropped negative witness from Lemma 3.20, by Theorem 4.1, we have a cropped negative witness for  $G^\circ$ :

$$|\hat{w}_{\mathcal{A}}^\circ\rangle = \sqrt{\frac{2}{r^i}} \sqrt{w_i} |\hat{w}_{\mathcal{A}}^i\rangle.$$

Thus, we have:

$$\hat{W}_-(G^\circ) \leq \frac{2w_i}{r^i} \hat{W}_-(G_{\varphi_i}) \leq \frac{C_i^-}{C_i^-} = 1, \quad \text{and so} \quad \frac{\hat{W}_-(G^\circ)}{r^\circ} \leq \sum_{i=1}^d C_i^- = C^-,$$

using the induction hypothesis, which says that  $\frac{\hat{W}_-(G_{\varphi_i})}{r^i} \leq C_i^-$ .

A similar argument about the basis efficiency and dimension of  $H_{G^\circ}$  as in the OR case completes the proof.  $\square$

We will prove a more easily-applied corollary for the case where  $\varphi$  is symmetric. We will use the following two lemmas.

**Lemma 4.11.** *Let  $\varphi$  be a symmetric formula (see Definition 2.3) 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  $\sigma \in \Sigma$ ,  $C_\sigma^- = L^-$  is a known upper bound on  $\hat{W}_-(G_\sigma)/r^\sigma$ , independent of  $\sigma$ ; and  $C_\sigma^+$  is a known upper bound on  $\hat{W}_+(G_\sigma)\hat{W}_-(G_\sigma)$ . Let  $\{C_\sigma^+, C_\sigma^-, C_\sigma\}_{\sigma \in \bar{\Sigma}}$  be as in Definition 4.9. Then for all  $\sigma \in \bar{\Sigma} \setminus \Sigma$  whose subtrees have depth  $D'$ ,*

$$C_\sigma^+ = \frac{C_\sigma^2}{L^- \prod_{j=1}^{\lceil D'/2 \rceil} d_{2j-1}} \quad \text{and} \quad C_\sigma^- = L^- \prod_{j=1}^{\lfloor D'/2 \rfloor} d_{2j-1}$$

*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  $C_\sigma = \sqrt{C_\sigma^+ C_\sigma^-}$ , it is sufficient to prove the statement about  $C_\sigma^-$ .

**Base Case:** Let  $\sigma \in \Sigma$ , so  $D' = 0$ . Then, indeed,  $C_\sigma^- = L^- = L^- \prod_{j=1}^0 d_{2j-1}$ .

**Odd Case:** Let  $D' > 0$  be odd. By the induction hypothesis, and since  $D' - 1$  is even, for all  $i \in [d_{D'}]$

$$C_{\sigma i}^- = L^{-} \prod_{j=1}^{(D'-1)/2} d_{2j-1}.$$

Since  $D'$  is odd,  $v(\sigma) = \wedge$ , and so

$$C_{\sigma}^- = \sum_{i=1}^{d_{D'}} C_{\sigma i}^- = d_{D'} L^{-} \prod_{j=1}^{(D'-1)/2} d_{2j-1} = L^{-} \prod_{j=1}^{\lceil D'/2 \rceil} d_{2j-1}.$$

**Even Case:** Let  $D' > 0$  be even. By the induction hypothesis, for all  $i \in [d_{D'}]$

$$C_{\sigma i}^+ = \frac{C_{\sigma i}^2}{C_{\sigma i}^-} = \frac{C_{\sigma i}^2}{L^{-} \prod_{j=1}^{\lceil (D'-1)/2 \rceil} d_{2j-1}} = \frac{C_{\sigma i}^2}{L^{-} \prod_{j=1}^{\lceil D'/2 \rceil} d_{2j-1}}.$$

Since  $D'$  is even,  $v(\sigma) = \vee$ , and so

$$C_{\sigma}^+ = \sum_{i=1}^{d_{D'}} C_{\sigma i}^+ = \frac{\sum_{i=1}^{d_{D'}} C_{\sigma i}^2}{L^{-} \prod_{j=1}^{\lceil D'/2 \rceil} d_{2j-1}} = \frac{C_{\sigma}^2}{L^{-} \prod_{j=1}^{\lceil D'/2 \rceil} d_{2j-1}}.$$

Thus

$$C_{\sigma}^- = \frac{C_{\sigma}^2}{C_{\sigma}^+} = L^{-} \prod_{j=1}^{\lceil D'/2 \rceil} d_{2j-1}. \quad \square$$

**Corollary 4.12.** Let  $\varphi$  be a symmetric formula (Definition 2.3) on  $\{0, 1\}^{\Sigma}$ . Let  $\{f_{\sigma}\}_{\sigma \in \Sigma}$  be Boolean functions, and for each  $\sigma \in \Sigma$ , let  $G_{\sigma}$  be an st-composable subspace graph (Definition 3.9) computing  $f_{\sigma}$ , with working bases that can be generated in time at most  $T$  with scaling factor  $r^{\sigma}$ , and  $\log \dim H_{G_{\sigma}} \leq S$ . Suppose that for each  $\sigma \in \Sigma$ ,  $L^{-}$  is a known upper bound on  $\hat{W}_{-}(G_{\sigma})/r^{\sigma}$ , and  $C_{\sigma}^+$  is a known upper bound on  $r^{\sigma} \hat{W}_{+}(G_{\sigma})$ . Let  $C_{\sigma} = \sqrt{C_{\sigma}^+ C_{\sigma}^-}$ , which is then a known upper bound on  $\hat{C}(G_{\sigma})$ . For any  $\sigma \in \bar{\Sigma} \setminus \Sigma$  (i.e. proper prefixes of the strings in  $\Sigma$ ), if the children of  $\sigma$  in  $\varphi$  are labelled by  $[d]$ , define

$$C_{\sigma} = \sqrt{\sum_{i \in [d]} C_{\sigma i}^2},$$

and suppose a state proportional to  $\sum_{i \in [d]} C_{\sigma i} |i\rangle$  can be generating in cost  $O(\log d)$ . Then there is an st-composable subspace graph  $G^{\circ}$  computing  $\varphi \circ (f_{\sigma})_{\sigma \in \Sigma}$  with  $\log \dim H_{G^{\circ}} = S + O(\log |\Sigma|)$  and

$$r^{\circ} \hat{W}_{+}(G^{\circ}) \leq \frac{\sum_{\sigma \in \Sigma} C_{\sigma}^2}{L^{-} \prod_{j=1}^{\lceil D'/2 \rceil} d_{2j-1}} \quad \text{and} \quad \hat{W}_{-}(G^{\circ})/r^{\circ} \leq L^{-} \prod_{j=1}^{\lceil D'/2 \rceil} d_{2j-1}$$

so

$$\hat{C}(G^{\circ})^2 \leq \sum_{\sigma \in \Sigma} C_{\sigma}^2.$$

Furthermore, the working bases of  $G^{\circ}$  can be generated in time  $T + O(\log |\Sigma|)$ .

*Proof.* By Lemma 4.11, since we can generate  $\sum_{i \in [d]} C_{\sigma i} |i\rangle$  in  $\log d$  complexity, our upper bounds  $C_{\sigma}^{\pm}$  incur logarithmic reflection cost, so we can apply Lemma 4.10.  $\square$



### 4.3 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<sup>+</sup>22]. Note that it is fairly natural to assume  $\varphi = \varphi'(z_1, \dots, z_a) \vee z_{a+1}$  for some formula  $\varphi'$ , 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 4.10 instead of Corollary 4.12, a similar statement, possibly with extra time overhead, holds for any balanced formula  $\varphi$ , as long as we can account for the possibly more complicated reflection costs.

**Theorem 4.13.** *Fix a function family  $f_{\ell,n} : D_{\ell,n} \rightarrow \{0,1\}$ . Fix unit-time-computable functions  $\lambda_1, \lambda_2 : \mathbb{N} \rightarrow \mathbb{N}$ , and a formula  $\varphi$  on  $\{0,1\}^{a+1}$  such that  $\varphi = \varphi'(z_1, \dots, z_a) \vee z_{a+1}$  for some symmetric formula  $\varphi'$ . 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  $f_{\text{aux},\ell,n} : D_{\ell,n} \rightarrow \{0,1\}$  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  $\tilde{O}(T(\ell,n))$ , 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  $\ell \leq \ell_0$ ,  $T(\ell,n) = 2T_{\text{aux}}(\ell,n)$ .

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  $\varphi'$ , 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  $\ell \leq \ell_0$ . For all  $(\ell,n)$ , let  $C^-(\ell,n)$  be defined recursively, as follows.

$$C^-(\ell,n) := \begin{cases} T_{\text{aux},0} & \text{if } \ell \leq \ell_0 \\ \frac{T(\ell,n)^2}{\bar{d}C^-(\lambda(\ell,n))^2 + 4T_{\text{aux}}(\ell,n)} & \text{else.} \end{cases}$$

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.  $\dim H_G \leq 2^{S(\ell,n)}$ ;
2.  $\hat{W}_-(G_{\ell,n})/r_{\ell,n} \leq C^-(\ell,n)$ , and  $\hat{C}(G_{\ell,n}) \leq T(\ell,n) =: C(\ell,n)$ ;
3.  $\log \dim H_{G_{\ell,n}} = S_{\ell,n} \leq cD \log(2a) + cS_{\text{aux}}(\ell,n) + c \log T_{\text{aux}}(\ell,n)$ ;
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  $\ell \leq \ell_0$ . Then we let  $G_{\ell,n}$  be the subspace graph from [Lemma 3.24](#) 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  $\dim H_{G_{\ell,n}} \leq c(S_{\text{aux}}(\ell, n) + \log T_{\text{aux}}(\ell, n))$ , and basis generation cost at most  $c \log T_{\text{aux}}(\ell, n)$ , for sufficiently large constant  $c$ . By [Lemma 3.24](#), we also have

$$\hat{W}_-(G_{\ell,n})/r_{\ell,n} \leq 2T_{\text{aux}}(\ell, n)/2 \leq T_{\text{aux},0}, \quad \text{and} \quad \hat{C}(G_{\ell,n}) \leq 2T_{\text{aux}}(\ell, n).$$

**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 4.12](#), where  $G_1, \dots, G_a$  are each copies of  $G_{\lambda(\ell,n)}$  – so  $\mathbf{C}_i := \mathbf{C}(\lambda(\ell, n)) = T(\lambda(\ell, n))$  and  $\mathbf{L}^- := \mathbf{C}^-(\lambda(\ell, n))$  are known upper bounds on  $\hat{C}(G_i)$  and  $\hat{W}_-(G_i)/r^i$ , by the induction hypothesis – to get a subspace graph  $G_{\ell,n}^{\varphi'}$ . In order to apply [Corollary 4.12](#), we have used the fact that  $\mathbf{L}^-$  does not depend on  $i$ . By [Corollary 4.12](#), we have

$$\begin{aligned} \hat{W}_-(G_{\ell,n}^{\varphi'})/r &\leq \mathbf{L}^- \bar{d} = \mathbf{C}^-(\lambda(\ell, n)) \bar{d} =: \mathbf{C}_0^-(\ell, n) \\ \text{and} \quad \hat{C}(G_{\ell,n}^{\varphi'})^2 &\leq \sum_{i=1}^a \mathbf{C}_i^2 = aT(\lambda(\ell, n))^2 =: \mathbf{C}_0(\ell, n)^2. \end{aligned}$$

Next, we will use [Lemma 4.10](#), with the formula  $x_0 \vee x_1$ , and subspace graphs  $G'_0 = G_{\ell,n}^{\varphi'}$  and  $G'_1$  a subspace graph computing  $f_{\text{aux},\ell,n}$  from [Lemma 3.24](#), using the known upper bounds  $\mathbf{C}_0(\ell, n)$  and  $\mathbf{C}_0^-(\ell, n)$  defined above, and  $\mathbf{C}_1^-(\ell, n) := T_{\text{aux}}(\ell, n) \geq \hat{W}_-(G'_1)r'_1$  and  $\mathbf{C}_1(\ell, n) := 2T_{\text{aux}}(\ell, n) \geq \hat{C}(G'_1)$  from [Lemma 3.24](#). This gives a subspace graph  $G_{\ell,n}$  computing  $f_{\ell,n}$  with

$$\begin{aligned} \hat{C}(G_{\ell,n})^2 &\leq \mathbf{C}_0(\ell, n)^2 + \mathbf{C}_1(\ell, n)^2 \leq aT(\lambda(\ell, n))^2 + 4T_{\text{aux}}(\ell, n)^2 = T(\ell, n)^2 \\ \hat{W}_-(G_{\ell,n})/r_{\ell,n} &\leq \frac{\mathbf{C}_0(\ell, n)^2 + \mathbf{C}_1(\ell, n)^2}{\frac{\mathbf{C}_0^-(\ell, n)^2}{\mathbf{C}_0^-(\ell, n)} + \frac{\mathbf{C}_1(\ell, n)^2}{\mathbf{C}_1^-(\ell, n)}} \leq \frac{T(\ell, n)^2}{\frac{aT(\lambda(\ell, n))}{d\mathbf{C}^-(\lambda(\ell, n))} + \frac{4T_{\text{aux}}(\ell, n)^2}{T_{\text{aux}}(\ell, n)}} = \mathbf{C}^-(\ell, n). \end{aligned}$$

In order to apply [Lemma 4.10](#), we must be able to generate a state proportional to

$$\sqrt{\frac{\mathbf{C}_0(\ell, n)^2}{\mathbf{C}_0^-(\ell, n)}}|0\rangle + \sqrt{\frac{\mathbf{C}_1(\ell, n)^2}{\mathbf{C}_1^-(\ell, n)}}|1\rangle$$

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  $G_1, \dots, G_a$  is at most

$$L_{\lambda(\ell,n)} = (D-1)c' \log(2a) + c \log T_{\text{aux}}(\ell, n)$$

by the induction hypothesis, and so by [Corollary 4.12](#), the cost to generate the bases for  $G_{\ell,n}^{\varphi'}$  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 3.24](#). Then by [Lemma 4.10](#) 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 4.12](#),

$$\log \dim H_{G_{\ell,n}^{\varphi'}} \leq c(D-1) \log(2a) + cS_{\text{aux}}(\ell, n) + c \log T_{\text{aux}}(\ell, n) + c \log(a).$$

By [Lemma 3.24](#),

$$\log \dim H_{G'_1} \leq cS_{\text{aux}}(\ell, n) + c \log T_{\text{aux}}(\ell, n).$$

Thus, by [Lemma 4.10](#), 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 4.2](#) to get a subspace graph  $G'_{\ell, n}$  with  $\hat{W}_+ \leq 1$ , and then apply [Theorem 3.13](#), 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  $S_{\ell, n} = O(S_{\text{aux}}(\ell, n) + \log T(\ell, n))$ .  $\square$

#### 4.4 Switching Networks with Subroutines

The following is similar to [Lemma 4.10](#), except that we generalize to any switching network, not just those arising from Boolean formulas (series-parallel switching networks), and we specialize to the case where the composed subspace graphs are of the form in [Section 3.5](#) (this specialization is not necessary, but it is not clear what we gain from the more general result).

**Lemma 4.14.** *Let  $G$  be any switching network computing a Boolean function  $f$ , with working bases that can be generated in time  $T_{\text{basis}}$ . 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  $T_{\text{max}}$  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  $G^\circ$  computing  $f \circ (f_e)_{e \in E}$  with complexities  $\hat{W}_+(G^\circ) = O(\max_{x \in f^{-1}(1)} \mathcal{R}_{s,t}(G(x)) \log T_{\text{max}})$  and  $\hat{W}_-(G^\circ) = O(\max_{x \in f^{-1}(0)} \sum_{e \in F_x} w_e T_e^2)$ , and its bases can be generated in time  $T_{\text{basis}} + O(\log T_{\text{max}})$ .*

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  $G^\circ$ , 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 3.24](#) derived from the algorithm  $\{U_r^e\}_r$ , using weights  $\alpha_r = r + 1$ . Thus  $G^e$  computes  $f_e$ . We will apply [Theorem 4.1](#) 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 4.1](#), there is a cropped positive witness for  $G^\circ$ :

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

since  $\bar{E} = E$ , where  $r^e = 2$  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 [\(4\)](#), so  $\langle \leftarrow, e | \hat{w} \rangle = \frac{\theta(e)}{\sqrt{w_e}}$ , so we have:

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

and

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

Because we have set  $\alpha_r = r + 1$ , by [Lemma 3.24](#), we have

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

so

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

**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  $|\hat{w}_{\mathcal{A}}^e\rangle$  be a negative witness for  $G^e$ . By [Theorem 4.1](#), there is a cropped negative witness for  $G^\circ$ :

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

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

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

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

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

**Basis Generation:** By [Theorem 4.1](#), since the bases of  $G^e$  can be generated in time  $O(\log T_e)$ , the bases of  $G^\circ$  can be generated in time  $T_{\text{basis}} + O(\log T_{\max})$ , completing the proof.  $\square$

Then the main theorem of this section follows directly.

**Theorem 4.15.** *Let  $G$  be any switching network computing a Boolean function  $f$ , with working bases that can be generated in time  $T_{\text{basis}}$ . 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 bounded error quantum algorithm for  $f$  with complexity*

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

*Proof.* Let  $G^\circ$  be the subspace graph from [Lemma 4.14](#). By [Corollary 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 3.13](#) to  $G'$  to get a bounded error quantum algorithm for  $f \circ (f_e)_e$  with complexity:

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

## 5 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 5.3](#) 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  $\text{poly}(n)$  time and  $O(\log n)$  space, and  $\text{poly}(n)$  factors are suppressed in  $2^{O(\log n)}$ .

**Problem 5.1** (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  $\text{PATH}_\ell(\mathcal{O}_G, u, v)$  that will be called recursively in the algorithm. This subroutine decides whether there is a path of length at most  $\ell$  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  $t$ :

<b>Subroutine 1:</b> $\text{PATH}_\ell(\mathcal{O}_G, u, v)$ for $\ell \geq 2$
<b>Input:</b> Oracle $\mathcal{O}_G$ , $u, v \in V$
<b>Output:</b> 1 if there is a path of length $\leq \ell$ from $u$ to $v$ in $G$ ; 0 otherwise.
<b>for</b> $w \in V$ <b>do</b>
$b_w := \text{PATH}(\ell/2, u, w) \wedge \text{PATH}(\ell/2, w, v)$ ;
<b>end</b>
<b>return</b> $\bigvee_{w \in V} b_w$ ;

Next, we describe the base case. When  $\ell = 1$ , the subroutine simply performs a single query  $\mathcal{O}_G(u, v)$  and outputs:

$$\text{PATH}_1(\mathcal{O}_G, u, v) = \begin{cases} 1 & \text{if } (u, v) \in E \\ 0 & \text{otherwise.} \end{cases}$$

Then Savitch's algorithm simply outputs  $\text{PATH}_n(\mathcal{O}_G, s, t)$ , where  $n = |V|$ , to decide if there is a path from  $s$  to  $t$ .

<b>Algorithm 2:</b> Savitch's algorithm for DSTCON
<b>Input:</b> Oracle $\mathcal{O}_G$ for $G = (V, E)$ with $ V  = n$ , $s, t \in V$
<b>Output:</b> 1 if there is a path from $s$ to $t$ in $G$ ; 0 otherwise.
<b>return</b> $\text{PATH}_n(\mathcal{O}_G, s, t)$ ;

The following result of Savitch is easy to verify.

**Theorem 5.2.** *Algorithm 2 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 4.13](#).

**Theorem 5.3.** *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  $\tilde{O}((\sqrt{2n})^{\log n}) = 2^{\frac{1}{2} \log^2 n + O(\log n)}$  and space  $O(\log^2 n)$ .*

*Proof.* To show existence of such a quantum algorithm, we will rephrase algorithm 2 in terms of the condition of [Theorem 4.13](#) and analyze its complexity. We start with defining a function corresponding to DSTCON.

$$f_{\ell,n} : \{0,1\}^{n^2} \times \{0,1\}^{\log n} \times \{0,1\}^{\log n} \rightarrow \{0,1\}$$

$$(G, u, v) \mapsto \begin{cases} 1 & \text{if there is a path from } u \text{ to } v \text{ in } G \text{ of length } \leq \ell \\ 0 & \text{otherwise,} \end{cases}$$

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  $\mathcal{P}_{aux,\ell,n}$  is a single edge query when  $\ell = 1$  and does nothing otherwise. This implies  $T_{aux}(\ell, n) = O(1)$  and  $S_{aux}(\ell, n) = O(\log n)$  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  $\varphi'$  in  $a = 2n$  variables:

$$f_{\ell,n}(G, u, v) = \underbrace{\bigvee_{w \in V} (f_{\ell/2,n}(G, u, w) \wedge f_{\ell/2,n}(G, w, v))}_{\varphi'} \vee f_{aux,\ell,n}(G, u, v).$$

Now we apply [Theorem 4.13](#) to everything described above and conclude that there is a recursive bounded error quantum algorithm that computes  $f_{\ell,n}$  with time complexity  $\tilde{O}(T(\ell, n))$  and space complexity  $O(S_{aux}(\ell, n) + \log T(\ell, n))$ , 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

$$T(n, n) = O((\sqrt{2n})^{\log n}) = O\left(2^{\frac{1}{2}(\log^2 n + \log n)}\right).$$

The space complexity of the algorithm for  $f_{n,n}$  is thus

$$O(S_{aux}(n, n) + \log T(n, n)) = O(\log^2 n).$$

Since  $f_{n,n}$  precisely describes the problem DSTCON, this concludes the proof.  $\square$

## 6 Acknowledgements

We thank Simon Apers for useful discussions about this work, and Troy Lee for useful comments on an early draft.

This work is supported by NWO Klein project number OCENW.Klein.061, and ARO contract no W911NF2010327. This work is funded/co-funded by the European Union (ERC, ASC-Q, 101040624). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them. SJ is a CIFAR Fellow in the Quantum Information Science Program.

This publication is part of the project Divide & Quantum (with project number 1389.20.241) of the research programme NWA-ORC which is (partly) financed by the Dutch Research Council (NWO).

This work is supported by the Dutch National Growth Fund (NGF), as part of the Quantum Delta NL programme.



## References

- [ABB<sup>+</sup>23] Jonathan Allcock, Jinge Bao, Aleksandrs Belovs, Troy Lee, and Miklos Santha. On the quantum time complexity of divide and conquer. arXiv: [2311.16401](#), 2023. 4
- [AJPW22] Simon Apers, Stacey Jeffery, Galina Pass, and Michael Walter. Quantum walks for USTCON. In *Proceedings of the 31st Annual European Symposium on Algorithms (ESA)*, pages 10:1–10:17, 2022. arXiv: [2212.00094](#) 4
- [Amb10] Andris Ambainis. Quantum search with variable times. *Theory of Computing Systems*, 47:786–807, 2010. arXiv: [quant-ph/0609168](#) 4
- [BB94] M. L. Bonet and S. R. Buss. Size-depth tradeoffs for boolean formulae. *Information Processing Letters*, 49:151–155, 1994. 3
- [BCJ<sup>+</sup>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. 7
- [Bel13] Aleksandrs Belovs. Quantum walks and electric networks. arXiv: [1302.3143](#), 2013. 7, 9
- [BJY23] Aleksandrs Belovs, Stacey Jeffery, and Duyal Yolcu. Taming quantum time complexity. arXiv: [2311.15873](#), 2023. 3, 5
- [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. 3, 4
- [CKKD<sup>+</sup>22] Andrew M. Childs, Robin Kothari, Matt Kovacs-Deak, Aarthi Sundaram, and Daochen Wang. Quantum divide and conquer. arXiv: [2210.06419](#), 2022. 2, 4, 36
- [Cor23] Arjan Cornelissen. *Quantum multivariate estimation and span program algorithms*. PhD thesis, University of Amsterdam, 2023. 4
- [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](#) 4
- [IJ19] Tsuyoshi Ito and Stacey Jeffery. Approximate span programs. *Algorithmica*, 79:2158–2195, 2019. 7
- [Jef22] Stacey Jeffery. Quantum subroutine composition. arXiv: [2209.14146](#), 2022. 1, 2, 4, 8, 20, 21, 22, 23, 24
- [JJKP18] Michael Jarret, Stacey Jeffery, Shelby Kimmel, and Alvaro Piedrafita. Quantum algorithms for connectivity and related problems. In *Proceedings of the 26th Annual European Symposium on Algorithms (ESA)*, pages 49:1–49:13, 2018. 3, 4, 13
- [JK17] Stacey Jeffery and Shelby Kimmel. Quantum algorithms for graph connectivity and formula evaluation. *Quantum*, 1(26), 2017. 3, 4, 13, 24, 31
- [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. arXiv: [2208.13492](#) 1, 2, 4, 5, 6, 7, 8
- [Kit96] Alexei Y. Kitaev. Quantum measurements and the Abelian stabilizer problem. *ECCC*, TR96-003, 1996. arXiv: [quant-ph/9511026](#) 6

- [Lee59] C. Y. Lee. Representation of switching functions by binary decision programs. *Bell Systems Technical Journal*, 38(4):985–999, 1959. 3
- [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](#) 9
- [Pot15] Aaron H. Potechin. *Analyzing monotone space complexity via the switching network model*. PhD thesis, Massachusetts Institute of Technology, 2015. 3
- [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. arXiv: [0904.2759](#) 3
- [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. 2
- [RŠ12] Ben W. Reichardt and Robert Špalek. Span-program-based quantum algorithm for evaluating formulas. *Theory of Computing*, 8(13):291–319, 2012. 7
- [Sav70] Walter J Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of computer and system sciences*, 4(2):177–192, 1970. 3, 40
- [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. 3, 13
- [Sha49] Claude E. Shannon. The synthesis of two-terminal switching circuits. *Bell System Technical Journal*, 28(1):59–98, 1949. 3, 13
- [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. arXiv: [quant-ph/0401053](#) 7, 9

## A Recovering the Local Structure

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

$$\mathcal{B}_{G^\circ} = \bigoplus_{u \in V} \tilde{\Lambda}(\mathcal{V}_u) \oplus \bigoplus_{e \in \bar{E}} \bigoplus_{u \in V^e \setminus \{s, t\}} \mathcal{V}_u^e + \bigoplus_{e \in \bar{E}} \bigoplus_{e' \in E^e} \Xi_{e'}^{e\mathcal{B}} \quad (22)$$

$$\bar{\mathcal{B}}_{G^\circ} = \underbrace{\text{span} \Lambda(\Psi_{\bar{\mathcal{B}}_G}^-) \cup \bigcup_{e \in \bar{E}} (\Psi_{\bar{\mathcal{B}}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\})}_{=: \Psi_{\bar{\mathcal{B}}_{G^\circ}}^-} \cup \underbrace{\left\{ \frac{1}{\sqrt{2}}(|\rightarrow, e'\rangle + |\leftarrow, e'\rangle) : e' \in \bar{E}^\circ = \bigcup_{e \in \bar{E}} \bar{E}^e \right\}}_{\text{spans } \Xi_{\bar{E}^\circ}^{e\mathcal{B}} = \bigoplus_{e \in \bar{E}} \bigoplus_{e' \in \bar{E}^e} \Xi_{e'}^{e\mathcal{B}}} \}. \quad (23)$$

**Lemma A.1.** For any  $e \in \bar{E}$ ,

$$\text{span} \Psi_{\bar{\mathcal{B}}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi_{\bar{E}^e}^{e\mathcal{B}} = \mathcal{B}_{G^e} \cap \{|\leftarrow, s^e\rangle, |\rightarrow, t^e\rangle\}^\perp = \bigoplus_{u \in V^e \setminus \{s, t\}} \mathcal{V}_u^e + \Xi_{\bar{E}^e}^{e\mathcal{B}}.$$

*Proof.* It is clear that

$$\begin{aligned} \text{span} \Psi_{\bar{\mathcal{B}}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi_{\bar{E}^e}^{e\mathcal{B}} &\subseteq \mathcal{B}_{G^e} = \text{span} \Psi_{\bar{\mathcal{B}}_{G^e}}^- \oplus \Xi_{\bar{E}^e}^{e\mathcal{B}} \\ \text{and also } \bigoplus_{u \in V^e \setminus \{s, t\}} \mathcal{V}_u^e + \Xi_{\bar{E}^e}^{e\mathcal{B}} &\subseteq \mathcal{B}_{G^e} = \bigoplus_{u \in V^e} \mathcal{V}_u^e + \Xi_{\bar{E}^e}^{e\mathcal{B}}. \end{aligned}$$

Since we also have everything in  $\text{span}\Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\}$ ,  $\bigoplus_{u \in V^e \setminus \{s,t\}} \mathcal{V}_u^e$  and  $\Xi_{\overline{E}^e}^{\mathcal{B}}$  orthogonal to both  $|\leftarrow, s^e\rangle$  and  $|\rightarrow, t^e\rangle$ , we have:

$$\begin{aligned} \text{span}\Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi_{\overline{E}^e}^{\mathcal{B}} &\subseteq \mathcal{B}_{G^e} \cap \{|\leftarrow, s^e\rangle, |\rightarrow, t^e\rangle\}^\perp \\ \text{and } \bigoplus_{u \in V^e \setminus \{s,t\}} \mathcal{V}_u^e + \Xi_{\overline{E}^e}^{\mathcal{B}} &\subseteq \mathcal{B}_{G^e} \cap \{|\leftarrow, s^e\rangle, |\rightarrow, t^e\rangle\}^\perp. \end{aligned}$$

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  $|\phi_-\rangle \in \text{span}\Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\}$  and  $|\phi_+\rangle \in \Xi_{\overline{E}^e}^{\mathcal{B}}$ , and scalars  $a_0$  and  $a_1$ , we can express it as:

$$\begin{aligned} |\phi\rangle &= a_0|b_0^e\rangle + a_1|b_1^e\rangle + |\phi_-\rangle + |\phi_+\rangle \\ &= a_0 \frac{1}{\sqrt{2}}(|\leftarrow, s^e\rangle + |\rightarrow, t^e\rangle) + a_1 \frac{|\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle + \sqrt{r^e}|\bar{b}_1\rangle}{\sqrt{2+r^e}} + |\phi_-\rangle + |\phi_+\rangle. \end{aligned}$$

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  $|\phi\rangle \in \text{span}\Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi_{\overline{E}^e}^{\mathcal{B}}$ .

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  $|\phi_V\rangle \in \bigoplus_{u \in V^e \setminus \{s,t\}} \mathcal{V}_u^e$  and  $|\phi'_+\rangle \in \Xi_{\overline{E}^e}^{\mathcal{B}}$ . By the same reasoning as above, we must have  $a'_0 = a'_1 = 0$ , so  $|\phi\rangle \in \bigoplus_{u \in V^e \setminus \{s,t\}} \mathcal{V}_u^e + \Xi_{\overline{E}^e}^{\mathcal{B}}$ .  $\square$

**Lemma A.2.** For any  $e \in \overline{E}$ ,

$$\tilde{\Lambda}(|\rightarrow, e\rangle + |\leftarrow, e\rangle) \in \mathcal{B}_{G^e} \cap \{|\leftarrow, s^e\rangle, |\rightarrow, t^e\rangle\}^\perp.$$

*Proof.* Since  $\mathcal{V}_s^e + \mathcal{V}_t^e \subset \mathcal{B}_{G^e}$ , 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),

$$\begin{aligned} \sqrt{r^e}\tilde{\Lambda}(|\rightarrow, e\rangle + |\leftarrow, e\rangle) &= |\psi_\star(s^e)\rangle + |\psi_\star(t^e)\rangle \\ &= |\leftarrow, s^e\rangle + |\psi_\star(s^e)\rangle + |\rightarrow, t^e\rangle + |\psi_\star(t^e)\rangle - (|\leftarrow, s\rangle + |\rightarrow, t\rangle), \end{aligned}$$

so  $\tilde{\Lambda}(|\rightarrow, e\rangle + |\leftarrow, e\rangle) \in \mathcal{B}_{G^e}$ . Clearly  $\tilde{\Lambda}(|\rightarrow, e\rangle + |\leftarrow, e\rangle)$  is also orthogonal to  $|\leftarrow, s^e\rangle$  and  $|\rightarrow, t^e\rangle$ , concluding the proof.  $\square$

**Lemma A.3.** Let  $\tilde{\Lambda}$  be as in (14), and define

$$\tilde{\mathcal{B}}_{G^\circ} = \underbrace{\text{span}\tilde{\Lambda}(\Psi_{\mathcal{B}_{G^e}}^-) \cup \bigcup_{e \in \overline{E}} (\Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\})}_{=:\tilde{\Psi}_{\mathcal{B}_{G^\circ}}^-} \oplus \bigoplus_{e \in \overline{E}} \bigoplus_{e' \in \overline{E}^e} \Xi_{e'}^{\mathcal{B}}.$$

Then  $\tilde{\mathcal{B}}_{G^\circ} = \overline{\mathcal{B}}_{G^\circ}$ .

*Proof.* Let  $|\tilde{b}^e\rangle = \frac{1}{\sqrt{r^e}}(|\psi_\star(s^e)\rangle - |\psi_\star(t^e)\rangle)$ . Then

$$|\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle + \sqrt{r^e}|\tilde{b}^e\rangle \in \mathcal{V}_s^e + \mathcal{V}_t^e \subseteq \text{span}\{|b_0^e\rangle, |b_1^e\rangle, \dots, |b_\ell^e\rangle\} \oplus \Xi_{\overline{E}^e}^{\mathcal{B}}.$$

Then since the only vector that overlaps  $|\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle$  is  $|b_1^e\rangle = \frac{1}{\sqrt{2+r^e}}(|\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle) + \sqrt{\frac{r^e}{2+r^e}}|\bar{b}_1^e\rangle$ , we must have:

$$\langle b_1^e | (|\leftarrow, s^e\rangle - |\rightarrow, t^e\rangle + \sqrt{r^e}|\tilde{b}^e\rangle) = \sqrt{2+r^e},$$

from which it follows that  $\langle \tilde{b}_1^e | \tilde{b}^e \rangle = 1$ , and we can write:

$$|\tilde{b}^e\rangle = |\bar{b}_1^e\rangle + |d^e\rangle \quad (24)$$

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 A.1](#),  $|d^e\rangle \in \text{span}\Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi_{\bar{E}^e}^{e\mathcal{B}}$ .

Suppose  $|\tilde{\psi}\rangle \in \tilde{\mathcal{B}}_{G^\circ}$ , so we can express it, for some scalars  $a_e$ ,  $|\psi_{E \setminus \bar{E}}\rangle \in \Xi_{E \setminus \bar{E}}$  and  $|\psi'\rangle \in \text{span}\Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi_{\bar{E}^e}^{e\mathcal{B}}$  as:

$$\begin{aligned} |\tilde{\psi}\rangle &= \tilde{\Lambda} \left( \underbrace{\sum_{e \in \bar{E}} a_e \frac{1}{\sqrt{2}} (|\rightarrow, e\rangle - |\leftarrow, e\rangle) + |\psi_{E \setminus \bar{E}}\rangle}_{\in \text{span}\Psi_{\mathcal{B}_G}^-} \right) + |\psi'\rangle \\ &= \sum_{e \in \bar{E}} a_e |\tilde{b}^e\rangle + |\psi_{E \setminus \bar{E}}\rangle + |\psi'\rangle \\ &= \sum_{e \in \bar{E}} a_e |\bar{b}_1^e\rangle + |\psi_{E \setminus \bar{E}}\rangle + \sum_{e \in \bar{E}} a_e |d^e\rangle + |\psi'\rangle \\ &= \Lambda \left( \underbrace{\sum_{e \in \bar{E}} a_e \frac{1}{\sqrt{2}} (|\rightarrow, e\rangle - |\leftarrow, e\rangle) + |\psi_{E \setminus \bar{E}}\rangle}_{\in \text{span}\Psi_{\mathcal{B}_G}^-} \right) + \underbrace{\sum_{e \in \bar{E}} a_e |d^e\rangle + |\psi'\rangle}_{\in \text{span}\Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi_{\bar{E}^e}^{e\mathcal{B}}} \in \bar{\mathcal{B}}_{G^\circ}. \end{aligned} \quad (25)$$

Thus,  $\tilde{\mathcal{B}}_{G^\circ} \subseteq \bar{\mathcal{B}}_{G^\circ}$ .

For the other direction, suppose  $|\psi\rangle \in \bar{\mathcal{B}}_{G^\circ}$ . Then similar to above, we can express it as

$$\begin{aligned} |\psi\rangle &= \Lambda \left( \underbrace{\sum_{e \in \bar{E}} a_e \frac{1}{\sqrt{2}} (|\rightarrow, e\rangle - |\leftarrow, e\rangle) + |\psi_{E \setminus \bar{E}}\rangle}_{\in \text{span}\Psi_{\mathcal{B}_G}^-} \right) + |\psi'\rangle \\ &= \sum_{e \in \bar{E}} a_e |\bar{b}_1^e\rangle + |\psi_{E \setminus \bar{E}}\rangle + |\psi'\rangle \\ &= \sum_{e \in \bar{E}} a_e |\tilde{b}_1^e\rangle + |\psi_{E \setminus \bar{E}}\rangle - \sum_{e \in \bar{E}} a_e |d^e\rangle + |\psi'\rangle \\ &= \tilde{\Lambda} \left( \underbrace{\sum_{e \in \bar{E}} a_e \frac{1}{\sqrt{2}} (|\rightarrow, e\rangle - |\leftarrow, e\rangle) + |\psi_{E \setminus \bar{E}}\rangle}_{\in \text{span}\Psi_{\mathcal{B}_G}^-} \right) - \underbrace{\sum_{e \in \bar{E}} a_e |d^e\rangle + |\psi'\rangle}_{\in \text{span}\Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi_{\bar{E}^e}^{e\mathcal{B}}} \in \tilde{\mathcal{B}}_{G^\circ}. \end{aligned} \quad (26)$$

Thus  $\bar{\mathcal{B}}_{G^\circ} \subseteq \tilde{\mathcal{B}}_{G^\circ}$ . □

**Theorem A.4.** Let  $\bar{\mathcal{B}}_{G^\circ} = \text{span}\Psi_{\mathcal{B}_{G^\circ}}$  where  $\Psi_{\mathcal{B}_{G^\circ}}$  is as in (19) (see also (23)), and  $\mathcal{B}_{G^\circ}$  as in (17) (see also (22)). Then  $\bar{\mathcal{B}}_{G^\circ} = \mathcal{B}_{G^\circ}$ .

*Proof.* We will show that  $\mathcal{B}_{G^\circ} = \tilde{\mathcal{B}}_{G^\circ}$ , which is sufficient, by [Lemma A.3](#). We first show that  $\tilde{\mathcal{B}}_{G^\circ} \subseteq \mathcal{B}_{G^\circ}$ . First, for any  $e \in \bar{E}$ , it follows from [Lemma A.1](#) that

$$\Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi_{\bar{E}^e}^{e\mathcal{B}} = \bigoplus_{u \in V^e \setminus \{s, t\}} \mathcal{V}_u^e + \Xi_{\bar{E}^e}^{e\mathcal{B}} \subseteq \mathcal{B}'_{G^\circ}. \quad (27)$$

Next, for any  $|\psi\rangle \in \Psi_{\mathcal{B}_G}^- \subseteq \bigoplus_{u \in V} \mathcal{V}_u$ , we have

$$\tilde{\Lambda}(|\psi\rangle) \in \tilde{\Lambda} \left( \bigoplus_{u \in V} \mathcal{V}_u \right) = \bigoplus_{u \in V} \tilde{\Lambda}(\mathcal{V}_u) \subseteq \mathcal{B}'_{G^\circ}. \quad (28)$$

From (27) and (28), it follows that  $\tilde{\mathcal{B}}_{G^\circ} \subseteq \mathcal{B}_{G^\circ}$ .

We now show the other direction,  $\mathcal{B}_{G^\circ} \subseteq \tilde{\mathcal{B}}_{G^\circ}$ . As in (27), for any  $e \in \bar{E}$ ,

$$\bigoplus_{u \in V^e \setminus \{s, t\}} \mathcal{V}_u^e + \Xi_{\bar{E}}^{e\mathcal{B}} = \Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi_{\bar{E}}^{e\mathcal{B}} \subseteq \tilde{\mathcal{B}}_{G^\circ}. \quad (29)$$

Next, for any  $u \in V$  and  $|\psi\rangle \in \mathcal{V}_u \subseteq \mathcal{B}_G = \text{span}\Psi_{\mathcal{B}_G}^- \oplus \Xi_{\bar{E}}^{\mathcal{B}}$ , we can write

$$|\psi\rangle = |\psi_{-}\rangle + \sum_{e \in \bar{E}} a_e (|\rightarrow, e\rangle + |\leftarrow, e\rangle)$$

for some  $|\psi_{-}\rangle \in \text{span}\Psi_{\mathcal{B}_G}^-$  and scalars  $a_e$ . Then:

$$\tilde{\Lambda}(|\psi\rangle) = \underbrace{\tilde{\Lambda}|\psi_{-}\rangle}_{\in \text{span}\tilde{\Lambda}(\Psi_{\mathcal{B}_G}^-) \subseteq \tilde{\mathcal{B}}_{G^\circ}} + \sum_{e \in \bar{E}} a_e \tilde{\Lambda}(|\rightarrow, e\rangle + |\leftarrow, e\rangle). \quad (30)$$

By Lemma A.2, for each  $e$ ,  $\tilde{\Lambda}(|\rightarrow, e\rangle + |\leftarrow, e\rangle) \in \mathcal{B}_{G^e} \cap \{|\leftarrow, s\rangle, |\rightarrow, t\rangle\}^\perp$ , which is equal to  $\text{span}\Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi_{\bar{E}}^{e\mathcal{B}}$  by Lemma A.1. Since  $\text{span}\Psi_{\mathcal{B}_{G^e}}^- \setminus \{|b_0^e\rangle, |b_1^e\rangle\} \oplus \Xi_{\bar{E}}^{e\mathcal{B}} \subseteq \tilde{\mathcal{B}}_{G^\circ}$ , we have  $\tilde{\Lambda}|\psi\rangle \in \tilde{\mathcal{B}}_{G^\circ}$ , from which it follows that

$$\tilde{\Lambda}(\mathcal{V}_u) \subseteq \tilde{\mathcal{B}}_{G^\circ}. \quad (31)$$

Combining (29) and (31) establishes  $\mathcal{B}_{G^\circ} \subseteq \tilde{\mathcal{B}}_{G^\circ}$ , completing the proof.  $\square$