


# Two-qubit Stabilizer Circuits with Recovery II: Analysis

**Wim van Dam**

Department of Computer Science, Department of Physics, University of California, Santa Barbara, CA, USA

vandam@ucsb.edu

 <https://orcid.org/0000-0001-7852-6158>

**Raymond Wong**

Department of Computer Science, University of California, Santa Barbara, CA, USA

rwong@ucsb.edu

---

## Abstract

We study stabilizer circuits that use non-stabilizer qubits and  $Z$ -measurements to produce other non-stabilizer qubits. These productions are successful when the correct measurement outcome occurs, but when the opposite outcome is observed, the non-stabilizer input qubit is potentially destroyed. In preceding work [arXiv:1803.06081 (2018)] we introduced protocols able to recreate the expensive non-stabilizer input qubit when the two-qubit stabilizer circuit has an unsuccessful measurement outcome. Such protocols potentially allow a deep computation to recover from such failed measurements without the need to repeat the whole prior computation. Possible complications arise when the recovery protocol itself suffers from a failed measurement. To deal with this, we need to use nested recovery protocols. Here we give a precise analysis of the potential advantage of such recovery protocols as we examine its optimal nesting depth. We show that if the expensive input qubit has cost  $d$ , then typically a depth  $O(\log d)$  recovery protocol is optimal, while a certain special case has optimal depth  $O(\sqrt{d})$ . We also show that the recovery protocol can achieve a cost reduction by a factor of at most two over circuits that do not use recovery.

**2012 ACM Subject Classification** Theory of computation → Quantum computation theory

**Keywords and phrases** stabilizer circuit, recovery circuit, magic state

**Digital Object Identifier** 10.4230/LIPIcs.TQC.2018.8

**Funding** This material is based upon work supported by the National Science Foundation under Grants No. 0917244 and 1719118.

## 1 Introduction

In [21] we saw another treatment of two-qubit stabilizer circuits for recovery purposes on a select set of input states. Here, we give a more thorough assessment of its potential to better determine its influence on quantum computations.

As of now, such studies are still necessary to address one major difficulty to building quantum computers, and that is the large overhead required to ensure a reliable system for handling noise [10]. Over the course of a long computation, a quantum state may encounter unwanted influences from the outside (the environment) and from within (faulty parts) that affect the qubits in undesirable ways. Any realistic solution must include quantum error correction and fault-tolerance to prevent an uncontrollable spread of errors, and often, stabilizer operations which consist of Clifford group unitaries, Pauli measurements, and ancilla  $|0\rangle$  preparation are considered a viable option to serve as the foundation of a fault-tolerant scheme. One of their most memorable characteristics is perhaps that which is famously



© Wim van Dam and Raymond Wong;

licensed under Creative Commons License CC-BY

13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2018).

Editor: Stacey Jeffery; Article No. 8; pp. 8:1–8:21

Leibniz International Proceedings in Informatics



Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

stated in the Gottesman-Knill theorem: that stabilizer operations are efficiently simulable on classical computers. On the other hand, it also means stabilizer operations are inadequate for universal quantum computation (UQC).

To resolve this, Bravyi and Kitaev introduced magic state distillation [5]. It is a technique in which noisy magic states are distilled to a higher quality, then consumed to implement quantum gates outside the Clifford group of operations e.g.  $\pi/4$  phase rotation  $T$ . This is entirely sufficient for UQC since any non-Clifford gate with stabilizer operations is enough to form a universal basis. Many improvements have appeared since its debut [4, 6, 8, 9, 11, 15, 16], but even more impressive is that some of these recent proposals [6, 8, 11, 15] support the distillation of multiple kinds of magic qubits, which enables the implementation of other non-Clifford gates and yields richer bases. Related work on circuit synthesis has also surged, using number theory as the foundation for designing efficient algorithms over universal gate sets [1, 2, 3, 18, 20]. For single qubit unitaries, optimal usage of  $T$ -gates is possible [14, 19].

Research originating from state distillation and gate synthesis has inspired other studies on stabilizer operations. One such example [21] expanded on ideas from [9, 12, 17] to produce some interesting results. In particular, van Dam and Wong [21] (and indirectly by Reichardt [17]) found that any stabilizer procedure generating a single qubit output from a two-qubit input can be realized by a postselected stabilizer circuit of single qubit Clifford gates and at most one CNOT or SWAP. Then for those involving a CNOT, there exist “recovery circuits” that essentially recycle a stabilizer circuit output back into a reusable form. Such operations pair nicely with processes that inject magic states toward the tail end of a long and expensive computation. Thus if the original state preparation is an extremely costly endeavor, recovery circuits provide a welcome alternative. For the moment, two conditions are required for recovery circuits to be of service: (1) the two-qubit input is a product state, and (2) one of the qubits is pure.

In this paper, we continue the evaluation of recovery circuits. Specifically we pursue a more rigorous examination of a nested recovery protocol previously described in [21] to answer questions about its optimal nesting depth. Though the current applications for such a recovery technique are limited, we cannot rule out the possibility of similarly defined recovery operations for larger stabilizer circuits and inputs. For that reason, it is worthwhile to know how helpful the nested recovery protocol will be even in the two-qubit domain. Through our analysis, we learn that for an initial preparation cost of about  $d$ , a protocol of depth  $O(\log d)$  is optimal in generic situations, while the depth is allowed to grow to  $O(\sqrt{d})$  in one special case (Theorem 16). Under this assumption, we discover up to a factor of two savings is achievable over a protocol that ignores recovery (Theorem 17).

## 2 Background

This section covers the main concepts and notation. We refer the reader to [21] for a more detailed account on the subjects presented in Subsections 2.2 and 2.3.

### 2.1 Pauli Matrices and Stabilizer Circuits

The Pauli group consists of  $n$ -qubit Pauli operators on the four matrices

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \quad (1)$$

An  $n$ -qubit stabilizer state is then a simultaneous  $+1$  eigenstate of  $n$  independent and commuting operators from the Pauli group; there are six such states when  $n = 1$ . The

normalizer of the Pauli group is known as the Clifford group and is generated by the Controlled-NOT (CNOT), Hadamard ( $H$ ), and Phase ( $P$ ) gates. The matrices of these three operators are

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \quad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad P = \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}. \quad (2)$$

A stabilizer circuit is therefore a quantum circuit of CNOT,  $H$ ,  $P$  gates and measurements in the  $Z$ -basis. In a quantum circuit diagram, we use



to represent a  $Z$ -measurement and a qubit swap, respectively.

## 2.2 Postselected Two-to-One Stabilizer Circuits

To reiterate, the following terminology appear in [21].

► **Definition 1** (postselected two-to-one stabilizer circuit). A *postselected two-to-one stabilizer circuit*  $(C, b)$  is a two-qubit quantum circuit that implements a Clifford unitary  $C$ , followed by a  $Z$ -measurement on the second qubit with an outcome  $b \in \{0, 1\}$ .

► **Definition 2** (probability and output). Let  $(C, b)$  be a postselected two-to-one stabilizer circuit and let  $\rho$  be a two-qubit state. Then the *probability*  $Q_b$  of outcome  $b$  on the transformed state  $C\rho C^\dagger$  is  $Q_b(C, \rho) = \text{Tr}((I \otimes \langle b|)C\rho C^\dagger(I \otimes |b\rangle))$ . If  $Q_b(C, \rho) > 0$ , then the *output*  $\Phi_b$  of a postselected circuit  $(C, b)$  on an input  $\rho$  is

$$\Phi_b(C, \rho) = \frac{(I \otimes \langle b|)C\rho C^\dagger(I \otimes |b\rangle)}{Q_b(C, \rho)}. \quad (3)$$

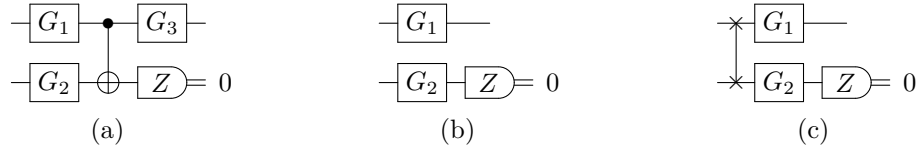
The expression *run circuit*  $C$  shall mean an application of unitary  $C$  on the initial state  $\rho$ , followed by a  $Z$ -measurement on the second qubit; *circuit*  $C$  shall reference the stabilizer circuit piece only of the postselected circuit  $(C, b)$ , including the measurement gate. Because different postselected stabilizer circuits may produce the same output on a given input state  $\rho$ , we have the following definition.

► **Definition 3** (equivalent postselected two-to-one stabilizer circuits). Two postselected two-to-one stabilizer circuits  $(C_1, b_1)$  and  $(C_2, b_2)$  are *Clifford equivalent*,  $(C_1, b_1) \sim (C_2, b_2)$ , if and only if there is a single qubit Clifford gate  $G$  such that for all two-qubit states  $\rho$ , we have the equality

$$(I \otimes \langle b_1|)C_1\rho C_1^\dagger(I \otimes |b_1\rangle) = G(I \otimes \langle b_2|)C_2\rho C_2^\dagger(I \otimes |b_2\rangle)G^\dagger. \quad (4)$$

Note that a Clifford equivalence implies that the probabilities of observing a  $b_1$  or  $b_2$  are the same for the two circuits i.e.  $Q_{b_1}(C_1, \rho) = Q_{b_2}(C_2, \rho)$ . The two postselected circuits are *equivalent*,  $(C_1, b_1) \equiv (C_2, b_2)$ , if and only if  $G = I$  in Equation 4.

We can classify a postselected circuit  $(C, b)$  into one of three types. More precisely, there are always single qubit Clifford gates  $G_1$  and  $G_2$  such that either  $(C, b) \sim (I \otimes G_1, 0)$ , or  $(C, b) \sim ((I \otimes G_1)\text{SWAP}, 0)$ , or  $(C, b) \sim (\text{CNOT}(G_1 \otimes G_2), 0)$ . If we know  $(C, b) \sim (C', b')$ , where  $C'$  is one of three previous forms, then  $(C, 1 - b) \sim ((I \otimes X)C', b') \equiv (C', 1 - b')$ . A summary of the configurations is provided in Figure 1. Depending on the type of circuit and input we are dealing with,  $(C, b)$  may be eligible for a *recovery circuit* [21].



■ **Figure 1** Any stabilizer procedure generating a single qubit output from a two-qubit input can be implemented by a postselected stabilizer circuit  $(C, b)$  taking on one of the three forms above. The exact single qubit Clifford gates  $G_1$ ,  $G_2$ , and  $G_3$  depend on  $C$  and  $b$  and are not unique.

### 2.3 Recovery Circuits

If  $\rho$  is the product state  $\varphi \otimes |\psi\rangle\langle\psi|$ , then only postselected circuits of the kind  $(C, b) \sim (\text{CNOT}(G_1 \otimes G_2), 0)$  qualify for a recovery circuit. For convenience, we use  $\psi$  in place of the density matrix  $|\psi\rangle\langle\psi|$  from this point on.

► **Definition 4** (interacting postselected circuit). A postselected two-to-one stabilizer circuit  $(C, b)$  is *interacting* if and only if there are single qubit Clifford gates  $G_1$  and  $G_2$  such that  $(C, b) \sim (\text{CNOT}(G_1 \otimes G_2), 0)$ . We say circuit  $C$  is *interacting* if and only if  $(C, 0)$  is interacting.

► **Definition 5** (recovery circuit). Let  $(C, b)$  be an interacting postselected circuit. Then a postselected two-to-one stabilizer circuit  $(C', b')$  is a *recovery circuit* of  $(C, b)$  if and only if for all two-qubit states  $\varphi \otimes \psi$ , we have  $\varphi = \Phi_{b'}(C', \Phi_{1-b}(C, \varphi \otimes \psi) \otimes \psi)$ .

Thus if an outcome  $b$  is more desirable than  $1 - b$ , we say an interacting circuit  $C$  is *successful* if the measurement on  $C(\varphi \otimes \psi)C^\dagger$  yields  $b$  and *unsuccessful* otherwise. If unsuccessful, then given a recovery circuit  $(C', b')$  of  $(C, b)$ , we may run circuit  $C'$  on the input state described above to try and recover  $\varphi$ . There is also a relatively simple construction to acquire a recovery circuit. If  $(C, b) \sim (\text{CNOT}(G_1 \otimes G), 0)$  for single qubit Clifford gates  $G_1$  and  $G$ , then there is a third Clifford gate  $G_2$  satisfying  $(C, 1 - b) \equiv ((G_2^\dagger \otimes I)\text{CNOT}(G_1 \otimes G), 1)$ . We may then use this to design a recovery circuit  $(C', 0)$  of  $(C, b)$ , where  $C' = (G_2^\dagger \otimes I)\text{CNOT}(G_2 \otimes G)$  [21].

The success probabilities of circuits  $C$  and  $C'$  on their respective input are also interrelated. If we start with a two qubit state  $\varphi_1 \otimes \psi$ , then the probability of recovering  $\varphi_1$  is

$$Q_0(C', \Phi_{1-b}(C, \varphi_1 \otimes \psi) \otimes \psi) = \frac{(1 - z^2)/4}{1 - Q_b(C, \varphi_1 \otimes \psi)} \tag{5}$$

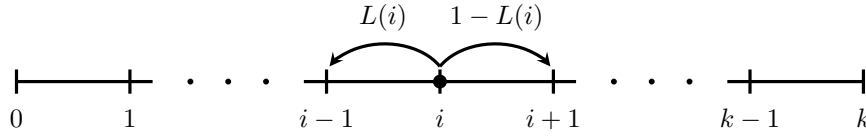
where  $z = \langle\psi|G^\dagger ZG|\psi\rangle$ . More than one recovery circuit of  $(C, b)$  exists, but all recovery circuits of  $(C, b)$  are equivalent to each other and hence have the same recovery success rate. Furthermore, recovery circuits are interacting postselected circuits as well, leading to the following corollary [21].

► **Corollary 6.** *Every recovery circuit  $(C', b')$  has its own recovery circuit  $(C'', b'')$ .*

Finally, there is a Clifford gate  $G$  such that  $\Phi_{1-b}(C, \varphi_1 \otimes \psi) = G\varphi_1G^\dagger$  whenever  $|\psi\rangle$  is a stabilizer qubit [21]. Since the output is essentially  $\varphi_1$ , recovery circuits are no longer helpful for this combination of input qubits.

## 3 Depth $k$ Protocol with Recovery

Suppose our goal is to produce the output of a postselected circuit  $(C_1, b_1)$  on a two-qubit state  $\varphi_1 \otimes \psi$ . By Corollary 6, we can derive a *depth  $k$*  protocol on  $k - 1$  interacting postselected



■ **Figure 2** The behavior of a depth  $k$  protocol corresponds to a random walk on integers  $\{0, \dots, k\}$  and starts at position 1. The random walk ends upon reaching 0 or  $k$ , with 0 representing success and  $k$  representing failure. The transition from  $i$  to  $i - 1$  is the success probability of the  $i$ -th circuit  $C_i$  from the protocol.

circuits such that  $(C_{i+1}, b_{i+1})$  is a recovery circuit of  $(C_i, b_i)$ . We may assume without loss of generality a desirable outcome  $b_i = 0$  for all  $k - 1$  circuits. Thus when circuit  $C_1$  is unsuccessful i.e. measure a 1, we fall back on circuit  $C_2$ . If circuit  $C_2$  is also unsuccessful, we depend on circuit  $C_3$ , and so on all the way down to circuit  $C_{k-1}$ . In more detail, our depth  $k$  protocol works as follows:

1. Let  $\varphi_1 \otimes \psi$  be the initial state, and let  $(C_1, 0), \dots, (C_{k-1}, 0)$  be interacting postselected circuits such that  $(C_{i+1}, 0)$  is a recovery circuit of  $(C_i, 0)$ .
2. Run circuit  $C_1$  on  $\varphi_1 \otimes \psi$ . If we measure 0, then we declare *success*. Otherwise, let  $\varphi_2$  be the output of  $(C_1, 1)$  on  $\varphi_1 \otimes \psi$ .
3. Run circuit  $C_2$  on  $\varphi_2 \otimes \psi$ . If we measure 0, then we recover  $\varphi_1$  and we repeat step 2. Otherwise we get the output  $\varphi_3$  of  $(C_2, 1)$  on  $\varphi_2 \otimes \psi$ .
4. Repeat step 3 as necessary for other circuits  $C_i$ . That is, let  $\varphi_i$  be the output of  $(C_{i-1}, 1)$  on  $\varphi_{i-1} \otimes \psi$ . Run circuit  $C_i$  on  $\varphi_i \otimes \psi$ . On measuring 0, the output is  $\varphi_{i-1}$  and we rerun circuit  $C_{i-1}$  on  $\varphi_{i-1} \otimes \psi$ . Otherwise, we proceed with circuit  $C_{i+1}$  on  $\varphi_{i+1} \otimes \psi$ .
5. If circuit  $C_{k-1}$  is unsuccessful on  $\varphi_{k-1} \otimes \psi$ , then we declare *failure* and stop.

We repeat this setup on  $k - 1$  circuits until we secure the output qubit  $\varphi_0 = \Phi_0(C_1, \varphi_1 \otimes \psi)$ . By involving more than one circuit, we prolong our attempts at gaining  $\varphi_0$  while reducing the number of times we rerun the prior computation on new copies of  $\varphi_1$ . As pointed out by the simulations in [21], we expect the protocol is more useful when  $\varphi_1$  is the result of a long and resource intensive preparation procedure. The depth  $k$  affects the amount of resource qubits  $|\psi\rangle$  our protocol consumes on each invocation. We give a more thorough explanation on how to choose  $k$  later in the paper.

We may view the process of generating  $\varphi_0$  as a random walk on  $k + 1$  integers  $\{0, \dots, k\}$ , starting at location 1. A step onto 0 signals success, and a step onto  $k$  indicates failure. The success probability of circuit  $C_i$  is the left step transition probability from position  $i$  to  $i - 1$ . Not surprisingly, we can compute the recovery probability for every circuit  $C_2$  to  $C_{k-1}$  if we know the first success probability  $Q_0(C_1, \varphi_1 \otimes \psi)$ . The next lemma is an extension of Equation 5.

► **Lemma 7.** *Consider a series of  $k - 1$  interacting postselected circuits  $(C_i, 0)$  such that  $(C_{i+1}, 0)$  is a recovery circuit of  $(C_i, 0)$ . Then given a two-qubit state  $\varphi_1 \otimes \psi$  and outputs  $\varphi_i = \Phi_1(C_{i-1}, \varphi_{i-1} \otimes \psi)$ , the success probability of each circuit  $C_i$  is*

$$L(i) = Q_0(C_i, \varphi_i \otimes \psi) = \begin{cases} Q_0(C_1, \varphi_1 \otimes \psi) & \text{if } i = 1 \\ \frac{(1 - z^2)/4}{1 - L(i - 1)} & \text{if } i \in \{2, \dots, k - 1\} \end{cases} \quad (6)$$

where  $z \in \{\langle \psi | X | \psi \rangle, \langle \psi | Y | \psi \rangle, \langle \psi | Z | \psi \rangle\}$ .

## 8:6 Recovery Circuits II: Analysis

**Proof.** We primarily need to explain why the numerator stays the same at every step  $i$ , since we can infer the form from Equation 5. Suppose  $(C_1, 1) \equiv ((G_2^\dagger \otimes I)\text{CNOT}(G_1 \otimes G), 1)$ , where  $G$ ,  $G_1$ , and  $G_2$  are single qubit Clifford gates. This means

$$(C_1, 0) \sim (\text{CNOT}(G_1 \otimes G), 0) \quad (7)$$

$$(C_2, 0) \equiv ((G_1^\dagger \otimes I)\text{CNOT}(G_2 \otimes G), 0). \quad (8)$$

Next, there is a Clifford gate  $G_3$  such that  $(C_2, 1) \equiv ((G_3^\dagger \otimes I)\text{CNOT}(G_2 \otimes G), 1)$ , which implies  $(C_3, 0) \equiv ((G_2^\dagger \otimes I)\text{CNOT}(G_3 \otimes G), 0)$ . Continuing in this manner, we find single qubit Clifford gates  $G_i$  and  $G_{i+1}^\dagger$  satisfying

$$(C_i, 1) \equiv ((G_{i+1}^\dagger \otimes I)\text{CNOT}(G_i \otimes G), 1) \quad (9)$$

$$(C_{i+1}, 0) \equiv ((G_i^\dagger \otimes I)\text{CNOT}(G_{i+1} \otimes G), 0) \quad (10)$$

for all  $i \geq 1$ . We study the effects of each postselected circuit  $(C_i, 1)$  on  $\varphi_i \otimes \psi$  and  $(C_{i+1}, 0)$  on  $\varphi_{i+1} \otimes \psi$  via the equivalent postselected circuits just described.

Consider the qubits  $|\psi'\rangle = G|\psi\rangle$  and  $\varphi'_i = G_i\varphi_iG_i^\dagger$ . From our  $G_{i+1}$  selection, this means

$$\varphi'_{i+1} = \Phi_1(\text{CNOT}, \varphi'_i \otimes \psi') = G_{i+1}\varphi_{i+1}G_{i+1}^\dagger \quad (11)$$

$$\varphi'_i = \Phi_0(\text{CNOT}, \varphi'_{i+1} \otimes \psi'). \quad (12)$$

Observe that both gates  $G_i$  and  $G_{i+1}^\dagger$  to the control qubit in  $((G_{i+1}^\dagger \otimes I)\text{CNOT}(G_i \otimes G), 1)$  are always neutralized by the recovery circuit  $((G_i^\dagger \otimes I)\text{CNOT}(G_{i+1} \otimes G), 0)$ . In other words, at each step  $i$ , we always apply CNOT on qubits  $\varphi'_i$  and  $|\psi'\rangle$  as if the rotations by  $G_i$  and  $G_{i+1}^\dagger$  never took place. In the last section (and [21]), we saw  $(C_1, 0) \sim (\text{CNOT}(G_1 \otimes G, 0))$  and  $(C_2, 0) \sim (\text{CNOT}(G_2 \otimes G, 0))$  pave the way to Equation 5. We apply the same arguments between  $(C_i, 0)$  and  $(C_{i+1}, 0)$  to obtain the recurrence above. ◀

We can also narrow the success probability of each circuit  $C_i$  to a more specific range.

► **Lemma 8.** *Consider a series of  $k - 1$  interacting postselected circuits  $(C_i, 0)$  such that  $(C_{i+1}, 0)$  is a recovery circuit of  $(C_i, 0)$ . Then given a two-qubit state  $\varphi_1 \otimes \psi$  and outputs  $\varphi_i = \Phi_1(C_{i-1}, \varphi_{i-1} \otimes \psi)$ , the success probability of each circuit  $C_i$  is bounded above and below by*

$$\frac{1 - \sqrt{1 - 4\lambda}}{2} \leq L(i) = Q_0(C_i, \varphi_i \otimes \psi) \leq \frac{1 + \sqrt{1 - 4\lambda}}{2} \quad (13)$$

where  $\lambda = (1 - z^2)/4$  and  $z \in \{\langle \psi|X|\psi\rangle, \langle \psi|Y|\psi\rangle, \langle \psi|Z|\psi\rangle\}$ .

**Proof.** Assume  $C_i = \text{CNOT}$  for simplicity. Then  $z = \langle \psi|Z|\psi\rangle$  and  $z_i = \text{Tr}(Z\varphi_i)$ . This gives

$$\frac{1 - |z|}{2} \leq L(i) = \frac{1 + z_i z}{2} \leq \frac{1 + |z|}{2} \quad (14)$$

since  $z_i \in [-1, 1]$ . But we can also say

$$\frac{1 + \sqrt{1 - 4\lambda}}{2} = \frac{1 + |z|}{2}, \quad \frac{1 - \sqrt{1 - 4\lambda}}{2} = \frac{1 - |z|}{2} \quad (15)$$

which implies the inequality. ◀

We only care for positive values of  $\lambda = (1 - z^2)/4 \leq 1/4$ . It equals zero if  $z = \pm 1$ , which occurs whenever  $|\psi\rangle$  undergoes a Clifford rotation  $G$  such that  $G|\psi\rangle = |0\rangle$  or  $|1\rangle$  prior to CNOT (see proof to Lemma 7 for greater details). Moreover, as  $1 - z^2 = x^2 + y^2$  for the Bloch vector  $(x, y, z)$  of  $G|\psi\rangle$ , we may interpret  $\lambda$  as the reduced overlap that  $G|\psi\rangle$  makes with the  $XY$ -plane.

## 4 Performance Analysis of Protocol

We consume a certain number of  $|\psi\rangle$  qubits every time we run the protocol. The amount we expend varies with the depth  $k$ , so it is imperative we find the ideal depth to minimize our  $|\psi\rangle$  usage.

### 4.1 Expected Cost

We first need to know the resource requirements of our protocol. To facilitate the presentation of our results, observe that abstractly our protocol is essentially a sequence of numbers  $L(1), \dots, L(k-1)$ , generated entirely by a recurrence relation  $L(i)$  defined on two real numbers which we call  $\lambda$  and  $\gamma$ . The depth  $k$  only serves to indicate a stopping point when generating that sequence, so our protocol is basically controlled by three parameters  $(\lambda, \gamma, k)$ . We will usually say that an instance of our protocol is set according to an assignment on these three values. As we alluded to a moment ago,  $\lambda$  is the *reduced XY-overlap* of resource qubit  $|\psi\rangle$ , and  $\gamma$  is the *starting success probability*  $Q_0(C_1, \varphi_1 \otimes \psi)$ . However, if we want to treat  $\lambda$  and  $\gamma$  simply as real numbers, we need these two parameters to comply with certain constraints for the  $L(i)$  numbers to be valid probabilities. Definition 9 brings together all relevant details about  $\lambda$  and  $\gamma$  that are necessary to define a difference equation adhering to Lemma 8.

► **Definition 9** (probability specification and boundary). Given real numbers  $(\lambda, \gamma)$ , let

$$\alpha = \frac{1 + \sqrt{1 - 4\lambda}}{2}, \quad \beta = 1 - \alpha = \frac{1 - \sqrt{1 - 4\lambda}}{2}. \quad (16)$$

Then  $(\lambda, \gamma)$  is a *probability specification* if and only if  $0 \leq \lambda \leq 1/4$  and  $\beta \leq \gamma \leq \alpha$ . A probability specification is *restricted* if and only if  $0 < \lambda < 1/4$  and  $\beta < \gamma < \alpha$ . The values  $(\alpha, \beta)$  are the *boundaries* of the probability specification.

► **Definition 10** (intermediate functions and rde). Let  $(\lambda, \gamma)$  be a probability specification and let  $(\alpha, \beta)$  be its boundaries. The following are the *intermediate* functions of  $(\lambda, \gamma)$ :

$$A_1(i) = \alpha^i - \beta^i, \quad A_2(i) = \alpha^i + \beta^i, \quad B_j(i) = A_j(i+1) - \gamma A_j(i), \quad (17)$$

and the following is a *rational difference equation (rde)* on  $(\lambda, \gamma)$ :

$$L(i) = \frac{\lambda B_1(i-2)}{B_1(i-1)} = \begin{cases} \gamma & \text{if } i = 1 \\ \frac{\lambda}{1 - L(i-1)} & \text{otherwise.} \end{cases} \quad (18)$$

As the name implies, the purpose of the intermediate functions is to help us build smaller results leading up to our main propositions. We also realize right away that because  $L(i)$  is a rational difference equation on a probability specification  $(\lambda, \gamma)$ , the boundaries  $\alpha$  and  $\beta$  are fixed points of  $L(i)$ . We end up with a similar situation to  $\lambda = 0$ . When  $\alpha = \gamma > 1/2$ , this suggests either input qubit  $|\psi\rangle$  or  $\varphi_1$  is a stabilizer state, and we have an analogous implication with  $\beta = \gamma < 1/2$ . Hence we define a restricted probability specification as satisfying both  $0 < \lambda < 1/4$  and  $\beta < \gamma < \alpha$ . On the other hand,  $\lambda = 1/4$  means  $\gamma$  no longer has the freedom to take on more than one value.

► **Lemma 11.** *There is only one probability specification with  $\lambda = 1/4$ . It forces  $\beta = \gamma = \alpha = 1/2$ , which leads to  $L(i) = 1/2$ .*

There are three ingredients to computing a protocol's expected cost.

► **Definition 12** (startup cost, success probability (of protocol), and expected demand). Consider a depth  $k$  protocol that starts by running circuit  $C_1$  on a two-qubit state  $\varphi_1 \otimes \psi$ . Then we define the following quantities of the protocol:

- i. *startup cost*: cost to prepare one  $\varphi_1$  qubit relative to the cost of one  $|\psi\rangle$  qubit
- ii. *success probability (of protocol)*: probability of declaring success before declaring failure
- iii. *expected demand*: expected number of  $|\psi\rangle$  states used in each execution, regardless of the final success or fail outcome.

► **Definition 13** (expected cost). The *expected cost* of a depth  $k$  protocol is determined by  $N = (d + s)/p$ , where  $d$  is the startup cost,  $p$  is the protocol's success probability, and  $s$  is the expected demand.

In the next lemma, we present the success probability and expected demand of a protocol in the general situation.

► **Lemma 14.** *Let  $A_1(i)$  and  $B_2(i)$  be intermediate functions of a restricted probability specification  $(\lambda, \gamma)$ . Then the success probability  $p$  and expected demand  $s$  of a protocol set to  $(\lambda, \gamma)$  and depth  $k$  are*

$$p = \frac{\gamma A_1(k-1)}{A_1(k)}, \quad s = \frac{A_1(k-1)(\gamma - 2\lambda) + (k-1)A_1(1)B_2(k-1)}{(A_1(1))^2 A_1(k)}. \quad (19)$$

**Proof.** As we mentioned earlier, we model our protocol as a random walk on the integers  $\{0, \dots, k\}$ . Since we are dealing with a restricted probability specification, we look towards Lemma 27 of Appendix B. Plugging  $i = 1$  into the equations returns the solutions above. ◀

A protocol given an assignment of  $(\lambda, \gamma, k)$  behaves quite differently when  $\lambda = 1/4$  versus the more general  $(\lambda, \gamma)$  a restricted probability specification. Because we have to treat the protocol specially when  $\lambda = 1/4$ , we end up with two expected cost equations.

► **Lemma 15.** *The expected cost of a protocol with startup cost  $d$  and set to a restricted probability specification  $(\lambda, \gamma)$  and depth  $k$  is*

$$N(k) = \frac{dA_1(k)}{\gamma A_1(k-1)} + \frac{(k-1)B_2(k-1)}{\gamma A_1(1)A_1(k-1)} + \frac{\gamma - 2\lambda}{\gamma (A_1(1))^2} \quad (20)$$

where  $A_1(i)$  and  $B_2(i)$  are intermediate functions of  $(\lambda, \gamma)$ . The expected cost of a protocol with  $\lambda = 1/4$  is

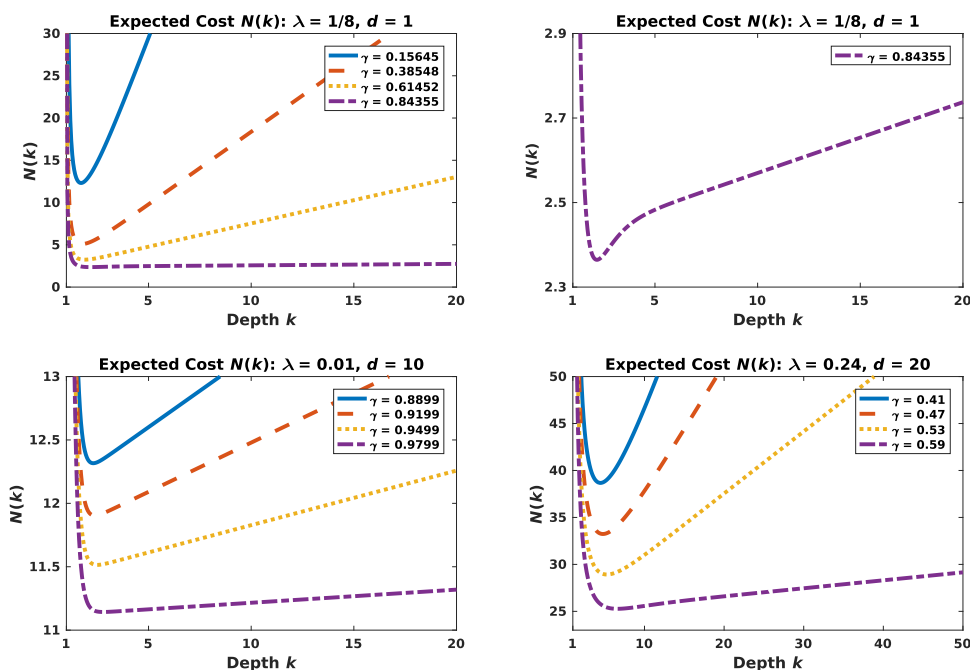
$$N(k) = \frac{k^2 + kd - k}{k-1}. \quad (21)$$

**Proof.** The proof is straightforward from  $N(k) = (d+s)/p$ , where  $s = k-1$  and  $p = (k-1)/k$  when  $\lambda = 1/4$  by Lemma 28, and by Lemma 14 when  $(\lambda, \gamma)$  is a restricted probability specification. ◀

## 4.2 Minimizing Expected Cost

We want to find the integer  $k \geq 2$  that minimizes the expected cost  $N(k)$ . That is, we wish to solve  $N_{\text{opt}} = \min_{k \in \{2, 3, \dots\}} N(k)$  and determine the depth  $k_{\text{opt}}$  such that  $N_{\text{opt}} = N(k_{\text{opt}})$ . Fortunately, there is evidence to suggest  $N(k)$  has a single critical point. Figure 3 shows the expected cost for several protocol instances set to varying restricted probability specifications  $(\lambda, \gamma)$  and startup costs  $d$ . The examples provide a convincing argument to assume  $N(k)$  has





■ **Figure 3** This figure contains plots of the expected cost  $N(k)$  for three choices of the reduced  $XY$ -overlap  $\lambda \in (0, 1/4)$  and varying starting probabilities  $\gamma$ . Although the curve of  $\gamma = 0.84355$  for  $\lambda = 1/8$  appear to reach a constant, the close-up in the top right graph suggests otherwise. Notice how every curve has a minimum at a point  $k > 1$  before a region of continuous increase. Equation 23 indicates that the rate of change eventually reaches a nonzero positive constant.

a single minimum. This means if we find the point  $k_{\min}$  that minimizes  $N(k)$ , we can easily find  $k_{\text{opt}}$ .

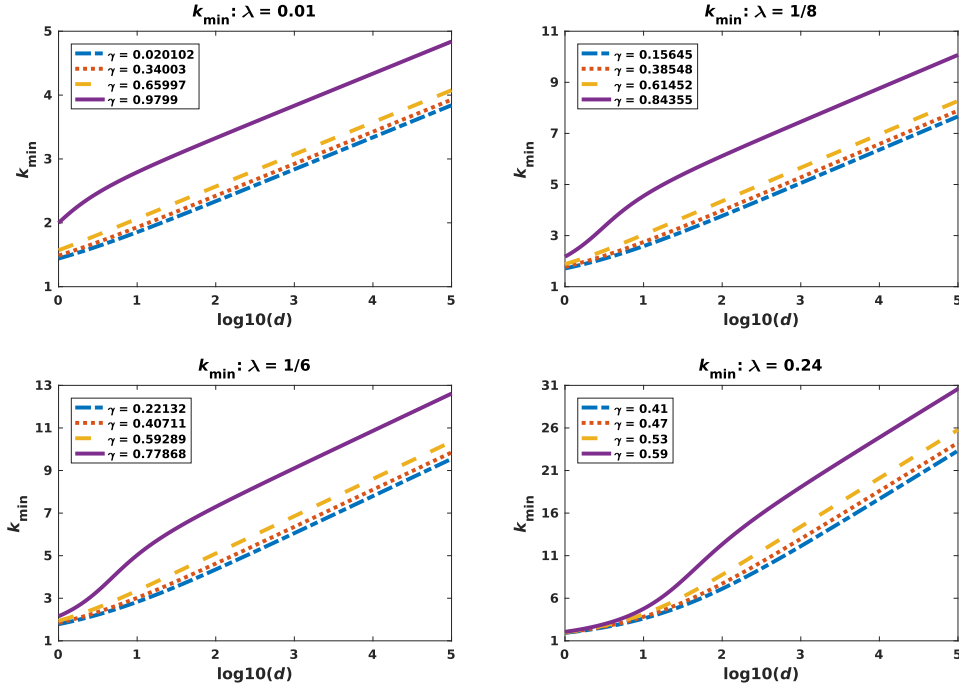
There is a good reason to running  $k_{\text{opt}} - 1$  circuits: if the depth  $k$  is too small, then we are stopping prematurely and not taking full advantage of the recovery ability of two-qubit stabilizer circuits; if  $k$  is too large, then we are putting more work into running the recovery than it is to start over.

#### 4.2.1 Optimal Depth: Generic Case

Given the nature of the expected cost functions from Lemma 15, we devote most of our efforts to answering  $k_{\text{opt}}$  for a protocol set to a restricted probability specification  $(\lambda, \gamma)$ . By the end, we propose that  $k_{\text{opt}}$  scales logarithmically with respect to the startup cost  $d$ . Let  $(\alpha, \beta)$  be the boundaries of  $(\lambda, \gamma)$ . Then the first derivative in its entirety is

$$N'(k) = - \frac{\ln(\alpha/\beta) ((\alpha - \beta)^2 d + (k - 1)(1 - 2\gamma))}{(\alpha - \beta) \left(1 - (\beta/\alpha)^{k-1}\right) \left((\alpha/\beta)^{k-1} - 1\right) \gamma} + \frac{\left(\alpha + (\beta/\alpha)^{k-1} \beta - \left(1 + (\beta/\alpha)^{k-1}\right) \gamma\right)}{(\alpha - \beta) \left(1 - (\beta/\alpha)^{k-1}\right) \gamma} \quad (22)$$

Seeing how  $N'(k)$  is transcendental, we rely on a combination of numerical and analytical approaches to justify our claim. A quick look at the limits of  $N'(k)$  reveals its behavior falls



■ **Figure 4** The data for  $k_{\min}$  suggests a protocol set to a restricted probability specification should use  $O(\log d)$  circuits to keep costs to a minimum, where  $d$  is the startup cost.

within our expectations. That is,  $N'(k) \rightarrow -\infty$  as  $k \rightarrow 1^+$  and

$$\lim_{k \rightarrow \infty} N'(k) = \lim_{k \rightarrow \infty} \frac{(\alpha + (\beta/\alpha)^{k-1} \beta - (1 + (\beta/\alpha)^{k-1}) \gamma)}{(\alpha - \beta) (1 - (\beta/\alpha)^{k-1}) \gamma} = \frac{\alpha - \gamma}{(\alpha - \beta) \gamma} > 0 \quad (23)$$

since  $\beta < \gamma < \alpha$ . The first term in Equation 22 also zeroes out as a consequence of  $\beta < \alpha$ . This is typical of a function with at least one minimum. If we let  $k' = k - 1$  and make some rearrangements, then we rewrite  $N'(k)$  as

$$N'(k') = - \frac{\ln(\alpha/\beta) \left( (\alpha - \beta)^2 d + (1 - 2\gamma) k' \right)}{(\alpha - \beta) \left( 1 - (\beta/\alpha)^{k'} \right) \left( (\alpha/\beta)^{k'} - 1 \right) \gamma} + \frac{(\alpha - \gamma) (\alpha/\beta)^{k'} + (\gamma - \beta) (\beta/\alpha)^{k'} - \alpha + \beta}{(\alpha - \beta) \left( 1 - (\beta/\alpha)^{k'} \right) \left( (\alpha/\beta)^{k'} - 1 \right) \gamma}. \quad (24)$$

We come up with a lower bound of  $N'(k')$  by dropping the term  $(\gamma - \beta)(\beta/\alpha)^{k'} \leq 1$ :

$$N'_{\text{lb}}(k') = \frac{(\alpha - \gamma) (\alpha/\beta)^{k'} - \alpha + \beta - \ln(\alpha/\beta) \left( (\alpha - \beta)^2 d + (1 - 2\gamma) k' \right)}{(\alpha - \beta) \left( 1 - (\beta/\alpha)^{k'} \right) \left( (\alpha/\beta)^{k'} - 1 \right) \gamma} \quad (25)$$

which may be used to locate an upper bound of  $k_{\min}$ . Starting with  $N'_{\text{lb}}(k') = 0$ , we get

$$\left( \frac{\alpha}{\beta} \right)^{k'} = \ln \left( \frac{\alpha}{\beta} \right) \left( \frac{1 - 2\gamma}{\alpha - \gamma} \right) k' + \frac{\ln(\alpha/\beta) (\alpha - \beta)^2 d + \alpha - \beta}{\alpha - \gamma}. \quad (26)$$

Making the substitution

$$-t = k' + \frac{\ln(\alpha/\beta) (\alpha - \beta)^2 d + \alpha - \beta}{\ln(\alpha/\beta) (1 - 2\gamma)} \quad (27)$$

turns Equation 26 into

$$t \left( \frac{\alpha}{\beta} \right)^t = -\frac{1}{t_0} \left( \frac{\alpha}{\beta} \right)^{-\frac{t_1}{t_0}} \quad (28)$$

where

$$t_0 = \ln \left( \frac{\alpha}{\beta} \right) \left( \frac{1 - 2\gamma}{\alpha - \gamma} \right), \quad t_1 = \frac{\ln(\alpha/\beta) (\alpha - \beta)^2 d + \alpha - \beta}{\alpha - \gamma}. \quad (29)$$

The solution  $t$  to Equation 28 indicates that

$$k_{\min} \leq k_{\text{up}} = -\frac{W \left( -\frac{\ln(\alpha/\beta)}{t_0} \left( \frac{\alpha}{\beta} \right)^{-\frac{t_1}{t_0}} \right)}{\ln(\alpha/\beta)} - \frac{t_1}{t_0} + 1 \quad (30)$$

where  $W$  is the Lambert  $W$  function. If in addition  $\gamma = 1/2$ , then  $N'(k') = 0$  is easier to solve, leading to

$$k_{\text{up}} = \frac{\ln \left( \ln(\alpha/\beta) (\alpha - \beta)^2 d + \alpha - \beta \right) - \ln(\alpha - 1/2)}{\ln(\alpha/\beta)} + 1. \quad (31)$$

Figure 4 contains plots of  $k_{\min}$  found using conventional optimization techniques. Aside from smaller values of the startup cost  $d$ , the graphs provide a compelling case that  $k_{\text{opt}} = O(\log d)$ . Equation 31 is a good starting point to begin a search for the exact value of  $k_{\text{opt}}$ .

#### 4.2.2 Optimal Depth: Special Case

The derivative of  $N(k)$  when  $\lambda = 1/4$  is much simpler by comparison:  $N'(k) = \frac{(k-1)^2 - d}{(k-1)^2}$ . The roots are  $1 \pm \sqrt{d}$ , of which only one is positive. From what we can gather, the optimal depth has a sublinear relationship with respect to the startup cost in both domains.

► **Theorem 16.** *Let  $d$  be the startup cost of a protocol set to a probability specification  $(\lambda, \gamma)$ . Then the optimal depth is  $k_{\text{opt}} = \min(\lceil 1 + \sqrt{d} \rceil, \lfloor 1 + \sqrt{d} \rfloor)$  when  $\lambda = 1/4$  and  $O(\log d)$  when  $(\lambda, \gamma)$  is a restricted probability specification.*

#### 4.3 Cost Ratio

To determine the effectiveness of our recovery, we compare  $N(2)$  – the method with no recovery whatsoever – against  $N(k_{\text{opt}})$ . We look at  $N(2)/N(k_{\text{opt}})$  under the assumptions of Theorem 16.

► **Theorem 17.** *Let  $k_{\text{opt}}$  be the optimal depth of a protocol with startup cost  $d$ . Then*

$$\lim_{d \rightarrow \infty} \frac{N(2)}{N(k_{\text{opt}})} \leq 2. \quad (32)$$

**Proof.** We consider a restricted probability specification  $(\lambda, \gamma)$  first. Let  $(\alpha, \beta)$  be its boundaries and let  $A_1(i)$ ,  $B_2(i)$  be its intermediate functions. Given that  $N(2) = (d+1)/\gamma$ , the exact ratio is

$$\frac{N(2)}{N(k)} = \frac{(d+1) A_1(k-1) (A_1(1))^2}{d A_1(k) (A_1(1))^2 + (k-1) B_2(k-1) A_1(1) + (\gamma - 2\lambda) A_1(k-1)}. \quad (33)$$

In addition to  $A_1(i) \leq 1$  and  $B_2(i) \leq 2$  for all integers  $i \geq 0$ , we can factor out  $\alpha^{k-1}$  from the top and bottom to say

$$\frac{N(2)}{N(k_{\text{opt}})} = \frac{(A_1(1))^2 \left(1 - (\beta/\alpha)^{k_{\text{opt}}-1}\right) (d+1)}{(A_1(1))^2 \left(1 - (\beta/\alpha)^{k_{\text{opt}}}\right) \alpha d + O(k_{\text{opt}})} \quad (34)$$

where we ignore lower order terms in the denominator. Since in this case  $k_{\text{opt}} = O(\log d)$  and  $\beta < \alpha$ , our conclusion now is more apparent:

$$\lim_{d \rightarrow \infty} \frac{(A_1(1))^2 \left(1 - (\beta/\alpha)^{O(\log d)}\right) \left(1 + \frac{1}{d}\right)}{(A_1(1))^2 \left(1 - (\beta/\alpha)^{O(\log d)}\right) \alpha + \frac{O(\log d)}{d}} = \frac{1}{\alpha}. \quad (35)$$

A protocol with uniform success probabilities  $L(i) = 1/2$  is very much the same. For simplicity, we use  $k_{\text{min}} = 1 + \sqrt{d}$ :

$$\lim_{d \rightarrow \infty} \frac{N(2)}{N(k_{\text{min}})} = \lim_{d \rightarrow \infty} \frac{2d\sqrt{d} + 2\sqrt{d}}{d\sqrt{d} + 2d + \sqrt{d}} = \frac{1}{\alpha} \quad (36)$$

since  $\alpha = 1/2$ . ◀

#### 4.4 Potential Improvements with Commonly Used Resource Qubits

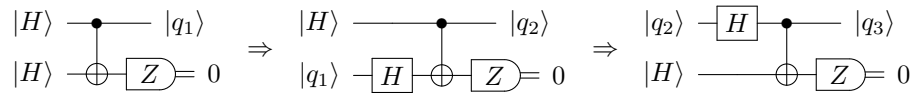
According to Theorem 17, the best scenario is when  $\lambda = 1/4$ , which translates to  $\alpha = 1/2$  and an expected cost reduction by up to half. We achieve this when performing phase rotations with a single CNOT and  $|\psi\rangle = |\theta\rangle = (|0\rangle + e^{i\theta})/\sqrt{2}$  at angles  $0 < \theta < \pi/2$  and  $\theta \neq \pi/4$ . The probability of rotating in either  $+\theta$  or  $-\theta$  direction is both  $1/2$ . An alternative to recovery is to try a correction with  $|2\theta\rangle$ . This shifts the cost to preparing  $|2\theta\rangle$  from two  $|\theta\rangle$  qubits but turns out to be actually less optimal. Observe that if we fail with  $|2\theta\rangle$ , then we need to prepare  $|2^2\theta\rangle$ , and so on up to some max power of 2 exponent  $j$ . Since the optimal depth is about  $\sqrt{d}$  for startup cost  $d$ , the gap between  $2^j$  and  $\sqrt{d}$  may be large, meaning this is worse than following the recovery protocol directly.

One particular example that may benefit are the  $V$ -basis gate implementations from [1]. For the non-Clifford operation

$$V_3 = \frac{1+2i}{\sqrt{5}} \begin{bmatrix} 1 & 0 \\ 0 & -\frac{3}{5} - i\frac{4}{5} \end{bmatrix}, \quad (37)$$

the idea is to inject  $|\theta_1\rangle$  such that  $\cos(\theta_1) = 7\sqrt{2}/10$  and  $\sin(\theta_1) = \sqrt{2}/10$ . Bocharov, Gurevich, and Svore [1] show that single qubit unitary approximations in the Clifford+ $V$  universal basis has the potential to be lower than Clifford+ $T$ , where  $T$  is the  $\pi/4$  phase rotation. If we have a long sequence of Clifford+ $V$  gates  $U_l \cdots U_1$ , then including recovery for the  $V$  gate implementations around  $U_l$  may prove helpful. More research is needed to determine one way or the other.

Previous work [21] also lists one concrete example in which the recovery protocol improves the average  $|H\rangle$  cost, where  $H|H\rangle = |H\rangle$  is a magic state. The procedure is provided in



■ **Figure 5** Approach to generate  $|q_3\rangle$  with three postselected circuits and four  $|H\rangle$  states as seen in [21]. Adding recovery for the last two-qubit circuit lowers the average  $|H\rangle$  usage.

Figure 5 for self-containment. In particular, the method without recovery uses 10.04  $|H\rangle$  qubits on average, but reduces slightly to about 9.45 with recovery. This represents a change of about 5.9%. If we now consider an even longer chain of postselected circuits to prepare an arbitrary resource  $\varphi_1$  from  $|H\rangle$  states, Theorem 17 says the savings grows more to about 17%. This is assuming  $(C_1, 0) \equiv (\text{CNOT}, 0)$  and  $|\psi\rangle = |H\rangle$  to yield  $\alpha \approx 0.8536$ . Direct use of

$$|\mathcal{T}\rangle\langle\mathcal{T}| = \frac{1}{2} \left[ I + \frac{1}{\sqrt{3}} (X + Y + Z) \right], \quad (38)$$

the  $+1$  eigenstate of  $e^{i\pi/4}PH$ , in  $(\text{CNOT}, 0)$  means  $1/\alpha \approx 1.267$ . But as far as we know, there are yet to be significant applications that directly use  $|\mathcal{T}\rangle$  besides to create  $|\pi/6\rangle$  [5]. This starts from  $|\mathcal{T}\rangle \otimes |\mathcal{T}\rangle$ , so our recovery operation is not beneficial in this use case.

## 5 Conclusion

We have proposed a protocol built on the recovery potential of two-qubit stabilizer circuits that has the capacity to lower the expected costs of obtaining some target qubit over the naive approach. To be of greater practical value, one direction of interest is how the protocol holds up in the face of noisy  $|\psi\rangle$ , since the errors may spread to  $\varphi_i$  and accumulate as it passes through each circuit  $C_i$ . A numerical study with  $|H\rangle$  states in [9] shows a decay for certain error rates, but whether this observation is retained for arbitrary non-stabilizer  $|\psi\rangle$  states is unknown. A related question is how the optimal depth is affected by the presence of errors, where we expect  $k_{\text{opt}}$  to decrease but by what amount.

In the long run, we predict our results are less likely to have a direct impact on current and future state distillation schemes, and are more suited toward resource intensive computations that require the injection of already finely distilled non-stabilizer states. Namely, that we have one resource qubit  $|\psi\rangle$ , and another relatively more costly  $\varphi_1$ , which may be entangled with another system and for which we have spent much effort to obtain. At the moment, we can only identify such setups to have any cost improvement when factoring in our approach. However we hope that our demonstration can serve as a starting point for an investigation into the reversibility of larger  $n$ -to- $k$  stabilizer circuits on arbitrary non-stabilizer states  $|\psi\rangle$ , and the viability of such operations for resource optimization.

## References

- 1 Alex Bocharov, Yuri Gurevich, and Krysta M. Svore. Efficient decomposition of single-qubit gates into  $V$  basis circuits. *Phys. Rev. A*, 88:012313, Jul 2013. doi:10.1103/PhysRevA.88.012313.
- 2 Alex Bocharov, Martin Roetteler, and Krysta M. Svore. Efficient synthesis of probabilistic quantum circuits with fallback. *Phys. Rev. A*, 91:052317, May 2015. doi:10.1103/PhysRevA.91.052317.
- 3 Alex Bocharov, Martin Roetteler, and Krysta M. Svore. Efficient Synthesis of Universal Repeat-Until-Success Quantum Circuits. *Phys. Rev. Lett.*, 114:080502, Feb 2015. doi:10.1103/PhysRevLett.114.080502.

- 4 Sergey Bravyi and Jeongwan Haah. Magic-state distillation with low overhead. *Phys. Rev. A*, 86:052329, Nov 2012. doi:10.1103/PhysRevA.86.052329.
- 5 Sergey Bravyi and Alexei Kitaev. Universal quantum computation with ideal clifford gates and noisy ancillas. *Phys. Rev. A*, 71:022316, Feb 2005. doi:10.1103/PhysRevA.71.022316.
- 6 Earl T Campbell and Joe O’Gorman. An efficient magic state approach to small angle rotations. *Quantum Science and Technology*, 1(1):015007, 2016. URL: <http://stacks.iop.org/2058-9565/1/i=1/a=015007>.
- 7 Peter G. Doyle and J. Laurie Snell. Random walks and electric networks, 2006. URL: <https://math.dartmouth.edu/~doyle/docs/walks/walks.pdf>.
- 8 Guillaume Duclos-Cianci and David Poulin. Reducing the quantum-computing overhead with complex gate distillation. *Phys. Rev. A*, 91:042315, Apr 2015. doi:10.1103/PhysRevA.91.042315.
- 9 Guillaume Duclos-Cianci and Krysta M. Svore. Distillation of nonstabilizer states for universal quantum computation. *Phys. Rev. A*, 88:042325, Oct 2013. doi:10.1103/PhysRevA.88.042325.
- 10 Austin G. Fowler, Matteo Mariantoni, John M. Martinis, and Andrew N. Cleland. Surface codes: Towards practical large-scale quantum computation. *Phys. Rev. A*, 86:032324, Sep 2012. doi:10.1103/PhysRevA.86.032324.
- 11 Jeongwan Haah, Matthew B. Hastings, D. Poulin, and D. Wecker. Magic State Distillation with Low Space Overhead and Optimal Asymptotic Input Count. *Quantum*, 1:31, Oct 2017. doi:10.22331/q-2017-10-03-31.
- 12 N Cody Jones, James D Whitfield, Peter L McMahon, Man-Hong Yung, Rodney Van Meter, Alán Aspuru-Guzik, and Yoshihisa Yamamoto. Faster quantum chemistry simulation on fault-tolerant quantum computers. *New Journal of Physics*, 14(11):115023, 2012. URL: <http://stacks.iop.org/1367-2630/14/i=11/a=115023>.
- 13 J.G. Kemény and J.L. Snell. *Finite markov chains*. University series in undergraduate mathematics. Springer-Verlag New York, 1976.
- 14 Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca. Fast and efficient exact synthesis of single qubit unitaries generated by Clifford and  $T$  gates. *Quantum Information and Computation*, 13(7-8):607–630, 2013. URL: <http://arxiv.org/abs/1206.5236>.
- 15 Andrew J. Landahl and Chris Cesare. Complex instruction set computing architecture for performing accurate quantum  $Z$  rotations with less magic, Feb 2013. arXiv:1302.3240.
- 16 Adam Meier, Bryan Eastin, and Emanuel Knill. Magic-state distillation with the four-qubit code. *Quantum Information and Computation*, 13:195–209, 2013. URL: <http://arxiv.org/abs/1204.4221>.
- 17 Ben Reichardt. Quantum universality by state distillation. *Quantum Information and Computation*, 9:1030–1052, 2009. URL: <http://arxiv.org/abs/quant-ph/0608085v2>.
- 18 Neil J. Ross. Optimal ancilla-free Clifford+ $V$  approximation of  $z$ -rotations. *Quantum Information and Computation*, 15(11-12):932–950, 2015. URL: <http://arxiv.org/abs/1409.4355>.
- 19 Neil J. Ross and Peter Selinger. Optimal ancilla-free Clifford+ $T$  approximation of  $z$ -rotations. *Quantum Information and Computation*, 16(11-12):901–953, 2016. URL: <http://www.rintonpress.com/xxqic16/qic-16-1112/0901-0953.pdf>.
- 20 Peter Selinger. Efficient Clifford+ $T$  approximation of single-qubit operators. *Quantum Information and Computation*, 15(1-2):159–180, 2015. URL: <http://arxiv.org/abs/1212.6253>.
- 21 Wim van Dam and Raymond Wong. Two-qubit Stabilizer Circuits with Recovery I: Existence, Mar 2018. arXiv:1803.06081.

## A Identities on Definition 10 Intermediate Functions

Appendix B relies heavily on the next lemma.

► **Lemma 18.** *Let  $(\lambda, \gamma)$  be a restricted probability specification. Then we have the following identities on its intermediate functions  $A_j(i)$  and  $B_j(i)$ :*

- i.  $A_j(i+1) = A_j(i) - \lambda A_j(i-1)$
- ii.  $B_j(i+1) = B_j(i) - \lambda B_j(i-1)$
- iii.  $A_1(j)A_1(i) = A_2(j+i) - \lambda^i A_2(j-i)$
- iv.  $A_2(j)A_1(i) = A_1(j+i) - \lambda^i A_1(j-i)$
- v.  $B_1(i)A_1(i+1)A_1(1) + \lambda B_1(2i) = B_1(2i+2) - 2\lambda^{i+1}A_1(1) + \gamma\lambda^i A_1(1)$
- vi.  $B_1(j-i)A_1(i) = B_2(j) - \lambda^i B_2(j-2i)$
- vii.  $\lambda^i A_2(j-2i)A_1(1) + A_2(j-i+1)A_1(i) = A_2(j-i)A_1(i+1)$
- viii.  $\lambda^i B_2(j-2i-1)A_1(1) + B_2(j-i)A_1(i) = B_2(j-i-1)A_1(i+1)$

**Proof.** Note that  $\lambda = \alpha\beta$  and  $A_2(1) = \alpha + \beta = 1$  for boundaries  $(\alpha, \beta)$  of  $(\lambda, \gamma)$ . The first equation is obvious from  $A_j(i) - \lambda A_j(i-1) = \alpha^i + (-1)^j \beta^i - \alpha^i \beta - (-1)^j \alpha \beta^i$ , and the second one follows immediately. The next two are just as easy to prove. The fifth identity looks a little more involved, but we just need to show

$$B_1(i)A_1(i+1) = A_1(i+1)A_1(i+1) - \gamma A_1(i+1)A_1(i) \quad (39)$$

$$= A_2(2i+2) - 2\lambda^{i+1} - \gamma A_2(2i+1) + \gamma\lambda^i \quad (40)$$

$$= B_2(2i+1) - 2\lambda^{i+1} + \gamma\lambda^i \quad (41)$$

$$B_2(2i+1)A_1(1) = A_2(2i+2)A_1(1) - \gamma A_2(2i+1)A_1(1) \quad (42)$$

$$= A_1(2i+3) - \lambda A_1(2i+1) - \gamma A_1(2i+2) + \gamma\lambda A_1(2i) \quad (43)$$

$$= B_1(2i+2) - \lambda B_1(2i) \quad (44)$$

and the result becomes clear. The following covers (vi):

$$B_1(j-i)A_1(i) = A_1(j-i+1)A_1(i) - \gamma A_1(j-i)A_1(i) \quad (45)$$

$$= A_2(j+1) - \lambda^i A_2(j-2i+1) - \gamma A_2(j) + \gamma\lambda^i A_2(j-2i) \quad (46)$$

$$= B_2(j) - \lambda^i B_2(j-2i) \quad (47)$$

while (vii) is based on (iv):

$$\begin{aligned} \lambda^i A_2(j-2i)A_1(1) + A_2(j-i+1)A_1(i) &= \alpha^i \beta^i (A_1(j-2i+1) - \alpha\beta A_1(j-2i-1)) \\ &\quad + A_1(j+1) - \alpha^i \beta^i A_1(j-2i+1) \end{aligned} \quad (48)$$

$$= A_1(j+1) - \alpha^{i+1} \beta^{i+1} A_1(j-2i-1) \quad (49)$$

$$= A_2(j-i)A_1(i+1). \quad (50)$$

The last one is a consequence of (vii). ◀

## B Random Walk Modeling of Depth $k$ Protocol

We model our depth  $k$  protocol as a 1-dimensional random walk on the integers  $\{0, \dots, k\}$ , with Equation 18 as the left step probability at each location on the number line. Every time we execute the protocol, we start a random walk at location  $i = 1$ . When we obtain  $\Phi_0(C_1, \varphi_1 \otimes \psi)$ , this represents a step onto the left boundary 0.

Random walk processes have been studied extensively in [7] and [13]. We borrow two functions from [7] to compute certain aspects about our protocol.

► **Definition 19** (success probability of random walk). Consider a random walk over the integers  $\{0, \dots, k\}$ . Define  $P(i)$  to be the probability that the walk, starting at  $i$ , successfully reaches 0 before it reaches  $k$ . Then the  $P(i)$  probabilities are described by

$$P(i) = \begin{cases} 1 & \text{if } i = 0 \\ 0 & \text{if } i = k \\ L(i)P(i-1) + (1-L(i))P(i+1) & \text{otherwise} \end{cases} \quad (51)$$

where  $L(i)$  is the probability of a left step from  $i$  to  $i-1$ .

► **Definition 20** (expected number of steps in random walk). Similar to Definition 19, define  $S(i)$  to be the expected number of steps that the walk, starting at  $i$ , takes to reach 0 or  $k$ . Then the  $S(i)$  expectations are described by

$$S(i) = \begin{cases} 0 & \text{if } i = 0 \text{ or } i = k \\ L(i)S(i-1) + (1-L(i))S(i+1) + 1 & \text{otherwise} \end{cases} \quad (52)$$

where  $L(i)$  is the probability of a left step from  $i$  to  $i-1$ .

We solve for the closed-form expressions of  $P(i)$  and  $S(i)$  with Equation 18 as the transition. Because of Lemma 11, there are two sets of solutions based on the nature of  $L(i)$ , which we present in Lemmas 27 and 28. We start with the general framework for computing  $P(i)$  and  $S(i)$  individually as it appears in [13].

A 1-dimensional random walk on integers  $\{0, \dots, k\}$  is also called an absorbing Markov chain, where the endpoints 0 and  $k$  serve as absorbing states. It has  $k-1$  transient (non-absorbing) states, and we may write down the transition matrix in canonical form as

$$\left[ \begin{array}{c|c} \overbrace{I}^2 & \overbrace{O}^{k-1} \\ \hline U & V \end{array} \right] \begin{matrix} \} 2 \\ \} k-1 \end{matrix} \quad (53)$$

where  $O$  contains all zeroes and  $I$  is the  $2 \times 2$  identity. Each row sums to 1, and the first and second rows represent transitions from the left and right boundaries. The block matrices  $U$  and  $V$  contain transition probabilities from transient to absorbing and transient to transient states, respectively. We arrange the rows and columns of  $V$  such that

$$V_{i,j} = \begin{cases} L(i) & \text{if } j = i-1 \\ R(i) & \text{if } j = i+1 \\ 0 & \text{otherwise} \end{cases} \quad (54)$$

where  $L(i)$  is the probability from  $i$  to  $i-1$  and  $R(i) = 1 - L(i)$ . It has nonzero entries only at places immediately adjacent to the main diagonal. The  $U$  matrix is mostly zeroes with the exception of two spots:  $U_{1,1} = L(1)$  and  $U_{k-1,2} = R(k-1)$ . As an example,

$$\left[ \begin{array}{cc|ccc} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ \hline L(1) & 0 & 0 & R(1) & 0 \\ 0 & 0 & L(2) & 0 & R(2) \\ 0 & R(3) & 0 & L(3) & 0 \end{array} \right] \quad (55)$$

is the canonical transition matrix of a random walk with  $k = 4$ .



At the heart of proving Lemma 27 is the inverse  $E = (I - V)^{-1}$  known as the *fundamental matrix*. According to [13], we may use  $E$  to obtain  $P(i) = (EU)_{i,1}$  and  $S(i) = (E\vec{1})_i$ , where  $\vec{1}$  is a column vector of ones. If  $S(i)$  is an expectation in the number of steps taken, then the variance is  $(2E - I)E\vec{1} - \text{Sq}(E\vec{1})$ , where  $\text{Sq}(E\vec{1})$  squares each entry of  $E\vec{1}$ . The fundamental matrix basically allows us to gather a number of meaningful statistics that we may want when evaluating a Markov chain.

The generic form of  $E = (I - V)^{-1}$  for the random walk can be found through various derivations, but regardless of which method we use, we may write an entry of the matrix in terms of the following recurrences:

$$F(i) = F(i - 1) - L(i)R(i - 1)F(i - 2), \quad F(0) = 1, \quad F(-1) = 0 \quad (56)$$

$$\bar{F}(i, k) = F(i + 1, k) - R(i)L(i + 1)\bar{F}(i + 2, k), \quad \bar{F}(k, k) = 1, \quad \bar{F}(k + 1, k) = 0. \quad (57)$$

The  $\bar{F}(i, k)$  function mirrors  $F(i)$ , with  $k$  acting as the base. To give an example, if  $k = 4$  and we start with

$$\left[ I - V \mid I \right] = \left[ \begin{array}{ccc|ccc} 1 & -R(1) & 0 & 1 & 0 & 0 \\ -L(2) & 1 & -R(2) & 0 & 1 & 0 \\ 0 & -L(3) & 1 & 0 & 0 & 1 \end{array} \right] \quad (58)$$

then Gaussian elimination eventually yields

$$E = \left[ \begin{array}{ccc} \frac{\bar{F}(2, 4)F(0)}{\bar{F}(1, 4)} & \frac{R(1)\bar{F}(3, 4)F(0)}{\bar{F}(1, 4)} & \frac{R(2)R(1)\bar{F}(4, 4)F(0)}{\bar{F}(1, 4)} \\ \frac{L(2)\bar{F}(3, 4)F(0)}{\bar{F}(1, 4)} & \frac{\bar{F}(3, 4)F(1)}{\bar{F}(1, 4)} & \frac{R(2)\bar{F}(4, 4)F(1)}{\bar{F}(1, 4)} \\ \frac{L(3)L(2)\bar{F}(4, 4)F(0)}{\bar{F}(1, 4)} & \frac{L(3)\bar{F}(4, 4)F(1)}{\bar{F}(1, 4)} & \frac{\bar{F}(4, 4)F(2)}{\bar{F}(1, 4)} \end{array} \right] \quad (59)$$

as our inverse. Substituting Equation 18 into the matrix leads to Lemma 23, but to realize this, we prove some identities on  $F(i)$  and  $\bar{F}(i, k)$  to make the algebra easier to handle later.

► **Lemma 21.** *Let  $F(i) = F(i - 1) - \alpha\beta F(i - 2)$  with initial conditions  $F(-1) = 0, F(0) = 1$  and positive real numbers  $\alpha, \beta$  such that  $\alpha + \beta = 1$ . Then*

$$F(i) = \sum_{j=0}^i \alpha^{i-j} \beta^j = \alpha^i + \alpha^{i-1}\beta + \dots + \alpha\beta^{i-1} + \beta^i. \quad (60)$$

Moreover,  $(\alpha - \beta)F(i) = \alpha^{i+1} - \beta^{i+1}$ .

**Proof.** We prove the lemma by induction on  $i$ . The base cases are trivial to see: the first is an empty sum, and the second is a single term. Assuming  $F(l)$  is true for all  $l < i$ , then

$$F(i) = (\alpha + \beta) \sum_{j=0}^{i-1} \alpha^{i-1-j} \beta^j - \alpha\beta \sum_{j=0}^{i-2} \alpha^{i-2-j} \beta^j \quad (61)$$

$$= \sum_{j=0}^{i-1} \alpha^{i-j} \beta^j + \beta \sum_{j=0}^{i-2} \alpha^{i-1-j} \beta^j - \beta \sum_{j=0}^{i-2} \alpha^{i-1-j} \beta^j + \beta^i \quad (62)$$

$$= \sum_{j=0}^i \alpha^{i-j} \beta^j. \quad (63)$$

For the second identity,

$$(\alpha - \beta) F(i) = \alpha^{i+1} + \sum_{j=0}^{i-1} \alpha^{i-j} \beta^{j+1} - \sum_{j=0}^{i-1} \alpha^{i-j} \beta^{j+1} - \beta^{i+1} = \alpha^{i+1} - \beta^{i+1} \quad (64)$$

which finishes the proof.  $\blacktriangleleft$

► **Lemma 22.** *Let  $\alpha, \beta$  be positive real numbers such that  $\alpha + \beta = 1$ . Let  $k \geq 2$  be an integer. Then the two recurrences*

$$F(i) = F(i-1) - \alpha\beta F(i-2), \quad F(0) = 1, \quad F(-1) = 0 \quad (65)$$

$$\bar{F}(i, k) = F(i+1, k) - \alpha\beta \bar{F}(i+2, k), \quad \bar{F}(k, k) = 1, \quad \bar{F}(k+1, k) = 0 \quad (66)$$

are related by  $\bar{F}(i, k) = F(k-i)$ .

**Proof.** The induction goes in decreasing values of  $i$ . Immediately, we see  $\bar{F}(k+1, k) = F(-1) = 0$  and  $\bar{F}(k, k) = F(0) = 1$ . Assuming  $\bar{F}(j, k) = F(k-j)$  holds for all  $j > i$ , then

$$\bar{F}(i, k) = \bar{F}(i+1, k) - \alpha\beta \bar{F}(i+2, k) \quad (67)$$

$$= F(k-(i+1)) - \alpha\beta F(k-(i+2)) \quad (68)$$

$$= F(k-i-1) - \alpha\beta F(k-i-2) = F(k-i). \quad (69)$$

This completes the proof.  $\blacktriangleleft$

Lemmas 23 and 24 describe what the fundamental matrix  $E$  will be in our application.

► **Lemma 23.** *Let  $L(i)$  be an rde on a restricted probability specification  $(\lambda, \gamma)$ . If  $L(i)$  determines the left step probabilities of a random walk over  $\{0, \dots, k\}$ , then the following describes the entries of the fundamental matrix  $E$ :*

$$E_{i,j} = \begin{cases} \frac{\lambda^{i-j} A_1(k-i) A_1(j) B_1(j-1)}{A_1(1) A_1(k) B_1(i-1)} & \text{if } i \geq j \\ \frac{A_1(k-j) A_1(i) B_1(j-1)}{A_1(1) A_1(k) B_1(i-1)} & \text{otherwise} \end{cases} \quad (70)$$

where  $A_1(i)$  and  $B_1(i)$  are intermediate functions of  $(\lambda, \gamma)$ .

**Proof.** Let  $(\alpha, \beta)$  be the boundaries of  $(\lambda, \gamma)$ . After we adapt the example matrix in Equation 59 with Equation 18, we check if what we get for  $E$  is in fact the inverse of  $I - V$  (the block matrix  $V$  comes from the canonical representation of the transition matrix).

The non-recursive formulas of  $L(i)$  and  $R(i)$  are

$$L(i) = \frac{\lambda B_1(i-2)}{B_1(i-1)}, \quad R(i) = \frac{B_1(i)}{B_1(i-1)}. \quad (71)$$

The pattern in Equation 59 suggests

$$E_{i,j} = \begin{cases} \frac{\bar{F}(j+1, k) F(i-1)}{\bar{F}(1, k)} \prod_{l=i}^{j-1} R(l) & \text{if } i < j \\ \frac{\bar{F}(i+1, k) F(i-1)}{\bar{F}(1, k)} & \text{if } i = j \\ \frac{\bar{F}(i+1, k) F(j-1)}{\bar{F}(1, k)} \prod_{l=j+1}^i L(l) & \text{if } i > j. \end{cases} \quad (72)$$

If we combine  $L(i)R(i-1) = \lambda = \alpha\beta$ , Lemma 21, Lemma 22, and

$$\prod_{l=j+1}^i L(l) = \frac{\lambda^{i-j} B_1(j-1)}{B_1(i-1)}, \quad \prod_{l=i}^{j-1} R(l) = \frac{B_1(j-1)}{B_1(i-1)} \quad (73)$$

then we obtain Equation 70 above.

We validate  $E_{i,j}$  as the last step in our proof. All rows and columns of  $I - V$  have at most three non-zero entries, all lying near the main diagonal. When we examine row  $i$  of  $I - V$  and column  $j$  of  $E$  such that  $i < j$ , we get

$$\begin{aligned} ((I - V)E)_{i,j} = & -\frac{\lambda B_1(i-2)}{B_1(i-1)} \frac{A_1(k-j)A_1(i-1)B_1(j-1)}{A_1(1)A_1(k)B_1(i-2)} \\ & + \frac{A_1(k-j)A_1(i)B_1(j-1)}{A_1(1)A_1(k)B_1(i-1)} \\ & - \frac{B_1(i)}{B_1(i-1)} \frac{A_1(k-j)A_1(i+1)B_1(j-1)}{A_1(1)A_1(k)B_1(i)} = 0 \end{aligned} \quad (74)$$

by Lemma 18(i). The special case  $i = 1 < j$  involves only two terms, but the result remains the same since  $A_1(2) = A_1(1)$ . The other situations follow similarly, where  $((I - V)E)_{i,j} = 1$  when  $i = j$  and  $((I - V)E)_{i,j} = 0$  when  $i > j$ . The same logic applies for  $E(I - V)$ . ◀

► **Lemma 24.** *The fundamental matrix  $E$  for a uniform random walk over  $\{0, \dots, k\}$  is*

$$E_{i,j} = \begin{cases} \frac{2(k-i)j}{k} & \text{if } i \geq j \\ \frac{2(k-j)i}{k} & \text{otherwise.} \end{cases} \quad (75)$$

**Proof.** We have  $F(i) = (i+1)/2^i$  as a consequence of  $\alpha = \beta = 1/2$  by Lemma 11. For  $i < j$ ,

$$E_{i,j} = \frac{\bar{F}(j+1, k)F(i-1)}{\bar{F}(1, k)} \prod_{l=i}^{j-1} R(l) = \frac{k-j}{2^{k-j-1}} \frac{i}{2^{i-1}} \frac{2^{k-1}}{k} \frac{1}{2^{j-i}} = \frac{(k-j)i}{2^{-1}k}. \quad (76)$$

The other case is similar. ◀

Given matrix  $E$ , we sum across row  $i$  to compute the expected steps  $S(i)$ . We separate the summation into two parts, one from column 1 to column  $i$  and another from  $i+1$  to  $k-1$ . We show a couple identities on these two smaller sums before the final proof.

► **Lemma 25.** *Let  $A_1(i)$  and  $B_1(i)$  be intermediate functions of a restricted probability specification  $(\lambda, \gamma)$ . Then for all integers  $i \geq 0$ ,*

$$J_1(i) = \sum_{j=0}^i \lambda^{i-j} B_1(j) A_1(j+1) = \frac{B_1(2i+2)}{A_1(1)} - ((2i+3)\lambda - (i+1)\gamma) \lambda^i. \quad (77)$$

**Proof.** Recognizing  $A_1(3) = A_2(2)A_1(1) + \lambda A_1(1)$  and  $A_1(2) = A_1(1)$ , we can show

$$J_1(0) = \frac{A_1(3) - \gamma A_1(2)}{A_1(1)} - 3\lambda + \gamma \quad (78)$$

$$= \frac{A_1(1)(A_2(2) + \lambda - \gamma)}{A_1(1)} - 3\lambda + \gamma \quad (79)$$

$$= A_2(2) - 2\lambda = A_1(1)A_1(1). \quad (80)$$

## 8:20 Recovery Circuits II: Analysis

This acts as a base case for an induction on  $J_1(i) = \lambda J_1(i-1) + B_1(i)A_1(i+1)$ . If we continue forward, then

$$J_1(i) = B_1(i)A_1(i+1) + \frac{\lambda B_1(2i)}{A_1(1)} - ((2i+1)\lambda - i\gamma)\lambda^i \quad (81)$$

$$= \frac{B_1(2i+2)}{A_1(1)} - 2\lambda^{i+1} + \gamma\lambda^i - (2i+1)\lambda^{i+1} + i\gamma\lambda^i \quad (82)$$

as a result of Lemma 18(v).  $\blacktriangleleft$

► **Lemma 26.** *Let  $A_1(i)$  and  $B_2(i)$  be intermediate functions of a restricted probability specification  $(\lambda, \gamma)$ . Let  $k \geq 2$  be an integer. Then for all integers  $i \geq 1$ ,*

$$J_2(i) = \sum_{j=1}^i B_1(k-j-1)A_1(j) = (i+1)B_2(k-1) - \frac{A_1(i+1)}{A_1(1)}B_2(k-i-1). \quad (83)$$

**Proof.** Again, we give a proof by induction. Starting with  $i = 1$ ,

$$J_2(1) = B_1(k-1-1)A_1(1) + B_2(k-1) - B_2(k-1) \quad (84)$$

$$= 2B_2(k-1) - \frac{A_1(1)(B_2(k-1) + \lambda B_2(k-3))}{A_1(1)} \quad (85)$$

$$= 2B_2(k-1) - \frac{A_1(2)}{A_1(1)}B_2(k-2) \quad (86)$$

using Lemma 18. Assuming  $J_2(j)$  is true for all  $j < i$ , let us look at

$$J_2(i) = B_1(k-i-1)A_1(i) + J_2(i-1) \quad (87)$$

$$= B_1(k-i-1)A_1(i) + iB_2(k-1) - \frac{A_1(i)}{A_1(1)}B_2(k-i). \quad (88)$$

By Lemma 18(vi), we end up with

$$J_2(i) = (i+1)B_2(k-1) - \lambda^i B_2(k-2i-1) - \frac{A_1(i)}{A_1(1)}B_2(k-i). \quad (89)$$

After gathering the last two terms under a common denominator, the numerator becomes

$$-\lambda^i B_2(k-2i-1)A_1(1) - B_2(k-i)A_1(i) = -B_2(k-i-1)A_1(i+1) \quad (90)$$

due to Lemma 18(viii).  $\blacktriangleleft$

We are ready to solve Equations 51 and 52.

► **Lemma 27.** *If the left step probabilities of a random walk over  $\{0, \dots, k\}$  are determined by an rde on a restricted probability specification  $(\lambda, \gamma)$ , then the following are solutions to Equations 51 and 52 of the random walk:*

$$P(i) = \frac{A_1(1)A_1(k-i)\gamma\lambda^{i-1}}{A_1(k)B_1(i-1)} \quad (91)$$

$$S(i) = \frac{A_1(k-i)(\gamma\lambda^{i-1} - 2\lambda^i)i + (k-i)A_1(i)B_2(k-1)}{A_1(1)A_1(k)B_1(i-1)} \quad (92)$$

where  $A_1(i)$  and  $B_j(i)$  are intermediate functions of  $(\lambda, \gamma)$ .

**Proof.** More formally,

$$S(i) = \sum_{j=1}^{k-1} E_{i,j} = \frac{A_1(k-i)J_1(i-1) + A_1(i)J_2(k-i-1)}{A_1(1)A_1(k)B_1(i-1)} \quad (93)$$

where  $J_1(i-1)$  and  $J_2(k-i-1)$  are defined in Lemmas 25 and 26. Note that  $J_2(k-i-1)$  starts the summation index from the right end of fundamental matrix  $E$  and moves inward. With the help of

$$A_1(i)B_2(i) = A_1(i)A_2(i+1) - \gamma A_1(i)A_2(i) \quad (94)$$

$$= A_1(2i+1) - \gamma A_1(2i) - \lambda^i A_1(1) \quad (95)$$

$$= B_1(2i) - \lambda^i A_1(1) \quad (96)$$

and Lemma 18, we arrive at

$$A_1(i)J_2(k-i-1) = (k-i)A_1(i)B_2(k-1) - \frac{A_1(k-i)}{A_1(1)}B_1(2i) + \lambda^i A_1(k-i). \quad (97)$$

Then combining it with

$$A_1(k-i)J_1(i-1) = \frac{A_1(k-i)}{A_1(1)}B_1(2i) - A_1(k-i)((2i+1)\lambda - i\gamma)\lambda^{i-1} \quad (98)$$

we see that a couple terms cancel out, leaving Equation 92 as desired.

The derivation of  $P(i)$  from  $E$  is easier to obtain. Recall that  $P(i) = (EU)_{i,1}$ , where  $U$  is a  $(k-1) \times 2$  matrix with  $U_{1,1} = \gamma$  and 0 for the rest of column 1. As such,

$$P(i) = \gamma E_{i,1} = \frac{A_1(k-i)A_1(1)B_1(0)\gamma\lambda^{i-1}}{A_1(1)A_1(k)B_1(i-1)} = \frac{A_1(1)A_1(k-i)\gamma\lambda^{i-1}}{A_1(k)B_1(i-1)} \quad (99)$$

since  $B_1(0) = A_1(1)$ . ◀

► **Lemma 28.** *The solutions to Equations 51 and 52 are  $P(i) = (k-i)/k$  and  $S(i) = ki - i^2$  for a uniform random walk over  $\{0, \dots, k\}$ .*

**Proof.** The solutions are already discussed in [7], but we can reach the same conclusion by way of Lemma 24. Accordingly,

$$S(i) = \sum_{j=1}^i \frac{2(k-i)j}{k} + \sum_{j=i+1}^{k-1} \frac{2(k-j)i}{k} \quad (100)$$

$$= \frac{2(k-i)(i+1)i}{k} + \frac{2i(k-i)(k-i-1)}{k} \quad (101)$$

$$= \frac{(i+1+k-i-1)(k-i)i}{k} = ki - i^2. \quad (102)$$

The  $P(i)$  solution is simpler to derive. ◀