13th Conference on the Theory of Quantum Computation, Communication and Cryptography

TQC 2018, July 16–18, 2018, Sydney, Australia

Edited by

Stacey Jeffery

Editor

Stacey Jeffery
QuSoft and CWI
Amsterdam
jeffery@cwi.nl

ACM Classification 2012
Theory of computation

ISBN 978-3-95977-080-4

Published online and open access by
Schloss Dagstuhl – Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken/Wadern, Germany. Online available at http://www.dagstuhl.de/dagpub/978-3-95977-080-4.

Publication date
July, 2018

Bibliographic information published by the Deutsche Nationalbibliothek
The Deutsche Nationalbibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data are available in the Internet at http://dnb.d-nb.de.

Digital Object Identifier: 10.4230/LIPIcs.TQC.2018.0

ISBN 978-3-95977-080-4 **ISSN 1868-8969** **http://www.dagstuhl.de/lipics**

LIPIcs – Leibniz International Proceedings in Informatics

LIPIcs is a series of high-quality conference proceedings across all fields in informatics. LIPIcs volumes are published according to the principle of Open Access, i.e., they are available online and free of charge.

ISSN 1868-8969

http://www.dagstuhl.de/lipics

Contents

13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2018).
Editor: Stacey Jeffery
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Quantum Ciphertext Authentication and Key Recycling with the Trap Code

Yfke Dulek
Qusoft, Centrum voor Wiskunde en Informatica, Amsterdam, the Netherlands
dulek@cwi.nl

Florian Speelman[1]
QMATH, Department of Mathematical Sciences, University of Copenhagen, Denmark
speelman@math.ku.dk

Abstract

We investigate quantum authentication schemes constructed from quantum error-correcting codes. We show that if the code has a property called *purity testing*, then the resulting authentication scheme guarantees the integrity of ciphertexts, not just plaintexts. On top of that, if the code is *strong* purity testing, the authentication scheme also allows the encryption key to be recycled, partially even if the authentication rejects. Such a strong notion of authentication is useful in a setting where multiple ciphertexts can be present simultaneously, such as in interactive or delegated quantum computation. With these settings in mind, we give an explicit code (based on the trap code) that is strong purity testing but, contrary to other known strong-purity-testing codes, allows for natural computation on ciphertexts.

2012 ACM Subject Classification Theory of computation → Cryptographic protocols, Theory of computation → Error-correcting codes, Security and privacy → Information-theoretic techniques, Security and privacy → Symmetric cryptography and hash functions, Theory of computation → Quantum information theory

Keywords and phrases quantum authentication, ciphertext authentication, trap code, purity-testing codes, quantum computing on encrypted data

Digital Object Identifier 10.4230/LIPIcs.TQC.2018.1

Related Version `https://arxiv.org/abs/1804.02237` (full version)

Acknowledgements We thank Gorjan Alagic, Christian Majenz, and Christian Schaffner for valuable discussions and useful input at various stages of this research. Aditionally, we thank Christian Schaffner for his comments on an earlier version of this manuscript.

1 Introduction

A central topic in cryptography is authentication: how can we make sure that a message remains unaltered when we send it over an insecure channel? How do we protect a file from being corrupted when it is stored someplace where adversarial parties can potentially access it? And, especially relevant in the current era of cloud computing, how can we let an untrusted third party compute on such authenticated data?

[1] European Research Council (ERC Grant Agreement no 337603), the Danish Council for Independent Research (Sapare Aude), Qubiz Quantum Innovation Center, and VILLUM FONDEN via the QMATH Centre of Excellence (Grant No. 10059)

13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2018).
Editor: Stacey Jeffery; Article No. 1; pp. 1:1–1:17
LIPICS Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Following extensive research on authentication of classical data, starting with the seminal work by Wegman and Carter [17], several schemes have been proposed for authenticating quantum states [5, 1, 6]. One notable such scheme is the trap code [6], an encoding scheme that surrounds the data with dummy qubits that function as *traps*, revealing any unauthorized attempts to alter the plaintext data. A client holding the classical encryption key can guide a third party in performing computations directly on the ciphertext by sending input-independent auxiliary quantum states that help bypass the traps, and updating the classical key during the computation. The result is an authenticated output ciphertext.

The trap code distinguishes itself from other quantum authentication schemes in two ways. First, individually-authenticated input qubits can be entangled during the computation, but still be de-authenticated individually. This contrasts for example the Clifford code [1], where de-authentication needs to happen simultaneously on all qubits that were involved in the computation, including any auxiliary ones. Second, the trap code allows for 'authenticated measurements': if a third party measures a ciphertext, the client can verify the authenticity of the result from the classical measurement outcomes only. It is not known how to perform authenticated measurements on other codes. These two qualities make the trap code uniquely suited for quantum computing on authenticated data. It was originally designed for its use in quantum one-time programs [6], but has found further applications in zero-knowledge proofs for QMA [7], and in quantum homomorphic encryption with verification [2].

The extraordinary structure of the trap code is simultaneously its weakness: an adversary can learn information about the secret key by altering the ciphertext in a specific way, and observing whether or not the result is accepted by the client. Thus, to ensure security after de-authentication, the key needs to be refreshed before another quantum state is authenticated. This need for a refresh inhibits the usefulness of the trap code, because computation on multiple qubits under the trap code requires these qubits to be authenticated under overlapping secret keys.

In recent years, several works have refined the original definition of quantum authentication by Barnum et al. [5]. The trap code is secure under the weakest of these definitions [10], where only authenticity of the plaintext is guaranteed. But, as argued, it is not under the stronger 'total authentication' [12], where no information about the key is leaked if the client accepts the authentication. As Portmann mentions in his work on authentication with key recycling in the abstract-cryptography framework [15], it is not even clear whether the trap code can be regarded as a scheme with *partial* key leakage, as defined in [12], because of the adaptive way in which it can be attacked. In a different direction, Alagic, Gagliardoni, and Majenz [3] define a notion of quantum ciphertext authentication (QCA), where also the integrity of the ciphertext is guaranteed, and not just that of the plaintext. Ciphertext authentication is incomparable with total authentication: neither one implies the other. Before the current work, it was unknown whether the trap code authenticates ciphertexts.

Barnum et al. [5] built schemes for authentication of quantum data based on quantum error-correcting codes that are *purity testing*, meaning that any bit or phase flip on the message is detected with high probability. Portmann [15], working in the abstract-cryptography framework, showed that if the underlying code satisfies a stronger requirement called *strong purity testing*, the resulting authentication scheme allows for complete key recycling in the accept case, and for partial key recycling in the reject case. The trap code can be seen as a purity-testing error-correcting code, but it is not strong purity testing. This is consistent with the observation that keys in the trap code cannot be recycled.

Quantum plaintext authentication with key recycling has been studied before. Oppenheim and Horodecki [14] showed partial key recycling for schemes based on purity testing codes,

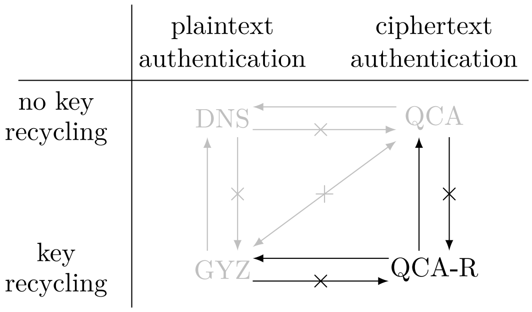


Figure 1 Overview of different definitions of quantum authentication. Three previously defined notions (in gray) and their relations were already known: DNS [10] is strictly weaker than GYZ [12] (total authentication) and QCA [3]. These last two are incomparable: there exist schemes that satisfy either one, but not the other. On the bottom right, our new definition QCA-R is displayed: it is strictly stronger than both GYZ and QCA. For justifications of the relations displayed in this figure, refer to pages 6 (for DNS ↛ GYZ), 6 (for DNS ↛ QCA), 6 (for GYZ ↮ QCA), and 9 (for GYZ ↛ QCA-R and QCA ↛ QCA-R).

under a weaker notion of security. Hayden, Leung, and Mayers [13] adapted Barnum et al.'s construction to use less key and show its authenticating properties in the universal-composability framework. Fehr and Salvail [11] develop a quantum authentication scheme for classical messages that achieves the same key-recycling rate as Portmann [15], but is not based on quantum error-correction and only requires the client to prepare and measure.

1.1 Our contributions

We investigate the relation between (strong) purity testing and quantum ciphertext authentication (QCA), and give a variation on the trap code with stronger security guarantees. We specify our contributions in more detail below.

Section 3: Definition of quantum ciphertext authentication with key recycling *(QCA-R)*. We give a new definition for quantum authentication, QCA-R, that provides both ciphertext authentication and key recycling, and is thereby strictly stronger than existing definitions. See Figure 1 for a comparison of different notions of authentication.

Section 3.1: Purity-testing codes give rise to QCA-secure encryption. We prove that Barnum et al.'s canonical construction of authentication schemes from purity-testing codes [5] produces schemes that are not only plaintext authenticating, but also ciphertext authenticating (QCA). The proof generalizes the proofs in [8] that the trap code and Clifford code are plaintext authenticating, using a different (but still efficient) simulator. Note that our result immediately implies that the trap code is ciphertext authenticating.

Section 3.1: Strong-purity-testing codes give rise to QCA-R-secure encryption. Purity-testing codes are generally not sufficient for constructing QCA-R schemes, but strong-purity-testing codes are: we prove that Barnum et al.'s canonical construction achieves QCA-R when a strong-purity-testing code is used as a resource. In case the authenticated message is accepted, the entire key can be reused. Otherwise, all but the quantum-one-time-pad key can be reused.

Section 4: A strong-purity-testing version of the trap code. We give an explicit construction of a strong-purity-testing code that is inspired by the trap code. In this *strong trap code*, the underlying error-correcting code is not only applied to the data qubits, but also to the trap qubits. The result is a quantum authentication scheme which satisfies the strong notion of QCA-R, but still maintains the computational properties that make the original trap code such a useful scheme.

Section 5: Security under parallel encryption. To illustrate the power of recycling key in the reject case, we consider a setting with a different type of key reuse: reusing (part of) a key immediately to authenticate a second qubit, even before the first qubit is verified. We show that, if multiple qubits are simultaneously authenticated using a scheme that is based on a strong-purity-testing code, then de-authenticating some of these qubits does not jeopardize the security of the others, even if their keys overlap. This property is especially important when using the computational capabilities of the strong trap code, since computing on authenticated qubits needs multiple qubits to use overlapping keys.

2 Preliminaries

2.1 Notation

We use conventional notation for unitary matrices (U or V), pure states ($|\psi\rangle$ or $|\varphi\rangle$), and mixed states (ρ or σ). We reserve the symbol τ for the completely mixed state $\mathbb{I}/d$, and $|\Phi^+\rangle$ for the EPR pair $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$. The m-qubit Pauli group is denoted with $\mathbb{P}_m$, and its elements with P_ℓ where ℓ is a $2m$-bit string indicating the bit-flip and phase-flip positions. By convention, P_0 is identity. The X-weight (, Y-weight, or Z-weight) of a Pauli is the number of qubits on which it acts as an X (, Y, or Z) operation.

We often specify the register(s) on which a unitary acts by gray superscripts (as in U^R); it is implicit that the unitary acts as identity on all other registers. The trace norm of a density matrix ρ is written as $\|\rho\|_{\text{tr}}$. The diamond norm of a channel Ψ is written as $\|\Psi\|_\diamond := \sup_\rho \|(\mathbb{I} \otimes \Psi)(\rho)\|_{\text{tr}}$. If we want to talk about the distance between two channels Ψ and Ψ', we use the normalized quantity $\frac{1}{2}\|\Psi - \Psi'\|_\diamond$, which we refer to as the *diamond-norm distance.*

2.2 Quantum authentication

A secret-key quantum encryption scheme consists of three (efficient) algorithms: key generation KeyGen, encryption $\mathsf{Encrypt}_k$, and decryption $\mathsf{Decrypt}_k$. Throughout this work, we will assume that KeyGen selects a key k uniformly at random from some set $\mathcal{K}$; our results still hold if the key is selected according to some other distribution. By Lemma B.9 in [4], we can characterize the encryption and decryption maps as being of the form

$$\mathsf{Encrypt}_k : \rho^M \mapsto U_k^{MT}(\rho \otimes \sigma_k^T)(U_k^\dagger)^{MT}, \tag{1}$$

$$\begin{aligned} \mathsf{Decrypt}_k : \rho^{MT} \mapsto \mathrm{Tr}_T\left[(\Pi_k^{\mathsf{acc}})^T \left(U_k^\dagger \rho U_k^{MT}\right)(\Pi_k^{\mathsf{acc}})^T\right] \\ + D_k^{MT}\left[(\Pi_k^{\mathsf{rej}})^T \left(U_k^\dagger \rho U_k^{MT}\right)(\Pi_k^{\mathsf{rej}})^T\right]. \end{aligned} \tag{2}$$

Here, M is the message register, σ_k is some key-dependent *tag* state in register T, and U_k is a unitary acting on both. Π_k^{acc} and Π_k^{rej} are orthogonal projectors onto the support of σ_k and its complement, respectively. Finally, D_k is any channel: we will usually assume that $D_k(\cdot) = \mathrm{Tr}_{MT}(\cdot) \otimes |\bot\rangle\langle\bot|^M$, i.e., it traces out the message and tag register entirely, and replaces the message with some dummy state that signifies a reject. Because of the above characterization, we will often talk about encryption schemes as a keyed collection $\{(U_k, \sigma_k)\}_{k\in\mathcal{K}}$ of unitaries and tag states.

There are several definitions of the authentication of quantum data. All definitions involve some parameter ε; unless otherwise specified, we require ε to be negligibly small in the size of the ciphertext.

The simplest definition is that of plaintext authentication, presented in [10] (although their definition was in phrased terms of the trace norm), where no guarantees are given about the recyclability of the key.

▶ **Definition 1** (Quantum plaintext authentication (DNS) [10]). A quantum encryption scheme $\{(U_k, \sigma_k)\}_{k\in\mathcal{K}}$ is *plaintext ε-authenticating* (or ε-DNS) if for all CP maps $\mathcal{A}$ (acting on the message register M, tag register T, and a side-information register R), there exist CP maps $\mathcal{S}_{\mathsf{acc}}$ and $\mathcal{S}_{\mathsf{rej}}$ such that $\mathcal{S} := \mathcal{S}_{\mathsf{acc}} + \mathcal{S}_{\mathsf{rej}}$ is trace-preserving, and

$$\frac{1}{2}\left\| \mathop{\mathbb{E}}_{k} \left[\mathsf{Decrypt}_k \circ \mathcal{A}^{MTR} \circ \mathsf{Encrypt}_k\right]^{MR} - \left(\mathbb{I}^{M} \otimes \mathcal{S}^{R}_{\mathsf{acc}} + |\bot\rangle\langle\bot|^{M}(\mathrm{Tr}_M \otimes\ \mathcal{S}^{R}_{\mathsf{rej}})\right)\right\|_{\diamond} \leq \varepsilon,$$

where $\mathsf{Encrypt}_k$ and $\mathsf{Decrypt}_k$ are of the form of equations (1) and (2).

The simulator in Definition 1 reflects the ideal functionality of an authentication scheme: in the accept case, the message remains untouched, whereas in the reject case, it is completely discarded and replaced with the fixed state $|\bot\rangle\langle\bot|$. Any action on the side-information register R is allowed.

▶ **The trap code.** An example of a plaintext-authenticating scheme is the trap code [6]. This scheme encrypts single-qubit messages by applying a fixed distance-d CSS code E to the message, producing n physical qubits, and then appending $2n$ "trap" qubits (n computational-basis traps in the state $|0\rangle\langle 0|$, and n Hadamard-basis traps in the state $|+\rangle\langle +|$). The resulting $3n$ qubits are permuted in a random fashion according to a key k_1, and one-time padded with a second key k_2. At decryption, the one-time pad and permutation are removed, the traps are measured in their respective bases, and the syndrome of the CSS code is checked.[2] The trap code, for a key $k = (k_1, k_2)$, is characterized by $U_k = P_{k_2}\pi_{k_1}(E \otimes \mathsf{I}^{\otimes n} \otimes \mathsf{H}^{\otimes n})$ and $\sigma_k = |0\rangle\langle 0|^{\otimes(3n-1)}$, where π_{k_1} is a unitary that permutes the $3n$ qubits. A proof that the trap code is plaintext $({}^2\!/_3)^{d/2}$-authenticating can be found in e.g. [8].

Another definition of quantum authentication is presented in [12] (where it is called 'total authentication'): in this definition, the key should be recyclable in the accept case. This is modeled by revealing the key to the environment after use, and requiring that it is indistinguishable from a completely fresh and uncorrelated key. If that is the case, it can be recycled for another round.

▶ **Definition 2** (Quantum plaintext authentication with key recycling (GYZ) [12]). A quantum encryption scheme $\{(U_k, \sigma_k)\}_{k\in\mathcal{K}}$ is *plaintext ε-authenticating with key recycling* (or ε-GYZ) if for all CP maps $\mathcal{A}$ (acting on the message register M, tag register T, and a side-information register R), there exist CP maps $\mathcal{S}_{\mathsf{acc}}$ and $\mathcal{S}_{\mathsf{rej}}$ such that $\mathcal{S} := \mathcal{S}_{\mathsf{acc}} + \mathcal{S}_{\mathsf{rej}}$ is trace preserving,

[2] We differ from the analysis by Broadbent and Wainewright [8] in that we consider the variant that uses error detection instead of error correction on the data qubits.

and

$$\frac{1}{2}\left\| \mathop{\mathbb{E}}_{k}\left[\rho^{MR} \mapsto \mathrm{Tr}_T\left(\Pi_k^{\mathsf{acc}} U_k^\dagger\left(\mathcal{A}^{MTR}\left(U_k(\rho\otimes\sigma_k^T)U_k^\dagger\right)\right)U_k\Pi_k^{\mathsf{acc}}\right)\otimes|k\rangle\langle k|\right] - \left(\mathbb{I}^M\otimes\mathcal{S}^R_{\mathsf{acc}}\otimes\tau_{\mathcal{K}}\right)\right\|_\diamond \le \varepsilon.$$

Note that Definition 2 only specifies what should happen in the accept case. Nevertheless, it is a strictly stronger definition than DNS authentication [4].

The trap code is not plaintext ε-authenticating with key recycling for sub-constant ε. To see this, consider an adversary $\mathcal{A}$ that applies X to (only) the first qubit of the MT register. With probability $2/3$, the attack lands on a data qubit or a $|0\rangle\langle 0|$ trap, and is detected. Thus, in the real accept scenario, the key register will contain a mixture of only those keys that permute a $|+\rangle\langle +|$ into the first position. All other keys are diminished by the projector Π_k^{acc}. Since the ideal scenario contains a mixture of *all* possible keys in the key register, the difference between the two channels is considerable. In practice, if an adversary learns whether the authentication succeeded, she gets information about the positions of the traps.

▶ **The Clifford code.** A simple yet powerful code that authenticates plaintexts with key recycling is the Clifford code [1]. In this code, we fix a parameter t, and set $\sigma_k = |0^t\rangle\langle 0^t|$ for all k, and U_k a uniformly random Clifford on $t+1$ qubits. The Clifford code (and any authentication code that is based on a 2-design) is plaintext ε-authenticating with key recycling for $\varepsilon = O(2^{-t})$ [4].

Strengthening Definition 1 in a different direction, Alagic, Gagliardoni, and Majenz recently introduced the notion of quantum ciphertext authentication [3]. This notion does not limit the amount of key leaked, but requires that if authentication accepts, the entire *ciphertext* was completely untouched. Ciphertext authentication is used as an ingredient for quantum encryption that is secure against chosen-ciphertext attacks [3].

▶ **Definition 3** (Quantum ciphertext authentication (QCA) [3])**.** A quantum encryption scheme $\{(U_k, \sigma_k = \sum_r p_{k,r}|\varphi_{k,r}\rangle\langle\varphi_{k,r}|)\}_{k\in\mathcal{K}}$ is *ciphertext ε-authenticating* (or ε-QCA) if it is plaintext ε-authenticating as in Definition 1, and the accepting simulator $\mathcal{S}_{\mathsf{acc}}$ is of the form

$$\mathcal{S}_{\mathsf{acc}} : \rho^R \mapsto \mathop{\mathbb{E}}_{k',r}\left[\langle\varphi_{k',r}|^T\langle\Phi^+|^{M_1M_2}U_{k'}^\dagger\mathcal{A}^{M_1TR}\left(U_{k'}^{M_1T}\rho_{k',r}^{RM_1M_2T}U_{k'}^\dagger\right)U_{k'}|\varphi_{k',r}\rangle|\Phi^+\rangle\right].$$

where $\rho_{k',r} := \rho^R \otimes |\Phi^+\rangle\langle\Phi^+|^{M_1M_2} \otimes |\varphi_{k',r}\rangle\langle\varphi_{k',r}|^T$ is the input state before (simulated) encryption.

In QCA, the accepting simulator tests whether the message remains completely untouched by encrypting half of an EPR pair (stored in register M_1) as a 'dummy message', under a key k' that it generates itself. It remembers the randomness r used in creating the tag state σ_k, so that it can test very accurately whether the tag state was untouched. Because $\mathcal{S}_{\mathsf{acc}}$ remembers the randomness, a scheme that appends a qubit at the end of its ciphertexts, but never checks its state at decryption time, cannot be ciphertext authenticating. The Clifford code *is* QCA [3], as is the trap code (see Section 3.1).

In general, key recycling as in Definition 2 does not imply QCA. To see this, take any scheme $\{(U_k, \sigma_k)\}_{k\in\mathcal{K}}$ that is plaintext authenticating with key recycling, and alter it by appending a qubit in the fully mixed state to σ_k (and extending U_k to act as identity on this qubit). This scheme still satisfies Definition 2, but cannot be ciphertext authenticating, because attacks on this last qubit are not noticed in the real scenario. Conversely, not all ciphertext-authenticating schemes have key recycling. Take any scheme that is QCA, and alter it by adding one extra bit b of key, and setting $\sigma_{kb} := \sigma_k \otimes |b\rangle\langle b|$ and $U_{kb} := U_k \otimes \mathsf{I}$, effectively appending the bit of key at the end of the ciphertext. This scheme still satisfies Definition 3, but leaks at least one bit of key.[3] For an overview of the relations between DNS, GYZ, and QCA, refer to Figure 1 on page 3.

2.3 (Strong) purity testing in quantum error correction

An $[[n, m]]$ quantum error-correcting code (QECC), characterized by a unitary operator V, encodes a message ρ consisting of m qubits into a codeword $V(\rho \otimes |0^t\rangle\langle 0^t|)V^\dagger$ consisting of n qubits, by appending $t := n - m$ tags $|0\rangle\langle 0|$, and applying the unitary V. Decoding happens by undoing the unitary V, and measuring the tag register in the computational basis. The measurement outcome is called the syndrome: an all-zero syndrome indicates that no error-correction is necessary. In this work, we will only use the error-detection property of QECCs, and will not worry about how to correct the message if a non-zero syndrome is measured. If that happens, we will simply discard the message (i.e., reject).

For any bit string $x \in \{0,1\}^m$, let $|x_L\rangle$ (for “logical $|x\rangle$") denote a valid encoding of $|x\rangle$, i.e., a state that will decode to $|x\rangle$ without error. A defining feature of any QECC is its distance: the amount of bit and/or phase flips required to turn one valid codeword into another. If we want to be explicit about the distance d of an $[[n, m]]$ code, we will refer to it as an $[[n, m, d]]$ code.

▶ **Definition 4** (Distance). The *distance* of an $[[n, m]]$ code is the minimum weight of a Pauli P such that $P|x_L\rangle = |y_L\rangle$ for some $x \neq y$, with $x, y \in \{0,1\}^m$.

In a cryptographic setting, it can be useful to select a code from a set of codes $\{V_k\}_{k\in\mathcal{K}}$ for some key set $\mathcal{K}$. We will again assume that the key k is selected uniformly at random.

Following [5] and [15], we restrict our attention to codes for which applying a Pauli to a codeword is equivalent to applying a (possibly different) Pauli directly to the message and tag register. In other words, the unitary V must be such that for any $P_\ell \in \mathbb{P}_{m+t}$, there exists a $P_{\ell'} \in \mathbb{P}_{m+t}$ and a $\theta \in \mathbb{R}$ such that $P_\ell V = e^{i\theta} V P_{\ell'}$. With our attention restricted to codes with this property, we can meaningfully define the following property:

▶ **Definition 5** (Purity testing [5]). A set of codes $\{V_k\}_{k\in\mathcal{K}}$ is *purity testing* with error ε if for any Pauli $P_\ell \in \mathbb{P}_{m+t} \backslash \{\mathsf{I}^{\otimes(m+t)}\}$,

$$\Pr_k \left[V_k^\dagger P_\ell V_k \in (\mathbb{P}_m \backslash \{\mathsf{I}^{\otimes m}\}) \otimes \{\mathsf{I}, \mathsf{Z}\}^{\otimes t} \right] \leq \varepsilon.$$

In words, for any non-identity Pauli, the probability (over the key) that the Pauli alters the message but is not detected (i.e., no tag bit is flipped) is upper bounded by ε.

The trap code (see page 5) based on an $[[n, 1, d]]$ CSS code, without the final quantum one-time pad, is a purity-testing code with error $({}^2\!/\!_3)^{d/2}$ [6]. In our framework, the trap code is described as a QECC with $m = 1$, $t = 3n - 1$, and $V_k = \pi_k(E \otimes \mathsf{I}^{\otimes n} \otimes \mathsf{H}^{\otimes n})$.

Note that purity-testing codes do not necessarily detect *all* Pauli attacks with high probability: it may well be that a Pauli attack remains undetected, because it acts as identity on the message. Flipping the first bit of a trap-code ciphertext is an example of

[3] We thank Gorjan Alagic and Christian Majenz for providing these example schemes that show the separation between Definitions 2 and 3.

such an attack: it remains undetected with probability 1/3 (if it hits a $|+\rangle$ trap), but unless it is detected, it also does not alter the message. An attacker may use this fact to learn information about the permutation π_k by observing whether or not the QECC detects an error.

The above exploitation of purity-testing codes has led Portmann to consider a stronger notion of purity testing that should allow for keys to be safely reusable. In this definition, even the Paulis that act as identity on the message should be detected:

▶ **Definition 6** (Strong purity testing [15]). A set of codes $\{V_k\}_{k\in\mathcal{K}}$ is *strong purity testing* with error ε if for any Pauli $P_\ell \in \mathbb{P}_{m+t}\backslash\{\mathsf{I}^{\otimes(m+t)}\}$,

$$\Pr_k\left[V_k^\dagger P_\ell V_k \in \mathbb{P}_m \otimes \{\mathsf{I},\mathsf{Z}\}^{\otimes t}\right] \leq \varepsilon.$$

The Clifford code is strong purity testing with error 2^{-t}, as is any other unitary 2-design [15]. As informally discussed above, the trap code is not strong purity testing for any small ε.

Barnum et al. [5] described a canonical method of turning a QECC set $\{V_{k_1}\}_{k_1\in\mathcal{K}_1}$ into a symmetric-key encryption scheme. The encryption key k consists of two parts: the key $k_1 \in \mathcal{K}_1$ for the QECC, and an additional one-time pad key $k_2 \in \{0,1\}^{2(m+t)}$. The encryption map is then defined by setting $U_{k_1,k_2} := P_{k_2}V_{k_1}$, and $\sigma_{k_1,k_2} = |0^t\rangle\langle 0^t|$. Since σ_{k_1,k_2} is key-independent, the projectors $\Pi^{\mathsf{acc}} = |0^t\rangle\langle 0^t|$ and $\Pi^{\mathsf{rej}} = \mathbb{I} - |0^t\rangle\langle 0^t|$ are key-independent as well. In Construction 1, the complete protocol is described. In [6], protocols of this form are called "encode-encrypt schemes".

■ **Construction 1** Barnum et al.'s canonical construction [5] of a symmetric-key encryption scheme from an $[[m+t,m]]$ quantum error-correcting code $\{V_{k_1}\}_{k_1\in\mathcal{K}_1}$.

```
Generate keys: sample $k_1 \leftarrow \mathcal{K}_1$ and $k_2 \leftarrow \mathcal{K}_2 = \{0,1\}^{2(m+t)}$.
Encrypt: $\rho^M \mapsto P_{k_2}^{MT} V_{k_1}^{MT} (\rho^M \otimes |0^t\rangle\langle 0^t|^T) V_{k_1}^{MT} P_{k_2}^{MT}$.
Decrypt: $\rho^{MT} \mapsto \langle 0^t| \left(V_{k_1}^\dagger P_{k_2}^\dagger \rho P_{k_2} V_{k_1}\right) |0^t\rangle + |\bot\rangle\langle\bot|^M \otimes \mathrm{Tr}_M \left[\sum_{i\neq 0^t} \langle i| \left(V_{k_1}^\dagger P_{k_2}^\dagger \rho P_{k_2} V_{k_1}\right) |i\rangle\right]$
```

When using Construction 1 with a strong-purity-testing code, plaintext authentication with key recycling is achieved, even with partial key recycling in the reject case [15]. If just a purity-testing code is used for the construction, the resulting encryption scheme is plaintext authenticating [5], but not necessarily with key recycling (the trap code is a counterexample).

3 Quantum ciphertext authentication with key recycling (QCA-R)

In this section, we will define a notion of quantum authentication that is stronger than all of Definitions 1, 2, and 3. We will show that Construction 1, when used with a strong-purity-testing code, results in an authentication scheme in this new, stronger sense.

▶ **Definition 7** (Quantum ciphertext authentication with key recycling (QCA-R)). A quantum encryption scheme $\{(U_k, \sigma_k = \sum_r p_{k,r}|\varphi_{k,r}\rangle\langle\varphi_{k,r}|)\}_{k\in\mathcal{K}}$ is *ciphertext ε-authenticating with key recycling* (or ε-QCA-R), with key recycling function f, if for all CP maps $\mathcal{A}$ (acting on the message register M, tag register T, and a side-information register R), there exists a CP map $\mathcal{S}_{\mathsf{rej}}$ such that

$$\begin{aligned}\mathfrak{R} : \rho^{MR} \mapsto \mathbb{E}_k\Big[&\mathrm{Tr}_T\left(\Pi^{acc}\left(U_k^\dagger \mathcal{A}^{MTR}\left(U_k^{MT}(\rho\otimes\sigma_k^T)U_k^\dagger\right)U_k\right)\Pi^{acc}\right)\otimes|k\rangle\langle k|\\ &+|\bot\rangle\langle\bot|^M\otimes\ \mathrm{Tr}_{MT}\left(\Pi^{rej}\left(U_k^\dagger \mathcal{A}^{MTR}\left(U_k^{MT}(\rho\otimes\sigma_k^T)U_k^\dagger\right)U_k\right)\Pi^{rej}\right)\otimes|f(k)\rangle\langle f(k)|\Big]\end{aligned}$$

is ε-close in diamond-norm distance to the ideal channel,

$$\mathfrak{I} : \rho^{MR} \mapsto \left(\mathbb{I}^{M} \otimes \mathcal{S}^{\mathsf{acc}}\right)\left(\rho^{MR}\right) \otimes \tau_{\mathcal{K}} \quad + \quad |\bot\rangle\langle\bot|^{M} \otimes \ \mathcal{S}^{\mathsf{rej}}(\rho^{R}) \otimes \mathbb{E}_{k}\left[|f(k)\rangle\langle f(k)|\right],$$

where $\mathcal{S} := \mathcal{S}_{\mathsf{acc}} + \mathcal{S}_{\mathsf{rej}}$ is trace preserving, and $\mathcal{S}_{\mathsf{acc}}$ is as in Definition 3 of QCA, that is,

$$\mathcal{S}_{\mathsf{acc}} : \rho^{R} \mapsto \mathop{\mathbb{E}}_{k',r}\left[\langle\varphi_{k',r}|^{T}\langle\Phi^{+}|^{M_1M_2}U_{k'}^{\dagger}\mathcal{A}^{M_1TR}\left(U_{k'}^{M_1T}\rho_{k',r}^{RM_1M_2T}U_{k'}^{\dagger}\right)U_{k'}|\varphi_{k',r}\rangle|\Phi^{+}\rangle\right]$$

for $\rho_{k',r} := \rho^{R} \otimes |\Phi^{+}\rangle\langle\Phi^{+}|^{M_1M_2} \otimes |\varphi_{k',r}\rangle\langle\varphi_{k',r}|^{T}$.

The first condition (closeness of the real and ideal channel) is a strengthening of Definition 2: following Portmann [15], we also consider which part of the key can be recycled in the reject case. If the recycling function f is the identity function, all of the key can be recycled. If f maps all keys to the empty string, then no constraints are put on key leakage in the reject case.

QCA-R strengthens both GYZ and QCA, but not vice versa: the schemes from Section 2.2 that separate the two older notions are immediately examples of schemes that are GYZ or QCA but cannot be QCA-R.

3.1 Constructing QCA-R from any strong-purity-testing code

It was already observed that if a set of quantum error-correcting codes $\{V_{k_1}\}_{k_1\in\mathcal{K}_1}$ is purity testing, then the encryption scheme resulting from Construction 1 is plaintext authenticating [5]. We strengthen this result by showing that the construction turns purity-testing codes into *ciphertext*-authenticating schemes (Theorem 8), and strong-purity-testing codes into QCA-R schemes (Theorem 9). Only purity testing is in general not enough to achieve QCA-R: the trap code is again a counterexample.

▶ **Theorem 8.** *Let $\{V_{k_1}\}_{k_1\in\mathcal{K}_1}$ be a purity-testing code with error ε. The encryption scheme resulting from Construction 1 is quantum ciphertext ε-authenticating (ε-QCA).*

Sketch. In order to prove Theorem 8, we define a simulator that runs the adversary on encrypted halves of EPR pairs, so that the simulator is of the correct form for Definition 3. We prove that the ideal and the real channel are close by considering the accept and the reject cases separately, and by showing that they are both $\varepsilon/2$-close. First, we decompose the adversarial attack into Paulis by Pauli twirling [9] it with the quantum-one-time-pad encryption from Construction 1. In the accept case, the difference between the real and the ideal scenario lies in those attacks that are accepted in the real case, but not in the ideal case. These are exactly those Paulis that, after conjugation with the key k_1 that indexes the purity-testing code, are in the set $(\mathbb{P}_m \otimes \{\mathsf{I},\mathsf{Z}\}^{\otimes t})\backslash(\{\mathsf{I}^{\otimes m}\}\otimes\{\mathsf{I},\mathsf{Z}\}^{\otimes t}) = (\mathbb{P}_m\backslash\{\mathsf{I}^{\otimes m}\})\otimes\{\mathsf{I},\mathsf{Z}\}^{\otimes t}$. The purity-testing property guarantees that the probability over k_1 of a Pauli attack landing in this set is small. The reject case is similar. ◀

A full proof of Theorem 8 can be found in the full version. The proof of Theorem 9 below uses the same techniques. It follows the proof structure of [15, Theorem 3.5], but with a simulator that is suitable for QCA-R.

▶ **Theorem 9.** *Let $\{V_{k_1}\}_{k_1\in\mathcal{K}_1}$ be a strong-purity-testing code with error ε. The encryption scheme resulting from Construction 1 is quantum ciphertext ($\sqrt{\varepsilon} + \frac{3}{2}\varepsilon$)-authenticating with key recycling (ε-QCA-R), with recycling function $f(k_1,k_2) := k_1$.*

Proof. Let $\mathcal{A}$ be an adversary as in Definition 7. Define a simulator $\mathcal{S}$ on the side-information register R as follows: prepare an EPR pair $|\Phi^+\rangle\langle\Phi^+|$ in the register M_1M_2 and encrypt the first qubit using a freshly sampled key $(k_1', k_2') \in \mathcal{K} := \mathcal{K}_1 \times \mathcal{K}_2$ (that is, initialize the tag register T in the state $|0^t\rangle\langle 0^t|$, and apply $P_{k_2'}V_{k_1'}$ to M_1T). Then, run the adversary on the registers M_1TR, keeping M_2 to the side. Afterwards, run the decryption procedure by undoing the encryption unitary and measuring whether the registers M_1M_2T are still in the state $|\Phi^+, 0^t\rangle\langle\Phi^+, 0^t|$ ($= |\Phi^+\rangle\langle\Phi^+| \otimes |0^t\rangle\langle 0^t|$). If so, accept, and if not, reject. Note that this simulator is of the required form in the accept case (see Definition 7).

We show that for this simulator, the distance $\frac{1}{2}\|\mathfrak{I} - \mathfrak{R}\|_\diamond$ between the ideal and the real channel is upper bounded by $\sqrt{\varepsilon} + \frac{3}{2}\varepsilon$. Let ρ^{MRE} be any quantum state on the message register, side-information register, and an environment register E. Let U^{MTR} be a unitary[4] map representing the adversarial channel $\mathcal{A}$, and let $\mu^{\mathsf{real}}_{k_1,k_2}$ and $\mu^{\mathsf{ideal}}_{k_1,k_2}$ be the effective output states in the real and ideal world, respectively:

$$\mu^{\mathsf{real}}_{k_1,k_2} := V^\dagger_{k_1} P^\dagger_{k_2} U^{MTR} P^{MT}_{k_2} V^{MT}_{k_1} (\rho \otimes |0^t\rangle\langle 0^t|) V^\dagger_{k_1} P^\dagger_{k_2} U^\dagger P_{k_2} V_{k_1}, \tag{3}$$

$$\mu^{\mathsf{ideal}}_{k_1,k_2} := V^\dagger_{k_1} P^\dagger_{k_2} U^{M_1TR} P^{M_1T}_{k_2} V^{M_1T}_{k_1} (\rho \otimes |0^t, \Phi^+\rangle\langle 0^t, \Phi^+|) V^\dagger_{k_1} P^\dagger_{k_2} U^\dagger P_{k_2} V_{k_1}. \tag{4}$$

Then we can write the result of the real and the ideal channels as

$$\begin{aligned}\mathfrak{R}(\rho) = \mathop{\mathbb{E}}_{k_1,k_2} \Bigg[&\langle 0^t|^T \mu^{\mathsf{real}}_{k_1,k_2} |0^t\rangle \otimes |k_1k_2\rangle\langle k_1k_2| \\ &+ \ |\bot\rangle\langle\bot|^M \otimes \ \mathrm{Tr}_M \left(\sum_{i \neq 0^t} \langle i|^T \mu^{\mathsf{real}}_{k_1,k_2} |i\rangle \right) \otimes |k_1\rangle\langle k_1| \Bigg],\end{aligned} \tag{5}$$

$$\begin{aligned}\mathfrak{I}(\rho) = \mathop{\mathbb{E}}_{k_1',k_2'} \Bigg[&\langle \Phi^+, 0^t|^{M_1M_2T} \mu^{\mathsf{ideal}}_{k_1',k_2'} |\Phi^+ 0^t\rangle \otimes \tau_{\mathcal{K}} \\ &+ \ |\bot\rangle\langle\bot|^M \otimes \ \mathrm{Tr}_M \left(\sum_{i \neq (\Phi^+, 0^t)} \langle i|^{M_1M_2T} \mu^{\mathsf{ideal}}_{k_1',k_2'} |i\rangle \right) \otimes \tau_{\mathcal{K}_1} \Bigg].\end{aligned} \tag{6}$$

These expressions are obtained simply by plugging in the description of the authentication scheme (see Construction 1) and the simulator into the channels of Definition 7. Since the accept states are orthogonal to the reject states in the M register, and since the key states are all mutually orthogonal, the distance $\frac{1}{2}\|\mathfrak{I}(\rho) - \mathfrak{R}(\rho)\|_{\mathrm{tr}}$ can be written as

$$\begin{aligned}&\mathop{\mathbb{E}}_{k_1,k_2} \frac{1}{2} \left\| \mathop{\mathbb{E}}_{k_1',k_2'} \left(\langle \Phi^+, 0^t| \mu^{\mathsf{ideal}}_{k_1',k_2'} |\Phi^+, 0^t\rangle \right) - \langle 0^t| \mu^{\mathsf{real}}_{k_1,k_2} |0^t\rangle \right\|_{\mathrm{tr}} \\ &+ \ \mathop{\mathbb{E}}_{k_1} \frac{1}{2} \left\| \mathop{\mathbb{E}}_{k_1',k_2'} \left(\mathrm{Tr}_M \sum_{i \neq (0^t, \Phi^+)} \langle i| \mu^{\mathsf{ideal}}_{k_1',k_2'} |i\rangle \right) - \mathop{\mathbb{E}}_{k_2} \left(\mathrm{Tr}_M \sum_{i \neq 0^t} \langle i| \mu^{\mathsf{real}}_{k_1,k_2} |i\rangle \right) \right\|_{\mathrm{tr}} .\end{aligned} \tag{7}$$

For a complete derivation, see the full version. We can thus focus on bounding the two terms in equation (7), for accept and reject, separately. Note the difference between the two terms: in the reject case, the expectation over the one-time pad key k_2 does not have to

[4] We can assume unitarity without loss of generality: if the adversary's actions are not unitary, we can dilate the channel into a unitary one by adding another environment and tracing it out afterwards. In the proof, the environment takes on the same role as the side-information register R, so we omit it for simplicity.

be brought outside of the trace norm, since it is not recycled after a reject. This will make bounding the second term in equation (7) the simpler of the two, so we will start with that one.

Decompose the attack as $U^{MTR} = \sum_\ell \alpha_\ell P_\ell^{MT} \otimes U_\ell^R$. Intuitively, the two states inside the second trace norm differ on those Paulis P_ℓ that are rejected in the ideal scenario, but not in the real one. The strong-purity-testing property promises that these Paulis are very few. However, we have to be careful, because the simulator independently generates its own set of keys. We will now bound the second term in equation (7) more formally.

By rearranging sums, commuting Paulis, and applying projectors (for details: see the full version), we can rewrite the second term inside the trace norm, the state in the real reject case for k_1, as

$$\mathop{\mathbb{E}}_{k_2}\left(\mathrm{Tr}_M \sum_{i\neq 0^t} \langle i|\mu^{\mathsf{real}}_{k_1,k_2}|i\rangle\right) = \mathrm{Tr}_M\left(\sum_{\ell\,:\,V_{k_1}^\dagger P_\ell V_{k_1}\notin \mathbb{P}_{\mathsf{real}}} |\alpha_\ell|^2\, U_\ell^R \rho^{MR} U_\ell^\dagger\right), \tag{8}$$

where $\mathbb{P}_{\mathsf{real}}$ contains the Paulis that are accepted by the real projector, i.e., $\mathbb{P}_{\mathsf{real}} := \mathbb{P}_m \otimes \{\mathsf{I},\mathsf{Z}\}^{\otimes t}$. Similarly, defining $\mathbb{P}_{\mathsf{ideal}} := \{\mathsf{I}^{\otimes m}\} \otimes \{\mathsf{I},\mathsf{Z}\}^{\otimes t}$ to be the set of Paulis that are allowed by the ideal projector, the resulting state in the reject case is

$$\mathop{\mathbb{E}}_{k'_1,k'_2}\left(\mathrm{Tr}_M \sum_{i\neq(0^t,\Phi^+)} \langle i|\mu^{\mathsf{ideal}}_{k'_1,k'_2}|i\rangle\right) = \mathrm{Tr}_M\left(\sum_{\ell\neq 0}\ \mathop{\mathbb{E}}_{\substack{k'_1\in\mathcal{K}_1\\ V_{k'_1}^\dagger P_\ell V_{k'_1}\notin\mathbb{P}_{\mathsf{ideal}}}} |\alpha_\ell|^2\, U_\ell^R \rho^{MR} U_\ell^\dagger\right) \tag{9}$$

$$\approx_\varepsilon \mathrm{Tr}_M\left(\sum_{\ell\neq 0}\ \mathop{\mathbb{E}}_{k'_1\in\mathcal{K}_1} |\alpha_\ell|^2\, U_\ell^R \rho^{MR} U_\ell^\dagger\right), \tag{10}$$

where the approximation sign means that the trace distance between the two states is upper bounded by ε. The closeness follows from the strong-purity-testing property of the code: the two states differ in those keys k'_1 for which $V_{k'_1}^\dagger P_\ell V_{k'_1} \in \mathbb{P}_{\mathsf{ideal}} \subseteq \mathbb{P}_{\mathsf{real}}$, and for any non-identity Pauli P_ℓ, this set is small by strong purity testing. Combined with the facts that $\mathrm{tr}(U_\ell \rho U_\ell^\dagger) = 1$ and $\sum_\ell |\alpha_\ell|^2 = 1$, it follows that the states in equations (9) and (10) are ε-close. Note that none of the terms in equation (10) depends on k'_1, so we can remove the expectation over it.

Applying the triangle inequality (twice), the second term in equation (7) is found to be small:

$$\mathop{\mathbb{E}}_{k_1}\frac{1}{2}\left\|\mathop{\mathbb{E}}_{k'_1,k'_2}\left(\mathrm{Tr}_M \sum_{i\neq(0^t,\Phi^+)} \langle i|\mu^{\mathsf{ideal}}_{k'_1,k'_2}|i\rangle\right) - \mathop{\mathbb{E}}_{k_2}\left(\mathrm{Tr}_M \sum_{i\neq 0^t}\langle i|\mu^{\mathsf{real}}_{k_1,k_2}|i\rangle\right)\right\|_{\mathrm{tr}} \tag{11}$$

$$\leq \frac{\varepsilon}{2} + \mathop{\mathbb{E}}_{k_1}\frac{1}{2}\left\|\mathrm{Tr}_M\left(\sum_{\ell\,:\,V_{k_1}^\dagger P_\ell V_{k_1}\in\mathbb{P}_{\mathsf{real}}\setminus\{\mathsf{I}^{\otimes(m+t)}\}} |\alpha_\ell|^2\, U_\ell \rho U_\ell^\dagger\right)\right\|_{\mathrm{tr}} \tag{12}$$

$$\leq \frac{\varepsilon}{2} + \frac{1}{2}\mathop{\mathbb{E}}_{k_1} \sum_{\ell\,:\,V_{k_1}^\dagger P_\ell V_{k_1}\in\mathbb{P}_{\mathsf{real}}\setminus\{\mathsf{I}^{\otimes(m+t)}\}} |\alpha_\ell|^2, \tag{13}$$

which we can upper bound by ε by applying the strong-purity-testing property once more.

Next, we bound the first term of equation (7): the difference between the ideal and the real channel in the accept case. The strategy is identical to the reject case that we just treated, but because we want to recycle both k_1 and k_2 in the accept case, we have to be more careful. The state in the real scenario, $\langle 0^t|\mu^{\mathsf{real}}_{k_1,k_2}|0^t\rangle$, cannot be rewritten into the compact form of, e.g., equation (8), because we cannot average over the Pauli key k_2. Using a technical lemma from [15] and Jensen's inequality in order to take the expectation over the keys inside, we obtain the bound

$$\mathop{\mathbb{E}}_{k_1,k_2}\left\| \mathop{\mathbb{E}}_{k_1',k_2'}\left(\langle\Phi^+,0^t|\mu^{\mathsf{ideal}}_{k_1',k_2'}|\Phi^+,0^t\rangle\right) - \langle 0^t|\mu^{\mathsf{real}}_{k_1,k_2}|0^t\rangle\right\|_{\mathrm{tr}} \leq \frac{\varepsilon}{2}+\sqrt{\varepsilon}. \tag{14}$$

For a complete derivation, see the full version.

We have now upper bounded $\frac{1}{2}\|\mathfrak{I}(\rho)-\mathfrak{R}(\rho)\|_{\mathrm{tr}} \leq \sqrt{\varepsilon}+\frac{3}{2}\varepsilon$ for any state ρ^{MRE}, resulting in $\frac{1}{2}\|\mathfrak{I}-\mathfrak{R}\|_{\diamond} \leq \sqrt{\varepsilon}+\frac{3}{2}\varepsilon$, as desired. ◀

4 A strong-purity-testing variation on the trap code

Theorem 9 already gives us a quantum-ciphertext-authenticating code with key recycling: the Clifford code. However, the Clifford code is not very well suited for quantum computing on authenticated data. Although all Clifford-group operations can be performed easily by updating the key client-side, it is not known how to perform non-Clifford gates and measurements on the Clifford code. Moreover, if an entangling operation is performed on two separately encoded qubits, their keys also have to be combined into a key for a single, bigger ciphertext. This prevents output qubits from being decrypted individually.

In this section, we therefore present a strong-purity-testing variation on the trap code, the *strong trap code*, which does allow for computation on the ciphertexts in a meaningful and efficient way. By Theorem 9, this construction immediately gives rise to a ciphertext authentication scheme with key recycling (QCA-R). Note that the strong trap code is also secure in Portmann's abstract-cryptography definition of quantum plaintext authentication with key recycling [15].

4.1 Benign distance and weight sparsity

The strong trap code requires the existence of a family of quantum error-correcting codes with two specific properties: a high benign distance, and weight sparsity. We specify these properties here.

If a QECC has distance d, it is not necessarily able to detect all Pauli errors of weight less than d. For example, if one of the qubits in a codeword is in the state $|0\rangle$, then a Pauli-Z remains undetected. In general, any Pauli error that stabilizes all codewords will remain undetected by the code. Of course, such an error does not directly cause harm or adds noise to the state, because it effectively performs the identity operation. However, in an adversarial setting, even such 'benign' Pauli errors indicate that someone tried to modify the state.

We consider an alternative distance measure for quantum error-correcting codes that describes the lowest possible weight of a stabilizer:

▶ **Definition 10** (Benign distance). The *benign distance* of an $[[n,m]]$ code is the minimum weight of a non-identity Pauli P_ℓ such that $P_\ell|x_L\rangle = |x_L\rangle$ for all $x \in \{0,1\}^m$. If such P_ℓ does not exist, the benign distance is ∞.

To distinguish the benign distance from the notion of difference defined in Definition 4, we will often use the term *conventional distance* to refer to the latter.

The benign distance in a fixed relation to the conventional distance. For example, the $[[7,4]]$ Steane code has distance 3, but benign distance 4. On the other hand, the $[[49,1]]$ concatenated Steane code has distance 9, but a benign distance of only 4 (any non-identity stabilizer for the $[[7,4]]$ Steane code is also a stabilizer on the $[[49,1]]$ code if it is concatenated with identity on the other blocks). Even though the two quantities do not bound each other in general, we observe that the benign distance of weakly self-dual *CSS codes* (i.e., CSS codes constructed from a weakly self-dual classical code) grows with their conventional distance: see the full version.

We define a second property of interest: *weight sparsity*. Intuitively, weight sparsity means that for any set of X-, Y-, and Z-weights, randomly selecting a Pauli operator with those weights only yields a stabilizer with very small probability. This probability should shrink whenever the codeword length grows; for this reason, we consider weight sparsity as a property of code *families* rather than of individual codes.

▶ **Definition 11** (Weight-sparse code family)**.** Let $(E_i)_{i\in\mathbb{N}}$ be a family of quantum error-correcting codes with parameters $[[n(i), m(i), d(i)]]$. For each $i \in \mathbb{N}$, and for all non-negative integers x, y, z such that $x + y + z \leq n(i)$, let $A_i(x, y, z)$ denote the set of $n(i)$-qubit Paulis with X-weight x, Y-weight y, and Z-weight z. Let $B_i(x, y, z)$ denote set of benign Paulis in $A_i(x, y, z)$.

The family $(E_i)_{i\in\mathbb{N}}$ is *weight-sparse* if the function

$$f(i) := \max_{x+y+z\leq n(i)} \frac{|B_i(x,y,z)|}{|A_i(x,y,z)|}$$

is negligible[5] in $n(i)$.

In the full version of this paper, we construct a weight-sparse family of weakly self-dual CSS codes that have benign distance $O(\sqrt{n(i)})$, where $n(i)$ is the codeword length of the ith code in the family. The CSS codes are constructed from a punctured version of classical Reed–Muller codes [16].

4.2 The strong trap code

We present a modified version of the trap code, which we call the *strong trap code*. Contrary to the regular trap code, which appends $2t$ trap qubits, the strong trap code only appends a single $|0\rangle$ trap and a single $|+\rangle$ trap. These two traps are subsequently encoded using a quantum error-correcting code that has the desired properties described above, resulting in a ciphertext of the same length as the original trap code.

▶ **Definition 12** (Strong trap code)**.** Let $(E_i)_{i\in\mathbb{N}}$ be a weight-sparse family of weakly self-dual CSS codes with parameters $[[n(i), 1, d(i) = \Omega(\sqrt{n(i)})]]$ and benign distance $\Omega(\sqrt{n(i)})$. Then the ith strong trap code $\{V_{i,k}\}_{k\in\mathcal{K}_i}$ encodes $m = 1$ qubit using $t = 3n(i) - 1$ tags with the unitaries $V_{i,k} := \pi_k E_i^{\otimes 3}\mathsf{H}_{2n(i)+1}$ (where $\mathsf{H}_{2n(i)+1} = \mathsf{I}^{\otimes 2n(i)} \otimes \mathsf{H} \otimes \mathsf{I}^{\otimes(n(i)-1)}$).

[5] A function $f(x)$ is negligible in x if for all $c \in \mathbb{N}$, there exists an x_0 such that for all $x \geq x_0$, $f(x) < x^{-c}$. This definition is extended by stating that a function $f(x)$ is negligible in another function $g(x)$ if for all $c \in \mathbb{N}$, there exists an x_0 such that for all $x \geq x_0$, $f(x) < (g(x))^{-c}$.

The strong trap code invokes two layers of security: the CSS codes E_i, which detect low-weight attacks, and the traps $|0\rangle$ and $|+\rangle$, which detect higher-weight attacks by revealing bit and phase flips, respectively.

One can verify that computing on quantum states authenticated with the strong trap code works in much the same way as for the original trap code. For details, see [6].[6]

▶ **Theorem 13.** *The strong trap code is a strong-purity-testing code with error* $\mathrm{negl}(n(i))$.

Proof. Consider an arbitrary i and non-identity Pauli $P_\ell \in \mathbb{P}_{3n(i)} \backslash \{\mathsf{I}^{\otimes 3n(i)}\}$. Let w_x and w_z denote the X-weight and Z-weight (respectively) of P_ℓ, and note that $\max(w_x, w_z) > 0$.

We bound the probability that $P_{\ell'} := \pi_k^\dagger P_\ell \pi_k$ remains undetected by the code E_i and the traps. Because E_i is a CSS code, it detects X and Z errors separately: let us write $P_{\ell'} = P_x P_z$ with $P_x \in \{\mathsf{I}, \mathsf{X}\}^{\otimes 3n(i)}$ and $P_z \in \{\mathsf{I}, \mathsf{Z}\}^{\otimes 3n(i)}$, and focus first on the probability that P_x remains undetected, i.e., the probability that $\mathsf{H}_{2n(i)+1}(E_i^\dagger)^{\otimes 3} P_x E_i^{\otimes 3} \mathsf{H}_{2n(i)+1} \in \mathbb{P}_1 \otimes \{\mathsf{I}, \mathsf{Z}\}^{\otimes 3n(i)-1}$.

Because of the permutation π_k, P_x is a random Pauli in $\{\mathsf{I}, \mathsf{X}\}^{\otimes 3n(i)}$ with weight w_x. (Note that P_z is also a random Pauli with weight w_z, but is correlated with P_x: any overlap in the locations of X and Z operators in P_ℓ is preserved by the permutation.)

Consider all possible values of $w_x = w_1 + w_2 + w_3$, where w_1 denotes the weight of P_x on the first (data) codeword, w_2 the weight on the second ($|0\rangle$-trap) codeword, and w_3 the weight on the third ($|+\rangle$-trap) codeword:

- If $w_x = 0$, then the Pauli P_x is identity, and remains undetected with probability 1.
- If $0 < w_x < d(i)$, then $0 < w_j < d(i)$ for at least one $j \in \{1, 2, 3\}$. E_i detects an error on the jth block with certainty, since the weight of the error is below the distance and the benign distance.
- If $d(i) \leq w_x \leq 3n(i) - d(i)$, the attack P_x will likely be detected on the second block, the $|0\rangle$-trap. We can be in one of four cases:
  - $w_2 > 0$ and P_x is detected in the second block by the CSS code E_i.
  - $w_2 > 0$ and P_x acts as a logical operation on the second block. Since P_x consists of only I's and X's, this logical operation can only be an X by the construction of CSS codes. In this case, P_x is detected by the projection that checks whether the trap is still in the $|0\rangle$ state.
  - $w_2 > 0$ and P_x acts as a stabilizer on the second block, and remains undetected on that block. However, by the weight-sparsity of the code family, the probability that this is the case is negligible in $n(i)$.
  - $w_2 = 0$. In this case, P_x acts as identity on the second block. The probability that this case occurs, however, is small:

$$\Pr_k[w_2 = 0] = \frac{\binom{2n(i)}{w_x}}{\binom{3n(i)}{w_x}} < \left(\frac{2}{3}\right)^{w_x} \leq \left(\frac{2}{3}\right)^{d(i)} . \tag{15}$$

    The first inequality holds in general for binomials, and the second one follows from the fact that $w_x \geq d(i)$. Since $d(i) = \Omega(\sqrt{n(i)})$, this probability is negligible in $n(i)$.

  In total, the probability of the attack remaining undetected for $d(i) \leq w_x \leq 3n(i) - d(i)$ is negligible in $n(i)$.

[6] For some applications, authenticating through measurement (cf. [6, Appendix B.2]) can be very useful. Our underlying code has all the requirements to achieve this in principle, but in this work we focus on quantum authentication and do not formulate the full security notions needed to properly describe this scenario.

- If $3n(i) - d(i) < w_x < 3n(i)$: as in the second case, there is at least one $j \in \{1,2,3\}$ such that $n(i) - d(i) < w_j < n(i)$, causing the attack to be detected (recall that $\mathsf{X}^{\otimes 3n(i)}$ is a logical X, and therefore this mirrors the $0 < w_x < d(i)$ case).
- If $w_x = 3n(i)$, then the logical content of the second block, the $|0\rangle$-trap, is flipped. This is detected with certainty as well.

We see that unless $w_x = 0$, the Pauli P_x remains undetected only with probability negligible in $n(i)$. A similar analysis can be made for P_z: it is always detected with high probability, unless $w_z = 0$. We stress that these probabilities are *not* independent. However, we can say that

$$\Pr_k[P_x \text{ and } P_z \text{ undetected}] \leq \min\left\{\Pr_k[P_x \text{ undetected}],\ \Pr_k[P_z \text{ undetected}]\right\}, \tag{16}$$

and since at least one of w_x and w_z is non-zero, this probability is negligible in $n(i)$. ◀

5 Simultaneous encryptions with key reuse

Earlier work on key reuse for quantum authentication deals explicitly with *key recycling*, the decision to reuse (part of) a key for a new encryption after completing the transmission of some other quantum message. The key is reused only *after* the honest party decides whether to accept or reject the first message, so recycling is a strictly sequential setting.

If Construction 1 is instantiated with a strong-purity-testing code (such as the strong trap code), the resulting scheme is able to handle an even stronger, parallel, notion of key reuse. As long as the one-time pads are independent, it is possible to encrypt multiple qubits under the same code key while preserving security. Even if the adversary is allowed to interactively decrypt a portion of the qubits one-by-one, the other qubits will remain authenticated. This property is especially important for the strong trap code: computing on data authenticated with the strong trap code requires all qubits to be encrypted under the same permutation key.

The original trap code is secure in this setting (as long as the one-time pads are fresh; see Section 5.2 of [6]), but only if all qubits are decrypted at the same time. If some qubits can be decrypted separately, the adversary can deduce the location of the $|+\rangle$ traps by applying single-qubit X operations to different ciphertexts at different locations, and observing which ones are rejected. Repeating this for the Z operator to learn about the $|0\rangle$ traps, the adversary can completely break the authentication on the remaining qubits.

Suppose we encrypt two messages using an authentication scheme based on a strong-purity-testing code $\{V_{k_0}\}_{\mathcal{K}_0}$, using the same code key k_0 but a fresh one-time pad. If we then decrypt the first message, the scheme is still QCA-R-authenticating on the second message with only slightly worse security.

▶ **Theorem 14** (informal). *Let* (Encrypt, Decrypt) *be an* ε-QCA-R-*authenticating scheme resulting from Construction 1, using a strong-purity-testing code* $\{V_{k_0}\}_{\mathcal{K}_0}$. *Let* M_1, M_2 *denote the plaintext registers of the two messages,* $C_1 = M_1T_1, C_2 = M_2T_2$ *the corresponding ciphertext registers, and* R *a side-information register. Let* $\mathcal{A}_1$, $\mathcal{A}_2$ *be arbitrary adversarial channels. Consider the setting where the adversary acts on the qubits, encrypted with keys* k_0, k_1, k_2, *as*

$$\mathsf{Decrypt}_{k_0,k_2}^{C_2\to M_2} \circ \mathcal{A}_2^{M_1,C_2,R} \circ \mathsf{Decrypt}_{k_0,k_1}^{C_1\to M_1} \circ \mathcal{A}_1^{C_1,C_2,R} \circ \left(\mathsf{Encrypt}_{k_0,k_1}^{M_1\to C_1} \otimes \mathsf{Encrypt}_{k_0,k_2}^{M_2\to C_2}\right),$$

so that the key k_0 *is used for both messages. Then, the scheme is* 2ε-QCA-R-*authenticating on the second qubit.*

Sketch. As a first step, we rewrite the encryption of the second qubit as using encoding and teleportation, by using the equivalence between applying a random quantum one-time pad and teleporting a state. The encryption of the second qubit can then be thought of as happening after decryption of the first qubit. Next, we apply QCA-R security of the first qubit, where we are using the property that k_0 is recycled both in the accept and the reject case. Finally we undo the rewrite and can directly apply QCA-R security on the remaining state. ◀

The complete proof can be found in the full version. The argument easily extends to any polynomial number of authenticated qubits.

6 Conclusion

We presented a new security definition, QCA-R, for ciphertext authentication with key recycling, and showed that schemes based on purity-testing codes satisfy quantum ciphertext authentication, while strong purity testing implies both ciphertext authentication and key recycling. This is analogous to the security of quantum plaintext-authentication schemes from purity-testing codes [5, 15].

Additionally, we constructed the *strong trap code*, a variant of the trap code which is a strong-purity-testing code and therefore is QCA-R secure (as well as secure under all notions of plaintext authentication). This new scheme can strengthen security and add key-recycling to earlier applications of the trap code. It is also applicable in a wider range of applications than the original trap code, because encrypted qubits remain secure even if other qubits sharing the same key are decrypted earlier.

A potential application of the strong trap code is the design of a quantum CCA2-secure encryption scheme (as in [3, Definition 9]) that allows for computation on the encrypted data. By only using the pseudo-random generator for the one-time-pad keys, and recycling the key for the underlying error-correcting code, this security level could be achieved.

As future work, our definition of QCA-R could be generalized in different ways. First, one can consider a variant of the definition in the abstract-cryptography or universal-composability framework, in order to ease the composition with other cryptographic primitives. Second, because it can be useful to authenticate measurements in delegated computation applications, one could extend the definition of QCA-R to deal with the measurement of authenticated data. We expect no real obstacles for this extension of the definition, and refer to [6, Appendix B.2] for comparable work on the original trap code.

References

1 Dorit Aharonov, Michael Ben-Or, and Elad Eban. Interactive proofs for quantum computations. *arXiv preprint arXiv:0810.5375*, 2008.

2 Gorjan Alagic, Yfke Dulek, Christian Schaffner, and Florian Speelman. Quantum fully homomorphic encryption with verification. In *Advances in Cryptology – ASIACRYPT 2017*, pages 438–467, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-70694-8_16.

3 Gorjan Alagic, Tommaso Gagliardoni, and Christian Majenz. Unforgeable quantum encryption. *arXiv preprint arXiv:1709.06539*, 2017.

4 Gorjan Alagic and Christian Majenz. Quantum non-malleability and authentication. In *Advances in Cryptology – CRYPTO 2017*, pages 310–341, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-63715-0_11.

5 Howard Barnum, Claude Crépeau, Daniel Gottesman, Adam Smith, and Alain Tapp. Authentication of quantum messages. In *Foundations of Computer Science, 2002. Proceedings. The 43rd Annual IEEE Symposium on*, pages 449–458. IEEE, 2002.

6 Anne Broadbent, Gus Gutoski, and Douglas Stebila. Quantum one-time programs. In *Advances in Cryptology–CRYPTO 2013*, pages 344–360. Springer, 2013.

7 Anne Broadbent, Zhengfeng Ji, Fang Song, and John Watrous. Zero-knowledge proof systems for QMA. In *57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 31–40, Oct 2016. `doi:10.1109/FOCS.2016.13`.

8 Anne Broadbent and Evelyn Wainewright. Efficient simulation for quantum message authentication. In *Information Theoretic Security*, pages 72–91, Cham, 2016. Springer International Publishing. `doi:10.1007/978-3-319-49175-2_4`.

9 Christoph Dankert, Richard Cleve, Joseph Emerson, and Etera Livine. Exact and approximate unitary 2-designs and their application to fidelity estimation. *Physical Review A*, 80(1):012304, 2009.

10 Frédéric Dupuis, Jesper Buus Nielsen, and Louis Salvail. Actively secure two-party evaluation of any quantum operation. In *Advances in Cryptology – CRYPTO 2012*, volume 7417, pages 794–811. Springer International Publishing, 2012. Full version on IACR eprint archive: `eprint.iacr.org/2012/304`. `doi:10.1007/978-3-642-32009-5_46`.

11 Serge Fehr and Louis Salvail. Quantum authentication and encryption with key recycling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 311–338. Springer, 2017.

12 Sumegha Garg, Henry Yuen, and Mark Zhandry. New security notions and feasibility results for authentication of quantum data. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 342–371, Cham, 2017. Springer International Publishing. `doi:10.1007/978-3-319-63715-0_12`.

13 Patrick Hayden, Debbie W Leung, and Dominic Mayers. The universal composable security of quantum message authentication with key recyling. *arXiv preprint arXiv:1610.09434*, 2016.

14 Jonathan Oppenheim and Michał Horodecki. How to reuse a one-time pad and other notes on authentication, encryption, and protection of quantum information. *Phys. Rev. A*, 72:042309, Oct 2005. `doi:10.1103/PhysRevA.72.042309`.

15 Christopher Portmann. Quantum authentication with key recycling. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 339–368. Springer, 2017.

16 John Preskill. Quantum computation, 1997. URL: `http://www.theory.caltech.edu/people/preskill/ph229/index.html`.

17 Mark N. Wegman and J. Lawrence Carter. New hash functions and their use in authentication and set equality. *Journal of Computer and System Sciences*, 22(3):265–279, 1981. `doi:10.1016/0022-0000(81)90033-7`.

On the Complexity of Two Dimensional Commuting Local Hamiltonians

Dorit Aharonov
School of Computer Science and Engineering, The Hebrew University, Jerusalem, Israel
doria@cs.huji.ac.il

Oded Kenneth
School of Computer Science and Engineering, The Hebrew University, Jerusalem, Israel
kenneth@ph.technion.ac.il

Itamar Vigdorovich
Faculty of Mathematics and Computer Science, Weizmann Institute of Science, Rehovot, Israel
itamar.vigdorovich@weizmann.ac.il

Abstract

The complexity of the commuting local Hamiltonians (CLH) problem still remains a mystery after two decades of research of quantum Hamiltonian complexity; it is only known to be contained in NP for few low parameters. Of particular interest is the tightly related question of understanding whether groundstates of CLHs can be generated by efficient quantum circuits. The two problems touch upon conceptual, physical and computational questions, including the centrality of non-commutation in quantum mechanics, quantum PCP and the area law. It is natural to try to address first the more physical case of CLHs embedded on a 2D lattice, but this problem too remained open apart from some very specific cases [4, 17, 24]. Here we consider a wide class of two dimensional CLH instances; these are k-local CLHs, for any constant k; they are defined on qubits set on the edges of any surface complex, where we require that this surface complex is not too far from being "Euclidean". Each vertex and each face can be associated with an arbitrary term (as long as the terms commute). We show that this class is in NP, and moreover that the groundstates have an efficient quantum circuit that prepares them. This result subsumes that of Schuch [24] which regarded the special case of 4-local Hamiltonians on a grid with qubits, and by that it removes the mysterious feature of Schuch's proof which showed containment in NP without providing a quantum circuit for the groundstate and considerably generalizes it. We believe this work and the tools we develop make a significant step towards showing that 2D CLHs are in NP.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases local Hamiltonian complexity, commuting Hamiltonians, local Hamiltonian problem, trivial states, toric code, ground states, quantum NP, QMA, topological order, multiparticle entanglement, logical operators, ribbon

Digital Object Identifier 10.4230/LIPIcs.TQC.2018.2

Related Version `https://arxiv.org/abs/1803.02213`

Funding The authors are grateful for the generous funding of the ERC grant number 280157, and of the Simon grant number 385590

13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2018).
Editor: Stacey Jeffery; Article No. 2; pp. 2:1–2:21
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

1.1 Commuting local Hamiltonians

The Local Hamiltonian (LH) problem is central to the theory of quantum complexity. In 1998 it was proved by Kitaev to be QMA-complete [21], initiating by that the area of quantum Hamiltonian complexity. This result is often considered as the quantum analogue of the celebrated Cook-Levin theorem, which states that the Boolean Satisfiability problem (SAT) is NP-complete [23]. In 2003 Bravyi and Vyalyi [9] raised the question of what is the complexity of the intermediate class in which all terms mutually commute (commuting local Hamiltonians, or CLHs). The question begs an answer not only because the commutation restriction is natural and often made in physics; but this is also a computational probe to the fundamental question: is the uncertainty exhibited by non-commuting operators necessary for quantum systems to exhibit their *full* quantum nature? or, perhaps, it happens to be the (much less expected) case that even commuting quantum systems can express full quantum power.

The CLH problem may seem at first sight to be trivially in NP, since by the commutation condition, there exists a common basis of eigenstates to all terms, where each constraint has a well defined value on each eigenstate; the problem seems like a classical constraint satisfaction problem (CSP). This hope breaks down when realizing that the eigenstates themselves maybe highly complex. While in CSP, a proof for satisfiability is simply a string, i.e. a satisfying assignment, in the quantum case the eigenstates themselves may be highly entangled. Indeed, a beautiful example is Kitaev's toric code [20], whose global entanglement is characterized by topological properties. In the general case, we do not no whether groundstates of CLHs have an efficient classical description at all (that is, a polynomial size classical representation from which the result of any local measurement can be deduced efficiently).

The question of CLHs touches upon some of the most important aspects of quantum many body systems: fundamental, physical and complexity theoretical. For a start, stabilizer codes can be viewed as ground spaces of CLHs; these constitute by far the most common framework for the study of quantum error correcting codes. CLHs are also a very convenient place to start with when studying open problems and toy examples; for example in the study of the quantum PCP conjecture [2, 3, 7] often CLHs are used as a case study (e.g. [5, 13, 18]). Moreover, CLH systems provide the simplest examples for systems obeying the *area law* bounding the entanglement in groundstates of gapped systems[1]. In the one dimensional case, the area law was recently shown in a breakthrough result to provide an efficient classical algorithm for constructing groundstates [22]. In two or higher dimensions such an algorithm cannot be expected, since CLHs become NP hard in 2D. However it is still possible that groundstates satisfying the area law have polynomial size quantum circuits (which may be hard to find). Understanding whether groundstates of 2D CLH systems have efficient descriptions is thus an essential first step towards clarifying how the area law affects the complexity of groundstates.

Despite the importance and fundamental nature of this class, and fourteen years after the problem was posed [9], the complexity of the CLH problem remains a mystery, even in the physically motivated case of 2D. A trivial upper bound to the complexity of the CLH problem is that it belongs to QMA. A simple lower bound exists as well: if we let d denote the dimension of the particles, and let k denote the maximal number of particles that each local

[1] The area law states that the entanglement in the groundstate between two regions grows like the size of the *boundary* between these two regions, rather than their volume.

term acts on, then we may define $CLH(k,d)$ accordingly. Using this notation $CLH(k,d)$ is NP-hard if $k,d \geq 2$. The question becomes then to distinguish between those cases which are within NP, those which are QMA hard, and possibly, the intermediate cases. However, excluding a few special cases of CLH, not much is known.

1.2 Previous results

Bravyi and Vyalyi proved that $CLH(2,d)$, namely the class of instances in which the particle dimensionality d is an arbitrary constant, whereas the interactions only involve two such particles (this is called *two-local CLHs*), is in NP [9]. The proof relies on a decomposition lemma based on the theory of finite dimensional C*-algebra representations [26]. This tool has become essential in all following results about this problem.

Aharonov and Eldar [4] then considered the 3-local case with qubits and qutrits. They showed that $CLH(3,2) \in NP$ and also that $NE{-}CLH(3,3) \in NP$ where NE is a geometrical restriction on the interaction called *nearly Euclidean* [4]. An important fact about the proofs for both of these results is that the witness which is sent by the prover is virtually a constant depth quantum circuit which prepares a groundstate for the system, starting from a product state. Hastings called states which can be generated by constant depth quantum circuits "trivial" [18]; the name is justified since indeed, local observables can be computed classically in an efficient way for such states, given the circuit that generates them, because the *light cone* of qubits affecting the output qubits of a local observable is of constant size. Thus, the above mentioned results not only prove containment in NP, but also show that such systems have groundstates with very restricted multi-particle entanglement which is in some sense *local*.

In this regard, Aharonov and Eldar [4] mentioned a tight "threshold" which can be drawn at this point: commuting systems with parameters as above are essentially classical; But, when raising k or d just by 1, i.e when considering $CLH(4,2)$ or $CLH(3,4)$, we arrive at a new regime in which the quantum system can exhibit *global* entanglement, namely, the groundstates are no longer trivial (by Hastings' definition). In fact, such systems can exhibit global entanglement even when the system is embedded on a square lattice: Kitaev's toric code [20] is a wonderful example, as it can indeed be shown that groundstates of this code with nearest neighbor interactions cannot be generated by a constant depth quantum circuit [8]. This raises the possibility [4] that general CLH systems with parameters above the "transition point" are too complex for containment in NP, as they allow global entanglement.

There are several examples beyond the transition point which indicate that though global entanglement is possible, it might still be the case that CLH systems remain "classically accessible" even in that regime. First, it is known that despite their global entanglement, toric code states can be constructed in *logarithmic* depth quantum circuits called MERA [1] which moreover, allow local measurements to be simulated classically efficiently. In addition, Schuch proved that CLHs in which all qubits and all 4-local constraints are embedded on a square lattice (generalizing the toric code to general interactions with the same geometry and dimensionality) also belong to NP [24]. Interestingly enough though, Schuch's proof bypasses the question of whether an efficient description of a groundstate exists; instead, the witness which is sent by the prover convinces the verifier that a low energy state exists without describing that state at all. Schuch's result thus leaves open the possibility, suggested in [4], that when crossing the transition point from local to global entanglement mentioned above, groundstates may in general become difficult to describe classically (not including the toric code special case).

Hastings provided two other results proving upper bounds on the complexity of the CLH problem in certain cases. In [18] he considered k-local CLHs whose interaction graphs are 1-*localizable*; roughly speaking, these are instances whose interaction graphs can be mapped to graphs continuously, such that the preimage of every point is of bounded diameter. This extends the result of [9] that two local Hamiltonians are in NP, to slightly more general constructions which are in some sense, two-local in every local region. In another result of Hastings [17], he considered CLHs on a planar lattice, and proved that the problem is in NP under certain restrictive conditions on the C*-algebraic decomposition (essentially, that when dividing the lattice to stripes, the transformation which disentangles adjacent stripes, a'la Bravyi and Vyalyi [9], is local). Hastings also provided parts of a proof that 2D CLH is in NP, and suggested that the proof will be completed elsewhere, however this was not done.

We note that an interesting clue pointing in fact in the other direction, namely suggesting that the CLH problem could be harder than NP, was given recently by Gosset, Mehta and Vidick [14]; they show that a certain problem regarding the connectivity of the ground space of CLHs is as hard as that of general LHs. It is suggested in [14] that this is probably true even for CLHs in 2D, though this remains to be worked out.

We are left with the mystery: possibly the above "classical" examples are just special cases, and in the general case above the low parameters threshold, global entanglement prevents an efficient description of the groundstates of CLHs; or maybe, the "classicality" of the entanglement in the toric code groundstates as well as in the other examples mentioned [17,24] is generic for all CLHs, and thus the problem lies in NP.

1.3 Results

We consider a wide subclass of CLH in 2D. Specifically, we consider $CLH(k,2)$ instances (i.e with qubits) where the qubits are arranged on the edges of a polygonal complex $\mathcal{K}$ whose underlying topological space is a surface. We refer to those as *2D complexes*[2]. The local terms live on the vertices of $\mathcal{K}$ (these are called stars), and on its faces (plaquettes), where each of these terms acts on the edges attached to the vertex or the face, respectively. In Section 2, this class is formally defined and denoted by $2D-CLH^*(k,2)$. We shall emphasize that the Hamiltonian terms need not be of the form of products of σ_x or σ_z Paulis as in Kitaev's surface codes, but can be general operators on the relevant qubits (as long as they commute). Moreover, the locality parameter k, which in this case equals the maximal degree of vertices and faces of $\mathcal{K}$ (a degree of a face is the number of its edges), is an arbitrary constant as well.

An example of a polygonal complex, where each vertex and each face has a degree of at most 5. One may define on this complex a $2D-CLH^*(5,d)$ instance by assigning to each star and plaquette a Hamiltonian acting on the attached edges, where those Hamiltonians mutually commute.

We note that there is no restriction whatsoever on the topology of the complex $\mathcal{K}$; it can be of any genus, and may or may not include a boundary. We impose one condition on $\mathcal{K}$, which is a metric-geometric condition that we call quasi-Euclidity (though of similar flavor, it shouldn't be confused with the nearly-Euclidean condition of [4]). This condition ensures that the surface induced by the complex admits a triangulation in which the triangles may be slim (as in hyperbolic geometry) and may be fat (as in elliptic geometry) but only up to some constant. This makes the complex in some sense Euclidean up to a constant distortion, and prevents "wild" situations. Any physically natural 2D setting should be covered by this.

[2] despite some friction with ordinary *simplicial 2-complexes* as in e.g [19] which do not necessarily define topologically a surface

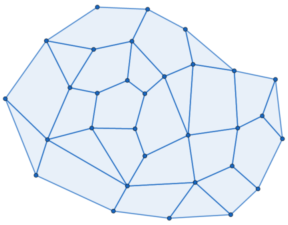


Figure 1.1 Polygonal complex.

Our main two results are:

▶ **Theorem 1.** *The $2D-CLH^*(k,2)$ problem on quasi-Euclidean complexes is in NP.*

▶ **Theorem 2.** *For any instance of $2D-CLH^*(k,2)$ defined on a quasi-Euclidean complex, there exists a polynomial depth quantum circuit which prepares a groundstate.*

Importantly, these results replace the mysterious feature of Schuch's result [24] providing a proof for containment in NP without an efficient groundstate description, by one in which the groundstate can be efficiently classically described; this seems to strengthen the common feeling that containment in NP should go hand in hand with efficient description for the groundstate. Moreover, our results hold for a wide class of cases, which includes not only the 4-local case in a square lattice of Schuch [24], but CLHs with arbitrary locality k, that are defined on any quasi-Euclidian 2D complex. We remark that our definition of $2D-CLH^*(k,2)$ unfortunately does not capture the most general k-local quantum systems of qubits embedded on a surface (see Section 2).

1.4 Proof overview

Our starting point is a folklore quantum algorithm for preparing the groundstates of the toric code. Recall that the toric code Hamiltonian [20] acts on qubits set on the edges of an $n \times n$ grid with boundary conditions which make it topologically a torus. The Hamiltonian has two types of constraints, one for each vertex (star) denoted s, and one for each face (plaquette) denoted p:

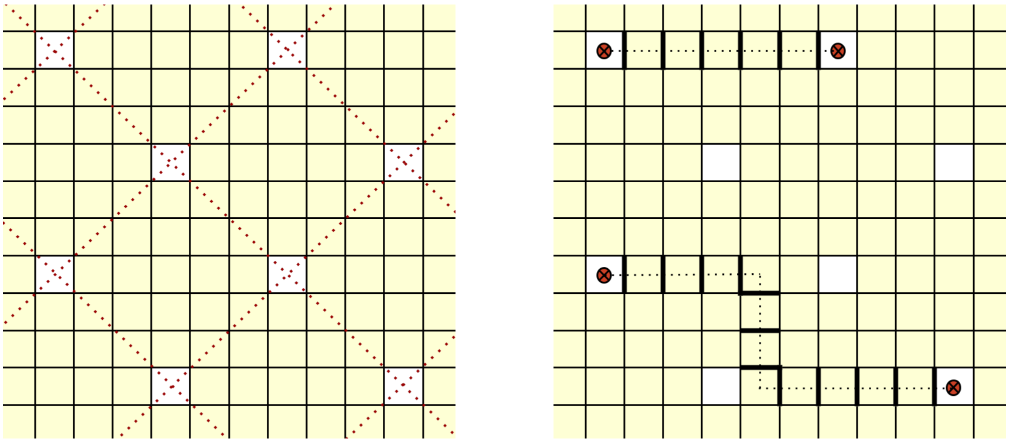
$$A_s = \bigotimes_{e\in s} \sigma_z^e, \qquad B_p = \bigotimes_{e\in p} \sigma_x^e, \qquad H = -\sum_s A_s - \sum_p B_p \tag{1.1}$$

The groundstates of this Hamiltonian form a code space, and exhibit global-entanglement.

Consider creating "holes" in the torus, by removing a small fraction of the plaquettes, in a regular manner. Figure 1.2 (A) shows how by removing enough plaquettes we are left with a *punctured Hamiltonian* $\tilde{H}$, which involves *two local* interactions between *super-particles* comprised each of constantly many qubits. By [9] there is a constant depth quantum circuit which prepares a groundstate (denote it $|\psi\rangle$) for $\tilde{H}$.

This doesn't seem at first as real progress, since $|\psi\rangle$ is a trivial state, whereas groundstates of the original Hamiltonian are globally entangled. The key idea is that now we can *correct for the plaquettes we have removed*, using the known idea of applying string operators connecting pairs of "holes".

To do this, we first *measure* in the state $|\psi\rangle$ each of the plaquette terms which were removed. Due to the commutation relations, the resulting state is *still* a groundstate of $\tilde{H}$ but now it is also an eigenstate of the toric code, with a known eigenvalue for each of the



(a) Punctured Hamiltonian

(b) Logical Operators

Figure 1.2 (A) The white squares are the holes. The dotted lines induce a partition of the set of qubits (edges) to squares (tilted in 45 degrees), which are the super-particles, each containing a constant number of qubits. Every local term (star or plaquette) of the punctured Hamiltonian acts on qubits which belong to at most 2 super-particles. (B): A hole with a spot inside indicates an excitation (i.e. a violation). The dotted lines are string logical operators (copaths) which annihilate particles in pairs. The edges in bold denote the qubits on which the logical operator acts.

terms. Viewing the toric code as a subcode of the punctured code (the groundspace of the punctured Hamiltonian $\tilde{H}$), what we now need is a set of *logical operators* in the punctured code, that act within it and can transform our state into a toric code groundstate.

To this end, we recall the notion of *string* operators which are Pauli operators acting on the paths (strings) connecting a pair of holes [20]. Such an operator changes the values of the measurements corresponding to the constraints in both holes, while keeping all the other values intact. Notice that this process always works on *pairs* of holes. The dependency relations between the local terms ($\prod_s A_s = \prod_p B_p = \mathbf{1}$) [20] imply that for any eigenstate of the toric code there is an *even* number of plaquette (and also star) terms which are in their excited states. Since all plaquettes in the punctured Hamiltonian are satisfied (i.e., not excited), it follows that there is an even number of excited plaquettes out of those which we removed, and thus such a pairing exists.

Note that we could have actually removed *all* plaquettes, resulting in a punctured Hamiltonian $\tilde{H}$ consisting only of A_s terms; Starting with the state $|0^n\rangle$, which is a groundstate of $\tilde{H}$, we could then proceed as in the above algorithm, to derive a groundstate of the toric code (without any help of the prover). We will make use of both approaches in this paper; the "regular holes" approach is the one we will generalize (conceptually) to more general instances, while the second more specific approach is used as a subroutine in our final algorithm, for technical reasons. We will thus present and prove it formally in Section 4.

1.4.1 Physical interpretation

The toric code has a physical interpretation which will be very useful for us [20]. The value of the edges in the σ_x and σ_z basis are interpreted as a $\mathbb{Z}_2$ vector potential or electric field, respectively. When a constraint is violated, we interpret this as if an elementary excitation, or a particle, is created. The star constraints can be viewed as requiring that the electric flux from the vertex (namely the values of the qubits in the computational basis) is zero, i.e., that this vertex will have no electrical charge. If a vertex constraint is violated, we say that there is an “electric charge” at that vertex. Likewise, the plaquette constraints require that the magnetic flux which passes through the face is zero (mod 2). If a plaquette constraint is violated we say that there is a "magnetric vortex" in this plaquette [20]. The toric code consists of the states in which neither electrical charges, nor magnetic vortices appear. The punctured system however allows particles to be created at the sites which we have removed. After measuring these terms, we know exactly where these particles are. It is left to annihilate them. Having a closed surface with no boundary, such as the torus, the total charge on it, as well as the total magnetic flux passing through it, must be zero (as Gauss and Stoke’s laws imply, respectively). This means that there must be an even number of electrical charges, and an even number of magnetic vortexes, which can then be annihilated in pairs, by what is called “string operators” connecting pairs of charges or pairs of vortexes (see [20]). In the above algorithm for the toric code we only needed to annihilate magnetic vortices (plaquettes).

1.4.2 From toric code to general $2D - CLH^*(k, 2)$

It is far from clear how the methods above concerning the toric code can be applied to general 2D CLH systems; after all, surface codes seem to be an extremely restricted type of 2D CLHs (where the local terms must take the form of tensor products of either σ_x or σ_z Pauli operators), whereas we are concerned with arbitrary commuting local terms. Theorem 13 in Section 5 provides our first main step in the proof: we show that all $2D - CLH^*(k, 2)$ instances are "equivalent to the toric code permitting boundaries". This in particular means that if all terms, stars and plaquettes, act non-trivially on all of their attached edges, (plus $\mathcal{K}$ is closed, i.e topologically has no boundary), then the instance is, up to a minor modification, equal to the toric code. In the general case, terms may act trivially on some of their qubits (edges); we will call such edges boundary/coboundary edges. Theorem 13 says that $2D - CLH^*(k, 2)$ instance are virtually the toric code, except for those essentially 1D behaving boundary areas (and thus the term "permitting boundaries"). The proof of this structure theorem relies heavily on the C*-algebraic techniques mentioned earlier. We emphasize that Theorem 13 holds only after some transformation of the instance to one with no "classical qubits" whose value is simply a classical bit which can be provided by the prover (see subsection 3.3).

1.4.3 Constructing the Punctured Hamiltonian

The above equivalence theorem raises the idea of using a similar algorithm as for the toric code groundstates, and somehow handling the special boundary/coboundary qubits. However, we encounter two challenges. First, we do not have sufficient control on operators near the boundary/coboundary. If we carelessly tear out holes in their vicinity, we might not know how to repair them- the correcting process of the toric code heavily relies on the specific commutation and anti-commutation relations between a string operator and the Hamiltonian terms (equation 1.1). We handle this difficulty by tearing out holes only in the interior regions (that is regions without boundary/coboundary qubits) where we do have resemblance to the toric code. It turns out that there is no need to tear holes close to boundary/coboundary qubits as in some sense these special qubits are already punctured: by definition such qubits are not surrounded by Hamiltonians acting on them non-trivially.

The second challenge is that we do no longer have the dependencies $\prod_s A_s = \prod_p B_p = \mathbf{1}$ that ensured earlier an even number of excitations of any given type, and so the idea of fixing holes in pairs is irrelevant. In the physical interpretation, the latter means that the

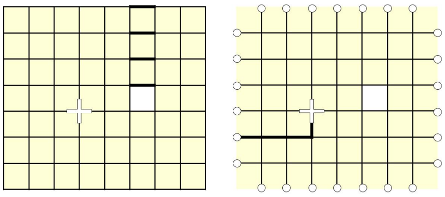


Figure 1.3 Logical operators.

total charge on the manifold can be different than 0 since now flux can escape through the boundary. In section 6 we show that the curse of boundaries is in fact a blessing, since now we can also dump excitations to the boundary/coboundary with string operators, similarly to logical operators in surface codes [10] (figure 1.4).

The latter idea, which can be viewed as the main conceptual idea in the paper, introduces a new challenge - we have two types of special qubits. Boundary qubits give rise to copath string logical operators whereas coboundary qubits give rise to path string logical operators. We cannot expect that puncturing only plaquette terms out of the surface will allow us to fix them later on. Figure 1.3 shows simple examples of systems in which only one type of term (star/plaquette) have access to the boundary/coboundary via copath/path. In short, plaquettes play nicely with boundary edges whereas stars play nicely with coboundary edges.

The white plaquette and the white plus indicate holes. In a complex with boundary but no coboundary only plaquette holes can be connected via a copath to utilize a logical operator, whereas in a complex with coboundary but no boundary only star holes can be connected via paths to utilize a logical operator.

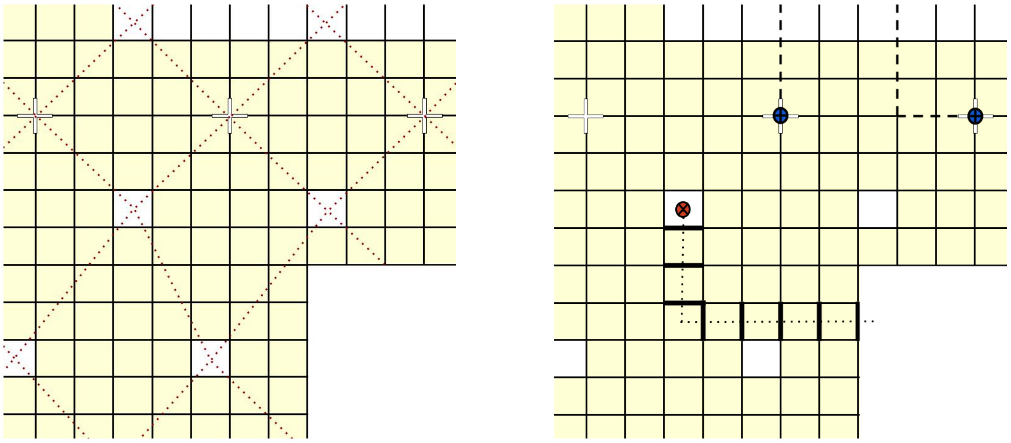
A major technical effort in the paper is proving Lemma 15 which roughly states that for any adjacent plaquette and star, at least one of them has access to the boundary/coboundary (unless they are both already touching the boundary/coboundary), hence a hole in one of them will be fixable

With this in mind, we construct the punctured Hamiltonian as follows: we start by considering the set $\mathcal{W}$ of "fixable" terms. These are terms which are not in the boundary of the system (and thus are in the form of a toric code term) and in addition have access to the boundary or coboundary via a copath or path depending on whether it is a plaquette or star term respectively (see Definition 14 and Figure 6.1). By Lemma 15 the fixable holes are very “dense”. We shall not hesitate to remove all of those terms since, by how the elements of the set $\mathcal{W}$ were chosen, we can correct their values later on.

We call the Hamiltonian obtained by removing all of the terms in $\mathcal{W}$ the *punctured Hamiltonian* $\tilde{H}$.

1.4.4 2-locality of the punctured Hamiltonian

Lemma 15 guarantees that at any large enough constant size area, either there are boundary qubits (recall these are qubits which are acted trivially by at least one of its surrounding terms) which may serve as a hole, or else there must be a fixable term in that area, i.e a member of $\mathcal{W}$, which was removed. In the case of the grid it is now very simple to generate a 2-local structure among constant size super-particles: just consider a coarse grained grid of 5 × 5, and use Lemma 15 to conclude that there must be some hole inside each 5 × 5 square. However we are allowing much more general geometries than the grid; it is here and only here, that we make use of the quasi-Euclidity condition. This is what allows us to follow a similar process, and to tear holes in some regular manner. Technically, we need



(a) Punctured Hamiltonian. **(b)** Logical Operators.

Figure 1.4 (A) Even when boundaries/coboundaries exist, one can tear out holes to obtain a 2-local instance w.r.t superparticles of constant size. (B) After measuring each hole, it remains to correct it if needed by connecting it to the boundary/coboundary via a string operator depending on the hole type (i.e plaquette/star).

to apply Moore's bound [6, 15] to bound the number of edges (qubits) which belong to any super-particle resulting from the process; together with some other combinatorial arguments the proof goes through.

Now that the punctured Hamiltonian is 2-local, we again are guaranteed that a groundstate can be generated by a constant depth quantum circuit [9]. This is the only place where the prover is needed. Note that this groundstate is in general not the groundstate of the original Hamiltonian, yet, the fact that we have torn out only terms of $\mathcal{W}$, namely the fixable terms, implies that we can apply the approach of measuring them and correcting them with string operators to the boundary/coboundary of the system (Figure 1.4 (B)).

1.5 Organization of the paper

In Section 2 we formalize the problem. Section 3 gives some background: "the induced algebra", "classical qubits", and notations. Section 4 provides the efficient algorithm for generating toric code states which we use as a subroutine. Section 5 contains Theorem 13, stating that $2D - CLH^*(k, 2)$ instances are "equivalent to the toric code permitting boundaries". Based on this, in Section 6 we prove lemma 15 which shows that many fixable terms (those with "access to the boundary") exist, and define the punctured Hamiltonian, in which all these terms are removed. In Section 7 we show that the punctured Hamiltonian is indeed 2-local with respect to super-particles of constant size. Section 8 combines all these results to prove Theorems 1,2. In Section 9 we discuss the results, their implications, and state open questions.

This version does not include all proofs in their complete form. Those can be found in the more techinical version of this paper [6].

2 Formulation of the problem

2.1 Definitions

▶ **Definition 1** (CLH instance)**.** An instance of $CLH(k, d)$ consists of a set of Hamiltonian terms (Hermitian matrices) acting on n qudits (particles of dimension d), where each term acts non-trivially on at most k of the n qudits. The norm of each term is bounded by 1, and the terms mutually commute.

To be precise, we note that as usual, the Hermitian matrices are given with entries represented by poly(n) bits.

We consider the cases where the CLH instance is defined on a 2D complex. The type of complexes we allow (see definition bellow) is a generalization of a simplicial 2-complex; while in simplicial complexes the 2-cells must be 2-simplexes (triangles), we allow the 2-cells to be any simple polygon. Topologically speaking, we may define a simple polygon to be any set homeomorphic to the closed disk $D = \{\mathbf{x} \in \mathbb{R}^2 \mid ||\mathbf{x}|| \leq 1\}$ with some choice of a finite amount (at least three) of points on its boundary to be called the vertices of the polygon. The arcs on the boundary which connect two adjacent vertices are called the sides of the polygon. Such complexes are often called *polygonal complexes* [15].

▶ **Definition 2** (polygonal complex)**.** A polygonal complex $\mathcal{K}$ is a collection of points (called 0-cells or vertices), line segments (1-cells, or edges), and simple polygons (2-cells, or faces) glued to each other such that:

1. Any side of a 2-cell in $\mathcal{K}$ is a 1-cell in $\mathcal{K}$. Every endpoint of a 1-cell in $\mathcal{K}$ is a 0-cell in $\mathcal{K}$.
2. The intersection of any two distinct 2-cells of $\mathcal{K}$ is either empty or else it is a single 1-cell (along with its endpoints). The intersection of any two distinct 1-cells of $\mathcal{K}$ is either empty or else it is a single 0-cell.

If all polygons have exactly three vertices then $\mathcal{K}$ is called a simplicial 2-complex. The 1-skeleton of $\mathcal{K}$ is by definition the graph obtained by removing all 2-cells from $\mathcal{K}$. Finally, $\mathcal{K}$ is called two dimensional (2D) if the topological space which it defines $\mathcal{S} = \bigcup \mathcal{K}$ is a surface.

By surface we mean the topological definition of a surface[3] allowing boundaries [25]; that is a topological space such that each point in the interior has a neighborhood homeomorphic to $\mathbb{R}^2$ whereas each point in the boundary has a neighborhood which is homeomorphic to the the upper plane $\{(x, y) \in \mathbb{R}^2 \mid y \geq 0\}$. We shall remark that if $\mathcal{K}$ is finite (which will be the only case we consider) then $\mathcal{S}$ is compact. If in addition $\mathcal{S}$ has no boundary (in the ordinary topological sense) then we say that $\mathcal{S}$ (and thus also $\mathcal{K}$) is closed.

Note that 2D polygonal complexes have the property that every 1-cell is the face of at most two 2-cells (one if that 1-cell is in the boundary, and two if it is in the interior). That is because if 3 or more 2-cells are attached at that 1-cell then the neighborhoods of points in the interior of that 1-cell are neither homeomorphic to $\mathbb{R}^2$ nor to the upper plane.

The 1-skeleton of $\mathcal{K}$ admit the natural graph metric in which the distance between any two vertices is the length of the minimal path between them, where the length of every edge is 1.

▶ **Definition 3** (triangulation)**.** A *triangulation* of a topological space X is a finite simplicial 2-complex $\mathcal{T}$ together with a homeomorphism $f : \mathcal{T} \to X$. The 2-cells of $\mathcal{T}$ are called the *triangles* of the triangulation.

[3] In many texts (e.g [19]) second countability and Hausdorff are required in the definition as well. In our case however, we are only considering finite polygonal complexes which always satisfy these two conditions.

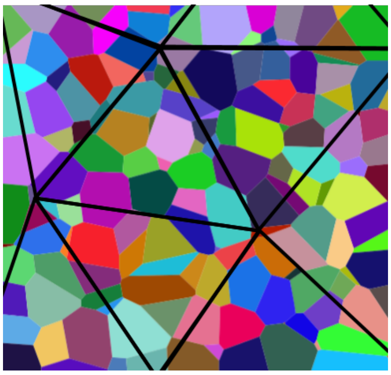


Figure 2.1 Quasi-Euclidean polygonal complex.

The following definition is inspired by metric geometry in which hyperbolic spaces are roughly defined to be metric spaces which have only r-slim triangles - triangles which do not contain any ball of radius r; whereas elliptic metric spaces are such which have a bound on the *diameter* of triangles [11].

▶ **Definition 4** (quasi-euclidean 2D complex)**.** Let $\mathcal{K}$ be a 2D polygonal complex with underlying surface S. A triangulation of $\mathcal{S}$ is said to be $(r, R)-$quasi-Euclidean for some $0 < r < R$ if each of its triangles contains a ball of radius r in $\mathcal{K}$ (w.r.t metric defined above) and the subgraph in it is of diameter at most R. The degree of a triangulation is by definition the maximal degree of its 1-skeleton. In the case where $\mathcal{S}$ admits such a triangulation we say that $\mathcal{K}$ is (r, R)-quasi-Euclidean.

We emphasize that there is no demand from the triangulation to be in any sort in accordance with the complex structure of $\mathcal{K}$ (e.g vertices of $\mathcal{T}$ do not need to be located on vertices of $\mathcal{K}$).

A triangulation $\mathcal{T}$ (dark lines) of the surface $\mathcal{S}$ on which the complex $\mathcal{K}$ lies. $\mathcal{T}$ is (r, R)-quasi-Euclidean with $r = 2$,$R = 12$ since each triangle contains a ball of radius 2 but its diameter is less than 12. The makes $\mathcal{K}$ a (r, R)-quasi-Euclidean complex. Having each triangle contain a ball of radius $r \geq 2k$ (here $k = 7$) ensures that there exists a polygon which is contained in the triangle, as well as all other polygons touching it. The fact that the diameter of each triangle is at most R implies that the number of edges in each triangle is bounded by a number dependent only on R and k, by Moore's bound [15].

▶ **Definition 5** ($2D - CLH^*(k, d)$ instance)**.** Consider instances x of $CLH(k, d)$ for which:

1. There exists a two dimensional polygonal complex $\mathcal{K}$.
2. There exists a 1-1 mapping between qudits of x and edges of $\mathcal{K}$.
3. There exists a 1-1 mapping between local terms of x and the set of vertices and faces of $\mathcal{K}$.
4. If h corresponds to a vertex v then the set of qudits $\{q_1, ..., q_r\}$ which h acts on corresponds to the set of edges $\{e_1, ..., e_r\}$ attached to v.
5. If h corresponds to a face f then the set of qudits $\{q_1, ..., q_r\}$ which h acts on corresponds to the set of edges $\{e_1, ..., e_r\}$ which are in the boundary of f.

We consider the restriction of this class to quasi-Euclidean complexes - those which admit a (r, R)-quasi-Euclidean triangulation of degree D, for some arbitrary constants $D > 0$ and $R > r > 2k$. We call such $2D - CLH^*(k, d)$ instances *quasi-Euclidean.*

The quasi-Euclidean condition doesn't limit the topology in any way. Specifically, for any compact surface $\mathcal{S}$ there exists a quasi-Euclidean polygonal complex $\mathcal{K}$ such that $\mathcal{S}$ is its underlying surface (i.e $\mathcal{S} = \bigcup \mathcal{K}$) [25]. This condition is needed only in Section 7. Hence in the following we ignore it and treat general $2D - CLH^*(k, 2)$ instances; only in Section 7 we will mention this condition again.

Another possible way to define a CLH on a 2D polygonal complex is to place the qudits on the *vertices* rather than the edges, and then local terms are associated with faces alone. We denote the class of such instances by $2D - CLH(k, d)$ (i.e without the star symbol). The latter definition captures the notion of a 2D system in a more general way: every $2D - CLH^*(k, 2)$ instance can be converted to a $2D - CLH(k, 2)$ whereas the converse is true only when the instance has no vertices of degree 3. In addition, if our results can be generalized to $2D - CLH^*(k, d)$ for arbitrary d, this will in fact imply that they also hold for $2D - CLH(k, d)$, under a mild condition similar to quasi-Euclidity (see [6]).

To each of those classes corresponds the local Hamiltonian problem of deciding, given $a < b$ with $b - a < \frac{1}{poly(n)}$, whether the ground energy of the system (i.e the sum of all local terms) is bellow a or above b, provided the promise that one of these cases hold. We use the same notation to denote both the class of such instances (as in Theorem 2) and the corresponding decision problem (as in Theorem 1).

3 Notation and Background

3.1 Notations

Throughout this paper we use $\mathcal{H}$ to denote Hilbert spaces, q to denote qubits, and accordingly $\mathcal{H}_q$ to denote the Hilbert space associated with the qubit q. $\mathcal{K}$ denotes the complex on which the $2D - CLH^*(k, 2)$ is defined whereas $\mathcal{S}$ denotes its underlying surface. We use s to denote stars, p to denote plaquettes and let $|s|$ and $|p|$ denote the degree of a star or a plaquette, i.e the number of edges which belong to s or to p. A_s denotes the local term which corresponds to s and B_p denotes the local term which corresponds to p. h denotes a local term in general. We say that two stars (plaquettes) are adjacent if they share an edge, and say that a star and plaquette are adjacent if they share two edges (which is the only way a star and a plaquette can intersect). When more geometrical aspects are discussed we will consider vertices instead of stars denoted by v, edges instead of qubits denoted by e and faces instead of plaquettes denoted by f. We let H denote the sum of all local terms $H = \sum_s A_s + \sum_p B_p$ where s and p range over the stars and plaquettes of the instance. When we construct a punctured Hamiltonian, i.e a Hamiltonian obtained by removing some terms from the original one, we will always denote it by $\tilde{H}$.

3.2 The induced algebra

▶ **Definition 6** (induced algebra)**.** Let h be an operator on a tensor product Hilbert space $\mathcal{H}_{q_1} \otimes \mathcal{H}_{q_2}$ and let $h = \sum_{i=1}^{m} h_{q_1}^i \otimes h_{q_2}^i$ be a Schmidt decomposition[4] of h. The induced algebra of h on $\mathcal{H}_{q_1}$ is denote by $\mathcal{A}_{\mathcal{H}_{q_1}}^h$ or in short $\mathcal{A}_{q_1}^h$ and is defined to be the C*-algebra generated by $\{I\} \cup \{h_{q_1}^i\}_{i=1}^m$ (I denotes the identity operator).

[4] That is to say: $h_{q_1}^i \in \mathcal{L}(\mathcal{H}_{q_1})$, $h_{q_2}^i \in \mathcal{L}(\mathcal{H}_{q_2})$ for each i and the that sets $\left\{h_{q_1}^i\right\}_{i=1}^m$, $\left\{h_{q_2}^i\right\}_{i=1}^m$ are orthogonal with respect to the Hilbert-Schmidt inner product i.e $tr({h_{q_l}^i}^\dagger \cdot h_{q_l}^j) = 0$ for any $i \neq j$ and $l = 1, 2$.

3.3 Classical qubits

The equivalence to the toric code which we are aiming for can be shown only after performing a certain reduction of removing "classical qubits". Classical qubits are classical in the sense that they do not participate in the entanglement of the system and consequently, the prover may hand us its correct value as a classical bit.

▶ **Definition 7** (trivial qubit). A qubit (or qudit) is called trivial, if no local term acts on it non-trivially.

▶ **Definition 8** (classical qubit). A qubit (or qudit) is called classical if its Hilbert space can be decomposed into a direct sum of 1-dimensional subspaces which are invariant under all local terms in the Hamiltonian H.

When we say that a Hamiltonian h acts trivially on a certain qubit we simply mean that it can be written as $h = I \otimes h'$ where I is the identity operator on that qubit, and h' acts only on other qubits.

Note that due to the low dimension of qubits, once such a non-trivial direct sum decomposition exists then the subspaces must be one dimensional and so the qubit is classical. Note also that every trivial qubit is in particular classical - any direct sum decomposition will do. The following claim says that whenever there is a classical qubit q, the instance can be reduced to a new instance in which it is a trivial qubit.

▶ **Claim 9** (removing classical qubits). *To derive theorems 1,2 it is sufficient to prove it under the restriction of $2D - CLH^*(k, 2)$ to instances with the condition that every classical qubit is trivial.*

This claim is the key idea in the proof that the 2-local commuting Hamiltonian problem lies in NP [9]; In fact, one can easily construct a formal proof for claim 9 using the same arguments as in [9] (see [6]).

Thus, we shall assume from now on that all classical qubits were turned to be trivial qubits.

4 Generating a toric code state

The toric code is a special case of a $2D - CLH^*(k, 2)$ instance. We shall not restrict to the particular setting of a grid on a torus, so by saying toric code we refer to any $2D - CLH^*(k, 2)$ instance defined on a closed complex $\mathcal{K}$ (i.e it topologically has no boundary) with the usual star and plaquette local terms (equation 1.1).

Starting with the state $|0\rangle^{\otimes n}$, we measure *all* plaquettes and record the measurement results by $\bar{\lambda} = (\lambda_p)_p$ (where $\lambda_p = \pm 1$). As a result, the system collapses to a state corresponding to the measured values: $|\psi_{\bar{\lambda}}\rangle$. Note that $|\psi_{\bar{\lambda}}\rangle$ is a toric code state (i.e a groundstate of the Hamiltonian given in equation 1.1) precisely when $\lambda_p = 1$ for each plaquette p.

Whenever we have two plaquettes p_1,p_2 with $\lambda_{p_1} = \lambda_{p_2} = -1$ we can connect them by a copath γ^*, apply $L^* = \bigotimes_{e \in \gamma^*} Z_e$, and obtain a new state $|\psi_{\bar{\lambda}'}\rangle$ where λ and λ' are the same except for the value on the plaquettes p_1,p_2 (see [6] for a more elaborate explanation of logical and string operators). In other words, a pair of plaquette terms which are in their excited state can always be relaxed. After matching pairs of excitations, and annihilating them by applying string operators between them, we obtain a toric code state. It is thus left to show that such a matching always exists:

▶ **Claim 10** (even amount of excitations). *The number of plaquettes p for which $\lambda_p = -1$ is even.*

Proof. Since $\mathcal{K}$ is closed so $\prod_p B_p = \mathbf{1}$ (and also $\prod_s A_s = \mathbf{1}$). Therefore:

$$|\psi_{\bar{\lambda}}\rangle = \mathbf{1}\,|\psi_{\bar{\lambda}}\rangle = \prod_p B_p\,|\psi_{\bar{\lambda}}\rangle = \prod_p \lambda_p\,|\psi_{\bar{\lambda}}\rangle = (\prod_p \lambda_p)\,|\psi_{\bar{\lambda}}\rangle$$

It follows that $\prod_p \lambda_p = 1$. ◀

This is summarized in the following algorithm:

Algorithm - constructing a toric code state (folklore):

1. Start with the tensor product state $|0\rangle^{\otimes n}$.
2. For each star p measure B_p and record the measured value λ_p.
3. As long as $-1 \in \{\lambda_p\}_p$ choose two stars p_1, p_2 with $\lambda_{p_1} = \lambda_{p_2} = -1$, find a copath γ^* connecting them (with some linear time path-finding classical algorithm) and apply Z along that copath, that is the operator $L = \bigotimes_{q\in\gamma} Z_q$. Then change the values of $\lambda_{p_1}, \lambda_{p_2}$ from -1 to 1.

It is not hard to be convinced that a similar approach works also for a variation of the toric code where each term is as in the toric code but with some scalar factor (see [6]). This remark is relevant since the equivalence to the toric code (which we formulate in the following section) allows such factors.

5 Equivalence to the toric code

We now formulate the notion of equivalence between general $2D - CLH^*(k,d)$ instances and the toric code.

▶ **Definition 11** (boundary/coboundary qubit). A qubit is said to be in the *boundary* of the system if it is acted non-trivially by at most one plaquette; it said to be in the *coboundary* of the system if it is acted non-trivially by at most one star. Other qubits are said to be in the *interior*. A local term which acts only on interior qubits is said to be in the *interior of the system.*

Qubits that live on edges which are topologically on the boundary of the manifold are of course in the boundary of the system; however qubits which are (topologically) in the interior of the manifold can also be in the boundary/coboundary of the *system* if a Hamiltonian term *acts trivially* on them. When this happens, these qubits serve, in spirit, as “holes”. We will later exploit this fact in order to tear out holes only in the *interior* of the system to obtain the 2-local punctured Hamiltonian and a constant depth circuit that generates groundstate for it.

Following [9], we will make use of the notion of induced algebras (Definition 6) of any term in the Hamiltonian, on any set of qubits it acts on. The induced algebra from a star (plaquette) term s (p) on qubits $q_1, ...q_r$ is denoted $\mathcal{A}^s_{q_1,...,q_r}$ ($\mathcal{A}^p_{q_1,...,q_r}$). In addition, given an operator h we denote by $\langle h\rangle$ the algebra generated by this operator. We can now state the definition of equivalence to the toric code:

▶ **Definition 12** (equivalence to the toric code permitting boundaries)**.** An instance of $2D - CLH^*(k,2)$ is said to be *equivalent to the toric code* if its underlying surface $\mathcal{S}$ is closed (it topologically doesn't have boundary) and there exists a choice of basis for each qubit such that $A_s \in \left\langle Z^{\otimes|s|}\right\rangle \backslash \mathbb{C}\cdot I$, $B_p \in \left\langle X^{\otimes|p|}\right\rangle \backslash \mathbb{C}\cdot I$ for any s,p.

An instance is said to be *equivalent to the toric code permitting boundaries* if there exists a choice of basis for each qubit such that:

1. $\mathcal{A}^s_{q_1,...,q_r} = \langle Z^{\otimes r}\rangle$ for any star s, for $(q_1,...,q_r)$ a copath of qubits of s which are not in the coboundary, with no two consecutive qubits in the boundary.
2. $\mathcal{A}^p_{q'_1,...,q'_r} = \langle X^{\otimes r}\rangle$ for any plaquette p, for $(q'_1,...,q'_r)$ a path of qubits of p which are not in the boundary, with no two consecutive qubits in the coboundary.

▶ **Theorem 13** (equivalence to the toric code permitting boundaries)**.** *Every $2D - CLH^*(k,2)$ instance (after removing all classical qubits as described in subsection 3.3) is equivalent to the toric code permitting boundaries. In particular, if it has no qubits which are in the boundary or in the coboundary then it is equivalent to the toric code.*

The first step in the proof is to classify the possible induced algebras of a Hamiltonian on a single qubit in the interior and show that these algebras are always generated by a *single* Pauli operator (i.e., an operator which is equal to a Pauli matrix up to a change of basis). This can be done quite easily using the ordinary C*-algebraic techniques as in [9, 24]. The main technical part is to establish severe restrictions on the induced algebras on *pairs* of qubits (which are in the interior, roughly), and essentially showing that they must be similar to those of the toric code. This analysis involves a close and fairly technical study of the implication of the commutation relations between the Hamiltonians on the algebras that they induce.

An immediate implication of Theorem 13 is that we now know how to generate a groundstate for any $2D - CLH^*(k,2)$ instance which has no qubits in the boundary or coboundary of the system, since such instances are equivalent to the toric code.

6 Construction of punctured Hamiltonian

We are now ready to show how we can generate a groundstate of an arbitrary quasi-Euclidean $2D - CLH^*(k,2)$ instance, even when there are qubits in the boundary/coboundary.

▶ **Definition 14** (access to the boundary/coboundary)**.** A star s is said to have *access to the coboundary* if there exists a path γ starting from s which ends at a coboundary edge such that $L = \bigotimes_{q\in\gamma} X_q$ anti-commutes with A_s and commutes with any other local term. Similarly, a plaquette p is said to have *access to the boundary* if there exists a copath γ^* starting from p which ends at a boundary edge such that $L^* = \bigotimes_{q\in\gamma^*} Z_q$ anti-commutes with B_p and commutes with any other local term.

Access to the boundary or coboundary means that either L^* or L serve as an appropriate logical operator for the corresponding plaquette or star respectively in the sense that it flips its value while keeping the value of all other constraints in tact.

▶ **Lemma 15** (**Main lemma:** access to the boundary/coboundary)**.** *Let s,p be adjacent star and plaquette which are in the interior of the system. Then either s has access to the coboundary, or p has access to the boundary.*

The proof of Lemma 15 relies on the a further study of the induced algebras near the boundary/coboundary of the system. The idea is to start with an edge shared by s and p and start drawing a ribbon from it which is briefly a juxtaposition of a path and an adjacent

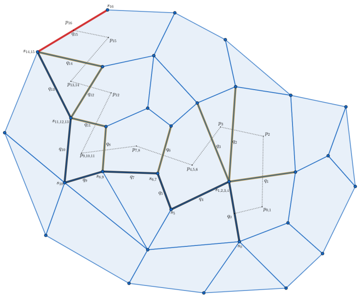


Figure 6.1 A ribbon to the boundary

copath (see Figure 6.1). We do this until we encounter a boundary/coboundary edge. At areas far from the boundary, we are in a regime which look like the toric code and thus the desired commutation and anti-commutation relations hold. Near the boundary/coboundary there are enough restrictions on the induced algebras to conclude that either the path or the copath within the ribbon can serve as the support for a logical operator which can correct p or s respectively (see [6]).

The labeled edges in bold form a ribbon. The stars and plaquettes which are shared between two adjacent edges in the ribbon are labeled as well. Ribbons include in them both a path which can easily be seen in the figure and a copath which is drawn as a dotted line. The last qubit of the ribbon is in the boundary and so we say that the first qubit has access to the boundary via a copath. See [6] for a full explanation.

Construction of punctured Hamiltonian: Let $\mathcal{W}$ denote the set consisting of all stars and plaquettes in the interior of the system which have access to the coboundary or to the boundary, respectively. This set can be thought of as the set of "fixable" terms. Let $\tilde{H}$ be the *punctured Hamiltonian:* the local Hamiltonian obtained by replacing all terms which are in $\mathcal{W}$ by the identity operator.

7 2-locality of the punctured Hamiltonian

We now show with the help of Lemma 15 that the punctured Hamiltonian $\tilde{H}$ has so many holes that it is 2-local.

The division to superparticles is based on the quasi-Euclidean condition (this is the only place we use this condition). Recall that by definition, the quasi-Euclidity condition (Definition 4) provides us with a triangulation $\mathcal{T}$ of $\mathcal{S}$ of degree $D = \mathcal{O}(1)$ such that each triangle contains a ball of radius $2k$ and is of diameter $R = \mathcal{O}(1)$ (with the ordinary graph metric with edge length 1).

We now construct a graph which will help us divide the qubits to superparticles. The vertices of this graph will be associated with terms in the Hamiltonian. A local term can be associated with a point in the surface in a natural way: each star is naturally realized as the vertex which is associated with it, and each plaquette p is associated with some arbitrarily chosen point in its interior to be called "the center of the plaquette". This allows us to precisely speak of a local term as a point on the surface.

▶ **Claim 16** (punctured triangles). *For each triangle $T \in \mathcal{T}$ there exists a term h of $\tilde{H}$ such that all of the edges attached to it (i.e the edges associated with the qubits which h acts on), are fully contained in T and moreover, h acts trivially on at least one of its qubits.*

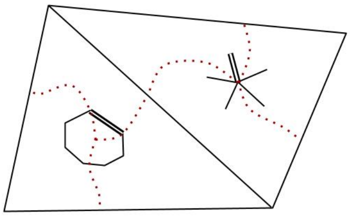


Figure 7.1 Choosing a holes in each triangle and separating to regions.

The idea of the proof is very simple: T contains a ball of radius $2k$ and thus must contain a pair of adjacent star and plaquette which are in the interior. By Lemma 15 at least one of them belongs to $\mathcal{W}$ so let h be that term.

Choose such a term h for every triangle $T \in \mathcal{T}$ and call it "the center of the triangle T". Such a term acts trivially on some edge e (when considered as a term in $\tilde{H}$; if h was removed, then this term is in fact the identity). In addition, for each 1-cell of $\mathcal{T}$, that is a side of a triangle $T \in \mathcal{T}$, choose some point in its interior to be called "the center of the 1-cell". Then connect each triangle center with 3 paths to the centers of the sides of T. Those paths should be non intersecting, contained in the interior of T (except at the end of the paths) and in addition must satisfy one more condition: clearly, those three non-intersecting paths divide T into 3 regions; the paths should be drawn such that e belongs to one region and all the other edges of h belong to the two other regions (that way h will act non-trivially on at most 2 regions). To be sure that such paths can always be drawn, it suffices to show it for an equilateral triangle - this can of course be done. Then the general case is obtained as a homeomorphism of the triangle (see figure 7.1).

According to Claim 16, every triangle includes a local term h which acts trivially on (at least) one of its edges e (this edge is marked as a double edge). Whether a star term or a plaquette term, we can connect it to the three triangle sides with three paths (dotted curves) such that e belongs to one region, and the other edges belong to the two other regions.

This construction gives rise to a graph G which highly resembles $\mathcal{T}^*$ the dual of $\mathcal{T}$. The vertices of G consist of the chosen triangles center of $\mathcal{T}$, as well as the centers of 1-cells of triangles in $\mathcal{T}$ which are on the boundary of $\mathcal{S}$. Between any two vertices of G corresponding to the centers of two triangle T_1, T_2 which share a side (i.e $T_1 \cap T_2$ is a 1-cell of $\mathcal{T}$) let there be an edge; in addition, for every triangle which has a side on the boundary of the surface, let there be an edge between the triangle center and the boundary. The edges of G are drawn on $\mathcal{S}$ as the paths constructed in the previous paragraph.

Consequently, vertices of $\mathcal{T}$ are in one-to-one correspondence with faces of G. Those faces induce a partition $\mathcal{P}$ of the set of qubits $\mathcal{Q}$ according to the face of G which they belong to (if an edge of $\mathcal{K}$ touches more then one face of $\mathcal{T}^*$ then join it to one of those faces arbitrarily) [12]. We accordingly have: $\mathcal{H} = \bigotimes_{q\in\mathcal{Q}} \mathcal{H}_q = \bigotimes_{P\in\mathcal{P}} \mathcal{H}_P$ with $\mathcal{H}_P := \bigotimes_{q\in P} \mathcal{H}_q$. We refer to each cluster $P \in \mathcal{P}$ and to its Hilbert space $\mathcal{H}_P$ as a super-particle.

The dark lines are the quasi-Euclidean triangulation. The dotted curves are the edges of the graph G realized as the chosen paths in $\mathcal{S}$. The faces of G induce a partition of the qubits into superparticles. The fatness of the triangles and the bounded degree of the triangulation implies that the superparticles' size is constant.

So far we have only used the "slimness" of a triangle condition in the definition of quasi-Euclidean condition. Here is where we need the bound on the fatness of triangles and the upper bound on its degree.

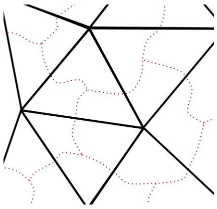


Figure 7.2 The graph embedding of G in $\mathcal{S}$.

▶ **Claim 17** (constant sized super-particles). *Each super-particle includes at most $D \cdot k^{R+2}$ qubits (in particular $\mathcal{O}(1)$).*

The proof is a straightforward consequence of Moore's bound [6, 15] which provides a bound on the number of vertices and edges which are contained in some ball in a graph of bounded degree. Since each vertex of $\mathcal{K}$ is of degree at most k, and the diameter of each triangle is at most R we immediately obtain a bound on the number of edges in each triangle and thus in each super-particle (since the number of triangles it intersects is at most D - the maximal degree of $\mathcal{T}$).

▶ **Claim 18** (punctured Hamiltonian is 2-local). *Each local term of $\tilde{H}$ acts on at most two super-particles.*

The proof follows by the observation that the only plaquettes/stars which act on 3 super-particles are the ones near the vertices of G which were removed (see Figure 7.2)

8 Completing the algorithm and the proofs for Theorems 1 & 2.

We now proof Theorems 1 & 2. By Claims 17, 18, it is possible to prepare a ground space $|\tilde{\psi}\rangle$ of $\tilde{H}$, using a constant depth quantum circuit. Given such a groundstate, we measure every $h \in \mathcal{W}$ one by one. Actually it will be simpler to measure $I - 2 \cdot \pi_h$ instead where π_h is the orthogonal projector onto the ground space of h. Record that result of the measurement by λ_h. Accordingly, having $\lambda_h = 1$ indicates that $|\tilde{\psi}\rangle$ is already a groundstate of h whereas $\lambda_h = -1$ indicates an excitation at that spot. The state we had $|\tilde{\psi}\rangle$ collapses by these measurements to a new state $|\psi\rangle$ which is an eigenstate of every $h \in \mathcal{W}$, while still being in the ground space of $\tilde{H}$. Recall that the set of terms we measured (the set $\mathcal{W}$) all have access to the boundary (Definition 14). Thus their value can be changed via string logical operators while not effecting the value of any other term. This is summarized by the following algorithm:

Algorithm (constructing a groundstate for an arbitrary quasi-Euclidean $2D - CLH^*(k, 2)$ instance):

1. If the instance has no boundary or coboundary qubits, then it is equivalent to the toric code, so apply algorithm 4 and terminate.
2. Else, generate a groundstate of $\tilde{H}$ with a constant depth quantum circuit.
3. For each term $h \in \mathcal{W}$ which was removed, measure $I - 2 \cdot \pi_h$, and record the measurement value as $\lambda_h = \pm 1$. (π_h is the orthogonal projector onto the groundspace of h).
4. Fix every $h \in \mathcal{W}$ for which $\lambda_h = -1$: if h is a star term s, find a path γ from s to the coboundary and apply $L = \bigotimes_{q \in \gamma} X_q$. If h is a plaquette term p, find a copath γ^* from p to the boundary and apply $L^* = \bigotimes_{q \in \gamma^*} Z_q$.

This proves Theorem 2. Theorem 1 follows as well: if the instance has no boundary/-coboundary qubits (and this can of course be checked efficiently by the verifier) then the system is equivalent to the toric code, so it's ground energy can be computed easily (see [6] for the case where the local terms are as in the toric code only upto a factor of a scalar). Otherwise, the problem of computing the ground energy of H reduces to computing the ground energy of $\tilde{H}$, since the verifier knows that any groundstate of $\tilde{H}$ can be corrected to a (possibly other) groundstate of $\tilde{H}$ such that all terms in $\mathcal{W}$ are satisfied (i.e the energy with respect to the terms in $\mathcal{W}$ is minimal). It is thus left to note that $\tilde{H}$ is a 2-local CLH, and this problem is in NP by [9].

9 Discussion

An interesting property of the algorithm is that all of the *quantum* operations are summed up to have only *constant* depth. Indeed, the algorithm consists of three steps: a constant depth quantum circuit that generates a groundstate for the punctured Hamiltonian, a non-constant depth computation of path finding which can be carried out in a *classical* manner, and finally a constant depth quantum circuit of logical operators (tensor product of Pauli operators).

This observation regards the complexity of the algorithm, but it is interesting also conceptually. While the quantum circuit presented here is of polynomial depth, it is enough for the verifier to obtain only a constant depth circuit description, and verify that it is indeed a groundstate of the punctured Hamiltonian, in order to be know the ground energy of the whole system (since the verifier knows that these holes can always be fixed). This means that while the time it takes to generate a groundstate for the system is concentrated on creating global entanglement, all the hardness and potential frustration of the groundstate comes into play only at the level of local entanglement of the groundstate of the 2-local punctured system.

Moreover, our results shed new light on the possible threshold phenomenon suggested in [4]. Recall that this threshold (described above in subsection 1.2) regards the fact that up until $k = 3, d = 3$, and also for $k = 2$ and arbitrary d, $CLH(k, d)$ always have trivial groundstate, which in turn implies that those problems are in NP. The threshold refers to the fact one cannot expect the exact same phenomenon for higher parameters since then there are systems with topological quantum order which are known to have no trivial groundstates. It is thus interesting that our proof extends this trivial state phenomenon even beyond this transition point into the regime of potentially global entanglements, in the sense that even here the prover hands us a description of a trivial state - a ground state of $\tilde{H}$ (even though it cannot in general be a groundstate of the actual instance). This raises the question of whether such a property holds for more general CLHs.

Can these results be extended to all 2D systems? A generalization from qubits to qudits of dimension larger than 2 would imply this, under the quasi-Euclidity assumption. Thus, the main open problem is to generalize our results to higher dimensional particles. We note that in any case one can still tear holes in a regular manner (using e.g the quasi-Euclidity assumption) to obtain a punctured Hamiltonian which is 2-local with respect to superparticles, and thus has a trivial groundstate. The problem is that we do not know how to fix those holes later on: our characterization of $2D - CLH^*(k, 2)$ instances (i.e Theorem 13) and of fixable terms (namely, the creation of logical operators in Lemma 15) strongly uses the fact that the particles are 2 dimensional. It is open whether further generalization could be derived using more general characterizations of commuting local Hamiltonians, perhaps over general finite groups (e.g the quantum double model [20]).

We mention that if indeed the results can be generalized to qudits, it might also be possible to generalize to 3D manifolds or more, perhaps in an inductive manner.

A more technical question is whether the quasi-Euclidity condition can be relaxed. Quasi-Euclidity seems closely related to the notion of 1-localizablity introduced in Hastings' paper [18] already mentioned (In fact, the quasi-Euclidity condition we use can be replaced by the technical assumption used in [18] regarding the girth of the complex; we could then deduce the existence of a groundstate for $\tilde{H}$ from 1-localizabilty instead of 2-locality). This raises the question of whether manifolds which are very non-Euclidean and which have low girths, can exhibit much more complex multi-particle entanglement (we mention in this context [16]).

References

1 Miguel Aguado and Guifre Vidal. Entanglement renormalization and topological order. *Physical review letters*, 100(7):070404, 2008.

2 Dorit Aharonov, Itai Arad, Zeph Landau, and Umesh Vazirani. The detectability lemma and quantum gap amplification. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 417–426. ACM, 2009.

3 Dorit Aharonov, Itai Arad, and Thomas Vidick. Guest column: the quantum pcp conjecture. *Acm sigact news*, 44(2):47–79, 2013.

4 Dorit Aharonov and Lior Eldar. On the complexity of commuting local hamiltonians, and tight conditions for topological order in such systems. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 334–343. IEEE, 2011.

5 Dorit Aharonov and Lior Eldar. The commuting local hamiltonian problem on locally expanding graphs is approximable in \mathsf {{NP}} np. *Quantum Information Processing*, 14(1):83–101, 2015.

6 Dorit Aharonov, Oded Kenneth, and Itamar Vigdorovich. On the complexity of two dimensional commuting local hamiltonians. *arXiv preprint arXiv:1803.02213*, 2018.

7 Dorit Aharonov and Tomer Naveh. Quantum np-a survey. *arXiv preprint quant-ph/0210077*, 2002.

8 S Bravyi, MB Hastings, and F Verstraete. Lieb-robinson bounds and the generation of correlations and topological quantum order. *Physical review letters*, 97(5):050401, 2006.

9 Sergey Bravyi and Mikhail Vyalyi. Commutative version of the k-local hamiltonian problem and common eigenspace problem. *arXiv preprint quant-ph/0308021*, 2003.

10 Sergey B Bravyi and A Yu Kitaev. Quantum codes on a lattice with boundary. *arXiv preprint quant-ph/9811052*, 1998.

11 Dmitri Burago, Yuriĭ Dmitrievich Burago, and Sergeĭ Ivanov. *A course in metric geometry.* American Mathematical Society, 2001. doi:10.1090/gsm/033.

12 Reinhard Diestel. *Graph theory.* Springer Publishing Company, Incorporated, 2017.

13 Michael H Freedman and Matthew B Hastings. Quantum systems on non-k-hyperfinite complexes: A generalization of classical statistical mechanics on expander graphs. *arXiv preprint arXiv:1301.1363*, 2013.

14 David Gosset, Jenish C Mehta, and Thomas Vidick. Qcma hardness of ground space connectivity for commuting hamiltonians. *Quantum*, 1:16, 2017.

15 Jonathan L Gross, Jay Yellen, and Ping Zhang. *Handbook of graph theory.* Chapman and Hall/CRC, 2013.

16 Larry Guth and Alexander Lubotzky. Quantum error correcting codes and 4-dimensional arithmetic hyperbolic manifolds. *Journal of Mathematical Physics*, 55(8):082202, 2014.

17 Matthew B Hastings. Matrix product operators and central elements: Classical description of a quantum state. *Geometry & Topology Monographs*, 18(115-160):276, 2012.

18 Matthew B Hastings. Trivial low energy states for commuting hamiltonians, and the quantum pcp conjecture. *arXiv preprint arXiv:1201.3387*, 2012.

19 Allen Hatcher. *Algebraic topology.* Cambridge University Press, 2002.

20 A Yu Kitaev. Fault-tolerant quantum computation by anyons. *Annals of Physics*, 303(1):2–30, 2003.

21 Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation.* Number 47 in Graduate studies in mathematics. American Mathematical Soc., 2002.

22 Zeph Landau, Umesh Vazirani, and Thomas Vidick. A polynomial time algorithm for the ground state of one-dimensional gapped local hamiltonians. *Nature Physics*, 11(7):566, 2015.

23 Leonid Anatolevich Levin. Universal sequential search problems. *Problemy Peredachi Informatsii*, 9(3):115–116, 1973.

24 Norbert Schuch. Complexity of commuting hamiltonians on a square lattice of qubits. *arXiv preprint arXiv:1105.2843*, 2011.

25 Richard Evan Schwartz. *Mostly surfaces*, volume 60. American Mathematical Soc., 2011.

26 Masamichi Takesaki. Theory of operator algebras. i. reprint of the first (1979) edition. encyclopaedia of mathematical sciences, 124. operator algebras and noncommutative geometry, 5, 2002.

Quantum Lower Bounds for Tripartite Versions of the Hidden Shift and the Set Equality Problems

Aleksandrs Belovs[1]
Faculty of Computing, University of Latvia, Raina 19, Riga, Latvia

Ansis Rosmanis[2]
Centre for Quantum Technologies, National University of Singapore,
Block S15, 3 Science Drive 2, Singapore

Abstract

In this paper, we study quantum query complexity of the following rather natural tripartite generalisations (in the spirit of the 3-sum problem) of the hidden shift and the set equality problems, which we call the 3-shift-sum and the 3-matching-sum problems.

The 3-shift-sum problem is as follows: given a table of $3\times n$ elements, is it possible to circularly shift its rows so that the sum of the elements in each column becomes zero? It is promised that, if this is not the case, then no 3 elements in the table sum up to zero. The 3-matching-sum problem is defined similarly, but it is allowed to arbitrarily permute elements within each row. For these problems, we prove lower bounds of $\Omega(n^{1/3})$ and $\Omega(\sqrt{n})$, respectively. The second lower bound is tight.

The lower bounds are proven by a novel application of the dual learning graph framework and by using representation-theoretic tools from [7].

2012 ACM Subject Classification Theory of computation → Quantum query complexity

Keywords and phrases Adversary Bound, Dual Learning Graphs, Quantum Query Complexity, Representation Theory

Digital Object Identifier 10.4230/LIPIcs.TQC.2018.3

Related Version `https://arxiv.org/abs/1712.10194`

Acknowledgements Part of this research was done while A.B. was visiting the Centre for Quantum Technologies at the National University of Singapore. A.B. would like to thank Miklos Santha for hospitality.

1 Introduction

One of the starting points of this paper was the following problem, posed by Aaronson and Ambainis [1]: construct a partial Boolean function with polylogarithmic quantum query complexity but whose randomised query complexity is $\omega(\sqrt{n})$, where n is the number of input variables. There are relatively many functions known with the required quantum query complexity and randomised query complexity $\Theta(\sqrt{n})$. For instance, one can take the forrelation problem of [1] with quantum query complexity 1, or the better-known hidden subgroup problem [13]. However, no function with polylogarithmic quantum query complexity

[1] This research is partially supported by the ERDF project number 1.1.1.2/I/16/113. Part of this work was done while supported by the ERC Advanced Grant MQC.

[2] This research is partially funded by the Singapore Ministry of Education and the National Research Foundation under grant R-710-000-012-135.

13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2018).
Editor: Stacey Jeffery; Article No. 3; pp. 3:1–3:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and randomised query complexity $\omega(\sqrt{n})$ is known. As shown in [10, 2], such a function would also yield a larger than 5/2-power separation between quantum and randomised query complexities for *total* Boolean functions.

Aaronson and Ambainis proposed a candidate function, which they call the k-fold forrelation. It has a very simple quantum $O(1)$-query algorithm, but it seems hard to lower bound its randomised query complexity. However, it is also possible to go in the opposite direction: find a function whose randomised query complexity is $\omega(\sqrt{n})$, and construct an efficient quantum algorithm computing this function. A potential candidate might be a modification of a function already known to be easy quantumly, preserving the hope the modification is still easy.

One particularly neat starting function, in our opinion, is the following *hidden shift problem*. Given two strings $x, y \in [q]^n$, the task is to distinguish two cases: in the positive case, x is a circular shift of y; in the negative case, all the input variables in x and y are distinct. This problem is equivalent to the hidden subgroup problem in the dihedral group [17], and its quantum query complexity is logarithmic.[3] It is also easy to see that its randomised query complexity is $\Theta(\sqrt{n})$.

In this paper we consider the following modification, which we call the *3-shift-sum* problem. We are given an input string $x \in [q]^{3n}$, which we treat as a $3 \times n$ table. In the positive case, it is possible to circularly shifts the rows of the table so that the sum of the elements in each column becomes divisible by q. In the negative case, no matter how we shift the rows, there is no column with the sum of its elements divisible by q. (In other words, there is no three elements, one from each row, whose sum is divisible by q.) This is a natural amalgamation of the hidden shift and the 3-sum problem, both studied quantumly.

It is easy to see that the randomised query complexity of this problem is $\Theta(n^{2/3})$. This raises the question of what its quantum query complexity is. Our first result is a simple proof that, unlike the hidden shift problem, the quantum query complexity of the 3-shift-sum problem is polynomial: $\Omega(n^{1/3})$. Thus, the 3-shift-sum problem fails to provide the desired separation.

Similarly as the 3-shift-sum problem is a tripartite version of the hidden shift problem, the *3-matching-sum* problem is a tripartite version of the *set equality* problem. In the set equality problem, the negative inputs are as in the hidden shift problem, but in a positive input, y is an arbitrary permutation of x, not necessary a circular shift. Unlike the hidden shift problem, the set equality problem has polynomial quantum query complexity: $\Theta(n^{1/3})$ [23, 25, 7]. In our tripartite version of it, the negative inputs are the same as in the 3-shift-sum problem, but for a positive input, there exists an arbitrary permutation of the elements within each row such that the sum of each column becomes divisible by q. Our second result is a complete characterisation of the quantum query complexity of this problem: it is $\Theta(\sqrt{n})$.

1.1 Techniques

Our main tool is the framework of dual learning graphs, which is "compiled" to the adversary lower bound.

[3] The canonical version of the Dihedral HSP also assumes that all the symbols in x are pairwise distinct and all the symbols in y are pairwise distinct. However, this condition is not relevant for the query complexity being logarithmic as easily follows from [14, Theorem 2]. Since it is not immediately obvious how to generalise this condition for the tripartite version, we omitted it from our definition of the hidden shift problem. Also note that this is the decision version of the problem, where it is not required to find the shift. The latter may be difficult if x contains repeated symbols.

The first version of the adversary method was developed by Ambainis [3]. This version, later known as the positive-weighted adversary, is easy to use, and it has found many applications, but it is also subject to some limitations: the certificate complexity [24, 26] and the property testing [15] barriers. The property testing barrier, which is relevant to our problems, states that, if any positive input differs from any negative input on at least ε fraction of the input variables, the positive-weighted adversary fails to prove a lower bound better than $\Omega(1/\varepsilon)$. In most cases $\varepsilon = \Omega(1)$, thus this only gives a trivial lower bound.

The next version of the bound, the negative-weighted adversary [15], is known to be tight [20], but it is also harder to apply. An application of the bound to the k-sum problem was obtained in [9]. This result was later stated in the framework of dual learning graphs [6], which we are about to describe.

Learning graphs is a model of computation introduced in [4, 5]. They are most naturally stated in terms of certificate structures, which describe where 1-certificates can be located in a positive input. Learning graphs capture quantum query complexity of certificate structures in the following sense. Let L be the learning graph complexity of a certificate structure $\mathcal{C}$. First, for *any* function with certificate structure $\mathcal{C}$, there exists a quantum algorithm solving it in $O(L)$ queries. Second, there exists *some* function with certificate structure $\mathcal{C}$ and quantum query complexity $\Omega(L)$. In general these functions are rather contrived, yet one example of them being natural are the following sum problems. A sum problem is a *total* function parametrised by a family $\mathcal{S}$ of $O(1)$-sized subsets of $[n]$. The task, given an input string $x \in [q]^n$, is to detect whether there exists $S \in \mathcal{S}$ such that $\sum_{i\in S} x_i$ is divisible by q. Note that our problems do not fall into this category, because every positive input is promised to have many such subsets.

While dual learning graphs give tight lower bounds for all of the above sum problems, in general, of course, they do not give lower bounds for all problems with a given certificate structure. For example, the learning graph complexity of the certificate structure corresponding to the hidden shift problem is $\Theta(n^{1/3})$, whereas its quantum query complexity is logarithmic. What about the 3-shift-sum problem? It turns out that dual learning graphs are still of help here, but in a slightly different way. The learning graph complexity of the corresponding certificate structure is $\Theta(\sqrt{n})$, yet we do not know whether it can be converted into a quantum query lower bound. However, a dual learning graph for a *different* certificate structure can be converted into, albeit not tight, but still a polynomial lower bound. This shows that dual learning graphs are more versatile than we thought.

Another interesting feature of our result is that it might be the simplest constructed example of the adversary bound surpassing the property testing barrier. Examples of the negative-weighted adversary breaking the certificate complexity and the property testing barriers were already obtained in [15]. But [15] did not cover the most interesting regime $\varepsilon = \Omega(1)$ of the property testing barrier. The sum problems of [6] are relatively simple examples of overcoming the certificate complexity barrier. An example for the $\varepsilon = \Omega(1)$ regime of the property testing barrier was constructed in [7], but the construction is quite technical. Our result gives a similar example by much simple means, comparable to that of [6].

Concerning the 3-matching-sum problem, our lower bound is an application of the technique developed for the set equality problem [7]. It is based on the representation theory of the symmetric group. Surprisingly, the technique can be used for the 3-matching-sum problem with essentially no modifications: our proof uses representation theory to a minimal extent, and mostly follows from combinatorial estimates involving the dual learning graph. This indicates that our technique has a potential to be used in proving lower bounds for other symmetric problems.

1.2 Results in property testing

In the property testing model, one is given some property (a set of positive inputs), and the task is to distinguish whether the input possesses the property, or is ε-far, in the relative Hamming distance, from any input that has the property.

Overcoming the property testing barrier automatically gives a lower bound for a property testing problem—that of testing whether the input is positive. But it is not always the most natural way to state the problem. We give an example of a lower bound for a problem that is most naturally stated in the setting of property testing.

The 3-shift-sum problem, as formulated above, must have relatively large q for the problem to be interesting. For instance, it is easy to see that for $q = 2$ there are almost no negative inputs. In our lower bound, we require that $q = \Omega(n^3)$. But it is possible to formulate a version of the problem that is interesting even when the input alphabet is Boolean. Define the set of positive inputs as before, and define the set of negative inputs as being at relative Hamming distance at least, say, $1/7$ to it. We prove a lower bound of $\Omega(n^{1/3})$ also for this version of the problem.

Although there is quite a number of quantum algorithms for property testing problems, there are not so many quantum lower bounds known. (An interested reader might consult a recent survey [19] for more information on the topic.) One of the main reasons, of course, is the property testing barrier for the positive-weighted adversary. Up to our knowledge, our result is the first property testing lower bound proven using the adversary method, which answers the problem mentioned in [19]. This shows yet another area of applications of dual learning graphs.

2 Preliminaries

For positive integers m and $\ell \geq m$, let $[m]$ denote the set $\{1, 2, ..., m\}$ and $[m..\ell]$ denote the set $\{m, m+1, \dots, \ell\}$. For P a predicate, we use 1_P to denote the variable that is 1 if P is true, and 0 otherwise.

For an $\mathcal{I} \times \mathcal{J}$-matrix A, $i \in \mathcal{I}$, and $j \in \mathcal{J}$, we denote by $A[\![i,j]\!]$ its (i,j)-th entry. For $\mathcal{I}' \subseteq \mathcal{I}$ and $\mathcal{J}' \subseteq \mathcal{J}$, $A[\![\mathcal{I}',\mathcal{J}']\!]$ denotes the corresponding submatrix. We use similar notation also for vectors. Next, $\|\cdot\|$ denotes the spectral norm (the largest singular value), and $\circ$ denotes the Hadamard (i.e., entry-wise) product of matrices. We often identify projectors with the spaces they project onto.

2.1 Adversary Bound

For background on quantum query complexity the reader may refer to [11]. In the paper, we only require the knowledge of the (negative-weighted) adversary bound for decision problems, which we are about to define.

Let $f\colon \mathcal{D} \to \{0,1\}$ with $\mathcal{D} \subseteq [q]^n$. An *adversary matrix* for f is a non-zero $f^{-1}(1) \times f^{-1}(0)$-matrix Γ. For any $j \in [n]$, the $f^{-1}(1) \times f^{-1}(0)$-matrix Δ_j is defined by

$$\Delta_j[\![x,y]\!] = \begin{cases} 0, & \text{if } x_j = y_j; \\ 1, & \text{if } x_j \neq y_j. \end{cases} \qquad (1)$$

▶ **Theorem 1** (Adversary bound, [15, 18, 9]). *In the above notation, the quantum query complexity of the function f is $\Theta\big(\mathrm{ADV}^{\pm}(f)\big)$, where $\mathrm{ADV}^{\pm}(f)$ is the optimal value of the semi-definite program*

$$\begin{aligned} &\text{maximise} && \|\Gamma\| && \text{(2a)}\\ &\text{subject to} && \|\Delta_j \circ \Gamma\| \le 1 \qquad \textit{for all } j \in [m], && \text{(2b)} \end{aligned}$$

where the maximisation is over all adversary matrices Γ for f.

We can choose any adversary matrix Γ and scale it so that the condition $\|\Delta_j \circ \Gamma\| \le 1$ holds. Thus, we often use the condition $\|\Delta_j \circ \Gamma\| = \mathrm{O}(1)$ instead of $\|\Delta_j \circ \Gamma\| \le 1$.

Working with the matrix $\Delta_j \circ \Gamma$ might be cumbersome, so we do the following transformation instead. We write $\Gamma \xmapsto{\Delta_j} \Gamma'$ if $\Gamma \circ \Delta_j = \Gamma' \circ \Delta_j$. In other words, we modify the entries of Γ with $x_j = y_j$. Now, from the fact [18] that $\gamma_2(\Delta_j) = \max_B \big\{\|\Delta_j \circ B\| : \|B\| \le 1\big\} \le 2$, we conclude that $\|\Delta_j \circ \Gamma\| \le 2\|\Gamma_j\|$, hence we can replace $\Delta_j \circ \Gamma$ with Γ' in (2b).

It is sometimes convenient [9] to allow several rows or columns corresponding to the same input x. We add labels to distinguish different rows corresponding to the same input.

2.2 Certificate Structures and Dual Learning Graphs

Let $f\colon \mathcal{D} \to \{0,1\}$ be a function with domain $\mathcal{D} \subseteq [q]^n$. For $x \in f^{-1}(1)$, a *certificate* for x is a subset $S \subseteq [n]$ such that $f(z) = 1$ for all $z \in \mathcal{D}$ satisfying $x_i = y_i$ for all $i \in S$. A *certificate structure* $\mathcal{C}$ is a collection of non-empty subsets of $2^{[n]}$. We say that f *has* certificate structure $\mathcal{C}$ if, for every $x \in f^{-1}(1)$, there exists $\mathcal{M} \in \mathcal{C}$ such that every $S \in \mathcal{M}$ is a certificate for x. It is natural to assume that all $\mathcal{M} \in \mathcal{C}$ are upward closed.

There are two formulations of the learning graph complexity: primal and dual. For the purposes of this paper, it is enough to state the dual one. A *dual learning graph* for a certificate structure $\mathcal{C}$ is a feasible solution to the following optimisation problem:

$$\begin{aligned} &\text{maximise} && \sqrt{\textstyle\sum_{\mathcal{M}\in\mathcal{C}} \alpha(\mathcal{M},\emptyset)^2} && && \text{(3a)}\\ &\text{subject to} && \sum_{\mathcal{M}\in\mathcal{C}} \big(\alpha(\mathcal{M},S) - \alpha(\mathcal{M},S\cup\{j\})\big)^2 \le 1 && \forall S \subseteq [n],\ \forall j \in [n]\setminus S; && \text{(3b)}\\ & && \alpha(\mathcal{M},S) = 0 && \forall S \in \mathcal{M}; && \text{(3c)}\\ & && \alpha(\mathcal{M},S) \in \mathbb{R} && \forall \mathcal{M} \in \mathcal{C},\ \forall S \subseteq [n]. && \text{(3d)} \end{aligned}$$

The optimal value of this optimisation problem is called the learning graph complexity of $\mathcal{C}$.

We call a solution to the dual learning graph for $\mathcal{C}$ any mapping $\alpha(\mathcal{M},S)$ satisfying (3d), where we implicitly assume (3c). A solution is feasible if it satisfies (3b). It is easy to see that any optimal solution $\alpha(\mathcal{M},S)$ to (3) is entry-wise non-negative and non-increasing in S. We will implicitly assume that any feasible solution satisfies these requirements.

Inspired by this optimisation problem, we define the norm of a solution α as $\|\alpha\| = \max_{S\subseteq[n]} \sqrt{\sum_{\mathcal{M}\in\mathcal{C}} \alpha(\mathcal{M},S)^2}$. It satisfies the usual axioms of a norm, although we will not use this fact. For $j \in [n]$, we define an operation ∂_j given by

$$\partial_j\alpha(\mathcal{M},S) = \begin{cases} \alpha(\mathcal{M},S) - \alpha(\mathcal{M},S\cup\{j\}), & \text{if } j \notin S;\\ 0, & \text{if } j \in S. \end{cases}$$

If α is a solution to the dual learning graph, so is $\partial_j\alpha$. Condition (3b) can be restated as $\|\partial_j\alpha\| \le 1$ for all $j \in [n]$. If $\alpha(\mathcal{M},S)$ is non-increasing in S, the objective value (3a) is given by $\|\alpha\|$.

Dual learning graphs have close connection to adversary matrices, which we discuss in Section 4.

2.3 Representation Theory

In this section, we introduce basic notions from the representation theory of finite groups with special emphasis on the symmetric group. For more background, the reader may refer to [12, 22] for general theory, and to [16, 21] for the special case of the symmetric group.

Assume G is a finite group. The *group algebra* $\mathbb{C}G$ is the complex vector space with the elements of G forming an orthonormal basis, where the multiplication law of G is extended to $\mathbb{C}G$ by linearity. A (left) *G-module*, also called a *representation* of G, is a complex vector space V with the left multiplication operation by the elements of $\mathbb{C}G$ satisfying the usual associativity and distributivity conditions. We can treat elements of $\mathbb{C}G$ as linear operators acting on V.

A *G-morphism* (or just morphism, if G is clear from the context) between two G-modules V and W is a linear operator $\theta\colon V \to W$ that commutes with all $\alpha \in \mathbb{C}G$: $\theta\alpha = \alpha\theta$, where the first α acts on V and the second one on W.

A G-module is called *irreducible* (or just irrep for irreducible representation) if it does not contain a non-trivial G-submodule. For any G-module V, one can define its *canonical* decomposition into the direct sum of *isotypic* subspaces, each spanned by all copies of a fixed irrep in V. Different isotypic subspaces in this decomposition are orthogonal. If an isotypic subspace contains at least one copy of the irrep, we say that V *uses* this irrep.

If G and H are finite groups, then the irreducible $G \times H$-modules are of the form $V \otimes W$ where V is an irreducible G-module and W is an irreducible H-module. And the corresponding group action is given by $(g, h)(v \otimes w) = gv \otimes hw$, with $g \in G$, $h \in H$, $v \in V$, and $w \in W$, which is extended by linearity.

We use the following results:

▶ **Lemma 2** (Schur's Lemma). *Assume $\theta\colon V \to W$ is a morphism between two irreducible G-modules V and W. Then, $\theta = 0$ if V and W are non-isomorphic; otherwise, θ is uniquely determined up to a scalar multiplier.*

▶ **Lemma 3** ([7]). *Let $\theta\colon V \to W$ be a morphism between two G-modules V and W. Then, there exists an irrep in V all consisting of right-singular vectors of θ of singular value $\|\theta\|$. (We call such right-singular vectors principal.)*

Let $\mathfrak{S}_L$ denote the *symmetric group* on a finite set L, that is, the group with the permutations of L as elements, and composition as the group operation. If m is a positive integer, $\mathfrak{S}_m$ denotes the isomorphism class of the symmetric groups $\mathfrak{S}_L$ with $|L| = m$. Representation theory of $\mathfrak{S}_m$ is closely related to *Young diagrams*, defined as follows.

A *partition* λ of an integer m is a non-increasing sequence $(\lambda_1, \dots, \lambda_t)$ of positive integers satisfying $\lambda_1 + \cdots + \lambda_t = m$. A partition $\lambda = (\lambda_1, \dots, \lambda_t)$ is often represented in the form of a Young diagram that consists, from top to bottom, of rows of $\lambda_1, \lambda_2, \dots, \lambda_t$ boxes aligned by the left side. We say that a partition *has k boxes below the first row* if $\lambda_1 = m - k$. For each partition λ of m, there exists an irreducible $\mathfrak{S}_m$-module S^λ, called the *Specht module*. All these modules are pairwise non-isomorphic, and give a complete list of all the irreps of $\mathfrak{S}_m$.

3 Formulation of the Problems and Easy Observations

In this section, we formulate the 3-shift-sum problem and define the closely related 3-matching-sum problem. We also sketch proofs of few simple observations about these problems.

Both the 3-shift-sum and the 3-matching-sum are partial Boolean functions defined on $[q]^{3n}$, with q and n positive integers. The $3n$ input variables are divided into three groups $A = [1..n]$, $B = [n+1..2n]$, and $C = [2n+1..3n]$. A *3-dimensional matching* is a partition μ of the set $[3n]$ into n triples, $\mu = \{T_1, \dots, T_n\}$, such that $|T_i \cap A| = |T_i \cap B| = |T_i \cap C| = 1$ for all i. This is a natural generalisation of the usual (2-dimensional) matching between sets A and B. We denote the set of 3-dimensional matchings by M_{M} (we omit n, assuming its value is clear from the context). We consider a special type of 3-dimensional matchings, we call *3-shifts*. A 3-shift is a matching $\mu = \{T_1, \dots, T_n\}$ such that there exist two numbers $b, c \in [n]$ such that $T_i = \bigl\{i,\ n+1+(i+b \bmod n),\ 2n+1+(i+c \bmod n)\bigr\}$ for all i. We denote the set of 3-shifts by M_{S}.

We define the *3-shift-sum* and the *3-matching-sum* problems as follows. Let M_{Q} stand for M_{S} in 3-shift-sum and for M_{M} in 3-matching-sum. In a positive input $x \in [q]^{3n}$, there exists $\mu \in M_{\mathrm{Q}}$ such that $x_a + x_b + x_c$ is divisible by q for every triple $\{a, b, c\} \in \mu$. We say that x is *of the form* μ in this case. In a negative input $y \in [q]^{3n}$, we have $y_a + y_b + y_c \not\equiv 0 \pmod q$ for any choice of $a \in A$, $b \in B$, and $c \in C$. The task is to determine whether the input is positive or negative, provided that one of the two options holds. Since 3-shift-sum is a special case of 3-matching-sum, the latter is a harder problem.

3.1 Randomised and Quantum Complexity

Let us describe what we can immediately say about quantum and randomised query complexities of these problems. Neither result will be relevant later in the paper.

▶ **Proposition 4.** *The quantum query complexity of the 3-shift-sum and the 3-matching-sum problems is $O(\sqrt{n})$.*

Proof sketch. Consider a positive input x, and let $\mu \in M_{\mathrm{Q}}$ be its form. Take random subsets $A' \subseteq A$ and $B' \subseteq B$ of size approximately $\sqrt{n}$, and query all the variables in $A' \cup B'$. With high probability, there exists $T \in \mu$ that intersects both A' and B'. Now use Grover's search to find an element $c \in C$ satisfying $x_a + x_b + x_c \equiv 0 \pmod q$ for some $a \in A'$ and $b \in B'$. ◀

▶ **Proposition 5.** *The randomised query complexity of the 3-shift-sum and the 3-matching-sum problems is $\Theta(n^{2/3})$.*

The proof is totally standard, and it can be found in the full version of the paper [8].

3.2 Certificate Structures

It is easy to describe the certificate structures $\mathcal{C}_{\mathrm{S}}$ and $\mathcal{C}_{\mathrm{M}}$ of the 3-shift-sum and the 3-matching-sum problems. For each $\mu \in M_{\mathrm{Q}}$, there is a corresponding $\mathcal{M}_\mu \in \mathcal{C}_{\mathrm{Q}}$ obtained as follows: a subset $S \subseteq [n]$ is in $\mathcal{M}_\mu$ if and only if there exists a triple $T \in \mu$ satisfying $T \subseteq S$.

The lower bound from the following proposition will be our main source of inspiration when constructing adversary bounds later in the paper.

▶ **Proposition 6.** *The learning graph complexity of the certificate structures $\mathcal{C}_{\mathrm{S}}$ and $\mathcal{C}_{\mathrm{M}}$ is $\Theta(\sqrt{n})$.*

Proof. The upper bound is similar to Proposition 4, and we omit the proof. The upper bound is stated here for completeness, and we do not use it further in the paper.

Let us prove the lower bound. For that we have to construct a feasible solution to the dual learning graph. For $\mathcal{M} \in \mathcal{C}_{\mathrm{Q}}$, define

$$\alpha(\mathcal{M}, S) = \frac{1}{\sqrt{|M_{\mathrm{Q}}|}} \max\bigl\{\sqrt{n} - |S|, 0\bigr\} \qquad \text{if } S \notin \mathcal{M}, \tag{4}$$

and as 0 otherwise. It is easy to see that the objective value (3a) is $\sqrt{n}$, and that (3c) holds.

It remains to check (3b). Fix S and j. If $|S| \ge \sqrt{n}$, then the left-hand side of (3b) is zero, so assume $|S| \le \sqrt{n}$. We have the following contributions to the left-hand side of (3b):

- If $S \cup \{j\} \notin \mathcal{M}$, then the value of $\alpha(\mathcal{M}, S)$ changes by $\frac{1}{\sqrt{|M_{\mathrm{Q}}|}}$ as $|S|$ increases by 1.
- If $\mu \in M_{\mathrm{Q}}$ is taken uniformly at random, the probability is $O\big((|S|/n)^2\big) = O(1/n)$ that $S \notin \mathcal{M}_\mu$ but $S \cup \{j\} \in \mathcal{M}_\mu$. In this case, $\alpha(\mathcal{M}_\mu, S)$ changes by at most $\sqrt{\frac{n}{|M_{\mathrm{Q}}|}}$.

Altogether we have:

$$\sum_{\mathcal{M}\in\mathcal{C}} \big(\alpha(\mathcal{M}, S) - \alpha(\mathcal{M}, S\cup\{j\})\big)^2 \le |M_{\mathrm{Q}}| \cdot \frac{1}{|M_{\mathrm{Q}}|} + O\left(\frac{|M_{\mathrm{Q}}|}{n}\right) \cdot \frac{n}{|M_{\mathrm{Q}}|} = O(1).$$

Scaling down α by a constant factor, we get a feasible solution with objective value $\Omega(\sqrt{n})$. ◀

4 Basic Definitions

In this section we introduce our basic notation, and describe a procedure of converting a solution to the dual learning graph into an adversary matrix. This is a general procedure from [6] tailored for the special case of the 3-shift-sum and the 3-matching-sum problems. This procedure does not immediately result in good adversary matrices for these problems, but we are able modify it in Sections 5 and 6 so that it works. Let again M_{Q} stand for either M_{S} or M_{M}.

4.1 Fourier Basis

Let $\mathcal{H} = \mathbb{C}^{\mathbb{Z}_q}$ and $e_0, \dots, e_{q-1}$ be the Fourier basis of $\mathcal{H}$. Recall that it is an orthonormal basis given by $e_i[\![j]\!] = \frac{1}{\sqrt{q}}\omega^{ij}$, where $\omega = \mathrm{e}^{2\pi\mathrm{i}/q}$. For m a positive integer, the Fourier basis of $\mathcal{H}^{\otimes m}$ is given by tensor products $e_{a_1} \otimes \cdots \otimes e_{a_m}$. A component e_{a_i} in this tensor product is called non-zero if $a_i \neq 0$. The weight of the Fourier basis element is the number of non-zero components.

We define two projectors in $\mathcal{H}$: $\Pi_0 = e_0 e_0^*$ and $\Pi_1 = I - \Pi_0 = \sum_{i=1}^{q-1} e_i e_i^*$. All the entries of Π_0 are equal to $1/q$. Important relations are $\Pi_0 \overset{\Delta}{\longmapsto} \Pi_0$ and $\Pi_1 \overset{\Delta}{\longmapsto} -\Pi_0$, where Δ is as in (1) and acts on the sole variable. For two sets $R \subseteq T$, we define a projector Π_R^T in the space $\mathcal{H}^T$ by $\Pi_R^T = \bigotimes_{j\in T} \Pi_{1_{j\in R}}$. As R ranges over all subsets of T, this gives an orthogonal decomposition of $\mathcal{H}^T$. By the above relations:

$$\Pi_R^T \overset{\Delta_j}{\longmapsto} \Pi_R^T \quad \text{if} \quad j \notin R \qquad \text{and} \qquad \Pi_R^T \overset{\Delta_j}{\longmapsto} -\Pi_{R\setminus\{j\}}^T \quad \text{if} \quad j \in R. \tag{5}$$

If $\mathcal{A}$ is a collection of subsets of T, we can define projector $\Pi_{\mathcal{A}}^T = \sum_{R\in\mathcal{A}} \Pi_R^T$. We clearly have $\Pi_{\mathcal{A}}\Pi_{\mathcal{B}} = \Pi_{\mathcal{A}\cap\mathcal{B}}$. We will use this construction only for some special cases, in particular, for a positive integer k, we define $\Pi_k^T = \sum_{R\subseteq T, |R|=k} \Pi_R^T$.

4.2 Basic Operators

Let $\mu = \{T_1, \dots, T_n\}$ be a 3-dimensional matching. Let $\mathcal{P}^\mu$ denote the set of positive inputs of form μ. We use $\mathcal{P}$ for the set of pairs (μ, x) with $\mu \in M_{\mathrm{Q}}$ and $x \in \mathcal{P}^\mu$. Think of $\mathcal{P}$ as the set of positive inputs with additional labels so that some inputs x can appear multiple times. We use $\mathcal{N}$ for the set of negative inputs, and $\mathcal{U} = [q]^{3n}$ for the set of all strings. Similarly to the proof of Proposition 5, $\mathcal{U}$ will be close to $\mathcal{N}$, and we use the former as a proxy for the latter.

Now assume T is a triple of elements. Think of it as an element of a 3-dimensional matching μ. Denote

$$P^T = \bigl\{(a,b,c) \in [q]^T \mid a+b+c \equiv 0 \pmod q\bigr\}. \tag{6}$$

Thus, $\mathcal{P}^\mu$ is the Cartesian product $\prod_{T\in\mu} P^T$. For $R \subseteq T$, define $\Psi^T_R = \sqrt{q}\,\Pi^T_R\,[\![P^T,[q]^T]\!]$, where the factor $\sqrt{q}$ is introduced to account for the reduced number of rows. For $S \subseteq [3n]$, let

$$\Psi^\mu_S = \bigotimes\nolimits_{T\in\mu} \Psi^T_{S\cap T} = q^{n/2}\,\Pi^{[3n]}_S[\![\mathcal{P}^\mu,\mathcal{U}]\!].$$

As for $\Pi^T_{\mathcal{A}}$, we will use $\Psi^\mu_{\mathcal{A}} = \sum_{S\in\mathcal{A}} \Psi^\mu_S$ for a family $\mathcal{A}$ of subsets of $[3n]$. Again, $\Psi^\mu_{\mathcal{A}}\Pi^{[3n]}_{\mathcal{B}} = \Psi^\mu_{\mathcal{A}\cap\mathcal{B}}$. Using (5), we have

$$\Psi^\mu_S \xmapsto{\Delta_j} \Psi^\mu_S \quad\text{if}\quad j\notin S \qquad\text{and}\qquad \Psi^\mu_S \xmapsto{\Delta_j} -\Psi^\mu_{S\setminus\{j\}} \quad\text{if}\quad j\in S. \tag{7}$$

4.3 From Dual Learning Graphs to Adversary Matrices

Now we explain how to convert a solution α to the dual learning graph (3) into a $\mathcal{P}\times\mathcal{U}$-matrix $G(\alpha)$. In [6], the adversary matrix Γ was obtained by restricting $\Gamma = G(\alpha)[\![\mathcal{P},\mathcal{N}]\!]$. It is convenient to allow all the columns corresponding to $\mathcal{U}$, and restrict them to $\mathcal{N}$ only at the very end.

If $\mu \in M_{\mathrm{Q}}$, let us for brevity write $\alpha(\mu,S)$ for $\alpha(\mathcal{M}_\mu,S)$. The matrix $G(\alpha)$ is defined block-wise by $G(\alpha)[\![\mathcal{P}^\mu,\mathcal{U}]\!] = G^\mu(\alpha) = \sum_{S\subseteq[n]} \alpha(\mu,S)\Psi^\mu_S$. Eq. (7) gives the following important relation:

$$G(\alpha) \xmapsto{\Delta_j} G(\partial_j\alpha). \tag{8}$$

4.4 Extended Matrices

Eq. (8) gives one connection between $G(\alpha)$ and the optimisation problem in (3). Here we give another one. For that, we define an extended version $\widetilde{G}(\alpha)$ of $G(\alpha)$.

Let $\widetilde{\mathcal{U}} = M_{\mathrm{Q}} \times \mathcal{U}$. We use $\widetilde{\mathcal{U}}^\mu$ to denote $\{\mu\}\otimes\mathcal{U}$. The $\widetilde{\mathcal{U}}\times\mathcal{U}$-matrix $\widetilde{G}(\alpha)$ is defined block-wise: $\widetilde{G}(\alpha)[\![\mathcal{U}^\mu,\mathcal{U}]\!] = \widetilde{G}^\mu(\alpha) = \sum_{S\subseteq[n]} \alpha(\mu,S)\Pi^{[3n]}_S$. Clearly, $G(\alpha) = q^{n/2}\widetilde{G}(\alpha)[\![\mathcal{P},\mathcal{U}]\!]$.

Using that $\bigl\{\Pi^{[3n]}_S\bigr\}$ is a decomposition of $\mathcal{H}^{3n}$ into orthogonal subspaces, we get

$$\widetilde{G}(\alpha)^*\widetilde{G}(\alpha) = \sum_{\mu\in M_{\mathrm{Q}}} \bigl(\widetilde{G}^\mu(\alpha)\bigr)^*\widetilde{G}^\mu(\alpha) = \sum_{S\subseteq[3n]} \Biggl[\sum_{\mu\in M_{\mathrm{Q}}} \alpha(\mu,S)^2\Biggr]\Pi^{[3n]}_S.$$

As $\|A\| = \sqrt{\|A^*A\|}$ for any matrix A, we obtain another important relation:

$$\|\widetilde{G}(\alpha)\| = \|\alpha\|. \tag{9}$$

Of course it also holds for $\partial_j\alpha$. If $\|\widetilde{G}(\alpha)\|$ and $\|G(\alpha)\|$ were close, then any feasible solution α would give an adversary matrix $\Gamma = G(\alpha)[\![\mathcal{P},\mathcal{N}]\!]$ with value $\|\alpha\|$. It is easy to lower bound $\|\Gamma\|$ in terms of $\|\alpha\|$, see Lemma 8 below, but, in general, $\|G(\partial_j\alpha)\|$ will be much larger than $\|\widetilde{G}(\partial_j\alpha)\|$. In particular, this is the case when α is the solution from Proposition 6. Our main challenge in the coming sections will be to find ways to reduce $\|G(\partial_j\alpha)\|$.

4.5 Reducing Extended Matrices

Here we will give a finer relation between $G(\alpha)$ and $\widetilde{G}(\alpha)$ than the trivial relation $G(\alpha) = q^{n/2}\widetilde{G}(\alpha)[\![\mathcal{P},\mathcal{U}]\!]$. For $\mathcal{M}_\mu \in \mathcal{C}_{\mathrm{Q}}$, we define

$$\Pi^\mu = \sum_{S\subseteq[3n], S\notin\mathcal{M}_\mu} \Pi_S^{[3n]} \qquad\text{and}\qquad \Psi^\mu = \sum_{S\subseteq[3n], S\notin\mathcal{M}_\mu} \Psi_S^\mu. \tag{10}$$

By condition (3c), we have $\widetilde{G}^\mu(\alpha) = \Pi^\mu\widetilde{G}^\mu(\alpha)$, and, thus, $G^\mu(\alpha) = \Psi^\mu\widetilde{G}^\mu(\alpha)$. If we define a linear operator $\Psi_{\mathrm{Q}}\colon \mathcal{H}^{\widetilde{\mathcal{U}}} \to \mathcal{H}^{\mathcal{P}}$ by $\Psi_{\mathrm{Q}} = \bigoplus_{\mu\in M_{\mathrm{Q}}} \Psi^\mu$, we get

$$G(\alpha) = \Psi_{\mathrm{Q}}\widetilde{G}(\alpha). \tag{11}$$

In the light of discussion after (9), it would help if we could upper bound the norm of Ψ_{Q}. Unfortunately, its norm is exponential. Indeed, we can write $\Psi^\mu = \bigotimes_{T\in\mu} \Psi^T_{\le 2}$,where $\Psi^T_{\le 2} = \sum_{R\subset T, R\ne T} \Psi^T_R$. We prove its basic properties in the next claim, where we also study the operator $\Psi^T_{\le 1} = \sum_{R\subset T, |R|\le 1} \Psi^T_R$.

▶ **Claim 7.** *We have the following estimates*

(a) $\left\|\Psi^T_{\le 2}\right\| = \sqrt{3}$,
(b) $\left\|\Psi^T_{\le 2}(\Pi_0 \otimes I_{\mathcal{H}} \otimes I_{\mathcal{H}})\right\| = \left\|\Psi^T_{\le 2}(I_{\mathcal{H}} \otimes \Pi_0 \otimes I_{\mathcal{H}})\right\| = \left\|\Psi^T_{\le 2}(I_{\mathcal{H}} \otimes I_{\mathcal{H}} \otimes \Pi_0)\right\| = 1$,
(c) $\left\|\Psi^T_{\le 1}\right\| = 1$,
(d) $(\Psi^T_\emptyset)^*\Psi^T_{\le 2} = \Pi^T_\emptyset$, *and* $\|\Psi^T_\emptyset\| = 1$.

The proof of the claim can be found in the full version of the paper. In the next sections, we will use points (b) and (c) of this claim to upper bound the norm of $G(\partial_j\alpha)$ using (11).

4.6 Restricting from $\mathcal{U}$ to $\mathcal{N}$

Finally, we give a general way of bounding the norm of $\Gamma = G(\alpha)[\![\mathcal{P},\mathcal{N}]\!]$ in terms of α. For our upcoming application in Section 6, we prove a slightly more general result. Note that the bound is related to the objective value (3a) of α.

▶ **Lemma 8.** *Let α be a solution to the dual learning graph of $\mathcal{C}_{\mathrm{Q}}$, and V is an arbitrary linear operator in $\mathbb{C}^{\mathcal{U}}$ satisfying $\Pi_\emptyset^{[3n]}V = \Pi_\emptyset^{[3n]}$. Then, $\left\|(G(\alpha)V)[\![\mathcal{P},\mathcal{N}]\!]\right\| \ge \sqrt{\frac{|\mathcal{N}|}{|\mathcal{U}|}\sum_{\mu\in M_{\mathrm{Q}}} \alpha(\mu,\emptyset)^2}$.*

An easy proof of this lemma can be found in the full version of the paper.

5 Lower Bound for the 3-Shift-Sum Problem

The goal is to prove a quantum query lower bound for the 3-shift-sum problem.

▶ **Theorem 9.** *Assume $q \ge 2n^3$. Then the quantum query complexity of the 3-shift-sum problem is $\Omega(n^{1/3})$.*

The main idea behind the lower bound is to use Claim 7(c). In order to do that, we perform a transition to a different certificate structure $\mathcal{C}'_{\mathrm{S}}$. For each $\mu \in M_{\mathrm{S}}$, there is a corresponding $\mathcal{M}'_\mu \in \mathcal{C}'_{\mathrm{S}}$ obtained as follows: a subset $S \subseteq [3n]$ is in $\mathcal{M}'_\mu$ if and only if there exists a triple $T \in \mu$ satisfying $|T \cap S| \ge 2$. Note that this is *not* the certificate structure for the 3-shift-sum problem. Rather it is the certificate structure of a problem one might call the 3-shift-equal problem. The input is a $3 \times n$-matrix. In the positive case, there exist circular shifts of rows such that the elements in each column become equal. In the negative case, any two elements from two different rows are different.

▶ **Proposition 10.** *The learning graph complexity of the certificate structure $\mathcal{C}'_{\mathrm{s}}$ is $\Omega(n^{1/3})$.*

Proof. The proof is similar to that of Proposition 12 from [6] for the hidden shift problem. We have $|\mathcal{C}'_{\mathrm{s}}| = n^2$. Define

$$\alpha(\mathcal{M}, S) = \frac{1}{n} \max\Bigl\{n^{1/3} - |S|, 0\Bigr\} \qquad \text{if } S \notin \mathcal{M}, \tag{12}$$

and as 0 otherwise. It is easy to see that the objective value (3a) is $n^{1/3}$, and that (3c) holds.

Fix S and j, and let us check (3b). If $|S| \geq n^{1/3}$, then the left-hand side of (3b) is zero, so assume $|S| \leq n^{1/3}$. There are n^2 choices of $\mathcal{M} \in \mathcal{C}'_{\mathrm{s}}$. If $S \cup \{j\} \notin \mathcal{M}$, then the value of $\alpha(\mathcal{M}, S)$ changes by $1/n$ as the size of S increases by 1. Also, there are at most $|S|n \leq n^{4/3}$ choices of $\mathcal{M}$ such that $S \notin \mathcal{M}$ but $S \cup \{j\} \in \mathcal{M}$. For each of them, the value of $\alpha(\mathcal{M}, S)$ changes by at most $n^{-2/3}$. Thus,

$$\sum_{\mathcal{M} \in \mathcal{C}} \bigl(\alpha(\mathcal{M}, S) - \alpha(\mathcal{M}, S \cup \{j\})\bigr)^2 \leq n^2 \cdot \frac{1}{n^2} + n^{4/3} \cdot n^{-4/3} = O(1). \qquad ◀$$

5.1 Regular Version

In this section we prove Theorem 9. Let α'_{s} be the feasible solution (12) for the $\mathcal{C}'_{\mathrm{s}}$ certificate structure. It is also a feasible solution for the $\mathcal{C}_{\mathrm{s}}$ certificate structure. As in Section 4, we define the adversary matrix by $\Gamma = G(\alpha'_{\mathrm{s}})[\![\mathcal{P}, \mathcal{N}]\!]$. By Lemma 8, we get $\|\Gamma\| = \Omega(n^{1/3})$ if we prove that $|\mathcal{N}| = \Omega(|\mathcal{U}|)$. But that is easy: for a uniformly random triple $(a, b, c) \in [q]^3$, the probability that $a + b + c$ is divisible by q is $1/q$. There are n^3 possible triples having one element in each of A, B, and C. Hence, by the union bound, a uniformly random input in $[q]^{3n}$ is negative with probability at least $1 - n^3/q \geq 1/2$. That is, $|\mathcal{N}| \geq q^{3n}/2$.

Now let us prove that $\|\Gamma \circ \Delta_j\| = O(1)$. By (8) and using that Γ is a submatrix of $G(\alpha'_{\mathrm{s}})$, it suffices to prove that $\|G(\partial_j \alpha'_{\mathrm{s}})\| = O(1)$.

Following (10), let us define an analogue of Ψ_{s} for our new certificate structure $\mathcal{C}'_{\mathrm{s}}$ by $\Psi'^{\mu} = \sum_{S \subseteq [3n], S \notin \mathcal{M}'_{\mu}} \Psi^{\mu}_S$ and $\Psi'_{\mathrm{s}} = \bigoplus_{\mu \in M_{\mathrm{s}}} \Psi'^{\mu}$. Similarly to (11), we get $G(\partial_j \alpha'_{\mathrm{s}}) = \Psi'_{\mathrm{s}} \widetilde{G}(\partial_j \alpha'_{\mathrm{s}})$. We have $\|\widetilde{G}(\partial_j \alpha'_{\mathrm{s}})\| = O(1)$ by (9) and Proposition 10. It suffices to prove that $\|\Psi'_{\mathrm{s}}\| = O(1)$. But it is easy to see that $\Psi'^{\mu} = \bigotimes_{T \in \mu} \Psi^{T}_{\leq 1}$, and, by Claim 7, $\|\Psi'^{\mu}\| = 1$, hence, $\|\Psi'_{\mathrm{s}}\| = 1$.

5.2 Property Testing Version

In this section, we prove a quantum lower bound for the property testing version of the 3-shift-sum problem. Unlike the original version of the 3-shift-sum problem, this problem makes sense even for $q = 2$, so, for concreteness, we will define it for Boolean alphabet, however, similar results also hold for larger alphabet sizes.

An input is a string in $\{0, 1\}^{3n}$. For a positive input x, there exists $\mu \in M_{\mathrm{s}}$ such that $x_a \oplus x_b \oplus x_c = 0$ for every triple $\{a, b, c\} \in \mu$. Here $\oplus$ stands for xor. The negative inputs are defined as being at relative Hamming distance at least ε to the set of positive inputs.

▶ **Theorem 11.** *For $\varepsilon \leq \frac{1}{7}$, the property testing version of the 3-shift-sum problem requires $\Omega(n^{1/3})$ quantum queries to solve.*

The construction is identical to that in Section 5.1. The proof of $\|\Gamma \circ \Delta_j\| = O(1)$ is identical. In this part of the proof only $\mathcal{P}$ and $\mathcal{U}$ are used, which are the same in the regular and the property testing versions of the problem, and the size of the alphabet is never used.

The only place where the size of the alphabet is used is in lower bounding $\|\Gamma\|$, where it is proven that $|\mathcal{N}| = \Omega(|\mathcal{U}|)$. If we prove this for this version of the problem, we will be done.

Recall that we treat x as an $3 \times n$-matrix. Fix the last two rows. The input x is negative if its first row is at relative Hamming distance at least $\frac{3}{7}$ from the xor of any of n^2 circular shifts of the last two rows. A simple application of the Chernoff and the union bounds shows that this is the case with probability $1 - o(1)$.

6 Lower Bound for the 3-Matching-Sum Problem

The goal of this section is to prove the following theorem:

▶ **Theorem 12.** *Assume $q \geq 2n^3$. Then the quantum query complexity of the 3-matching-sum problem is $\Omega(\sqrt{n})$.*

Let α_{M} be the feasible solution (4) to the dual learning graph of $\mathcal{C}_{\mathrm{M}}$ from Proposition 6. We will obtain an adversary matrix to the 3-matching-sum problem multiplying $G(\alpha_{\mathrm{M}})$ by a suitably chosen projector V. We define it using symmetries of the problem.

The group $\mathfrak{S}_n$ acts on the set $[q]^n$ in the natural way: $\pi \in \mathfrak{S}_n$ maps $x = (x_1, \dots, x_n)$ to $\pi x = (x_{\pi^{-1}(1)}, \dots, x_{\pi^{-1}(n)})$, and by linearity we extend this action to $\mathcal{H}^n$, the latter thus becoming an $\mathfrak{S}_n$-module.

Similarly, the group $\mathfrak{G} = \mathfrak{S}_A \times \mathfrak{S}_B \times \mathfrak{S}_C$ acts on $\mathcal{U}$: a group element $(\pi_A, \pi_B, \pi_C) \in \mathfrak{G}$ acts on $x = (x_A, x_B, x_C) \in \mathcal{U}$ by mapping it to $(\pi_A x_A, \pi_B x_B, \pi_C x_C)$. This action renders $\mathbb{C}^{\mathcal{U}}$ a $\mathfrak{G}$-module. Let $\mathfrak{G}$ act on $\mu \in M_{\mathrm{M}}$ by mapping each triple $(a_1, a_2, a_3) \in \mu$ to $(\pi_A(a_1), \pi_B(a_2), \pi_C(a_3))$. Together with the action on inputs of length $3n$, this gives us an action of $\mathfrak{G}$ on $\mathcal{P}$, hence, $\mathbb{C}^{\mathcal{P}}$ is also a $\mathfrak{G}$-module.

The 3-matching-sum problem is invariant under this action of $\mathfrak{G}$: positive inputs are mapped to positive inputs, and negative inputs are mapped to negative inputs. This means that $\mathbb{C}^{\mathcal{N}}$ is a $\mathfrak{G}$-submodule of $\mathbb{C}^{\mathcal{U}}$. It is easy to see that α_{M} is symmetric with respect to $\mathfrak{G}$, hence, $G(\alpha_{\mathrm{M}})$ is symmetric with respect to $\mathfrak{G}$. In other words, it commutes with any element of $\mathfrak{G}$, or, different still, it is a $\mathfrak{G}$-morphism.

Let T be a finite set. It is easy to see that Π_k^T is a $\mathfrak{S}_T$-submodule of Π^T. From [7], the module Π_k^T only contains irreps with at most k boxes below the first row. Denote by $\bar{\Pi}_k^T$ the projector onto the span of all irreps with *exactly* k boxes below the first row. In particular, $\bar{\Pi}_0^T = \Pi_0^T$.

In order to simplify statements of some results, in particular Lemma 15, let us assume there is a cutting point K such that

$$\alpha(\mu, S) = 0 \quad \text{whenever } |S| > K. \tag{13}$$

For α_{M}, we take $K = \lfloor \sqrt{n} \rfloor$. Define the projectors $V^T = \sum_{k=0}^{K} \bar{\Pi}_k^T$ and $V = V^A \otimes V^B \otimes V^C$. Note that $(\Pi_{k_A} \otimes \Pi_{k_B} \otimes \Pi_{k_C})V = (\bar{\Pi}_{k_A} \otimes \bar{\Pi}_{k_B} \otimes \bar{\Pi}_{k_C})$ for all k_A, k_B, k_C between 0 and K.

The adversary matrix Γ is obtained as $\Gamma = \big(G(\alpha_{\mathrm{M}})V\big)[\![\mathcal{P}, \mathcal{N}]\!]$.

We know from Section 5.1 that $|\mathcal{N}| = \Omega(|\mathcal{U}|)$. Also $\Pi_\emptyset^{[3n]} V = \bar{\Pi}_\emptyset^{[3n]} = \Pi_\emptyset^{[3n]}$. Hence, by Lemma 8 and Proposition 6, we have $\|\Gamma\| = \Omega(\sqrt{n})$.

It remains to prove that $\|\Delta_j \circ \Gamma\| = O(1)$ for any j, which we do in the remaining part of this section. Due to symmetry, $\|\Delta_j \circ \Gamma\|$ is the same for all j, so it suffices to consider $j = 1$. Note that $\Delta_1 \circ \Gamma$ is a $\mathfrak{G}'$-morphism, where $\mathfrak{G}' = \mathfrak{S}_{[2..n]} \otimes \mathfrak{S}_B \otimes \mathfrak{S}_C$.

We have to understand how Δ_1 acts on V, or, in particular, how it acts on $\bar{\Pi}_k^{[n]}$. For the usual projector, $\Pi_k^{[n]}$, we have the identity $\Pi_k^{[n]} = \Pi_0 \otimes \Pi_k^{[2..n]} + \Pi_1 \otimes \Pi_{k-1}^{[2..n]}$. Ref. [7] gives the following analogue of this identity for $\bar{\Pi}_k^{[n]}$, where one should think of $\Phi_k^{[n]}$ as an error term.

▶ **Lemma 13.** *Let $\Phi_k^{[n]} = \bar{\Pi}_k^{[n]} - \Pi_0 \otimes \bar{\Pi}_k^{[2..n]} - \Pi_1 \otimes \bar{\Pi}_{k-1}^{[2..n]}$. If $k < n/3$, then $\|\Phi_k^{[n]}\| = O(1/\sqrt{n})$.*

Define $\Phi^A = \sum_{k=1}^{K} \Phi^A_k$. It is easy to see that $\Pi^A_k \Phi^A_k \Pi^A_k = \Phi^A_k$, hence, $\|\Phi^A_k\| = O(1/\sqrt{n})$. Let $\Phi = \Phi^A \otimes V^B \otimes V^C$ and $V' = \Pi_0 \otimes V^{[2..n]} \otimes V^B \otimes V^C$.

From Lemma 13, we get the following variant of relation (8), whose proof is relatively straightforward and can be found in the full version of the paper.

▶ **Lemma 14.** *Let α be a solution to $\mathcal{C}_{\mathrm{M}}$ satisfying (13). Then, $G(\alpha)V \overset{\Delta_1}{\longmapsto} G(\partial_1\alpha)V' + G(\alpha)\Phi$.*

Applying Lemma 14 to $G(\alpha_{\mathrm{M}})$, we obtain $G(\alpha_{\mathrm{M}})V \overset{\Delta_1}{\longmapsto} G(\partial_1\alpha_{\mathrm{M}})V' + G(\alpha_{\mathrm{M}})\Phi$. Denote $W = I^A \otimes V^B \otimes V^C$, where I^A is the identity operator on $\mathcal{H}^A$. Note that $V' = WV'$ and $\Phi = W\Phi$. Also, $\|V'\| = 1$ and $\|\Phi\| = O(1/\sqrt{n})$. Thus, since Γ is a submatrix of $G(\alpha_{\mathrm{M}})V$, it suffices to prove that

$$\big\|G(\partial_1\alpha_{\mathrm{M}})W\big\| = O(1) \qquad \text{and} \qquad \big\|G(\alpha_{\mathrm{M}})W\big\| = O(\sqrt{n}), \tag{14}$$

which is reasonable since $\|\partial_1\alpha_{\mathrm{M}}\| = O(1)$ and $\|\alpha_{\mathrm{M}}\| = O(\sqrt{n})$. We prove this using the following somewhat technical estimate on the norm of $G(\alpha)W$, whose proof can be found in the full version of the paper.

▶ **Lemma 15.** *Let α be a solution to the dual learning graph of $\mathcal{C}_{\mathrm{M}}$ satisfying (13) and symmetric with respect to $\mathfrak{S}_B \times \mathfrak{S}_C$. Then $\|G(\alpha)W\| \le \max_{k_B,k_C\in[0..K]} \Lambda_{k_B,k_C}(\alpha)$, where $\Lambda_{k_B,k_C}(\alpha)$ is defined in the following way. Let $R_B = [n+1..n+2k_B]$ and $R_C = [2n+1..2n+2k_C]$. Let $L(\mu, R_B, R_C)$ be the number of triples in the matching μ that intersect both R_B and R_C. Then*

$$\Lambda_{k_B,k_C}(\alpha) = \sqrt{\sum_{\mu\in M_{\mathrm{M}}} 3^{L(\mu,R_B,R_C)} \max_{S\subseteq A\cup R_B\cup R_C} \alpha(\mu,S)^2}. \tag{15}$$

Now we show how to use Lemma 15 to prove the estimates in (14). The exponential term in (15) might be somewhat of a concern, but we prove that the fraction of matchings with large $L(\mu, R_B, R_C)$ decreases even faster.

▶ **Lemma 16.** *Assume $|R_B|, |R_C| \le 2\sqrt{n}$. Then, $\Pr_\mu\big[L(\mu,R_B,R_C) = \ell\big] \le 8^\ell/\ell!$, where the probability is over uniformly random $\mu \in M_{\mathrm{M}}$.*

Proof. Fix ℓ elements in each R_B and R_C. The probability that these elements are mutually matched by a random μ is $\binom{n}{\ell}^{-1}$. Hence, by the union bound, the probability that for a randomly chosen μ there are ℓ (or more) elements in R_B matched to elements in R_C is at most $\binom{|R_B|}{\ell}\binom{|R_C|}{\ell}\Big/\binom{n}{\ell} \le \frac{(2\sqrt{n})^{2\ell}}{\ell!(n/2)^\ell} \le \frac{8^\ell}{\ell!}$, where we have assumed that n is large enough so that $\ell \le 2\sqrt{n} < n/2$. ◀

▶ **Claim 17.** *We have $\big\|G(\alpha_{\mathrm{M}})W\big\| = O(\sqrt{n})$.*

Proof. We apply Lemma 15. By (4), we have $\alpha_{\mathrm{M}}(\mu,S)^2 \le n/|M_{\mathrm{M}}|$ for all μ and S. Hence, $\Lambda_{k_B,k_C}(\alpha_{\mathrm{M}}) \le \sqrt{n}\sqrt{\mathbb{E}_{\mu\in M_{\mathrm{M}}}\big[3^{L(\mu,R_B,R_C)}\big]}$. And, using Lemma 16:

$$\mathbb{E}_{\mu\in M_{\mathrm{M}}}\big[3^{L(\mu,R_B,R_C)}\big] \le \sum_{\ell=0}^{\infty} 3^\ell \cdot \frac{8^\ell}{\ell!} = \mathsf{e}^{24} = O(1), \tag{16}$$

which gives the required bound. ◀

▶ **Claim 18.** *We have $\big\|G(\partial_1\alpha_{\mathrm{M}})W\big\| = O(1)$.*

Proof. We apply Lemma 15. By the analysis in Proposition 6, we see that

$$\max_{S\subseteq A\cup R_B\cup R_C} \partial_1\alpha_{\mathrm{M}}(\mu,S)^2 \le \frac{1}{|M_{\mathrm{M}}|}\begin{cases} n, \text{ if 1 is matched by } \mu \text{ to elements in both } R_B \text{ and } R_C;\\ 1, \text{ otherwise.}\end{cases}$$

Let us call the event in the first case above $Z(\mu)$. Then,

$$\Lambda_{k_A,k_B}(\partial_1\alpha_{\mathrm{M}})^2 \le \mathbb{E}_{\mu\in M_{\mathrm{M}}}\Bigl[3^{L(\mu,R_B,R_C)}\Bigr] + \Pr_\mu[Z(\mu)]\cdot 3n\,\mathbb{E}_{\mu\in M_{\mathrm{M}}}\Bigl[3^{L(\mu,R_B,R_C)-1}\Bigm| Z(\mu)\Bigr].$$

The first term is $O(1)$ by (16). For the second term, it is easy to see that $\Pr_\mu[Z(\mu)] = |R_B||R_C|/n^2 = O(1/n)$, and the conditioned expectation the same as in (16), because we can remove the triple containing 1 from consideration thus reducing to the same problem with n, $|R_B|$, and $|R_C|$ smaller by 1. This gives the required bound of $O(1)$. ◀

7 Open Problems

The obvious open problem is to resolve the quantum query complexity of the 3-shift-sum problem. So far we only have an $\Omega(n^{1/3})$ lower bound and an $O(n^{1/2})$ upper bound, however, it is not clear how to improve on either of them.

For the 3-matching-sum problem, we have proven matching upper and lower bounds of $\Theta(\sqrt{n})$. An interesting problem is to generalise this to the k-matching-sum problem for arbitrary k. The main limitation seems to be the norm of the error term in Lemma 13.

Some other open problems can be formulated. What functions with randomised query complexity $\omega(\sqrt{n})$ could potentially have poly-logarithmic quantum query complexity? Or, can a relatively general result be proven that excludes some of such functions? For what other problems can the dual learning graph framework be useful?

References

1 Scott Aaronson and Andris Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. In *Proc. of 47th ACM STOC*, pages 307–316, 2015.

2 Scott Aaronson, Shalev Ben-David, and Robin Kothari. Separations in query complexity using cheat sheets. In *Proc. of 48th ACM STOC*, pages 863–876, 2016.

3 Andris Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002.

4 Aleksandrs Belovs. Span programs for functions with constant-sized 1-certificates. In *Proc. of 44th ACM STOC*, pages 77–84, 2012.

5 Aleksandrs Belovs. *Applications of the Adversary Method in Quantum Query Algorithms*. PhD thesis, University of Latvia, 2014.

6 Aleksandrs Belovs and Ansis Rosmanis. On the power of non-adaptive learning graphs. *Computational Complexity*, 23(2):323–354, 2014.

7 Aleksandrs Belovs and Ansis Rosmanis. Adversary lower bounds for the collision and the set equality problems. *Quantum Information & Computation*, 18(3&4):198–222, 2018.

8 Aleksandrs Belovs and Ansis Rosmanis. Quantum lower bounds for tripartite versions of the hidden shift and the set equality problems. Full version. Avaiable at arXiv:1712.10194, 2018.

9 Aleksandrs Belovs and Robert Špalek. Adversary lower bound for the k-sum problem. In *Proc. of 4th ACM ITCS*, pages 323–328, 2013.

10 Shalev Ben-David. A super-Grover separation between randomized and quantum query complexities, 2015. `arXiv:1506.08106`.

11 Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288:21–43, 2002.

12 Charles W. Curtis and Irving Reiner. *Representation theory of finite groups and associative algebras.* AMS, 1962.

13 Mark Ettinger, Peter Høyer, and Emanuel Knill. The quantum query complexity of the hidden subgroup problem is polynomial. *Information Processing Letters*, 91(1):43–48, 2004.

14 Aram W. Harrow, Cedric Yen-Yu Lin, and Ashley Montanaro. Sequential measurements, disturbance and property testing. In *Proc. of 28th ACM-SIAM SODA*, pages 1598–1611, 2017.

15 Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Proc. of 39th ACM STOC*, pages 526–535, 2007.

16 Gordon James and Adalbert Kerber. *The Representation Theory of the Symmetric Group*, volume 16 of *Encyclopedia of Mathematics and its Applications.* Addison-Wesley, 1981.

17 Greg Kuperberg. A subexponential-time quantum algorithm for the dihedral hidden subgroup problem. *SIAM Journal on Computing*, 35:170–188, 2005.

18 Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy. Quantum query complexity of state conversion. In *Proc. of 52nd IEEE FOCS*, pages 344–353, 2011.

19 Ashley Montanaro and Ronald de Wolf. A survey of quantum property testing. *Theory of Computing Graduate Surveys*, 7:1–81, 2016.

20 Ben W. Reichardt. Span programs and quantum query complexity: The general adversary bound is nearly tight for every Boolean function. In *Proc. of 50th IEEE FOCS*, pages 544–551, 2009.

21 Bruce E. Sagan. *The symmetric group: representations, combinatorial algorithms, and symmetric functions*, volume 203 of *Graduate Texts in Mathematics.* Springer, 2001.

22 Jean-Pierre Serre. *Linear Representations of Finite Groups*, volume 42 of *Graduate Texts in Mathematics.* Springer, 1977.

23 Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. In *Proc. of 43th IEEE FOCS*, pages 513–519, 2002.

24 Robert Špalek and Mario Szegedy. All quantum adversary methods are equivalent. *Theory of Computing*, 2:1–18, 2006.

25 Mark Zhandry. A note on the quantum collision and set equality problems. *Quantum Information & Computation*, 15(7&8):557–567, 2015.

26 Shengyu Zhang. On the power of Ambainis lower bounds. *Theoretical Computer Science*, 339(2):241–256, 2005.

The Quantum Complexity of Computing Schatten p-norms

Chris Cade
School of Mathematics, University of Bristol, UK
chris.cade@bristol.ac.uk

Ashley Montanaro
School of Mathematics, University of Bristol, UK
ashley.montanaro@bristol.ac.uk

Abstract

We consider the quantum complexity of computing Schatten p-norms and related quantities, and find that the problem of estimating these quantities is closely related to the one clean qubit model of computation. We show that the problem of approximating $\mathrm{Tr}(|A|^p)$ for a log-local n-qubit Hamiltonian A and $p = \mathrm{poly}(n)$, up to a suitable level of accuracy, is contained in DQC1; and that approximating this quantity up to a somewhat higher level of accuracy is DQC1-hard. In some cases the level of accuracy achieved by the quantum algorithm is substantially better than a natural classical algorithm for the problem. The same problem can be solved for arbitrary sparse matrices in BQP. One application of the algorithm is the approximate computation of the energy of a graph.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory, Theory of computation → Complexity classes

Keywords and phrases Schatten p-norm, quantum complexity theory, complexity theory, one clean qubit model

Digital Object Identifier 10.4230/LIPIcs.TQC.2018.4

Related Version A full version of the paper is available at `https://arxiv.org/abs/1706.09279`.

Funding CC was supported by the EPSRC. AM was supported by an EPSRC Early Career Fellowship (EP/L021005/1). No new data were created during this study.

1 Introduction

It is widely believed that quantum computers will be capable of solving certain computational problems more efficiently than any classical computer. However, the exact characterisation of the class of problems that allow for a quantum speedup is the subject of ongoing research. In complexity theory, this class is known as BQP [27] – the set of languages efficiently decidable by a uniform family of polynomial-size quantum circuits with bounded error. A useful way to understand and identify the types of problems that are efficiently solvable by a quantum computer, but unlikely to be efficiently solvable by a classical computer, is to find problems that are complete[1] for BQP; that is, problems that can be solved by a polynomial-time

[1] We note that what we are really referring to here are PromiseBQP-complete problems, since there are in fact no known BQP-complete problems. For a detailed discussion on this point see [14, 9].

13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2018).
Editor: Stacey Jeffery; Article No. 4; pp. 4:1–4:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

quantum computer, and that any other problem in BQP can be reduced to. Intuitively, these are the very hardest problems in BQP.

Several BQP-complete problems are known, including approximating the Jones polynomial [1], estimating quadratically signed weight enumerators (QSWEs)[18], and estimating diagonal entries of powers of sparse matrices [14]. The latter problem is particularly interesting, since it is a relatively natural problem that is not obviously 'quantum' in nature.

Knill and Laflamme [18] showed that a more constrained version of the QSWE problem is efficiently solvable in the one clean qubit model of computation – an apparently non-universal model of quantum computation that is weaker than full quantum computation, but that can seemingly solve some problems more efficiently than a classical computer [25]. Understanding the power of such intermediate classes of computation could shed light on the types of problems that are efficiently solvable by a fully universal quantum computer.

We consider the computational complexity of estimating Schatten p-norms of matrices. We find that for certain values of p and certain families of matrices, this problem is closely related to the one clean qubit model of computation: depending on the accuracy of the estimation, the problem can be efficiently solved in the one clean qubit model, or is hard for this model of computation. We also consider similar quantities related to the spectra of matrices, such as the so-called "energy" of graphs [19, 10], and provide quantum algorithms for estimating them that are more efficient than any known classical algorithms.

1.1 The One Clean Qubit Model of Computation

The one clean qubit model of quantum computation initially arose as an idealised model for computation on highly mixed initial states, such as those that appear in NMR implementations [17]. In this model, we are given a quantum state consisting of a single 'clean' qubit in the pure state $|0\rangle$, and n qubits in the maximally mixed state. This can be represented by the density matrix

$$\rho = |0\rangle\langle 0| \otimes \frac{I_{2^n}}{2^n}.$$

We then apply an arbitrary polynomial-sized quantum circuit to ρ, and measure the first qubit in the computational basis. Following [17], we will refer to the class of problems that can be solved in polynomial time using this model of computation as DQC1 – deterministic quantum computation with a single clean qubit.

The canonical problem that can be solved in this model is that of estimating the normalised trace of a $2^n \times 2^n$ unitary matrix U corresponding to a polynomial-size quantum circuit. This is achieved by applying a controlled version of U to ρ, where the clean qubit is used as the control qubit and is put into the state $(|0\rangle + |1\rangle)/\sqrt{2}$ using a Hadamard gate. More precisely, we apply the controlled-U operator to the state

$$\rho' = \frac{1}{2}(|0\rangle + |1\rangle)(\langle 0| + \langle 1|) \otimes \frac{I_{2^n}}{2^n}$$

and then apply a Hadamard gate to the first qubit, before measuring it. The probability of measuring zero is $\frac{1}{2} + \frac{1}{2}\frac{\mathrm{Re}(\mathrm{Tr}(U))}{2^n}$, which can be estimated up to accuracy ϵ by repeating the procedure $O(1/\epsilon^2)$ times. The imaginary part of the trace of U can be estimated similarly by starting with the first qubit in the state $\frac{1}{\sqrt{2}}(|0\rangle - i\,|1\rangle)$. This problem has been shown to be complete for the class DQC1 [26].

More generally, we might consider the DQCk model of computation. That is, deterministic quantum computation with k pure qubits. If $k = O(\log(n))$, then the DQCk model is equivalent to DQC1 [26]. This result is important for us since the quantum circuit that we apply to the initial state may require a number of ancilla qubits initialised to $|0\rangle$ in order to correctly perform its computation. For example, if the quantum circuit implementing the unitary U performs the phase estimation routine, then it will usually require an additional $O(\log n)$ clean qubits. In the context of estimating the trace of a unitary matrix, this result tells us that it is possible in DQC1 to compute the trace of a sub-matrix whose size is an inverse-polynomially large fraction of the size of the input matrix.

1.1.1 DQC1-complete Problems

Knill and Laflamme [17] showed that the problem of estimating a coefficient in the Pauli decomposition of a quantum circuit, up to polynomial accuracy, is complete for the class DQC1. In fact, the aforementioned problem of estimating the normalised trace of a quantum circuit is a special case of this problem [26]. Shor and Jordan [26] added to the relatively short list of DQC1-complete problems by showing that the problem of estimating the 'trace closure' of Jones polynomials is also complete for the class DQC1. Finally, Brandão [6] showed that two problems related to Hamiltonians were DQC1-complete: computing the partition function of a class of (quantum) Hamiltonians, and computing the sum of all eigenvalues of a Hamiltonian that fall between two given energy levels.

These quantities appear to be hard to compute classically, and therefore the one clean qubit model of computation seems to be more powerful than classical computation. However, it is unlikely that DQC1 contains all of BQP [25], and thus this model of computation appears to have a computational power that is somewhere in between BPP and BQP. Some evidence in this direction was recently provided by Morimae [21], who built on earlier work ([22]) to show that the output distribution of the one clean qubit model is difficult to sample from classically up to constant total variation distance error, provided that some complexity theoretic conjectures hold.

Here we show that the problem of computing Schatten p-norms of matrices is also closely related to the class DQC1.

1.2 Schatten p-norms and Graph Energy

Schatten p-norms are ubiquitous in Quantum Information theory (see for example [24, 3, 12]). This family of matrix norms includes the three most commonly used norms in quantum information theory: the Schatten 1-norm is more commonly called the trace norm, the Schatten 2-norm is also known as the Frobenius norm, and the Schatten ∞-norm is called the operator norm or spectral norm. Here we consider the normalised Schatten p-norm, defined as

$$\|A\|_p := \left(\frac{\sum_j |\lambda_j|^p}{2^n}\right)^{1/p}$$

for a $2^n \times 2^n$ Hermitian matrix A, where the sum ranges over the eigenvalues of A.

For instance, the Schatten 1-norm is the average of the absolute values of the eigenvalues of A,

$$\|A\|_1 = \frac{\mathrm{Tr}(|A|)}{2^n} = \frac{\sum_j |\lambda_j|}{2^n}.$$

If we consider the matrix A to be the adjacency matrix of a graph, this quantity is known as the 'Graph Energy', and has applications in chemistry, where it is related to the total electron energy of a class of organic molecules [19, 10]. More generally, quantities relating to the spectra of adjacency matrices are used throughout Graph Theory to reveal information about the graphs that they represent. In the present work, we consider some 'global' properties of the spectra of matrices and graphs – i.e. those of the form $\mathrm{Tr}(f(A))/2^n$, for some suitably chosen function f. The Schatten p-norms are examples of such quantities.

1.3 Our results

We study the complexity of approximately computing Schatten p-norms of sparse matrices and relate this to quantum computation. We consider Hermitian matrices of size $2^n \times 2^n$, where at most $d = \mathrm{poly}(n)$ entries in each row are non-zero, and call such matrices d-sparse. One fairly natural class of sparse matrices that can be expressed concretely is the class of 'log-local' Hamiltonians. That is, k-local n-qubit Hamiltonians, with $k = O(\log n)$ - i.e. Hermitian matrices that can be written as a sum $A = \sum_{j=1}^{m} A_j$, for some m, where each A_j is a Hermitian matrix that acts non-trivially on at most $k = O(\log n)$ qubits. We assume that we are given the individual matrices A_j directly, that $\|A_j\| = O(\mathrm{poly}(n))$ for all j, and that $m = \mathrm{poly}(n)$. Throughout this work, we use $\|A\|$ to denote the operator norm of A.

▶ **Theorem 1.** *Let A be a sparse Hermitian matrix on n qubits, and let $p, 1/\epsilon = O(\mathrm{poly}(n))$. Then the problem of estimating $\frac{\mathrm{Tr}(|A|^p)}{2^n}$ up to additive accuracy $\epsilon\|A\|^p$ is contained in BQP. If the matrix A is* log*-local, then this problem is also contained in DQC1.*

▶ **Theorem 2.** *Let A be a* log*-local Hermitian matrix on n qubits. Then the problem of estimating $\frac{\mathrm{Tr}(|A|^p)}{2^n}$ up to additive accuracy $\epsilon\left(\frac{\|A\|}{2}\right)^p$ for arbitrary $p, 1/\epsilon = O(\mathrm{poly}(n))$ is hard for the class DQC1.*

The BQP case of Theorem 1 follows from the result of Janzing and Wocjan [13], who gave a BQP algorithm for estimating diagonal entries of $f(A)$, for a sparse matrix A and an appropriate function f which can be taken to be $f(x) = |x|^p$.

We therefore see that the problem of computing Schatten p-norms for $p = O(\mathrm{poly}(n))$ is closely related to the one clean qubit model of computation. By contrast, for different values of p the problem is related to other classes of computation. For instance, $\|A\|_\infty$ is the operator norm of A, and the problem of computing it approximately is QMA-complete[2], even for 2-local Hamiltonians. To see this, suppose we have some upper bound $\Delta = O(\mathrm{poly}(n))$ on the largest eigenvalue of a 2-local n-qubit Hamiltonian A. Define the matrix $B := \Delta I_{2^n} - A$. Then the largest eigenvalue of B (in absolute value) corresponds to the smallest eigenvalue of A. Hence, if we can compute the smallest eigenvalue of A, then we can compute $\|B\|$, and vice versa. Since the problem of estimating the smallest eigenvalue of a k-local Hamiltonian is QMA-complete for $k \geq 2$ [15], this implies QMA-completeness of the problem of estimating the operator norm of a 2-local Hamiltonian.

Note that the required accuracies of the estimates in Theorems 1 and 2 differ by a factor of $1/2^p$. Unfortunately, we were unable to reconcile this difference, and therefore we did not find a variant of the problem that is *complete* for DQC1.

Theorem 1 gives us the following corollary:

▶ **Corollary 3.** *Let A be a* log*-local matrix corresponding to the adjacency matrix of a 2^n-vertex graph G, and let $p, 1/\epsilon = O(\mathrm{poly}(n))$. The normalised Graph Energy of G, $\mathrm{Tr}(|A|)/2^n$, can be estimated up to additive accuracy $\epsilon\|A\|$ in DQC1.*

[2] For a definition of the class QMA, see [27].

In proving Theorem 1, we also show that there exists a polynomial-time quantum algorithm (in DQC1) for estimating $\mathrm{Tr}(A^p)/2^n$ up to error $\epsilon\|A\|^p$ for $1/\epsilon, p \in O(\mathrm{poly}(n))$. This is useful in the context of graph theory because it allows for an estimation of the expected number of closed walks that start from each vertex in a 2^n-vertex graph. To obtain these algorithms, we prove a more general result:

▶ **Lemma 4.** *For a log-local Hamiltonian A, and any* log*-space polynomial-time computable function $f : I \to [-1, 1]$ (where I contains the spectrum of A) that is Lipschitz continuous with constant K (i.e. $|f(x)-f(y)| \leq K|x-y|$ for all $x, y \in I$), there exists a DQC1 algorithm to estimate $\mathrm{Tr}(f(A))/2^n = \sum_j f(\lambda_j)/2^n$ up to additive accuracy $\epsilon(K+1)$, where λ_j denote the eigenvalues of A, and $\epsilon = \Omega(1/\mathrm{poly}(n))$.*

Often, one is interested in calculating the properties of general sparse matrices. We note that it is easy to give a quantum algorithm for estimating the above quantities for sparse matrices by making use of a result of Janzing and Wocjan [14, 13], who give a BQP algorithm for estimating the diagonal entries of $f(A)$, for some function f that satisfies certain continuity constraints, but this comes at the expense of moving to the class BQP.

It is interesting to note that the results from [6] also make use of log-local Hamiltonians. In both these and our results, it is not clear how to drop the restriction of log-locality without losing the fact that the various problems are contained in DQC1.

1.3.1 Estimating $\|A\|_p$

Given a log-local n-qubit Hamiltonian A, the algorithm of section 3 outputs

$$\mathrm{Tr}(|A|^p)/2^n \pm \epsilon\|A\|^p.$$

By taking the pth root, we obtain an estimate of $\|A\|_p$ of the form

$$\left(\frac{\mathrm{Tr}(|A|^p)}{2^n} \pm \epsilon\|A\|^p\right)^{1/p} = \|A\|_p \left(1 + \frac{2^n\epsilon\|A\|^p}{\mathrm{Tr}(|A|^p)}\right)^{1/p}.$$

The error will be small when $\mathrm{Tr}(|A|^p)$ takes a value close to its maximum of $2^n\|A\|^p$. In the best case, the relative error is close to $(1+\epsilon)^{1/p}$. This suggests that in these 'good' cases, our algorithm can estimate $\|A\|_p$ up to a reasonable additive error in polynomial time. On the other hand, we can always bound

$$\frac{2^n\|A\|^p}{\mathrm{Tr}(|A|^p)} \leq \frac{2^n\|A\|^p}{2^n|\lambda_{\min}|^p} = \kappa(A)^p,$$

where $\lambda_{\min}$ is the minimal eigenvalue of A in absolute value, and $\kappa(A) = \|A\|\|A^{-1}\|$ is the condition number of A. In this case the relative error is at most $(1+\epsilon\kappa(A)^p)^{1/p}$.

Since we consider $p = \mathrm{poly}(n)$, the algorithm allows us to achieve relative error close to $\kappa(A)$ by taking $\epsilon = 1 - 1/\kappa(A)^p \approx 1$. Alternatively, we could achieve relative error $(1+\delta)$ for some $\delta = O(1/\mathrm{poly}(n))$ by setting $\epsilon = ((1+\delta)^p - 1)/\kappa(A)^p$. In this case, we sacrifice the run-time of the algorithm in order to improve the accuracy.

1.4 Relation to Previous Work

Our techniques are similar to those used in [14] and [11]. In particular, we use the same combination of Hamiltonian simulation and phase estimation for estimating and manipulating the eigenvalues of a Hermitian matrix. To show DQC1-hardness, we use techniques from the Hamiltonian complexity literature, and in particular ideas due to Kitaev et al. [16, 15].

By using a previous result of Janzing and Wocjan [14], we can obtain a BQP algorithm for estimating $\mathrm{Tr}(A^p)/2^n$ for general sparse matrices; however, it is not clear how to implement this algorithm in DQC1, since it uses $O(n)$ ancilla qubits for the Hamiltonian simulation step. In [14], the authors describe a polynomial-time quantum algorithm for estimating the diagonal entries of the matrix A^p up to error $\epsilon\|A\|^p$, for $\epsilon = O(1/\operatorname{poly}(n))$, and show that this problem is in fact BQP-complete for sparse symmetric matrices. The problem remains BQP-complete even for matrices with only $0, \pm 1$ entries.

1.5 Comparison with Classical Algorithms

We were not able to find any previous results in the literature regarding the complexity of estimating the above quantities for sparse matrices. In Appendix B, we give a classical algorithm for estimating the normalised trace of a sparse matrix raised to some power, and prove some bounds on the accuracy that this algorithm can achieve.

We find that for some types of matrix, the value $\mathrm{Tr}(A^p)/2^n$ can be estimated efficiently classically, and for others, a quantum algorithm appears to have some advantage over a classical one. In Appendix B, we prove the following:

▶ **Theorem 5.** *Given a $2^n \times 2^n$, d-sparse matrix A, there exists a classical algorithm to estimate $\mathrm{Tr}(A^p)/2^n$ up to accuracy $\epsilon d^p\|A\|_{\max}^p$ in time that is polynomial in n, p and $1/\epsilon$, where $\epsilon = O(1/\operatorname{poly}(n))$ and $\|A\|_{\max}$ is used to denote the maximum absolute size of an entry in A.*

Therefore, in the cases where $\|A\| \ll d\|A\|_{\max}$, we can get an advantage by making use of the algorithm of Theorem 1. We find that for certain classes of random graph (namely power-law graphs), the BQP algorithm for computing $\mathrm{Tr}(A^p)/2^n$ obtains a quadratic improvement in accuracy over the corresponding classical algorithm.

For log-local Hamiltonians and constant p, there exists an efficient *exact* classical algorithm for computing $\mathrm{Tr}(A^p)$. By using conventional matrix multiplication, it is possible to calculate the value of A^p by multiplying the individual matrices A_j. This can be seen from the expression for $\mathrm{Tr}(A^p)$:

$$\mathrm{Tr}(A^p) = \sum_{j_1, j_2, \ldots, j_p} \mathrm{Tr}(H_{j_1} H_{j_2} \cdots H_{j_p}),$$

where each index j_i ranges from 1 to m. Every H_{j_i} is k-local, and the complexity of multiplying a k-local matrix by an l-local matrix is $O(2^{3(k+l)})$ (using a naive algorithm), and results in a $(k+l)$-local matrix. If we perform the matrix multiplications from left to right, then, for each term in the sum, the first multiplication will take time $O(2^{3(2k)})$, the second $O(2^{3(3k)})$, and so on, until the final multiplication takes time $O(2^{3(pk)})$. There will be $p-1$ of these multiplications performed in total, with each taking at most $O(2^{3 \cdot pk})$ time, and hence the trace of $A_{j_1} A_{j_2} \cdots A_{j_p}$ can be calculated in $O(2^{3 \cdot pk})$ steps. There are m^p terms in the sum, and therefore the complexity of the entire computation is $O(m^p 2^{3 \cdot pk})$.

If we take $k = O(\log n)$ (i.e. take A to be a log-local Hamiltonian), the time complexity is $m^p n^{O(p)}$. For $p = O(1)$, this time complexity is polynomial and the output of this algorithm is better than the corresponding quantum algorithm, as it computes the desired value exactly.

Note that the problem of computing $\mathrm{Tr}(|A|^p)$ appears to be substantially harder classically for odd p, since it cannot be found by simply computing powers of a matrix, and instead requires more knowledge about the eigenvalues of A.

1.6 Organisation

We begin by providing a proof of Theorem 2 in Section 2. Section 3 describes a proof of Theorem 1, via an algorithm in the one clean qubit model that can estimate $\mathrm{Tr}(f(A))/2^n$ for a $2^n \times 2^n$ log-local matrix A and an appropriately continuous function f. Following this, in Section 4, we compare the performance of the quantum algorithm with its classical counterpart, which is described in Appendix B. Finally, appendix A contains a detailed analysis of the quantum algorithm used in Section 3.

2 Estimating $\mathrm{Tr}(|A|^p)/2^n$ is DQC1-hard

In this section, we show that the problem of estimating $\mathrm{Tr}(|A|^p)/2^n$ for a $2^n \times 2^n$ log-local Hamiltonian A up to a given accuracy is hard for the class DQC1. More precisely, we assume that we have access to an algorithm that can estimate $\mathrm{Tr}(|A|^p)/2^n$ up to accuracy $\epsilon \left(\frac{\|A\|}{2}\right)^p$, for $\epsilon = O(1/\operatorname{poly}(n))$ and $p = \operatorname{poly}(n)$, and show that this implies that we can solve any problem contained in DQC1.

To do this we show that, given as input a real unitary U (implemented by some polynomial-sized quantum circuit acting on n qubits), it is possible to construct a log-local Hamiltonian A such that $\mathrm{Tr}(|A|^p)/2^n = \mathrm{Tr}(U)/2^n$, for some $p = \operatorname{poly}(n)$. Furthermore, we show that an estimation accuracy of $\epsilon \left(\frac{\|A\|}{2}\right)^p$ is sufficient to provide an estimate of $\mathrm{Tr}(U)/2^n$ up to accuracy $1/\operatorname{poly}(n)$. This problem is complete for the class DQC1 [26], which implies that the problem of estimating $\mathrm{Tr}(|A|^p)/2^n$ up to the stated accuracy is DQC1-hard.

The construction is based on ideas from Hamiltonian complexity, and in particular Kitaev's clock construction for the local Hamiltonian problem [2]. We assume that we have a decomposition $U = U_{M-1}...U_1U_0$ of the circuit into M elementary gates. Since U is described by a polynomial-sized circuit, we have $M = \operatorname{poly}(n)$. We add $\lceil \log M \rceil$ additional qubits to act as a 'clock' register, which is used to control the application of the individual unitaries, and define a unitary operator

$$W := \sum_{l=0}^{M-1} |l+1\rangle\langle l| \otimes U_l,$$

where addition is taken to be modulo M. It is straightforward to check that

$$W^M = \sum_{l=0}^{M-1} |l\rangle\langle l| \otimes U_{l+M}...U_{l+2}U_{l+1}.$$

Then we have

$$\mathrm{Tr}(W^M) = \sum_{l=0}^{M-1} \mathrm{Tr}(|l\rangle\langle l|) \cdot \mathrm{Tr}(U_{l+M}...U_{l+2}U_{l+1}) = \sum_{l=0}^{M-1} \mathrm{Tr}(U_M...U_2U_1) = M\,\mathrm{Tr}(U),$$

where the second step follows from invariance of the trace under cyclic permutations.

W is log-local with $m = \operatorname{poly}(n)$ terms, since each clock operator $|l+1\rangle\langle l|$ acts on $\lceil \log M \rceil$ qubits, and each of the unitaries U_l act on at most $O(1)$ qubits each. Define the Hermitian matrix

$$A := \frac{1}{2}(W + W^\dagger).$$

Then the trace of A^M gives the real part of the trace of $\frac{2W^M}{2^M}$, since A^M equals $1/2^M(W^M + {W^\dagger}^M)$ plus some other powers of W and $W^\dagger$ that are traceless (since the clock unitaries can only have a trace if they return the clock state back to its initial state, which takes at least M applications of W), and therefore do not contribute to the trace of A^M.

W is a $2^{n+\lceil \log M \rceil} \times 2^{n+\lceil \log M \rceil}$ unitary matrix, and so we have $\|A\| \leq 1$. Thus, given the ability to estimate the normalised trace of A^p up to accuracy $\left(\frac{\|A\|}{2}\right)^p \epsilon$, we can estimate the value of $\mathrm{Re}[\mathrm{Tr}(U)]/2^n$ up to accuracy $1/\operatorname{poly}(n)$, which is the level of accuracy required for the class DQC1. To see this, we observe that, taking $p = M$ and assuming (without loss of generality) that M is a power of 2 (which also means that $|A|^M = A^M$),

$$\frac{\mathrm{Tr}(A^M)}{2^{n+\log M}} \pm \frac{\epsilon}{2^M} = \frac{2\,\mathrm{Re}(\mathrm{Tr}(W^M))}{2^M 2^{n+\log M}} \pm \frac{\epsilon}{2^M} = \frac{2M\,\mathrm{Re}(\mathrm{Tr}(U))}{M 2^M 2^n} \pm \frac{\epsilon}{2^M}.$$

Multiplying by 2^M, we obtain $\frac{\mathrm{Re}(\mathrm{Tr}(U))}{2^n} \pm \epsilon$, which is precisely the quantity that is DQC1-hard to compute. This is sufficient to show that the problem of estimating $\mathrm{Tr}(A^p)/2^n$ up to accuracy $\left(\frac{\|A\|}{2}\right)^p \epsilon$ for a log-local n-qubit Hamiltonian is hard for the class DQC1.

Note that we were not able to use standard techniques from the Hamiltonian complexity literature to make this construction work for k-local Hamiltonians with constant k [15, 16]. These techniques involve the introduction of a larger clock space that is then acted upon by k-local Hamiltonians. A term is then added to the Hamiltonian to 'penalise' invalid clock states and prevent them from contributing to the ground state energy. In our case, we care about the entire space on which the Hamiltonian acts and not just the subspace containing the valid clock states, and therefore the invalid clock states contribute to the trace of A^M in a non-trivial way.

3 Estimating $\mathrm{Tr}(|A|^p)/2^n$ is in DQC1

Here we show that the problem of estimating $\mathrm{Tr}(|A|^p)/2^n$ for a log-local Hamiltonian A, up to reasonable error, is in DQC1. In precise terms, we are given a k-local n-qubit Hamiltonian A, with $k = O(\log n)$; then the problem is to estimate $\mathrm{Tr}(|A|^p)/2^n$ up to error $\epsilon\|A\|^p$, for some integer $p = O(\operatorname{poly}(n))$ and accuracy $\epsilon = \Omega(1/\operatorname{poly}(n))$. Our approach is to construct a unitary U such that the normalised trace of U approximates the normalised trace of $|A|^p$. We show that this construction can be performed in polynomial time (that is, the unitary U takes $\operatorname{poly}(n, p, 1/\epsilon)$ time to implement). Using this approach, we can use the DQC1 model to compute the normalised trace of the matrix $|A|^p$, hence showing that this problem is contained in DQC1. We will use the following corollary of Lemma 4:

▶ **Corollary 6.** *For a log-local Hamiltonian A, and any* log*-space polynomial-time computable function $f : I \to \mathbb{R}$ (where I contains the spectrum of A) that is Lipschitz continuous with constant K' (i.e. $|f(x) - f(y)| \leq K'|x - y|$ for all $x, y \in I$), there exists a DQC1 algorithm to estimate $\mathrm{Tr}(f(A))/2^n = \sum_j f(\lambda_j)/2^n$ up to additive accuracy $\epsilon(K' + f_{\max})$, where λ_j denote the eigenvalues of A, $\epsilon = \Omega(1/\operatorname{poly}(n))$, and $f_{\max}$ is the supremum of $|f|$ on the interval I.*

The proof of Lemma 4 (and hence the above corollary) is split into roughly three parts. The first part, in Section 3.1, describes how the algorithm works, via a description of the unitary that is constructed from the input matrix. Following this, Section 3.2 discusses the accuracy and failure probability of the algorithm, and finally, Section 3.3 shows that the number of ancilla qubits required (and therefore the number of pure qubits needed) to implement the algorithm is at most $O(\log n)$.

3.1 Constructing the Unitary

We are given a log-local Hamiltonian A with eigenvectors $|\psi_j\rangle$ and corresponding eigenvalues λ_j. The basic idea is to construct a unitary U whose eigenvalues correspond to the eigenvalues of A in a useful way. In particular, we construct a polynomial-sized circuit whose associated unitary has eigenvalues λ'_j such that $\lambda'_j = f'(\lambda_j)$, for some function f' that depends on f.

The first step is to use Hamiltonian simulation to implement the unitary e^{iA}, which has eigenvalues $e^{i\lambda_j}$ for each eigenvector $|\psi_j\rangle$ of A. Section 3.4 discusses the time complexity of this part of the circuit. Then the circuit performs the following sequence of operations, which we will describe in terms of their effects on an eigenvector $|\psi_j\rangle$ of A and an arbitrary single qubit state of the form $\alpha|0\rangle + \beta|1\rangle$. We use $|0\dots0\rangle$ to denote an arbitrarily large ancilla register (with each qubit initialised to 0), and assume that both the phase estimation and Hamiltonian simulation parts of the circuit work perfectly.

1. Apply phase estimation on e^{iA} with the input $|\psi_j\rangle$, to obtain an estimate of the eigenvalue λ_j:
$$|\psi_j\rangle(\alpha|0\rangle + \beta|1\rangle)|0\dots0\rangle \mapsto |\psi_j\rangle(\alpha|0\rangle + \beta|1\rangle)|\lambda_j\rangle$$
2. Perform controlled phase rotations, where the phase depends on a function f of λ_j contained in the 3rd register (for example, $f(x) = x^p$):
$$|\psi_j\rangle(\alpha|0\rangle + \beta|1\rangle)|\lambda_j\rangle \mapsto |\psi_j\rangle(\alpha e^{i\arccos(f(\lambda_j))}|0\rangle + \beta e^{-i\arccos(f(\lambda_j))}|1\rangle)|\lambda_j\rangle$$
3. Undo the phase estimation to uncompute the value in the 3rd register:
$$\mapsto |\psi_j\rangle(\alpha e^{i\arccos(f(\lambda_j))}|0\rangle + \beta e^{-i\arccos(f(\lambda_j))}|1\rangle)|0\dots0\rangle$$

This gives us a unitary U that performs the mapping

$$|\psi_j\rangle(\alpha|0\rangle + \beta|1\rangle)|0\dots0\rangle \mapsto (\alpha e^{+i\arccos(f(\lambda_j))}|\psi_j\rangle|0\rangle + \beta e^{-i\arccos(f(\lambda_j))}|\psi_j\rangle|1\rangle)|0\dots0\rangle$$

for each eigenvector $|\psi_j\rangle$ of A. Therefore, for each eigenvalue λ_j of A, U has two corresponding eigenvalues $e^{\pm i\arccos(f(\lambda_j))}$.

By using the results described in Section 1.1, we can compute the trace of a sub-matrix of U in the one clean qubit model, provided that the number of ancilla qubits used is $O(\log n)$ (we check that this is indeed the case at the end of this section). In particular, we compute the trace of U', the sub-matrix of U obtained by fixing the ancilla qubits (except the one explicitly mentioned above) to $|0\rangle$. Then the trace of U' is

$$\begin{aligned}\mathrm{Tr}(U') &= \sum_j e^{\pm i\arccos(f(\lambda_j))} = \sum_j \cos(\pm\arccos(f(\lambda_j))) + i\sin(\pm\arccos(f(\lambda_j)))\\ &= \sum_j 2\cos(\arccos(f(\lambda_j))) + i\sin(\arccos(f(\lambda_j))) - i\sin(\arccos(f(\lambda_j)))\\ &= \sum_j 2f(\lambda_j).\end{aligned}$$

3.2 Error Analysis

Errors can arise in three places. Firstly, we will have some error in the Hamiltonian simulation part of the circuit. Secondly, there will be errors in estimating eigenvalues by using the phase estimation routine. And finally, there will be some error in the estimation of the normalised trace of U from using the one clean qubit model. In Appendix A, we consider the effect of all three sources of error, and show that we can estimate $\frac{1}{2^n}\sum_j f(\lambda_j)$ with additive error at most $\epsilon(K+1)$, for any $\epsilon = \Omega(1/\operatorname{poly}(n))$, where K is the Lipschitz constant of f. The analysis in this section is analogous to that of [14], since we use the same method for estimating an eigenvalue of A via simulation of e^{iA}, but uses different methods to bound the errors introduced by phase estimation and Hamiltonian simulation.

3.3 How many clean qubits are needed?

Here we consider how many clean qubits are required to implement the circuit described in Section 3.1 up to the desired accuracy. Any time the circuit uses ancilla qubits, these qubits will generally need to be initialised in the all-zeros state – that is, they must be under our control, and be 'clean'. As discussed in Section 1.1, we can use $O(\log n)$ clean qubits without changing the model of computation. In this section we argue that the implementation of the circuit described above requires no more than $O(\log n)$ ancilla qubits.

The two main parts of the circuit are the phase estimation routine, and Hamiltonian simulation. The rest of the circuit consists of more basic operations that require only a constant number of ancilla qubits (provided that the function f we choose is sufficiently easy to compute).

To achieve the accuracy stated in the previous section, we show in Appendix A that the phase estimation part of the circuit must be able to achieve an additive accuracy of $\frac{1}{2}\epsilon(K+1)$, for which it only requires $O(\log n)$ ancilla qubits. Hence, the number of clean qubits required to implement the phase estimation part of the circuit is $O(\log n)$.

In order to implement the simulation of the Hamiltonian A, we can use techniques based on the Lie-Trotter product formula [20]. This requires no more than a constant number of ancilla qubits, and, since we assume that we are given the Hamiltonian directly as a set of m individual Hamiltonians that each act on $O(\log n)$ qubits, there are no ancilla qubits required to 'load' the input into the system, which would be the case if we considered the case where the input Hamiltonian is specified by an oracle (it is precisely for this reason that we define the problem in terms of a log-local Hamiltonian rather than a sparse Hamiltonian). In our case, we can run a polynomial-time classical algorithm to compute the quantum circuit required to implement the unitary e^{iA}, given such a description of A. This is discussed more fully in the following section.

3.4 Simulating log-local Hamiltonians

We are required to implement the unitary e^{iA} for some log-local Hamiltonian A. We are limited to using at most $O(\log n)$ ancilla qubits, which rules out the more advanced Hamiltonian simulation techniques that are based on quantum walks (e.g. [5]). Instead, we use the 'vanilla' version of Hamiltonian simulation, which is based on the Lie-Trotter product formula [20].

We are given a log-local n-qubit Hamiltonian A, and wish to implement a unitary operator that approximates e^{iAt} for some value of t, up to a specified accuracy δ (in the operator norm). That is, we want to construct, in classical polynomial time, a quantum circuit that implements a unitary operator V such that

$$\|V - e^{iAt}\| \leq \delta.$$

It is straightforward to check that the standard techniques, which are usually presented for $O(1)$-local Hamiltonians, indeed work for log-local Hamiltonians, and allow us to simulate e^{iAt} up to accuracy δ in time

$$O(\operatorname{poly}(m, n, \tau, 1/\delta)),$$

where $\tau = t\|A\|$, using a circuit that can be computed by a polynomial-time classical algorithm[3]. The time complexity could be improved by the use of more complicated simulation techniques [4], but we do not consider this here.

In the circuit described in Section 3.1, we set $t = 1$, and require that $\delta = O(1/\operatorname{poly}(n))$. Thus, the time taken to implement the Hamiltonian simulation part of the circuit will be $O(\operatorname{poly}(n))$.

3.5 Proof that computing $\operatorname{Tr}(|A|^p)/2^n$ is in DQC1

The proof of Theorem 1, which states that the problem of estimating $\operatorname{Tr}(|A|^p)/2^n$ up to error $\epsilon\|A\|^p$ is in DQC1 for $p, 1/\epsilon = \operatorname{poly}(n)$, follows almost immediately from Lemma 4. The same proof also applies to the problem of estimating $\operatorname{Tr}(A^p)/2^n$. It is straightforward to check that, on the interval $[-b, b]$, both $f(x) = x^p$ and $f(x) = |x|^p$ are Lipschitz continuous with Lipschitz constant $K = pb^{p-1}$. Furthermore, $f_{\max} = b^p$ for both functions. In our case we can take $b = \|A\|$, since f is a function of the eigenvalues of A. Putting these values into Corollary 6, and replacing ϵ with $\frac{\epsilon}{p/\|A\|+1}$, we obtain an estimate of $\frac{\operatorname{Tr}(|A|^p)}{2^n}$ up to accuracy $\epsilon\|A\|^p$. Furthermore, this estimate can be obtained in DQC1 in time that is polynomial in n and inverse polynomial in ϵ.

4 Quantum vs. Classical

Here we compare the complexities of the (BQP) quantum and classical algorithms for computing $\operatorname{Tr}(A^p)$, for random $N \times N$ matrices. Recall that the quantum algorithm has an accuracy of $\epsilon\|A\|^p$, and that the classical algorithm has an accuracy of $\epsilon d^p \|A\|_{\max}^p$, where $p = \operatorname{polylog}(N)$.

In the event that $\|A\| \ll d$, the quantum algorithm achieves an improvement in accuracy over the classical algorithm. However, since the quantum algorithm requires the matrix A to be sparse, we must restrict our attention to only sparse matrices that have this property. Towards this end, we will begin by considering a general model for random graphs, and introduce some results that relate the degrees of the vertices of the graph to the eigenvalues of the adjacency matrix. Following this, we will consider how these results apply to sparse graphs.

4.1 Random Graphs

We consider a general model for unweighted random graphs (see e.g. [7]), in which each vertex v is associated with a weight w_v. Then a random graph G is constructed by assigning an edge independently to each pair of vertices (i, j) with probability $\frac{w_i w_j}{\sum_i w_i}$, such that the expected degree of vertex v is given by w_v. Denote by d the maximum expected degree, and by $\tilde{d}$ the value

$$\tilde{d} := \frac{\sum_{i=1}^N w_i^2}{\sum_{i=1}^N w_i}.$$

Then we have the following results from [7]:

[3] The details of this simulation can be found in the full version of this paper.

▶ **Theorem 7.** *If $\tilde{d} > \sqrt{d}\ln N$, then as $N \to \infty$ the largest eigenvalue of a random graph $G(w)$ is almost surely $(1+o(1))\tilde{d}$.*

▶ **Theorem 8.** *If $\sqrt{d} > \tilde{d}\ln^2 N$, then as $N \to \infty$ the largest eigenvalue of a random graph $G(w)$ is almost surely $(1+o(1))\sqrt{d}$*

Intuitively, $\|A\|$ is (asymptotically) the maximum of $\sqrt{d}$ and $\tilde{d}$ if the two values $\sqrt{d}$ and $\tilde{d}$ are far apart (i.e. by a power of $\log N$).

4.2 Restriction to Sparse Graphs

We are interested in sparse graphs – i.e. those in which the degree of every vertex is $O(\text{polylog}(N))$. If we use the random graph model above, and set $d = \Theta(\log^2 N)$, then if we allow all vertices to have an expected degree similar to d, then by Theorem 7, $\|A\| = (1+o(1))d$ almost surely, and the accuracies of both the classical and quantum algorithms are the same. Therefore, we are only going to see an advantage when we restrict the number of vertices that are allowed to have degree close to the maximum (which will be $O(\text{polylog}\, N)$ by necessity). In general, in an effort to make $\|A\| = o(d)$, we should only allow at most $O(\log N)$ vertices to have degree close to the maximum, and the others must have asymptotically smaller (e.g. constant) degree. A class of graphs that satisfies this requirement is the class of power law graphs.

A distribution on power-law graphs is given in [7] for which $d, \bar{d}$ and β are parameters that can be varied freely. In graphs of this type, the number of vertices with degree k is proportional to $k^{-\beta}$, and d is the maximum expected degree of a vertex in the graph, while $\bar{d}$ is the average degree. We have the following results, also from [7]:

1. For $\beta > 3$ and $d > \bar{d}^2 \log^{3+\epsilon} N$, the largest eigenvalue of the graph is almost surely $(1+o(1))\sqrt{d}$, for some $\epsilon = O(1)$, and where $\bar{d}$ denotes the average degree.
2. For $2.5 < \beta < 3$ and $d > \bar{d}^{(\beta-2)/(\beta-2.5)} \log^{3/(\beta-2.5)} N$, the largest eigenvalue of the graph is almost surely $(1+o(1))\sqrt{d}$.
3. For $2 < \beta < 2.5$ and $m > \log^{3/2.5-\beta} N$, the largest eigenvalue is almost surely $(1+o(1))\tilde{d}$.

Note that in all of the above, the bounds still apply when the graph is sparse (i.e. $d = O(\text{polylog}\, N)$). Hence, for power law graphs with exponent $\beta > 2.5$, we almost always get a quadratic improvement in accuracy over the classical algorithm. As the exponent decreases, so does the advantage gained by the quantum algorithm.

Some interesting subclasses of power law graphs have exponents between 2 and 2.5. For example, 'internet graphs' have exponents between 2.1 and 2.4, and the 'Hollywood' graph has exponent ≈ 2.3 [8]. In these cases, we might expect some quantum improvement over a classical approach, but not the full square root improvement.

References

1 D. Aharonov, V. Jones, and Z. Landau. A polynomial quantum algorithm for approximating the Jones polynomial. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 427–436. ACM, 2006. arXiv:quant-ph/0511096.

2 D. Aharonov and T. Naveh. Quantum NP-a survey. arXiv:quant-ph/0210077, 2002.

3 A. Ben-Aroya, O. Regev, and R. de Wolf. A hypercontractive inequality for matrix-valued functions with applications to quantum computing and LDCs. In *FOCS'08*, pages 477–486. IEEE, 2008. arXiv:0705.3806.

4 D. Berry, G. Ahokas, R. Cleve, and B. Sanders. Efficient quantum algorithms for simulating sparse Hamiltonians. *Communications in Mathematical Physics*, 270(2):359–371, 2007. arXiv:quant-ph/0508139.

5 D. W Berry, A. Childs, and R. Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 792–809. IEEE, 2015. arXiv:1312.1414.

6 F. Brandão. Entanglement theory and the quantum simulation of many-body physics. arXiv:0810.0026, 2008.

7 F. Chung, L. Lu, and V. Vu. Spectra of random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 100(11):6313–6318, 2003.

8 Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *ACM SIGCOMM computer communication review*, volume 29, pages 251–262. ACM, 1999.

9 O. Goldreich. On promise problems: A survey. In *Theoretical computer science*, pages 254–290. Springer, 2006.

10 I. Gutman. The energy of a graph: old and new results. In *Algebraic combinatorics and applications*, pages 196–211. Springer, 2001.

11 A. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103(15):150502, 2009. arXiv:0811.3171.

12 P. Hayden and A. Winter. Counterexamples to the maximal p-norm multiplicativity conjecture for all $p > 1$. *Communications in mathematical physics*, 284(1):263–280, 2008. arXiv:0807.4753.

13 D. Janzing and P. Wocjan. BQP-complete problems concerning mixing properties of classical random walks on sparse graphs. arXiv:quant-ph/0610235, 2006.

14 D. Janzing and P. Wocjan. A Simple PromiseBQP-complete Matrix Problem. *Theory of computing*, 3(1):61–79, 2007.

15 J. Kempe, A. Kitaev, and O. Regev. The complexity of the local Hamiltonian problem. *SIAM Journal on Computing*, 35(5):1070–1097, 2006.

16 A. Kitaev, A. Shen, and M. Vyalyi. *Classical and quantum computation*, volume 47. American Mathematical Society Providence, 2002.

17 E. Knill and R. Laflamme. Power of one bit of quantum information. *Physical Review Letters*, 81(25):5672, 1998. arXiv:quant-ph/9802037.

18 E. Knill and R. Laflamme. Quantum computing and quadratically signed weight enumerators. *Information Processing Letters*, 79(4):173–179, 2001. arXiv:quant-ph/9909094.

19 X. Li, Y. Shi, and I. Gutman. *Graph energy*. Springer Science & Business Media, 2012.

20 S. Lloyd. Universal quantum simulators. *Science*, 273(5278):1073, 1996. quant-ph/9703054.

21 T. Morimae. Hardness of classically sampling one clean qubit model with constant total variation distance error. arXiv:1704.03640, 2017.

22 T. Morimae, K. Fujii, and Joseph F. Fitzsimons. Hardness of classically simulating the one-clean-qubit model. *Physical review letters*, 112(13):130502, 2014. arXiv:1312.2496.

23 M. Nielsen and I. Chuang. *Quantum computation and quantum information*. Cambridge university press, 2010.

24 D. Perez-Garcia, M. Wolf, D. Petz, and M. Ruskai. Contractivity of positive and trace-preserving maps under L_p norms. *Journal of Mathematical Physics*, 47(8):083506, 2006. arXiv:math-ph/0601063.

25 D. Shepherd. Computation with unitaries and one pure qubit. arXiv:quant-ph/0608132, 2006.

26 P. Shor and S. Jordan. Estimating Jones polynomials is a complete problem for one clean qubit. *Quantum Information & Computation*, 8(8):681–714, 2008. arXiv:0707.2831.

27 J. Watrous. Quantum computational complexity. In *Encyclopedia of complexity and systems science*, pages 7174–7201. Springer, 2009. arXiv:0804.3401.

A Error Analysis of DQC1 Algorithm

A.1 Error from Hamiltonian Simulation

First we consider the error that arises in the circuit from Hamiltonian simulation. We assume that the Hamiltonian simulation step implements a unitary V that approximates e^{iA} in the sense that $||V - e^{iA}|| \leq \delta$, so that the eigenvalues of V and e^{iA} can differ by at most δ. For now, we will assume that the phase estimation routine works perfectly (i.e. introduces no error). Recall that this part of the circuit outputs an estimate for an eigenvalue of A in the range $[-\pi, \pi)$. Denote by λ_j and μ_j the output of the phase estimation routine when it is run using e^{iA} and V, respectively. We have

$$\left|e^{i\lambda_j} - e^{i\mu_j}\right| \leq \delta$$

by the bound on the error of the Hamiltonian simulation, where we can assume $|\mu_j - \lambda_j| \leq \pi$, by adding multiples of 2π to λ_j if necessary. The left hand side can be written as

$$\begin{aligned}
\left|1 - e^{i(\mu_j - \lambda_j)}\right| &= \left|e^{i\frac{(\mu_j-\lambda_j)}{2}}\left(e^{-i\frac{(\mu_j-\lambda_j)}{2}} - e^{i\frac{(\mu_j-\lambda_j)}{2}}\right)\right| \\
&= \left|e^{-i\frac{(\mu_j-\lambda_j)}{2}} - e^{i\frac{(\mu_j-\lambda_j)}{2}}\right| \\
&= 2\left|\sin\left(\frac{\mu_j-\lambda_j}{2}\right)\right| = 2\sin\left|\frac{\mu_j-\lambda_j}{2}\right| \qquad \text{(since } |\mu_j - \lambda_j| \leq 2\pi\text{)}.
\end{aligned}$$

We will use the inequality $(2/\pi)\theta \leq \sin\theta$ for $0 \leq \theta \leq \pi/2$. Therefore, we have that

$$(4/\pi)\frac{|\mu_j - \lambda_j|}{2} \leq 2\sin\left|\frac{\mu_j - \lambda_j}{2}\right| \leq \delta$$

and hence

$$|\mu_j - \lambda_j| \leq \pi\delta/2.$$

To see how this affects the accuracy of the algorithm, we consider the difference in the trace of U' when using V in place of e^{iA}.

$$\begin{aligned}
2\left|\sum_j f(\lambda_j) - \sum_j f(\mu_j)\right| &\leq 2\sum_j |f(\lambda_j) - f(\mu_j)| \\
&\leq 2\sum_j K\,|\lambda_j - \mu_j| \qquad \text{by the Lipschitz condition} \\
&\leq 2\sum_j K\pi\delta/2 = 2^n K\pi\delta.
\end{aligned}$$

Choosing the simulation accuracy to be $\delta \leq \epsilon/(2\pi)$, this contributes an error term of $2^n\epsilon K/2$. Thus, we have

$$2\left|\sum_j f(\lambda_j) - \sum_j f(\mu_j)\right| \leq 2^n\epsilon K/2. \tag{1}$$

A.2 Error from Phase Estimation

Here we consider the error that arises from using the phase estimation routine to estimate the eigenvalues μ_j of the unitary V from the previous sub-section. The phase estimation routine requires the addition of a ancilla qubits, which are used to control the application of powers of V on an n-qubit register. More precisely, the lth ancilla qubit is used to control the application of the unitary V^{2^l}, so that we apply the controlled gate

$$W_l := |0\rangle\langle 0|_l \otimes I + |1\rangle\langle 1|_l \otimes V^{2^l}$$

where the subscript l denotes that the projector acts on the lth ancilla/control qubit (and as the identity everywhere else). Let $W := W_1 W_2 \cdots W_a$. Then the phase estimation routine consists of applying Hadamard gates to all of the control qubits, applying W, and then applying the inverse quantum Fourier transform to the control qubits.

If we apply phase estimation to an eigenvector of V with eigenvalue $e^{i2\pi\theta}$, and measure the control register, we obtain some output $x \in \{0, 1, ..., 2^a - 1\}$ such that

$$\Pr(|\theta - x/2^a| < \eta) > 1 - \varphi \tag{2}$$

for $\varphi, \eta > 0$. To obtain this level of accuracy and probability of failure, it is sufficient [23] to set

$$a = \lceil \log(1/\eta) \rceil + \lceil \log(2 + (1/(2\varphi))) \rceil. \tag{3}$$

Let ϕ be defined as follows:

$$\phi(x) := \begin{cases} x2\pi/2^a & \text{if } x \leq 2^{a-1} \\ x2\pi/2^a - 2\pi & \text{otherwise} \end{cases}$$

Then let $\phi(x_j)$ be our estimate of the eigenvalue μ_j corresponding to the eigenvector $|\psi_j\rangle$, which, by the definition of ϕ above, lies in the interval $[-\pi, \pi)$. By Equation (2), if we apply phase estimation to an eigenvector $|\psi_j\rangle$ of V with corresponding eigenvalue $e^{i\mu_j}$, and measure, we have

$$\Pr(|\mu_j - \phi(x_j)| < 2\pi\eta) > 1 - \varphi \tag{4}$$

where the extra factor of 2π results from rescaling the value of x_j by 2π.

In our case, we do not measure the control register, and therefore we do not collapse the superposition over eigenvalues that phase estimation produces. Here we consider the effect that this has on the output of the algorithm, and simultaneously bound the error introduced by this part of the circuit. When phase estimation does not work perfectly, the algorithm consists of the following steps, implementing a unitary $\tilde{U}$:

1. Apply phase estimation on $V \approx e^{iA}$ with the input $|\psi_j\rangle$, to obtain a superposition over estimates $\phi(k)$ of the eigenvalue $\mu_j = 2\pi\theta_j$:

   $$|\psi_j\rangle (\alpha |0\rangle + \beta |1\rangle) |0 \dots 0\rangle \mapsto |\psi_j\rangle (\alpha |0\rangle + \beta |1\rangle) \sum_k \gamma_{k|j} |\phi(k)\rangle$$

   where $\gamma_{k|j} = \frac{1}{N} \sum_a e^{2\pi i a(\theta_j - k/N)}$.
2. Perform controlled phase rotations:

   $$|\psi_j\rangle (\alpha |0\rangle + \beta |1\rangle) \sum_k \gamma_{k|j} |\phi(k)\rangle$$
   $$\mapsto |\psi_j\rangle \sum_k \gamma_{k|j} (\alpha e^{i \arccos(f(\phi(k)))} |0\rangle + \beta e^{-i \arccos(f(\phi(k)))} |1\rangle) |\phi(k)\rangle .$$

3. Undo the phase estimation to uncompute the value in the 3rd register. To undo phase estimation we: a) apply the QFT to the register containing the $\phi(k)$'s, b) apply controlled powers of the unitary $V^\dagger \approx e^{-iA}$, and c) apply Hadamard gates to all qubits in the third register.

   a. Apply the QFT:

$$|\psi_j\rangle \frac{1}{\sqrt{N}} \sum_k \gamma_{k|j} (\alpha e^{i \arccos(f(\phi(k)))} |0\rangle + \beta e^{-i \arccos(f(\phi(k)))} |1\rangle) \sum_w e^{2\pi i w k/N} |w\rangle .$$

   b. Apply the controlled (on the third register) $V^\dagger$ gates:

$$|\psi_j\rangle \frac{1}{\sqrt{N}} \sum_k \gamma_{k|j} (\alpha e^{i \arccos(f(\phi(k)))} |0\rangle + \beta e^{-i \arccos(f(\phi(k)))} |1\rangle) \sum_w e^{2\pi i w k/N} e^{-2\pi i \theta_j w} |w\rangle .$$

   c. Apply Hadamard gates to each of the ancilla qubits:

$$\begin{aligned} &|\psi_j\rangle \frac{1}{N} \sum_k \gamma_{k|j} (\alpha e^{i \arccos(f(\phi(k)))} |0\rangle \\ &\qquad + \beta e^{-i \arccos(f(\phi((k)))} |1\rangle) \sum_w \sum_x e^{2\pi i w k/N} e^{-2\pi i \theta_j w} (-1)^{w \cdot x} |x\rangle . \end{aligned}$$

This means that $\tilde{U}$ performs the mapping

$$\begin{aligned} &|\psi_j\rangle (\alpha |0\rangle + \beta |1\rangle) |0 \ldots 0\rangle \\ &\mapsto |\psi_j\rangle \frac{1}{N} \sum_k \gamma_{k|j} (\alpha e^{i \arccos(f(\phi(k)))} |0\rangle \\ &\qquad + \beta e^{-i \arccos(f(\phi(k)))} |1\rangle) \sum_x \left(\sum_w e^{2\pi i w k/N} e^{-2\pi i \theta_j w} (-1)^{w \cdot x} \right) |x\rangle \end{aligned}$$

for each eigenvector $|\psi_j\rangle$ of V.

Let $\{|\psi_j\rangle |b\rangle |\phi\rangle,\ b \in \{0,1\}\}$ be a basis for the tensor product of the three registers. By design, the only states that contribute to the trace of U' are those of the form $|\psi_j\rangle |b\rangle |0 \ldots 0\rangle$. Hence, we can consider the trace of $\tilde{U'}$ – the submatrix of $\tilde{U}$ in which the third register is in the state $|0 \ldots 0\rangle$) – which is given by:

$$\begin{aligned} \mathrm{Tr}(\tilde{U'}) &= \sum_j \langle\psi_j| \langle 0| \left(|\psi_j\rangle \frac{1}{N} \sum_k \gamma_{k|j} \sum_w e^{2\pi i w k/N} e^{-2\pi i \theta_j w} e^{i \arccos(f(\phi(k)))} |0\rangle \right) \\ &+ \sum_j \langle\psi_j| \langle 1| \left(|\psi_j\rangle \frac{1}{N} \sum_k \gamma_{k|j} \sum_w e^{2\pi i w k/N} e^{-2\pi i \theta_j w} e^{-i \arccos(f(\phi(k)))} |1\rangle \right) \\ &= \frac{1}{N} \sum_{j,k} \gamma_{k|j} \sum_w e^{2\pi i w k/N} e^{-2\pi i \theta_j w} \left(e^{i \arccos(f(\phi(k)))} + e^{-i \arccos(f(\phi(k)))} \right) \\ &= \frac{1}{N} \sum_{j,k} \gamma_{k|j} 2 f(\phi(k)) \sum_w e^{2\pi i w (k/N - \theta_j)} \\ &= 2 \sum_{j,k} \left|\gamma_{k|j}\right|^2 f(\phi(k)) \\ &= 2 \sum_k f(\phi(k)) \sum_j \left|\gamma_{k|j}\right|^2 . \end{aligned}$$

Suppose that $\theta_j = z_j/N$ for some z_j – that is, each θ_j can be represented precisely by an n-bit rational number z_j/N. Then $\gamma_{k|j} = \delta_{k,z_j}$, and so $\mathrm{Tr}(\tilde{U'}) = 2\sum_j f(\mu_j)$. This corresponds to the case in which phase estimation works perfectly; in reality, we will not be able to express all eigenvalues precisely as n-bit rational numbers. Instead, suppose that $\theta_j = \tilde{z}_j/N + \delta_j$, where $\tilde{z}_j/N$ is the closest n-bit approximation of θ_j, and so $0 \le \delta_j \le 1/(2N)$. The difference between the trace in the two cases is given by

$$\begin{aligned} 2\left|\sum_j f(\mu_j) - \sum_j\sum_k \left|\gamma_{k|j}\right|^2 f(\phi(k))\right| &\le 2\sum_j \left|f(\mu_j) - \sum_k \left|\gamma_{k|j}\right|^2 f(\phi(k))\right| \\ &= 2\sum_j \left|\sum_k \left|\gamma_{k|j}\right|^2 (f(\mu_j) - f(\phi(k)))\right| \\ &\le 2\sum_j\sum_k \left|\gamma_{k|j}\right|^2 \left|f(\mu_j) - f(\phi(k))\right|, \end{aligned}$$

where the second step follows because $\sum_k \left|\gamma_{k|j}\right|^2 = 1$. The coefficient $\left|\gamma_{k|j}\right|^2$ is precisely the probability of measuring $\phi(k)$ on the ancilla register when the true eigenvalue is μ_j. By the promises of phase estimation (Equation (4)), with probability $\le \varphi$ we have $|\mu_j - \phi(k)| > 2\pi\eta$, in which case $|f(\mu_j) - f(\phi(k))| \le 2f_{\max}$; and with probability $\ge 1-\varphi$ we have $|\mu_j - \phi(k)| \le 2\pi\eta$, in which case $|f(\mu_j) - f(\phi(k))| \le 2\pi K\eta$. Hence, the error from this part of the circuit is bounded above by

$$2\left|\sum_j f(\mu_j) - \sum_j\sum_k \left|\gamma_{k|j}\right|^2 f(\phi(k))\right| \le 4\sum_j(\pi K\eta + \varphi f_{\max}) = 2^{n+2}(\pi K\eta + \varphi f_{\max}).$$

Choosing $\eta < \epsilon/(8\pi)$ and $\varphi < \epsilon/8$, and assuming that $f_{\max} \le 1$ (as stated earlier), this becomes

$$2\left|\sum_j f(\mu_j) - \sum_j\sum_k \left|\gamma_{k|j}\right|^2 f(\phi(k))\right| \le 2^n\frac{1}{2}\epsilon(K+1). \tag{5}$$

Now we consider how this contributes to the overall error. As before, let λ_j denote the eigenvalues of e^{iA}. Then the error of the algorithm, taking into account both the Hamiltonian simulation and phase estimation steps, is

$$\begin{aligned} &2\left|\sum_j f(\lambda_j) - \sum_j\sum_k \left|\gamma_{k|j}\right|^2 f(k)\right| \\ &\le 2\left|\sum_j f(\lambda_j) - \sum_j f(\mu_j)\right| + 2\left|\sum_j f(\mu_j) - \sum_j\sum_k \left|\gamma_{k|j}\right|^2 f(k)\right| \end{aligned}$$

where the first term on the right corresponds to the error from the Hamiltonian simulation part of the circuit (i.e. the difference between the trace of the circuit when using V instead of e^{iA}), and the second term corresponds to the error introduced by phase estimation. A bound on the first term is given by Equation (1), and the second term is bounded via Equation (5). Therefore, the difference in the trace of U' in the case where Hamiltonian simulation and phase estimation both work perfectly, and when they do not, is bounded by

$$2\left|\sum_j f(\lambda_j) - \sum_j\sum_k \left|\gamma_{k|j}\right|^2 f(k)\right| \le 2^n\epsilon(K+1/2). \tag{6}$$

A.3 Error from estimating $\mathrm{Tr}(U')/2^n$ in the DQC1 model

The one clean qubit model can estimate the normalised trace of a $2^n \times 2^n$ sub-matrix of a $2^{n+O(\log n)} \times 2^{n+O(\log n)}$ unitary matrix (implemented by a poly(n)-sized circuit) up to accuracy $\zeta = \Omega(1/\operatorname{poly}(n))$. Therefore, using the one clean qubit model to estimate the trace of U' will introduce an extra error term ζ. Let $\widetilde{\mathrm{Tr}(U')}/2^n$ be the output from the one clean qubit algorithm. Then choosing $\zeta = \epsilon/2$, and using the bound from Equation (6), we have

$$\left| \frac{2}{2^n} \sum_j f(\lambda_j) - \widetilde{\mathrm{Tr}(U')}/2^n \right| \leq \epsilon(K+1). \tag{7}$$

Hence, we can estimate $\frac{1}{2^n}\sum_j f(\lambda_j)$ in polynomial time with accuracy $\epsilon(K+1)$ for any $\epsilon = \Omega(1/\operatorname{poly}(n))$.

B Classical Algorithms

Here we describe a classical algorithm for diagonal entry estimation, which is the problem of estimating an entry on the diagonal of the matrix A^p, up to reasonable error. Given the ability to estimate the diagonal entries of a matrix, we are able to estimate the normalised trace of that matrix.

We first present an algorithm for the special case where A contains only $0, 1$ entries, and then in Appendix B.1 discuss how it can be extended to work for arbitrary real matrices. In the first case, the matrix A defines an unweighted, undirected graph with N vertices. The value of $(A^p)_{jj}$ is equivalent to the number of distinct walks (i.e. traversals around the graph that may traverse any edge more than once, or not at all) of length p starting and ending at vertex j.

We begin by observing that $(A^p)_{jj}$ can be re-interpreted as the total number of walks of length p leaving j multiplied by the probability that such a walk ends at vertex j. We can obtain an estimate of the latter by performing a number of random walks of length p, beginning at vertex j, and counting how many of them return to vertex j on the final step.

In order to obtain an estimate of the total number of walks of length p leaving a given vertex, we can do the following: given an upper bound d on the degree of the graph, we generate a number of sequences of p integers chosen independently and uniformly at random from the range $[0, d]$. Any given sequence provides a 'candidate' walk of length p on the graph, which may or may not be realisable on the graph defined by A. Given a candidate walk of the form $(n_0, n_1, ..., n_p)$, we test whether or not it is realisable by starting a walk at vertex j, and then moving to the n_0th neighbour of j. We then move to the n_1th neighbour of that vertex, and so on. If, at any step i of the walk, a vertex does not have a neighbour n_i, we terminate the process and conclude that the candidate is not realisable.

If we tried all d^p possible candidate walks from vertex j, then by counting the number of successes we would know the exact value of the number of walks of length p that leave vertex j; however, this would require $O(d^p)$ walks to be performed. If instead we sample from the set of all possible walks by generating a number of sequences at random, we can obtain a close estimate of the true number of walks. Below is the full algorithm for diagonal entry estimation. We assume that we are given some bound d on the degree of the graph, and that we wish to estimate $(A^p)_{jj}$.

1. Estimate the total number of walks of length p leaving vertex j:
   a. Define variables X_i for $i \in [k]$, for some value of k to be determined later.

   b. For $i = 1$ to k:
      i. Generate a sequence $(n_0, n_1, ..., n_p)$, where each $n_l \in [d]$.
      ii. Attempt to follow the walk defined by the sequence.
      iii. If the walk was successful, set $X_i = 1$, otherwise set $X_i = 0$.
   c. Then $\overline{X} = \frac{d^p}{k}(X_1 + X_2 + ... + X_k)$ provides an estimate of the total number of walks of length p leaving vertex j.
2. Estimate the probability that a given walk returns to vertex j:
   a. Define variables Y_i for $i \in [k']$, for some value of k' to be determined later.
   b. For $i = 1$ to k':
      i. Perform a random walk of length p starting at vertex j.
      ii. If the walk returns to vertex j (as its final step), then set $Y_i = 1$, otherwise set it to 0.
   c. Then $\overline{Y} = \frac{1}{k'}(Y_1 + Y_2 + ... + Y_{k'})$ gives an estimate of the probability that a given walk returns to vertex j.
3. Multiplying the two values together gives us our desired estimate: $(\tilde{A^p})_{jj} = \overline{X} \cdot \overline{Y}$.

To analyse the accuracy of this estimation, we will look at the errors in the two estimates $\overline{X}$ and $\overline{Y}$.

In both steps, we are essentially aiming to estimate the success probability of some Bernoulli process: in step 1 we aim to estimate the probability with which a randomly generated sequence of 'moves' succeeds in generating a valid walk around the graph, and in step 2 we are estimating the probability that a given (valid) walk of length p succeeds in returning to its starting vertex on the final step of the walk. In both cases, we can estimate the appropriate probability up any desired accuracy ϵ by choosing the number of samples (k in step 1, and k' in step 2) to be inverse polynomial in ϵ.

We use Hoeffding's inequality to bound the accuracy of both estimates. For step 1, we absorb the factor of d^p into the random variables X_i, and use the general form of the bound:

$$\Pr\left[|\overline{X} - \mathbb{E}[\overline{X}]| \geq \epsilon d^p\right] \leq 2e^{-2\epsilon^2 k}.$$

And for step 2, we have

$$\Pr\left[|\overline{Y} - \mathbb{E}[\overline{Y}]| \geq \epsilon'\right] \leq 2e^{-2\epsilon'^2 k'}.$$

Therefore, by choosing $k = \text{poly}(1/\epsilon)$ and $k' = \text{poly}(1/\epsilon')$, we can estimate $(A^p)_{jj}$ up to additive error that is at most $d^p(\epsilon + \epsilon' + \epsilon\epsilon) = d^p\delta$ for $\delta = 1/\text{poly}(n)$, with a constant probability of failure.

B.1 Extension to real matrices

In this section we extend the diagonal entry estimation algorithm of the previous section to work for arbitrary real matrices. Recall that this algorithm works for matrices with $0, 1$ entries by interpreting the input matrix as the adjacency matrix for an unweighted, undirected graph. More general (symmetric) matrices may be viewed as undirected graphs with weighted edges, and a similar interpretation of the value of $(A^p)_{jj}$ holds in these cases.

In the case of general matrices, the value of $(A^p)_{jj}$ depends not only on the number of closed walks (i.e. those that return to their start vertex) leaving vertex j, but also on the 'weight' of those walks. Let $\mathcal{C}_p^j$ be the set of all *closed* walks of length p leaving vertex j, and $E(\omega)$ be the set of edges that make up a given walk ω.

Then we have

$$(A^p)_{jj} = \sum_{c \in \mathcal{C}_p^j} \prod_{e \in E(c)} \text{weight}(e).$$

In order to estimate this quantity, we proceed similarly to the above case.

Let us denote the set of all (not necessarily closed) walks of length p originating at vertex j by $\mathcal{W}_p^j$. Then we can re-write the above quantity as

$$(A^p)_{jj} = W_p \, \mathbb{E}_{\omega \in \mathcal{W}_p^j} \left[\prod_{e \in E(\omega)} \text{weight}(e) \right],$$

by using the same reasoning as before – i.e. that the jth diagonal entry of A^p is given by the total number of walks of length p leaving vertex j multiplied by the expected 'weight' of each walk, where we assign a weight of 0 if the walk does not return to vertex j.

We can estimate the expectation on the right by sampling from the set of closed walks of length p originating at vertex j. This can be done by performing random walks of length p starting at vertex j, and recording the total weights of those walks that return to vertex j. This is easily incorporated into the existing algorithm: we set the variable Y_i to 0 if the ith walk does not return to vertex j, and otherwise we set it to the total weight of the walk (i.e. the product over the weights of the edges of the walk). W_p can be estimated as before, up to error ϵd^p. The error in estimating the expectation value depends upon the largest total weight of a closed walk in the graph. This is smaller than or equal to $\|A\|_{\max}^p$, where $\|A\|_{\max}$ is the maximum absolute size of an entry in A. A bound on the accuracy of estimating the expectation value is once again given by Hoeffding's inequality:

$$\Pr[|\overline{Y} - \mathbb{E}[\overline{Y}]| \geq \epsilon' \|A\|_{\max}^p] \leq 2e^{-2\epsilon'^2 k'}.$$

Multiplying the two estimates together, we obtain an estimate of $(A^p)_{jj}$ up to accuracy $\delta d^p \|A\|_{\max}^p$ with constant probability.

B.2 Estimating $\mathrm{Tr}(A^p)/N$ Classically

We can use the classical version of diagonal entry estimation to estimate the normalised trace of a matrix. More precisely, we obtain the empirical mean of $(A^p)_{jj}$ over a sample of values of j chosen uniformly at random. To see that the mean value of $(A^p)_{jj}$ for $j \in [N]$ does indeed give us the desired value, we observe that

$$\mathbb{E}_j[(A^p)_{jj}] = \frac{1}{N} \sum_{j=0}^{N-1} (A^p)_{jj} = \frac{\mathrm{Tr}(A^p)}{N}.$$

Let the output of the diagonal entry estimation algorithm be $\widetilde{(A^p)}_{jj}$ (which is an estimate of $(A^p)_{jj}$ up to additive error $\delta d^p \|A\|_{\max}^p$). Then let $\overline{\widetilde{(A^p)}_{jj}}$ be the mean value of the variable $\widetilde{(A^p)}_{jj}$ after sampling k times for randomly chosen values of j. The value of $\widetilde{(A^p)}_{jj}$ is bounded in the interval $[-(d\|A\|_{\max})^p, (d\|A\|_{\max})^p]$. Then by Hoeffding's inequality:

$$\Pr\left[\left|\overline{\widetilde{(A^p)}_{jj}} - \mathbb{E}[\widetilde{(A^p)}_{jj}]\right| \geq \delta d^p \|A\|_{\max}^p\right] \leq 2\exp\left(\frac{-\delta^2}{2}k\right).$$

Thus, choosing k to be inverse polynomial in δ allows us to obtain an estimate of $\mathbb{E}[(A^p)_{jj}] = \mathrm{Tr}(A^p)/N$ up to error $\delta d^p \|A\|_{\max}^p$. Note that for $0, 1$ and $-1, 0, +1$ matrices, $\|A\|_{\max} = 1$ and therefore the accuracy of the estimation in this case is just δd^p.

Subset Sum Quantumly in 1.17^n

Alexander Helm[1]
Horst Görtz Institute for IT-Security
Ruhr University Bochum, Germany
alexander.helm@rub.de

Alexander May
Horst Görtz Institute for IT-Security
Ruhr University Bochum, Germany
alex.may@rub.de

Abstract

We study the quantum complexity of solving the subset sum problem, where the elements $a_1, \dots, a_n$ are randomly chosen from $\mathbb{Z}_{2^{\ell(n)}}$ and $t = \sum_i a_i \in \mathbb{Z}_{2^{\ell(n)}}$ is a sum of $n/2$ elements. In 2013, Bernstein, Jeffery, Lange and Meurer constructed a quantum subset sum algorithm with heuristic time complexity $2^{0.241n}$, by enhancing the classical subset sum algorithm of Howgrave-Graham and Joux with a quantum random walk technique. We improve on this by defining a quantum random walk for the classical subset sum algorithm of Becker, Coron and Joux. The new algorithm only needs heuristic running time and memory $2^{0.226n}$, for almost all random subset sum instances.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases Subset sum, Quantum walk, Representation technique

Digital Object Identifier 10.4230/LIPIcs.TQC.2018.5

Acknowledgements We are grateful to Stacey Jeffery for many insightful discussions.

1 Introduction

The subset sum (aka knapsack) problem is one of the most famous NP-hard problems. Due to its simpleness, it inspired many cryptographers to build cryptographic systems based on its hardness. In the 80s, many attempts for building secure subset sum based schemes failed [20], often because these schemes were built on subset sum instances $(a_1, \dots, a_n, t)$ that turned out to be solvable efficiently.

Let $a_1, \dots, a_n$ be randomly chosen from $\mathbb{Z}_{2^{\ell(n)}}$, $I \subset \{1, \dots, n\}$ and $t \equiv \sum_{i \in I} a_i \bmod 2^{\ell(n)}$. The quotient $n/\ell(n)$ is usually called the *density* of a subset sum instance. In the *low density case* where $\ell(n) \gg n$, I is with high probability (over the randomness of the instance) a unique solution of the subset sum problem. Since unique solutions are often desirable for cryptographic constructions, most initial construction used low-density subset sums. However, Brickell [8] and Lagarias, Odlyzko [17] showed that low-density subset sums with $\ell(n) > 1.55n$ can be transformed into a lattice shortest vector problem that can be solved in practice in small dimension. This bound was later improved by Coster et al. [9] and Joux, Stern [15] to $\ell(n) > 1.06n$. Notice that this transformation does not rule out the hardness of subset sum in the low-density regime, since computing shortest vectors is in general also known to be NP-hard [2].

[1] Founded by NRW Research Training Group SecHuman.

13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2018).
Editor: Stacey Jeffery; Article No. 5; pp. 5:1–5:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

In the high-density regime with $\ell = \mathcal{O}(\log n)$ dynamic programming solves subset sum efficiently, see [11]. However, for the case $\ell(n) \approx n$ only exponential time algorithms are known. Impagliazzo and Naor showed constructions of cryptographic primitives in AC^0 that can be proven as hard as solving random subset sums around density 1. Many efficient cryptographic constructions followed, see e.g. [18, 10] for some recent constructions – including a CCA-secure subset sum based encryption scheme – and further references.

Classical complexity of subset sum

Let us assume that $\ell = \mathrm{poly}(n)$ such that arithmetic in $\mathbb{Z}_{2^{\ell(n)}}$ can be performed in time $\mathrm{poly}(n)$. Throughout this paper, for ease of notation we omit polynomial factors in exponential running times or space consumptions.

For solving subset sum with $\mathbf{a} = (a_1, \ldots, a_n)$, one can naively enumerate all $\mathbf{e} \in \{0,1\}^n$ and check whether $\langle \mathbf{e}, \mathbf{a} \rangle \equiv t \bmod 2^{\ell(n)}$ in time 2^n.

Let $\mathbf{a}^{(1)} = (a_1, \ldots, a_{n/2})$ and $\mathbf{a}^{(2)} = (a_{n/2+1}, \ldots, a_n)$. In the Meet-in-the-Middle approach of Horowitz and Sahni [13], one enumerates all $\mathbf{e}^{(1)}, \mathbf{e}^{(2)} \in \{0,1\}^{n/2}$ and checks for an identity $\langle \mathbf{e}^{(1)}, \mathbf{a}^{(1)} \rangle \equiv t - \langle \mathbf{e}^{(2)}, \mathbf{a}^{(2)} \rangle \bmod 2^{\ell(n)}$. This improves the time complexity to $2^{n/2}$, albeit using also space $2^{n/2}$.

Schroeppel and Shamir [21] later improved this to time $2^{n/2}$ with only space $2^{n/4}$. It remains an open problem, whether time complexity $2^{n/2}$ can be improved in the worst case [4]. However, when studying the complexity of random subset sum instances in the average case, the algorithm of Howgrave-Graham and Joux [14] runs in time $2^{0.337n}$. This is achieved by representing $\mathbf{e} = \mathbf{e}^{(1)} + \mathbf{e}^{(2)}$ with $\mathbf{e}^{(1)}, \mathbf{e}^{(2)} \in \{0,1\}^n$ ambiguously, also called the representation technique. In 2011, Becker, Coron and Joux [5] showed that the choice $\mathbf{e}^{(1)}, \mathbf{e}^{(2)} \in \{-1,0,1\}^n$ leads to even more representations, which in turn decreases the running time on average case instances to $2^{0.291n}$, the best classical running time currently known.

Quantum complexity of subset sum

In 2013, Bernstein, Jeffery, Lange and Meurer [6] constructed quantum subset sum algorithms, inspired by the classical algorithms above. Namely, Bernstein et al. showed that quantum algorithms for the naive and Meet-in-the-Middle approach achieve run time $2^{n/2}$ and $2^{n/3}$, respectively. Moreover, a first quantum version of Schroeppel-Shamir with Grover search [12] runs in time $2^{3n/8}$ using only space $2^{n/8}$. A second quantum version of Schroeppel-Shamir using quantum walks [1, 3] achieves time $2^{0.3n}$. Eventually, Bernstein, Jeffery, Lange and Meurer used the quantum walk framework of Magniez et al. [19] to achieve a quantum version of the Howgrave-Graham, Joux algorithm with time and space complexity $2^{0.241n}$.

Our result

Interestingly, Bernstein et al. did not provide a quantum version of the best classical algorithm – the BCJ-algorithm by Becker, Coron and Joux [5] – that already classically has some quite tedious analysis. We fill this gap and complete the complexity landscape quantumly, by defining an appropriate quantum walk for the BCJ-algorithm within the framework of Magniez et al. [19]. Our run time analysis relies on some unproven conjecture that we make explicit in Section 4. Under this conjecture, we show that all but a negligible fraction of instances of subset sum can be solved quantumly in time and space $2^{0.226n}$, giving polynomial speedups over the best classical complexity $2^{0.291n}$ and the best quantum complexity $2^{0.241n}$.

In a nutshell, our conjecture states that in the run-time analysis we can replace in a quantum walk an update with expected constant cost by an update with polynomially upper-bounded cost (that might stop), without significantly affecting the error probability and the structure of the random walk graph. While it might be legitimate to use an unproven non-standard conjecture to say something reasonable on the quantum complexity of problems in post-quantum cryptography, especially in the context of the present NIST standardization process, our conjecture is somewhat unsatisfactory from a theoretical point of view. We hope that our work encourages people to base this conjecture on more solid theoretical foundations.

Apart from that our result holds for random subset sums with $\ell = \text{poly}(n)$, i.e. with polynomial density. However, our algorithm behaves worst for subset sum instances with unique solution, i.e. in the case $\ell(n) \geq n$. In the high-density case $\ell(n) < n$, our analysis is non-optimal and might be subject to improvements.

The complexity $2^{0.226n}$ is achieved for subset sum solutions $t \equiv \sum_{i \in I} a_i \bmod 2^{\ell(n)}$ with worst case $|I| = n/2$. We also analyze the complexity for $|I| = \beta n$ with arbitrary $\beta \in [0,1]$. For instance for $\beta = 0.2$, our quantum-BCJ algorithm runs in time and space $2^{0.175n}$.

The paper is organized as follows. Section 2 defines some notation. In Section 3, we repeat the BCJ algorithm and its classical complexity analysis that we later adapt to the quantum case. In Section 4, we analyze the cost of a random walk on the search space defined by the BCJ algorithm and define an appropriate data structure. In Section 5, we put things together and analyze the complexity of the BCJ algorithm, enhanced by a quantum walk technique.

2 Preliminaries

Let $D = \{-1, 0, 1\}$ be a digit set, and let $\alpha, \beta \in \mathbb{Q} \cap [0,1]$ with $2\alpha + \beta \leq 1$. We use the notation $\mathbf{e} \in D^n[\alpha, \beta]$ to denote that $\mathbf{e} \in D^n$ has αn (-1)-entries, $(\alpha + \beta)n$ 1-entries and $(1 - 2\alpha - \beta)n$ 0-entries. Especially, $\mathbf{e} \in D^n[0, \beta]$ is a binary vector with βn 1-entries. Throughout the paper we ignore rounding issues and assume that αn and $(\alpha + \beta)n$ take on integer values.

We naturally extend the binomial coefficient notation $\binom{n}{k} = \frac{n!}{k!(n-k)!}$ to a multinomial coefficient notation

$$\binom{n}{k_1, \dots, k_r} = \frac{n!}{k_1! \dots k_r!(n - k_1 - \dots - k_r)!}.$$

Let $H(x) = -x \cdot \log_2(x) - (1-x) \cdot \log_2(1-x)$ denote the binary entropy function. From Stirling's formula one easily derives

$$\binom{\alpha n}{\beta n} \approx 2^{\alpha \cdot H(\frac{\beta}{\alpha})n},$$

where the $\approx$-notation suppresses polynomial factors.

Analogous, let $g(x,y) := -x \cdot \log_2(x) - y \cdot \log_2(y) - (1-x-y) \cdot \log_2(1-x-y)$. Then

$$\binom{\alpha n}{\beta n, \gamma n} \approx 2^{\alpha \cdot g(\frac{\beta}{\alpha}, \frac{\gamma}{\alpha})n}.$$

Let $\mathbb{Z}_{2^{\ell(n)}}$ be the ring of integers modulo $2^{\ell(n)}$. For the n-dimensional vectors $\mathbf{a} = (a_1, \dots, a_n) \in (\mathbb{Z}_{2^{\ell(n)}})^n$, $\mathbf{e} = (e_1, \dots, e_n) \in D^n[\alpha, \beta]$ the inner product is denoted

$$\langle \mathbf{a}, \mathbf{e} \rangle = \sum_{i=1}^{n} a_i e_i \bmod 2^{\ell(n)}.$$

We define a random weight-β (solvable) subset sum instance as follows.

▶ **Definition 1** (Random Subset Sum). Let $\mathbf{a}$ be chosen uniformly at random from $(\mathbb{Z}_{2^{\ell(n)}})^n$. For $\beta \in [0,1]$, choose a random $\mathbf{e} \in D^n[0,\beta]$ and compute $t = \langle \mathbf{a}, \mathbf{e} \rangle \in \mathbb{Z}_{2^{\ell(n)}}$. Then $(\mathbf{a}, t) \in (\mathbb{Z}_{2^{\ell(n)}})^{n+1}$ is a *random subset sum instance*. For $(\mathbf{a}, t)$, any $\mathbf{e}' \in \{0,1\}^n$ with $\langle \mathbf{a}, \mathbf{e}' \rangle \equiv t \bmod 2^{\ell(n)}$ is called a *solution*.

3 Subset Sum Classically – The BCJ Algorithm

Let $D = \{-1, 0, 1\}$ and let $(\mathbf{a}, t) = (a_1, \ldots, a_n, t) \in (\mathbb{Z}_{2^{\ell(n)}})^{n+1}$ be a subset sum instance with solution $\mathbf{e} \in D^n[0, \frac{1}{2}]$. That is $\langle \mathbf{e}, \mathbf{a} \rangle \equiv t \bmod 2^{\ell(n)}$, where $n/2$ of the coefficients of $\mathbf{e}$ are 1 and $n/2$ coefficients are 0.

Representations

The core idea of the Becker-Coron-Joux (BCJ) algorithm is to represent the solution $\mathbf{e}$ *ambiguously* as a sum

$$\mathbf{e} = \mathbf{e}_1^{(1)} + \mathbf{e}_1^{(2)} \text{ with } \mathbf{e}_1^{(1)}, \mathbf{e}_1^{(2)} \in D^n[\alpha_1, 1/4].$$

This means that we represent $\mathbf{e} \in D^n[0, 1/2]$ as a sum of vectors with $\alpha_1 n$ (-1)-entries, $(1/4 + \alpha_1)n$ 1-entries and $(3/4 - 2\alpha_1)n$ 0-entries. We call $(\mathbf{e}_1^{(1)}, \mathbf{e}_1^{(2)})$ a *representation* of $\mathbf{e}$.

Thus, every 1-coordinate e_i of $\mathbf{e}$ can be represented as either $1 + 0$ or $0 + 1$ via the i^{th}-coordinates of $\mathbf{e}_1^{(1)}, \mathbf{e}_1^{(2)}$. Since we have $n/2$ 1-coordinates in $\mathbf{e}$, we can fix among these $n/4$ 0-coordinates and $n/4$ 1-coordinates in $\mathbf{e}_1^{(1)}$, determining the corresponding entries in $\mathbf{e}_2^{(1)}$. This can be done in $\binom{n/2}{n/4}$ ways.

Analogously, the 0-coordinates in $\mathbf{e}$ can be represented as either $(-1)+1$, $1+(-1)$ or $0+0$. Again, we can fix among these $n/2$ coordinates $\alpha_1 n$ (-1)-coordinates, $\alpha_1 n$ 1-coordinates and $n/2 - 2\alpha_1 n$ 0-coordinates in $\mathbf{e}_1^{(1)}$. This can be done in $\binom{n/2}{\alpha_1 n, \alpha_1 n}$ ways.

Thus, in total we represent our desired solution $\mathbf{e}$ in

$$R_1 = \binom{n/2}{n/4}\binom{n/2}{\alpha_1 n, \alpha_1 n} \text{ ways.}$$

However, notice that constructing a *single representation* of $\mathbf{e}$ is sufficient for solving subset sum. Thus, the main idea of the BCJ algorithm is to compute only a $1/R_1$-fraction of all representations such that on expectation a single representation survives.

This is done by computing only those representations $(\mathbf{e}_1^{(1)}, \mathbf{e}_1^{(2)})$ such that the partial sums

$$\langle \mathbf{e}_1^{(1)}, \mathbf{a} \rangle \text{ and } t - \langle \mathbf{e}_1^{(2)}, \mathbf{a} \rangle$$

attain a fixed value modulo 2^{r_1}, where $r_1 = \lfloor \log R_1 \rfloor$. This value can be chosen randomly, but for simplicity of notation we assume in the following that both partial sums are 0 modulo 2^r.

More precisely, we construct the lists

$$\begin{aligned} L_1^{(1)} &= \{(\mathbf{e}_1^{(1)}, \langle \mathbf{e}_1^{(1)}, \mathbf{a} \rangle) \in D^n[\alpha_1, 1/4] \times \mathbb{Z}_{2^{\ell(n)}} \mid \langle \mathbf{e}_1^{(1)}, \mathbf{a} \rangle \equiv 0 \bmod 2^{r_1}\} \text{ and} \\ L_1^{(2)} &= \{(\mathbf{e}_1^{(2)}, \langle \mathbf{e}_1^{(2)}, \mathbf{a} \rangle) \in D^n[\alpha_1, 1/4] \times \mathbb{Z}_{2^{\ell(n)}} \mid t - \langle \mathbf{e}_1^{(2)}, \mathbf{a} \rangle \equiv 0 \bmod 2^{r_1}\}. \end{aligned}$$

Hence, $L_1^{(1)}, L_1^{(2)}$ have the same expected list length, which we denote shortly by

$$\mathbb{E}[|L_1|] = \frac{\binom{n}{\alpha_1 n, (1/4 + \alpha_1)n}}{2^{r_1}}.$$

Constructing the lists

$L_1^{(1)}, L_1^{(2)}$ are constructed recursively, see also Fig. 1. Let us first explain the construction of $L_1^{(1)}$. We represent $\mathbf{e}_1^{(1)} \in D^n[\alpha_1, 1/4]$ as

$$\mathbf{e}_1^{(1)} = \mathbf{e}_2^{(1)} + \mathbf{e}_2^{(2)} \text{ with } \mathbf{e}_2^{(1)}, \mathbf{e}_2^{(2)} \in D^n[\alpha_2, 1/8], \text{ where } \alpha_2 \geq \alpha_1/2.$$

By the same reasoning as before, the number of representations is

$$R_2 = \binom{\alpha_1 n}{\alpha_1/2n}\binom{(1/4+\alpha_1)n}{(1/8+\alpha_1/2)n}\binom{(3/4-2\alpha_1)n}{(\alpha_2-\alpha_1/2)n, (\alpha_2-\alpha_1/2)n},$$

where the three factors stand for the number of ways of representing (-1)-, 1- and 0-coordinates of $\mathbf{e}_1^{(1)}$. Let $r_2 = \lfloor \log R_2 \rfloor$. We define

$$\begin{aligned} L_2^{(j)} &= \{(\mathbf{e}_2^{(j)}, \langle \mathbf{e}_2^{(j)}, \mathbf{a}\rangle) \in D^n[\alpha_2, 1/8] \times \mathbb{Z}_{2^{\ell(n)}} \mid \langle \mathbf{e}_2^{(j)}, \mathbf{a}\rangle \equiv 0 \bmod 2^{r_2}\} \text{ for } j = 1,2,3, \\ L_2^{(4)} &= \{(\mathbf{e}_2^{(4)}, \langle \mathbf{e}_2^{(4)}, \mathbf{a}\rangle) \in D^n[\alpha_2, 1/8] \times \mathbb{Z}_{2^{\ell(n)}} \mid t - \langle \mathbf{e}_2^{(4)}, \mathbf{a}\rangle \equiv 0 \bmod 2^{r_2}\}. \end{aligned}$$

Thus, we obtain on level 2 of our search tree in Fig. 1 expected list sizes

$$\mathbb{E}[|L_2|] = \frac{\binom{n}{\alpha_2 n, (1/8+\alpha_2)n}}{2^{r_2}}.$$

An analogous recursive construction of level-3 lists $L_3^{(j)}$ from our level-2 lists yields

$$\mathbb{E}[|L_3|] = \frac{\binom{n}{\alpha_3 n, (1/16+\alpha_3)n}}{2^{r_3}},$$

where $r_3 = \lfloor \log R_3 \rfloor$ with

$$R_3 = \binom{\alpha_2 n}{\alpha_2/2n}\binom{(1/8+\alpha_2)n}{(1/16+\alpha_2/2)n}\binom{(7/8-2\alpha_2)n}{(\alpha_3-\alpha_2/2)n, (\alpha_3-\alpha_2/2)n}.$$

The level-3 lists are eventually constructed by a standard Meet-in-the-Middle approach from the following level-4 lists (where we omit the definition of $L_4^{(15)}, L_4^{(16)}$ that is analogous with $\mathbf{t} - \langle \mathbf{e}_4^{(\cdot)}, \mathbf{a}\rangle$)

$$\begin{aligned} L_4^{(2j-1)} &= \{(\mathbf{e}_4^{(2j-1)}, \langle \mathbf{e}_4^{(2j-1)}, \mathbf{a}\rangle) \in D^{n/2}[\alpha_3/2, 1/32] \times 0^{n/2} \times \mathbb{Z}_{2^{\ell(n)}}\} \text{ and} \\ L_4^{(2j)} &= \{(\mathbf{e}_4^{(2j)}, \langle \mathbf{e}_4^{(2j)}, \mathbf{a}\rangle) \in 0^{n/2} \times D^{n/2}[\alpha_3/2, 1/32] \times \mathbb{Z}_{2^{\ell(n)}}\} \text{ for } j = 1, \ldots, 7 \end{aligned}$$

of size

$$|L_4| = \binom{n/2}{(\alpha_3/2)n, (1/32+\alpha_3/2)n}.$$

Let us define indicator variables

$$X_{i,j} = \langle \mathbf{e}_i^{(2j-1)}, \mathbf{a}\rangle \text{ and } X_{i,j}^{+} = \langle \mathbf{e}_i^{(2j)}, \mathbf{a}\rangle \text{ for } i = 1,2,3,4 \text{ and } j = 1, \ldots, 2^{i-1}.$$

By the randomness of $\mathbf{a}$, we have $\Pr[X_{i,j} = c] = \Pr[X_{i,j}^{+} = c] = \frac{1}{2^{\ell(n)}}$ for all $c \in \mathbb{Z}_{2^{\ell(n)}}$. Thus, all $X_{i,j}$, $X_{i,j}^{+}$ are uniformly distributed in $\mathbb{Z}_{2^{\ell(n)}}$, and therefore also uniformly distributed modulo 2^{r_i} for any $r_i \leq \ell(n)$. Unfortunately, for fixed i, j the pair $X_{i,j}, X_{i,j}^{+}$ is not independent. We assume in the following that this (mild) dependence does not affect the run time analysis.

▶ **Heuristic 1.** *For the BCJ runtime analysis, we can treat all pairs $X_{i,j}, X_{i,j}^{+}$ as independent.*

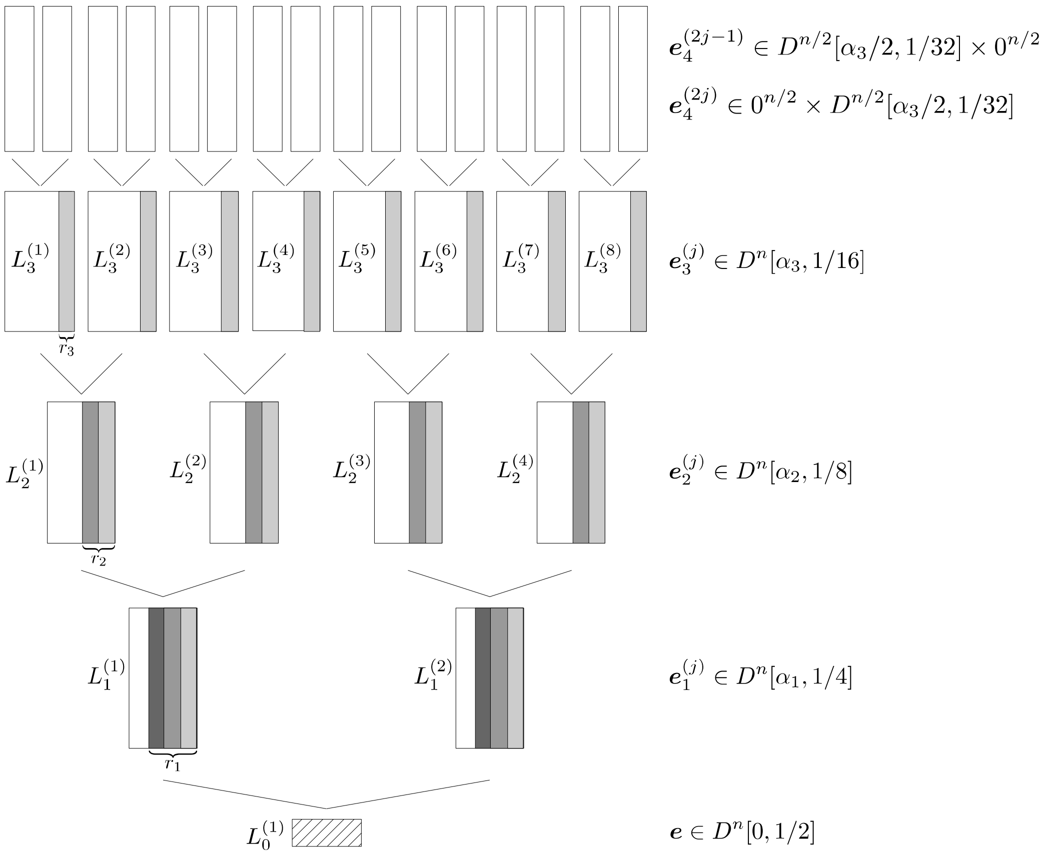

$\mathbf{e}_4^{(2j-1)} \in D^{n/2}[\alpha_3/2, 1/32] \times 0^{n/2}$
$\mathbf{e}_4^{(2j)} \in 0^{n/2} \times D^{n/2}[\alpha_3/2, 1/32]$
$L_3^{(1)}$ $L_3^{(2)}$ $L_3^{(3)}$ $L_3^{(4)}$ $L_3^{(5)}$ $L_3^{(6)}$ $L_3^{(7)}$ $L_3^{(8)}$
r_3
$\mathbf{e}_3^{(j)} \in D^n[\alpha_3, 1/16]$
$L_2^{(1)}$ $L_2^{(2)}$ $L_2^{(3)}$ $L_2^{(4)}$
r_2
$\mathbf{e}_2^{(j)} \in D^n[\alpha_2, 1/8]$
$L_1^{(1)}$ $L_1^{(2)}$
r_1
$\mathbf{e}_1^{(j)} \in D^n[\alpha_1, 1/4]$
$L_0^{(1)}$
$\mathbf{e} \in D^n[0, 1/2]$

Figure 1 Tree structure of the BCJ-Algorithm.

Under Heuristic 1 it can easily be shown that for all but a negligible fraction of random subset sum instances the lists sizes are sharply concentrated around their expectation. More precisely, a standard Chernoff bound shows that for all but a negligible fraction of instances the list size of $L_i^{(j)}$ lies in the interval

$$\mathbb{E}(|L_i|) - \mathbb{E}(|L_i|)^{1/2} \leq |L_i| \leq \mathbb{E}(|L_i|) + \mathbb{E}(|L_i|)^{1/2} \text{ for } i = 1, 2, 3. \tag{1}$$

In other words, for all but some pathological instances we have $|L_i| = \mathcal{O}(\mathbb{E}(|L_i|)$.

We give a description of the BCJ algorithm in Algorithm 1. Here we assume in more generality that a subset sum instance $(\mathbf{a}, t)$ has a solution $\mathbf{e} \in D^n[0, \beta]$. As one would expect, Algorithm 1 achieves its worst-case complexity for $\beta = \frac{1}{2}$ with a balanced number of zeros and ones in $\mathbf{e}$. However, one can also analyze the complexity for arbitrary β, as we will do for our quantum version of BCJ.

For generalizing our description from before to arbitrary β, we have to simply replace $\mathbf{e}_i^{(j)} \in D^n[\alpha_i, \frac{1}{2}2^{-i}]$ by $\mathbf{e}_i^{(j)} \in D^n[\alpha_i, \beta 2^{-i}]$.

By the discussion before, the final condition $|L_0^{(1)}| > 0$ in Algorithm 1 implies that we succeed in constructing a representation $(\mathbf{e}_1^{(1)}, \mathbf{e}_1^{(2)}) \in (D^n[\alpha_1, \beta n/2])^2$ of $\mathbf{e} \in D^n[0, \beta]$, where the $\mathbf{e}_1^{(j)}$ recursively admit representations $(\mathbf{e}_2^{(2j-1)}, \mathbf{e}_2^{(2j-1)}) \in (D^n[\alpha_2, \beta n/4])^2)$, and so forth.

Algorithm 1: Becker-Coron-Joux (BCJ) algorithm.

```
Input      : $(\mathbf{a}, t) \in (\mathbb{Z}_{2^{\ell(n)}})^{n+1}, \beta \in [0,1]$
Output     : $\mathbf{e} \in D^n[0, \beta]$
Parameters : Optimize $\alpha_1, \alpha_2, \alpha_3$.
Construct all level-4 lists $L_4^{(j)}$ for $j = 1, \ldots, 16$.
for $i = 3$ down to $0$ do
    Compute $L_i^{(j)}$ from $L_{i+1}^{(2j-1)}$, $L_{i+1}^{(2j)}$ for $j = 1, \ldots, 2^i$.
end
if $|L_0^{(1)}| > 0$ then output an arbitrary element from $L_0^{(1)}$.
```

Thus, one can eventually express

$$\mathbf{e} = \mathbf{e}_4^{(1)} + \mathbf{e}_4^{(2)} + \ldots + \mathbf{e}_4^{(16)}.$$

However, notice that we constructed all lists in such a way that on expectation at least one representation survives for every list $L_i^{(j)}$ from the for-loop of Algorithm 1. This implies that the BCJ algorithm succeeds in finding the desired solution $\mathbf{e}$, and therefore the leaves of our search tree in Fig. 1 contain elements that sum up to $\mathbf{e}$. The following theorem and its proof show how to optimize the parameters α_i, $i = 1, 2, 3$ such that BCJ's running time is minimized while still guaranteeing a solution.

▶ **Theorem 2** (BCJ 2011). *Under Heuristic 1 Algorithm 1 solves all but a negligible fraction of random subset sum instances* $(\mathbf{a}, t) \in (\mathbb{Z}_{2^{\ell(n)}})^{n+1}$ *(Definition 1) in time and memory* $2^{0.291n}$.

Proof. Numerical optimization yields the parameters

$$\alpha_1 = 0.0267, \ \alpha_2 = 0.0302, \ \alpha_3 = 0.0180.$$

This leads to

$$R_3 = 2^{0.241n}, \ R_2 = 2^{0.532n}, \ R_1 = 2^{0.799n} \text{ representations,}$$

which in turn yield expected list sizes

$$|L_4| = 2^{0.266n}, \ \mathbb{E}(|L_3|) = 2^{0.2909n}, \ \mathbb{E}(|L_2|) = 2^{0.279n}, \ \mathbb{E}(|L_1|) = 2^{0.217n}, \ \mathbb{E}(|L_0|) = 1.$$

For $i = 1, 2, 3$ the level-i lists $L_i^{(j)}$ can be constructed in time $2^{0.2909n}$ by looking at all pairs in $L_{i-1}^{(2j-1)} \times L_{i-1}^{(2j)}$. Under Heuristic 1, we conclude by Eq. (1) that for all but a negligible fraction of instance we have $|L_i| = \mathcal{O}(\mathbb{E}(|L_i|)$ for $i = 1, 2, 3$. Thus, the total running time and memory complexity can be bounded by $2^{0.291n}$. ◀

4 From Trees to Random Walks to Quantum Walks

In Section 3, we showed how the BCJ algorithm builds a search tree t whose root contains a solution $\mathbf{e}$ to the subset sum problem. More precisely, the analysis of the BCJ algorithm in the proof of Theorem 2 shows that the leaves of t contain a representation $(\mathbf{e}_4^{(1)}, \ldots, \mathbf{e}_4^{(16)}) \in L_4^{(1)} \times \ldots \times L_4^{(16)}$ of $\mathbf{e}$, i.e. $\mathbf{e} = \mathbf{e}_4^{(1)} + \ldots + \mathbf{e}_4^{(16)}$.

Idea of Random Walk

In a random walk, we no longer enumerate the lists $L_4^{(j)}$ completely, but only a random subset $U_4^{(j)} \subseteq L_4^{(j)}$ of some fixed size $|U_4| := |U_4^{(j)}|$, that has to be optimized. We run on these projected leaves the original BCJ algorithm, but with parameters $\alpha_1, \alpha_2, \alpha_3$ that have to be optimized anew. On the one hand, a small $|U_4|$ yields small list sizes, which in turn speeds up the BCJ algorithm. On the other hand, a small $|U_4|$ reduces the probability that BCJ succeeds. Namely, BCJ outputs the desired solution $\mathbf{e}$ iff $(\mathbf{e}_4^{(1)}, \ldots, \mathbf{e}_4^{(16)}) \in U_4^{(1)} \times \ldots \times U_4^{(16)}$, which happens with probability

$$\epsilon = \left(\frac{|U_4|}{|L_4|}\right)^{16}. \tag{2}$$

The graph $G = (V, E)$ of our Random Walk

We define vertices V with labels $U_4^{(1)} \times \ldots \times U_4^{(16)}$. Each vertex $v \in V$ contains the complete BCJ search tree with leaf lists defined by its label. Two vertices with labels $\ell = U_4^{(1)} \times \ldots \times U_4^{(16)}$ and $\ell' = V_4^{(1)} \times \ldots \times V_4^{(16)}$ are adjacent iff their symmetric difference is $|\Delta(\ell, \ell')| = 1$. I.e., we have $U_4^{(j)} = V_4^{(j)}$ for all j but one $V_4^{(i)} \neq V_4^{(i)}$ for which $U_4^{(i)}, V_4^{(i)}$ differ by only one element.

▶ **Definition 3** (Johnson graph)**.** Given an N-size set L the Johnson graph $J(N, r)$ is an undirected graph $G_J = (V_J, E_J)$ with vertices labeled by all r-size subsets of L. An edge between two vertices $v, v' \in V_J$ with labels ℓ, ℓ' exists iff $|\Delta(\ell, \ell')| = 1$.

In our case, we define $N = |L_4|, r = |U_4|$ and for each of our 16 lists $L_4^{(j)}$ its corresponding Johnson graph $J_j(N, r)$. However, by our construction above we want that two vertices are adjacent iff they differ in only one element throughout all 16 lists.

Let us therefore first define the Cartesian product of graphs. We will then show that our graph $G = (V, E)$ is exactly the Cartesian product

$$J^{16}(N, r) := J_1(N, r) \times \ldots \times J_{16}(N, r).$$

▶ **Definition 4.** Let $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$ be undirected graphs. The Cartesian product $G_1 \times G_2 = (V, E)$ is defined via

$$\begin{aligned} V &= V_1 \times V_2 = \{v_1 v_2 \mid v_1 \in V_1,\ v_2 \in V_2\} \text{ and} \\ E &= \{(u_1 u_2, v_1 v_2) \mid (u_1 = v_1 \wedge (u_2, v_2) \in E_2) \vee ((u_1, v_1) \in E_1 \wedge u_2 = v_2)\} \end{aligned}$$

Thus, in $J_1(n, r) \times J_2(n, r)$ the labels $v_1 v_2$ are Cartesian products of the labels $U_4^{(1)}, U_4^{(2)}$. An edge in $J_1(n, r) \times J_2(n, r)$ is set between two vertices with labels $U_4^{(1)} \times U_4^{(2)}$, $V_4^{(1)} \times V_4^{(2)}$ iff $U_4^{(1)} = V_4^{(1)}$ and $U_4^{(2)}$, $V_4^{(2)}$ differ by exactly one element or vice versa, as desired.

Mixing time

The mixing time of a random walk depends on its so-called spectral gap.

▶ **Definition 5** (Spectral gap)**.** Let G be an undirected graph. Let λ_1, λ_2 be the eigenvalues with largest absolute value of the transition matrix of the random walk on G. Then the *spectral gap* of a random walk on G is defined as $\delta(G) := |\lambda_1| - |\lambda_2|$.

For Johnson graphs it is well-known that $\delta(J(N,r)) = \frac{N}{r(N-r)} = \Omega(\frac{1}{r})$. The following lemma shows that for our graph $J^{16}(N,r)$ we have as well

$$\delta(J^{16}(N,r)) = \Omega\left(\frac{1}{r}\right) = \Omega\left(\frac{1}{|U_4|}\right). \tag{3}$$

▶ **Lemma 6** (Kachigar, Tillich [16]). *Let $J(N,r)$ be a Johnson graph, and let $J^m(N,r) := \bigtimes_{i=1}^{m} J(n,r)$. Then $\delta(J^m) \geq \frac{1}{m}\delta(J)$.*

Walking on G

We start our random walk on a random vertex $v \in V$, i.e. we choose random $U_4^{(j)} \subseteq L_4^{(j)}$ for $j = 1, \ldots, 16$ and compute the corresponding BCJ tree t_v on these sets. This computation of the starting vertex v defines the *setup cost* T_S of our random walk.

Let us quickly compute T_S for the BCJ algorithm, neglecting all polynomial factors. The level-4 lists $U_4^{(j)}$ can be computed and sorted with respect to the inner products $\langle \mathbf{e}_4^{(j)}, \mathbf{a} \rangle \bmod 2^{r_3}$ in time $|U_4|$. The level-3 lists contain all elements from their two level-4 children lists that match on the inner products. Thus we expect $\mathbb{E}(|U_3|) = |U_4|^2/2^{r_3}$ elements that match on their inner products. Analogous, we compute level-2 lists in expected time $|U_3|^2/2^{r_2-r_3}$. However, now we have to filter out all $\mathbf{e}_2^{(j)}$ that do not possess the correct weight distribution, i.e. the desired number of (-1)s, 0s, and 1s. Let us call any level-i $\mathbf{e}_i^{(j)}$ *consistent* if $\mathbf{e}_i^{(j)}$ has the correct weight distribution on level i. Let $p_{3,2}$ denote the probability that a level-2 vector constructed as a sum of two level-3 vectors is consistent. From Section 3 we have

$$\frac{|L_3|^2}{2^{r_2-r_3}} \cdot p_{3,2} = \mathbb{E}(|L_2|),$$

which implies

$$p_{3,2} := \frac{\binom{n}{\alpha_2 n, (1/8+\alpha_2)n}}{\binom{n}{\alpha_3 n, (1/16+\alpha_3)n}^2} \cdot 2^{r_2-r_3}.$$

Thus, after filtering for the correct weight distribution we obtain an expected level-2 list size of $\mathbb{E}(|U_2|) = |U_3|^2/2^{r_2-r_3} \cdot p_{3,2}$. Analogous, on level 1 we obtain expected list size $\mathbb{E}(|U_1|) = |U_2|^2/2^{r_1-r_2} \cdot p_{2,1}$ with

$$p_{2,1} := \frac{\binom{n}{\alpha_1 n, (1/4+\alpha_1)n}}{\binom{n}{\alpha_2 n, (1/8+\alpha_2)n}^2} \cdot 2^{r_1-r_2}.$$

The level-0 list can be computed in expected time $|U_1|^2/2^{n-r_1}$. In total we obtain

$$\mathbb{E}[T_S] = \max\left\{|U_4|, \frac{|U_4|^2}{2^{r_3}}, \frac{|U_3|^2}{2^{r_2-r_3}}, \frac{|U_2|^2}{2^{r_1-r_2}}, \frac{|U_1|^2}{2^{n-r_1}}\right\}$$

Analogous to the reasoning in Section 3 (see Eq. 1), for all but a negligible fraction of random subset sum instances we have $|U_i| = \mathcal{O}(\mathbb{E}(|U_i|))$. Thus, for all but a negligible fraction of instances and neglecting constants we have

$$\begin{aligned} T_S = &\quad \max\left\{|U_4|, \tfrac{|U_4|^2}{2^{r_3}}, \tfrac{\mathbb{E}(|U_3|)^2}{2^{r_2-r_3}}, \tfrac{\mathbb{E}(|U_2|)^2}{2^{r_1-r_2}}, \tfrac{\mathbb{E}(|U_1|)^2}{2^{n-r_1}}\right\} & (4)\\ \leq &\quad \max\left\{|U_4|, \tfrac{|U_4|^2}{2^{r_3}}, \tfrac{|U_4|^4}{2^{r_2+r_3}}, \tfrac{|U_4|^8}{2^{r_1+r_2+2r_3}}, \tfrac{|U_4|^{16}}{2^{n+r_1+2r_2+4r_3}}\right\} := \tilde{T}_S. & (5)\end{aligned}$$

If t_v contains a non-empty root with subset-sum solution $\mathbf{e}$, we denote v *marked.* Hence, we walk our graph $G = J_1(|L_4|, |U_4|) \times \ldots \times J_{16}(|L_4|, |U_4|)$ until we hit a marked vertex, which solves subset sum.

The cost for checking whether a vertex v is marked is denoted *checking cost* T_C. In our case checking can be done easily by looking at t_v's root. Thus, we obtain (neglecting polynomials)

$$T_C = 1. \tag{6}$$

Since any neighboring vertices v, v' in G only differ by one element in some leaf $U_4^{(j)}$, when walking from v to v' we do not have to compute the whole tree $t_{v'}$ anew, but instead we update t_v to $t_{v'}$ by changing the nodes on the path from list $U_4^{(j)}$ to its root accordingly. The cost of this step is therefore called *update cost* T_U. Our cost T_U heavily depends on the way we internally represent t_v. In the following, we define a data structure that allows for optimal *update cost* per operation.

4.1 Data Structure for Updates

Let us assume that we have a data structure that allows the three operations search, insertion and deletion in time logarithmic in the number of stored elements. In Bernstein et al. [7], it is e.g. suggested to use radix trees. Since our lists have exponential size and we ignore polynomials in the run time analysis, every operation has cost 1. This data structure also ensures the uniqueness of quantum states $|U_4^{(1)}, \ldots, U_4^{(16)}\rangle$, which in turn guarantees correct interference of quantum states with identical lists.

Definition of data structure

Recall from Section 3, that BCJ level-4 lists are of the form $L_4^{(j)} = \{(\mathbf{e}_4^{(j)}, \langle \mathbf{e}_4^{(j)}, \mathbf{a}\rangle)\}$. For our $U_4^{(j)} \subset L_4^{(j)}$ we store in our data structure the $\mathbf{e}_4^{(j)}$ and their inner products with $\mathbf{a}$ separately in

$$E_4^{(j)} = \{\mathbf{e}_4^{(j)} \mid \mathbf{e}_4^{(j)} \in U_4^{(j)}\} \text{ and } S_4^{(j)} = \{(\langle \mathbf{e}_4^{(j)}, \mathbf{a}\rangle, \mathbf{e}_4^{(j)}) \mid \mathbf{e}_4^{(j)} \in U_4^{(j)}\}, \tag{7}$$

where in $S_4^{(j)}$ elements are addressed via their first datum $\langle \mathbf{e}_4^{(j)}, \mathbf{a}\rangle$. Analogous, for $U_i^{(j)}$, $i = 3, 2, 1$ we also build separate $E_i^{(j)}$ and $S_i^{(j)}$. For the root list $U_0^{(1)}$, it suffices to build $E_0^{(1)}$.

We denote the operations on our data structure as follows. Insert$(E_i^{(j)}, \mathbf{e})$ inserts $\mathbf{e}$ into $E_i^{(j)}$, whereas Delete$(E_i^{(j)}, \mathbf{e})$ deletes one entry $\mathbf{e}$ from $E_i^{(j)}$. Furthermore, $\{\mathbf{e}_i\} \leftarrow$ Search$(S_i^{(j)}, \langle \mathbf{e}_i^{(j)}, \mathbf{a}\rangle)$ returns the list of all $\mathbf{e}_i$ with first datum $\langle \mathbf{e}_i^{(j)}, \mathbf{a}\rangle$.

Deletion/Insertion of an element

Our random walk replaces a list element in exactly one of the leaf lists $U_4^{(j)}$. We can perform the update by first deleting the replaced element and update the path to the root accordingly, and second adding the new element and again updating the path to the root.

Let us look more closely at the deletion process. On every level we delete a value, and then compute via the sibbling vertex, which values we have to be deleted recursively on the parent level. For illustration, deletion of $\mathbf{e}$ in $U_4^{(3)}$ triggers the following actions.

```
- Delete $(E_4^{(3)}, \mathbf{e})$.
- $\{\mathbf{e}_4^{(4)}\} \leftarrow \text{Search}(S_4^{(4)}, \langle \mathbf{e}, \mathbf{a} \rangle \bmod 2^{r_3})$    // $\mathbb{E}(|\{\mathbf{e}_4^{(4)}\}|) = \frac{|U_4|}{2^{r_3}}$
- For all $\mathbf{e}_3^{(2)} = \mathbf{e} + \mathbf{e}'$ with $\mathbf{e}' \in \{\mathbf{e}_4^{(4)}\}$
    - Delete $(E_3^{(2)}, \mathbf{e}_3^{(2)})$
    - $\{\mathbf{e}_3^{(1)}\} \leftarrow \text{Search}(S_3^{(1)}, \langle \mathbf{e}_3^{(2)}, \mathbf{a} \rangle \bmod 2^{r_2})$    // $\mathbb{E}(|\{\mathbf{e}_3^{(1)}\}|) = \frac{|U_3|}{2^{r_2 - r_3}}$
    - For all $\mathbf{e}_2^{(1)} = \mathbf{e}_3^{(2)} + \mathbf{e}'$ with $\mathbf{e}' \in \{\mathbf{e}_3^{(1)}\}$
        * Delete $(E_2^{(1)}, \mathbf{e}_2^{(1)})$
        * $\{\mathbf{e}_2^{(2)}\} \leftarrow \text{Search}(S_2^{(2)}, \langle \mathbf{e}_2^{(1)}, \mathbf{a} \rangle \bmod 2^{r_1})$    // $\mathbb{E}(|\{\mathbf{e}_2^{(2)}\}|) = \frac{|U_2|}{2^{r_1 - r_2}}$
        * For all $\mathbf{e}_1^{(1)} = \mathbf{e}_2^{(1)} + \mathbf{e}'$ with $\mathbf{e}' \in \{\mathbf{e}_2^{(2)}\}$
            · Delete $(E_1^{(1)}, \mathbf{e}_1^{(1)})$.
            · $\{\mathbf{e}_1^{(2)}\} \leftarrow \text{Search}(S_1^{(2)}, \langle \mathbf{e}_1^{(1)}, \mathbf{a} \rangle \bmod 2^{n})$    // $\mathbb{E}(|\{\mathbf{e}_1^{(2)}\}|) = \frac{|U_1|}{2^{n - r_1}}$
            · For all $\mathbf{e}_0^{(1)} = \mathbf{e}_1^{(1)} + \mathbf{e}'$ with $\mathbf{e}' \in \{\mathbf{e}_1^{(2)}\}$
            o Delete $(E_0^{(1)}, \mathbf{e}_0^{(1)})$.
```

Insertion of an element is analogous to deletion. Hence, the expected *update cost* is

$$\begin{aligned}
\mathbb{E}(T_U) &= \max\left\{1, \frac{|U_4|}{2^{r_3}}, \frac{|U_4|\mathbb{E}(|U_3|)}{2^{r_2}}, \frac{|U_4|\mathbb{E}(|U_3|)\mathbb{E}(|U_2|)}{2^{r_1}}, \frac{|U_4|\mathbb{E}(|U_3|)\mathbb{E}(|U_2|)\mathbb{E}(|U_1|)}{2^{n}}\right\} && (8)\\
&\leq \max\left\{1, \frac{|U_4|}{2^{r_3}}, \frac{|U_4|^3}{2^{r_2+r_3}}, \frac{|U_4|^7}{2^{r_1+r_2+2r_3}}, \frac{|U_4|^{15}}{2^{n}}\right\} := \tilde{T}_U. && (9)
\end{aligned}$$

Notice that for the upper bounds $\tilde{T}_S$, $\tilde{T}_U$ from Eq. (5) and (9) we have

$$\tilde{T}_S = |U_4| \cdot \tilde{T}_U. \tag{10}$$

Quantum Walk Framework

While random walks take time $T = T_S + \frac{1}{\epsilon}\left(T_C + \frac{1}{\delta} T_U\right)$, their quantum counterparts achieve some significant speedup due to their rapid mixing, as summarized in the following theorem.

▶ **Theorem 7** (Magniez et al. [19]). *Let $G = (V, E)$ be a regular graph with eigenvalue gap $\delta > 0$. Let $\epsilon > 0$ be a lower bound on the probability that a vertex chosen randomly of G is marked. For a random walk on G, let T_S, T_U, T_C be the setup, update and checking cost. Then there exists a quantum algorithm that with high probability finds a marked vertex in time*

$$T = T_S + \frac{1}{\sqrt{\epsilon}}\left(T_C + \frac{1}{\sqrt{\delta}} T_U\right).$$

Stopping unusually long updates

Recall that for setup, we showed that all instances but an exponentially small fraction finish the construction of the desired data structure in time T_S. However, the update cost is determined by the maximum cost over all *superexponentially many* vertices in a superposition. So even one node with unusually slow update may ruin our run time.

Therefore, we modify our quantum walk algorithm QW by imposing an upper bound of $\kappa = \text{poly}(n)$ steps for the update. After κ steps, we simply stop the update of all nodes and proceed as if the update has been completed. We denote by Stop-QW the resulting algorithm.

A first drawback of stopping is that some nodes that would get marked in QW, might stay unmarked in Stop-QW. However, since the event of stopping should not dependent on whether a node is marked or not, the ratio between marked and unmarked nodes and thus the success probability ϵ should not change significantly between QW and Stop-QW. Moreover, under Heuristic 1 and a standard Chernoff argument the probability of a node not finishing his update properly after κ steps is exponentially small.

A second drawback of stopping is that unfinished nodes partially destroy the structure of the Johnson graph, since different (truncated) representations of the same node do no longer interfere properly in a quantum superposition. We conjecture that this only mildly affects the spectral gap of the graph. A possible direction to prove such a conjecture might be to allow some kind of *self-repairing process* for a node. If a node cannot finish its update in time in one step, it might postpone the remaining work to subsequent steps to amortize the cost of especially expensive updates. After the repair work, a node then again joins the correct Johnson graph data structure in quantum superposition.

In the following heuristic, we assume that the change from QW to Stop-QW changes the success probability ϵ and the bound δ for the spectral gap only by a polynomial factor. This in turn allows us to analyze Stop-QW with the known parameters ϵ, δ from QW.

▶ **Heuristic 2.** *Let ϵ be the fraction of marked states and δ be the spectral gap of the random walk in* QW*. Then the fraction of marked states in* Stop-QW *is at least $\epsilon_{stop} = \frac{\epsilon}{poly(n)}$, and the spectral gap of the random walk on the graph in* StopQW *is at least $\delta_{stop} = \frac{\delta}{poly(n)}$. Moreover, the stationary distribution of* Stop-QW *is close to the distribution of its setup. Namely, we obtain with high probability a random node of the Johnson graph with correctly built data structure.*

With the upcoming NIST standardization for post-quantum cryptography, there is an even stronger need to analyze quantum algorithms for cryptographic problems. There is a strong need to provide more solid theoretical foundations that justify assumptions like Heuristic 2, since cryptographic parameter selections will be based on best quantum attacks. Hence, any progress in proving Heuristic 2 finds a broad spectrum of applications in the cryptographic community.

5 Results

In this section, we describe the BCJ algorithm enhanced by a quantum random walk, see Algorithm 2. Our following main theorem shows the correctness of our quantum version of the BCJ algorithm and how to optimize the parameters for achieving the stated complexity.

▶ **Theorem 8** (BCJ-QW Algorithm)**.** *Under Heuristic 1 and Heuristic 2, Algorithm 2 solves with high probability all but a negligible fraction of random subset sum instances $(\mathbf{a}, t) \in (\mathbb{Z}_{2^{\ell(n)}})^{n+1}$ (as defined in Definition 1) in time and memory $2^{0.226n}$.*

Proof. By Theorem 7, the running time T of Algorithm 2 can be expressed as

$$T = T_S + \frac{1}{\sqrt{\epsilon_{stop}}} \left(T_C + \frac{1}{\sqrt{\delta_{stop}}} T_U \right).$$

We recall from Heuristic 2, Eq. (2), (3) and (6)

$$\epsilon_{stop} \approx \epsilon = \left(\frac{|U_4|}{|L_4|} \right)^{16}, \ \delta_{stop} \approx \delta = \Omega \left(\frac{1}{|U_4|} \right) \text{ and } T_C = 1,$$

where the $\approx$-notation suppresses polynomial factors.

Algorithm 2: BCJ-QW ALGORITHM.

```
Input      : $(\mathbf{a}, t) \in (\mathbb{Z}_{2^{\ell(n)}})^{n+1}, \beta \in [0,1]$
Output     : $\mathbf{e} \in D^n[0,\beta]$
Parameters : Optimize $\alpha_1, \alpha_2, \alpha_3$.
Construct all level-4 lists $E_4^{(j)}$ and $S_4^{(j)}$ for $j = 1, \dots, 16$.    ▷ SETUP (see Eq. (7))
Construct all level-3 lists $E_3^{(j)}$ and $S_3^{(j)}$ for $j = 1, \dots, 8$.
Construct all level-2 lists $E_2^{(j)}$ and $S_2^{(j)}$ for $j = 1, \dots, 4$.
Construct all level-1 lists $E_1^{(j)}$ and $S_1^{(j)}$ for $j = 1, 2$.
Construct level-0 list $E_0$.
while $E_0 \neq \emptyset$ do                                                     ▷ CHECK
    for $1/\sqrt{\delta}$ times (via phase estimation) do
        Take a quantum step of the walk.                                          ▷ UPDATE
        Update the data structure accordingly, stop after $\kappa = \mathrm{poly}(n)$ steps.
    end
end
Output $\mathbf{e} \in E_0$.
```

Let us first find an optimal size of $|U_4|$. Plugging ϵ, δ and T_C into T and neglecting constants yields run time

$$T = T_S + |L_4|^8 |U_4|^{-15/2} T_U.$$

Let us substitute T_U by its expectation $\mathbb{E}[T_U]$. We later show that T_U and $\mathbb{E}[T_U]$ differ by only a polynomial factor, and thus do not change the analysis. We can upper bound the right hand side using our bounds $\tilde{T}_S \geq T_S$, $\tilde{T}_U \geq \mathbb{E}[T_U]$ from Eq. (5) and (9). We minimize the resulting term by equating both summands

$$\tilde{T}_S = |L_4|^8 |U_4|^{-15/2} \tilde{T}_U.$$

Using the relation $\tilde{T}_S = |U_4| \cdot \tilde{T}_U$ from Eq. (10) results in

$$|U_4| = |L_4|^{16/17}.$$

Therefore, $|L_4|^8 |U_4|^{-15/2} \cdot \mathbb{E}[T_U] = |U_4| \cdot \mathbb{E}[T_U]$. Thus for minimizing the runtime T of Algorithm 2, we have to minimize the term $\max\{T_S, |U_4| \cdot \mathbb{E}[T_U]\}$, which equals T up to a factor of at most 2. Recall from Eq. (4), which holds under Heuristic 1 and for all but a negligible fraction of instances, and Eq. (8) that

$$\begin{aligned} T_S &= \max\left\{|U_4|, \frac{|U_4|^2}{2^{r_3}}, \frac{\mathbb{E}(|U_3|)^2}{2^{r_2-r_3}}, \frac{\mathbb{E}(|U_2|)^2}{2^{r_1-r_2}}, \frac{\mathbb{E}(|U_1|)^2}{2^{n-r_1}}\right\}, \\ \mathbb{E}[T_U] &= \max\left\{1, \frac{|U_4|}{2^{r_3}}, \frac{|U_4|\mathbb{E}(|U_3|)}{2^{r_2}}, \frac{|U_4|\mathbb{E}(|U_3|)\mathbb{E}(|U_2|)}{2^{r_1}}, \frac{|U_4|\mathbb{E}(|U_3|)\mathbb{E}(|U_2|)\mathbb{E}(|U_1|)}{2^{n}}\right\}. \end{aligned}$$

Numerical optimization for minimizing $\max\{T_S, |U_4| \cdot \mathbb{E}[T_U]\}$ leads to parameters

$$\alpha_1 = 0.0120, \ \alpha_2 = 0.0181, \ \alpha_3 = 0.0125.$$

This gives

$$2^{r_3} = 2^{0.2259n}, \ 2^{r_2} = 2^{0.4518n}, \ 2^{r_1} = 2^{0.6627n} \text{ representations},$$

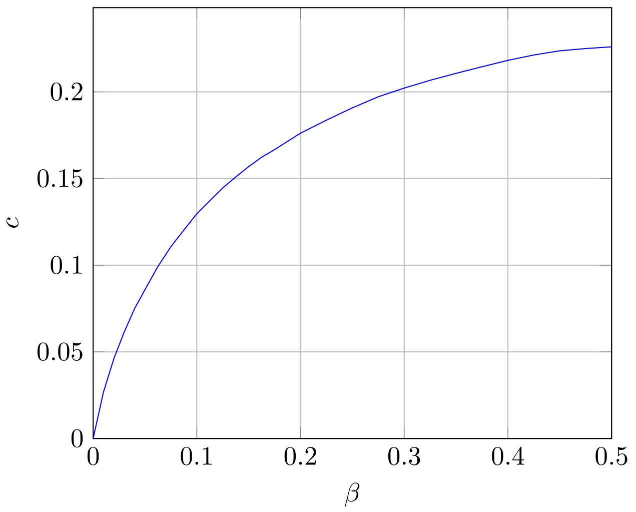


Figure 2 $c = \frac{\log T}{n}$ as a function of β for BCJ-QW.

which in turn yield expected list sizes

$$|U_4| = 2^{0.2259n},\ \mathbb{E}(|U_3|) = 2^{0.2259n},\ \mathbb{E}(|U_2|) = 2^{0.2109n},\ \mathbb{E}(|U_1|) = 2^{0.1424n}.$$

Plugging these values into our formulas for T_S, $\mathbb{E}[T_U]$ gives

$$\begin{aligned} T_S &= \max\{2^{0.2259n}, 2^{0.2259n}, 2^{0.2259n}, 2^{0.2109n}, 2^{-0.0524n}\} \text{ and} \\ |U_4| \cdot \mathbb{E}[T_U] &= \max\{2^{0.2259n}, 2^{0.2259n}, 2^{0.2259n}, 2^{0.2259n}, 2^{0.0310n}\}. \end{aligned}$$

It follows that $\mathbb{E}[T_U] = 1$. Since we have $T_U \leq \kappa = \text{poly}(n)$ by definition in Algorithm 2, the values T_U and $\mathbb{E}[T_U]$ differ by only a polynomial factor that we can safely ignore (by rounding up the runtime exponent). Thus, we conclude that Algorithm 2 runs in time $T = 2^{0.226n}$ using $|U_4| = 2^{0.226n}$ memory. ◀

▶ Remark. As in the classical BCJ case, a tree depth of 4 seems to be optimal for BCJ-QW. When analyzing varying depths, we could not improve over the run time from Theorem 8.

Complexity for the unbalanced case

We also analyzed subset sum instances with $t = \sum_{i \in I} a_i$, where $|I| = \beta n$ for arbitrary $\beta \in [0,1]$. Notice that w.l.o.g. we can assume $\beta \leq 1/2$, since for $\beta > 1/2$ we can solve a subset sum instance with target $t' = \sum_{i=1}^{n} a_i - t$. Hence, the complexity graph is symmetric around $\beta = 1/2$. Fig. 2 shows the run time exponent c for our BCJ-QW algorithm with time $T = 2^{cn}$ as a function of β.

References

1 Dorit Aharonov, Andris Ambainis, Julia Kempe, and Umesh Vazirani. Quantum walks on graphs. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 50–59. ACM, 2001.

2 Miklós Ajtai. The shortest vector problem in l2 is np-hard for randomized reductions. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 10–19. ACM, 1998.

3 Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM J. Comput.*, 37(1):210–239, 2007. doi:10.1137/S0097539705447311.

4 Per Austrin, Mikko Koivisto, Petteri Kaski, and Jesper Nederlof. Dense subset sum may be the hardest. *arXiv preprint arXiv:1508.06019*, 2015.

5 Anja Becker, Jean-Sébastien Coron, and Antoine Joux. Improved generic algorithms for hard knapsacks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 364–385. Springer, 2011.

6 Daniel J Bernstein, Stacey Jeffery, Tanja Lange, and Alexander Meurer. Quantum algorithms for the subset-sum problem. In *International Workshop on Post-Quantum Cryptography*, pages 16–33. Springer, 2013.

7 Daniel J. Bernstein, Stacey Jeffery, Tanja Lange, and Alexander Meurer. *Quantum Algorithms for the Subset-Sum Problem*, pages 16–33. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013. doi:10.1007/978-3-642-38616-9_2.

8 Ernest F Brickell. Solving low density knapsacks. In *Advances in Cryptology*, pages 25–37. Springer, 1984.

9 Matthijs J Coster, Brian A LaMacchia, Andrew M Odlyzko, and Claus P Schnorr. An improved low-density subset sum algorithm. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pages 54–67. Springer, 1991.

10 Sebastian Faust, Daniel Masny, and Daniele Venturi. Chosen-ciphertext security from subset sum. In *Public-Key Cryptography–PKC 2016*, pages 35–46. Springer, 2016.

11 Zvi Galil and Oded Margalit. An almost linear-time algorithm for the dense subset-sum problem. *SIAM Journal on Computing*, 20(6):1157–1189, 1991.

12 Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, pages 212–219. ACM, 1996.

13 Ellis Horowitz and Sartaj Sahni. Computing partitions with applications to the knapsack problem. *Journal of the ACM (JACM)*, 21(2):277–292, 1974.

14 Nick Howgrave-Graham and Antoine Joux. New generic algorithms for hard knapsacks. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 235–256. Springer, 2010.

15 Antoine Joux and Jacques Stern. Improving the critical density of the lagarias-odlyzko attack against subset sum problems. In *International Symposium on Fundamentals of Computation Theory*, pages 258–264. Springer, 1991.

16 Ghazal Kachigar and Jean-Pierre Tillich. Quantum information set decoding algorithms. *CoRR*, abs/1703.00263, 2017. URL: http://arxiv.org/abs/1703.00263, arXiv:1703.00263.

17 Jeffrey C Lagarias and Andrew M Odlyzko. Solving low-density subset sum problems. *Journal of the ACM (JACM)*, 32(1):229–246, 1985.

18 Vadim Lyubashevsky, Adriana Palacio, and Gil Segev. Public-key cryptographic primitives provably as secure as subset sum. In *Theory of Cryptography Conference*, pages 382–400. Springer, 2010.

19 Frédéric Magniez, Ashwin Nayak, Jérémie Roland, and Miklos Santha. Search via quantum walk. *SIAM Journal on Computing*, 40(1):142–164, 2011.

20 Andrew M Odlyzko. The rise and fall of knapsack cryptosystems. *Cryptology and computational number theory*, 42:75–88, 1990.

21 Richard Schroeppel and Adi Shamir. A T=O(2ˆn/2). *SIAM journal on Computing*, 10(3):456–464, 1981.

Trading Inverses for an Irrep in the Solovay-Kitaev Theorem

Adam Bouland[1]
Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, Berkeley, CA, USA
adam@csail.mit.edu
https://orcid.org/0000-0002-8556-8337

Māris Ozols[2]
QuSoft and University of Amsterdam, Amsterdam, Netherlands
marozols@gmail.com
https://orcid.org/0000-0002-3238-8594

Abstract

The Solovay-Kitaev theorem states that universal quantum gate sets can be exchanged with low overhead. More specifically, any gate on a fixed number of qudits can be simulated with error ϵ using merely $\mathrm{polylog}(1/\epsilon)$ gates from any finite universal quantum gate set $\mathcal{G}$. One drawback to the theorem is that it requires the gate set $\mathcal{G}$ to be closed under inversion. Here we show that this restriction can be traded for the assumption that $\mathcal{G}$ contains an irreducible representation of any finite group G. This extends recent work of Sardharwalla *et al.* [29], and applies also to gates from the special linear group. Our work can be seen as partial progress towards the long-standing open problem of proving an inverse-free Solovay-Kitaev theorem [16, 23].

2012 ACM Subject Classification Theory of computation → Quantum computation theory

Keywords and phrases Solovay-Kitaev theorem, quantum gate sets, gate set compilation

Digital Object Identifier 10.4230/LIPIcs.TQC.2018.6

Acknowledgements We thank Aram Harrow, Vadym Kliuchnikov, and Zoltán Zimborás for helpful discussions, and Adam Sawicki for pointing us to reference [35].

1 Introduction

Quantum computing promises to solve certain problems exponentially faster than classical computers. For instance, quantum computers can factor integers [34], simulate quantum mechanics [7], and compute certain knot invariants [4] exponentially faster than the best known classical algorithms. The power of quantum computing is formalized using the notion of quantum circuits, in which polynomial number of quantum gates are applied to a standard input state, and the answer to the computational problem is obtained by measuring the final state. This results in the complexity class BQP (see [24, 20] for an introduction).

Each gate in the circuit is a unitary transformation drawn from some finite *gate set* $\mathcal{G}$; it represents elementary quantum operations that can be performed in hardware and which

[1] AB was partially supported by the NSF GRFP under Grant No. 1122374, by the Vannevar Bush Fellowship from the US Department of Defense, by ARO Grant W911NF-12-1-0541, by NSF Grant CCF-1410022, and by the NSF Waterman award under grant number 1249349.

[2] Part of this work was done while MO was at the University of Cambridge where he was supported by a Leverhulme Trust Early Career Fellowship (ECF-2015-256). MO also acknowledges hospitality of MIT where this project was initiated.

13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2018).
Editor: Stacey Jeffery; Article No. 6; pp. 6:1–6:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

may vary between different realizations of quantum computing. Each gate can act on at most some finite number k of quantum systems at a time, where each individual system (or *qudit*) has d levels. A gate set $\mathcal{G}$ is called *universal*[3] if it is capable of approximately generating any quantum transformation on a sufficiently large number of qudits [15].

In general, the computational power of a quantum device may depend on the gate set $\mathcal{G}$ at its disposal. Clearly, if a gate set is not universal, it may have restricted computational power.[4] But *a priori*, the computational power of different universal gate sets could vary as well. This is because universality simply implies that the gates from $\mathcal{G}$ densely generate all unitaries, but it does not specify how quickly one can approximate arbitrary gates.[5]

While BQP consists of those computations that use $\operatorname{poly}(n)$ gates on an n-bit input, the degree of the polynomial for a specific algorithm could in principle depend on the actual gate set used. For example, if we are given an $O(n)$-gate algorithm over some gate set and we want to implement it using another gate set $\mathcal{G}$, we have to compile each gate to accuracy $O(1/n)$ in terms of $\mathcal{G}$. However, if our compiler uses, say, $O(1/\epsilon)$ gates to achieve accuracy $O(\epsilon)$, the total number of gates would become $O(n^2)$. This would be a strange situation for quantum computation, since the runtime of polynomial-time algorithms would be defined only up to polynomial factors. In particular, this would render Grover's speed-up useless.

Fortunately, this is not the case since the *Solovay-Kitaev theorem* [19, 20, 24, 16] (see also [14, 25]) provides a better compiler, so long as the universal gate set $\mathcal{G}$ is closed under inversion. More specifically, this theorem states that any universal gate set $\mathcal{G}$ can be used to simulate any gate U from any other universal gate set to accuracy ϵ using only $\operatorname{polylog}(1/\epsilon)$ gates from $\mathcal{G}$. Furthermore, there is an efficient algorithm, the *Solovay-Kitaev algorithm*, to perform this conversion between the gate sets.

Before formally stating the Solovay-Kitaev theorem, let us make a few remarks. First, we can assume without loss of generality that all gates in $\mathcal{G}$ are single-qudit gates in some fixed dimension d. Indeed, if $\mathcal{G}$ contains multi-qudit gates or if $\mathcal{G}$ becomes universal only on some larger number of qudits, we can simply set the new dimension to be d^k (for a sufficiently large constant k) and replace $\mathcal{G}$ by a larger gate set that consists of the original gates acting on all ordered subsets of k systems. Second, as we are now dealing with a single system, we can replace the universality of $\mathcal{G}$ by a requirement that $\mathcal{G}$ generates a *dense* subgroup of $\mathrm{SU}(d)$ [31]. Third, we can assume that $\mathcal{G}$ is itself a subset of the *special* unitary group $\mathrm{SU}(d)$ rather than $\mathrm{U}(d)$, since the global phase of a quantum gate has no physical effect. In fact, $\mathrm{U}(1)$ actually does *not* satisfy the Solovay-Kitaev theorem, hence the theorem does not hold for $\mathrm{U}(d)$ either, because in general we cannot approximate the elements of $\mathrm{U}(d)$ accurately enough due to their global phase.

With this fine print aside, we are now ready to state the theorem.

▶ **Theorem** (Solovay-Kitaev theorem [16]). *For any fixed $d \geq 2$, if $\mathcal{G} \subset \mathrm{SU}(d)$ is a finite gate set which is **closed under inverses** and densely generates $\mathrm{SU}(d)$, then there is an algorithm which outputs an ϵ-approximation to any $U \in \mathrm{SU}(d)$ using merely $O\big(\log^{3.97}(1/\epsilon)\big)$ elements from $\mathcal{G}$.*

Therefore if one wishes to change the gate set used for a BQP computation (which requires compiling each gate to $1/\operatorname{poly}$ accuracy), a change of gate set only incurs polylogarithmic

[3] This is also known as *physical universality*.

[4] But not always! Some gate sets which are not physically universal are nevertheless capable of universal quantum computing via an encoding; this is known as *encoded universality* [24].

[5] By a simple counting argument, generic unitaries on an n-qubit system require $\tilde{\Omega}(2^n)$ gates to implement (even approximately) irrespectively of the gate set [24, Section 4.5.4].

overhead in the input size n. In particular this implies the runtimes of quantum algorithms based on inverse-closed gate sets are well-defined up to polylog factors in n; an $O(n^c)$ algorithm using one particular universal gate set implies an $\tilde{O}(n^c)$ algorithm using any other (inversion-closed) universal gate set. It also implies that the choice of a particular universal gate set is unimportant for quantum computation; changing between gate sets incurs low overhead.

Given the central importance of the Solovay-Kitaev theorem to quantum computing, prior works have improved the theorem in various directions. For instance, a number of works (see, e.g., [21, 26, 32, 8, 9, 30, 28, 22, 27]) have decreased the overheads of the Solovay-Kitaev theorem for particular inverse-closed gate sets by improving the exponent in the logarithm from 3.97 to 1 (which is optimal) or even by improving the hidden constants in front of the logarithm. Such works are important steps towards making compilation algorithms practically efficient. Additionally prior work has shown a version of the Solovay-Kitaev algorithm for inverse-closed non-unitary matrices [2] and as well more general Lie groups [23]. Note that there is also an information-theoretic non-algorithmic version of the Solovay-Kitaev theorem with exponent 1 for generic inverse-closed gate sets [18]. This has subsequently been extended also to inverse-free gate sets [35].

In this work, rather than improving the overheads of the Solovay-Kitaev theorem, we work towards removing the assumption that the gate set contains inverses of all its gates. This is important for several reasons. First, on a theoretical level it would be surprising if the power of noiseless quantum computers could be gate set dependent. Of course, in the real world one could apply fault-tolerance [3] to allow the use of approximate inverses in place of exact inverses, but it seems strange to have to resort to such a powerful technique to deal with a seemingly minor issue which is easily stated in a noiseless setting. Furthermore, this would not answer the original mathematical question about how fast unitary gate sequences fill the space of all unitaries, since a fault-tolerant implementation corresponds to a completely positive rather than a unitary map (it implements the desired map on an encoded subspace of a larger-dimensional Hilbert space).

Second, an inverse-free Solovay-Kitaev theorem would be very helpful towards classifying the computational power of quantum gate sets. It remains open[6] to prove a classification theorem describing which gate sets $\mathcal{G}$ are capable of universal quantum computing, which are efficiently classically simulable, and which can solve difficult sampling problems like BosonSampling or IQP [1, 12]. A number of recent works have made partial progress on this problem [11, 10]. However, a common bottleneck in these proofs is that they need to invoke the Solovay-Kitaev theorem on various “postselection gadgets” to argue that one can perform hard sampling problems, and the set of these gadgets is not necessarily closed under inversion. In the above works this problem is tackled on an ad-hoc, case specific basis. An inverse-free Solovay-Kitaev theorem would simplify these proofs and expand the frontier for gate classification.

Finally, such theorem would enable further progress in quantum Hamiltonian complexity where universal gate sets are used to encode computational instructions by the interaction terms of local Hamiltonians. The ground states of such Hamiltonians have very complicated structure and computing their ground energy is typically $\mathsf{QMA}_{\mathsf{EXP}}$-complete [17], a phenomenon that can occur even when the local dimension of each individual subsystem is relatively small [5, 6]. Since low local dimension is physically more relevant, it is desirable to minimize the dimensions of these constructions even further. A significant roadblock in this is

[6] Even for the case of two-qubit gate sets [15, 11]!

the size of the universal gate set used to encode the computation. Since each gate contributes additional dimensions, one would like to have as few gates as possible. Considering how intricate and hard to optimize the known constructions [5, 6] are, getting rid of inverses would be an easy way forward.

For the reasons outlined above, we believe this longstanding open problem (noted in [16, 23]) is an important one to resolve. In this work, we make partial progress towards this goal by replacing the inverse-closedness assumption with the requirement that the gate set contains any (projective) irreducible representation (a.k.a. *irrep*) of a finite group. Roughly speaking, a *projective irrep* is a set of unitary matrices that form a group (up to a global phase) and that do not leave any non-trivial subspace invariant. A canonical example is the set of *Pauli matrices* $\{I, X, Y, Z\}$.

▶ **Theorem 1** (Solovay-Kitaev theorem with an irrep instead of inverses)**.** *For any fixed $d \geq 2$, suppose $\mathcal{G} \subset \mathrm{SU}(d)$ is a finite gate set which densely generates $\mathrm{SU}(d)$, and furthermore $\mathcal{G}$ contains a (projective) irrep of some finite group G. Then there is an algorithm which outputs an ϵ-approximation to any $U \in \mathrm{SU}(d)$ using merely $O(\mathrm{polylog}(1/\epsilon))$ elements from $\mathcal{G}$.*

In other words, the inverses of some of the gates of $\mathcal{G}$—namely those which constitute an irrep of G—are also in $\mathcal{G}$, but the inverses of the remaining gates may not be in $\mathcal{G}$. So we are trading inverses for some other structure in the gate set $\mathcal{G}$. This extends recent work of Sardharwalla, Cubitt, Harrow and Linden [29] which proved this theorem in the special case that G is the Weyl (or generalized Pauli)[7] group. Sardharwalla *et al.*'s result has already found application in gate set classification [10]. We therefore expect that our result will likewise enable further progress on the gate set classification problem. We also extend our theorem to the non-unitary case (see Theorem 4 in Appendix C), thus generalizing the (inverse-closed) non-unitary Solovay-Kitaev theorem of [2] (this may further extend to more general Lie groups as well following [23]). We expect that this version of the theorem will be particularly useful in gate classification as postselection gadgets are often non-unitary [11].

1.1 Proof techniques

Our proof works in a similar manner to those in [16, 29]. The basic idea is to take an ϵ_0-approximation V of some gate U and improve it to an $O(\epsilon_0^2)$-approximation of U, while taking the length of the approximation from ℓ_0 to $c\ell_0$ for some constant c. Iterating this improvement step allows one to obtain a polylogarithmic overhead for compilation.[8] The key in any proof of a Solovay-Kitaev theorem is to make use of V in this construction in such a way that one does not incur $O(\epsilon_0)$ error in the resulting approximation, as one would naively have from the triangle inequality. In other words, one needs the error in the approximation of U to cancel out to lowest order in ϵ_0.

In the proof of the regular (inverse-closed) Solovay-Kitaev theorem, this is achieved using group commutators [16], which manifestly require inverses in the gate set. Sardharwalla *et al.* [29] instead achieve this by applying a group averaging function over the Weyl group. They show by direct computation that the lowest order error term in ϵ cancels out (at least in a neighborhood of the identity).

[7] Note the Weyl operators only form a group up to global phase, but as we only require a *projective irrep* they meet the criteria of our theorem.

[8] One can easily see the lengths of the gate sequences increase exponentially with each application of this operation, while the error decreases doubly exponentially, which implies the desired polylog dependence of the error.

In our proof, we also consider a group averaging function $f : \mathrm{SU}(d) \to \mathrm{SU}(d)$ based on some (projective) irrep $R : G \to \mathrm{SU}(d)$ of a finite group G:

$$f(W) := \prod_{g \in G} R(g) W R(g)^\dagger. \tag{1}$$

Our main technical contribution consists in showing that the lowest order error term cancels here, due to certain orthogonality relations obeyed by irreducible representations. We show this follows from the fact that the multiplicity of the trivial irrep in the adjoint action of any irrep is one. Therefore our proof both shows that efficient compilation can occur with a wider family of gate sets than was previously known, and also explains the mathematical reason that Sardharwalla *et al.*'s proof works as it does.

2 Proof of the main result

To aid the understanding of our main result, let us first briefly define the relevant notions from representation theory (see [33, 13] for further details).

A d-dimensional *representation* of a group G is a map $R : G \to \mathrm{U}(d)$ such that $R(g_1)R(g_2) = R(g_1 g_2)$ for all $g_2, g_2 \in G$. Similarly, R is a *projective representation* if it obeys this identity up to a global phase, i.e. $R(g_1)R(g_2) = e^{i\theta(g_1,g_2)} R(g_1 g_2)$ for some function $\theta : G \times G \to \mathbb{R}$. A representation R is *reducible* if there is a unitary map $U \in \mathrm{U}(d)$ and two other representations R_1 and R_2 of G such that $U R(g) U^\dagger = R_1(g) \oplus R_2(g)$ for all $g \in G$. If this is not the case, R is called *irreducible* (or *irrep* for short). Finally, if $A \subset B$ are two sets, we say that A is *dense* in B if for any $\varepsilon > 0$ and any $b \in B$ there exists $a \in A$ such that $\|a - b\| \leq \varepsilon$ for some suitable notion of distance $\|\cdot\|$.

▶ **Theorem 1** (Solovay-Kitaev theorem with an irrep instead of inverses). *For any fixed $d \geq 2$, suppose $\mathcal{G} \subset \mathrm{SU}(d)$ is a finite gate set which densely generates $\mathrm{SU}(d)$, and furthermore $\mathcal{G}$ contains a (projective) irrep of some finite group G. Then there is an algorithm which outputs an ϵ-approximation to any $U \in \mathrm{SU}(d)$ using merely $O(\mathrm{polylog}(1/\epsilon))$ elements from $\mathcal{G}$.*

Proof. By assumption, our gate set is of the form

$$\mathcal{G} := R(G) \cup \{U_1, U_2, \ldots, U_N\} \tag{2}$$

where $R(G) := \{R(g) : g \in G\}$ and $N \geq 0$ is some integer. Here

- $R : G \to \mathrm{SU}(d)$ is a projective irreducible representation of some finite group G,
- $U_i \in \mathrm{SU}(d)$ are some additional elements whose inverses $U_i^\dagger$ are not necessarily in $\mathcal{G}$.

For the sake of simplicity, we will assume that R is an actual irrep rather than a projective irrep (we describe how to generalize the proof to projective irreps in Appendix B). Note that by the $\mathcal{G} \subset \mathrm{SU}(d)$ assumption we implicitly require that the representation R is in $\mathrm{SU}(d)$ rather than in $\mathrm{U}(d)$. While many irreps are ruled out by this restriction, one can deal with such irreps by first converting them to projective irreps and then applying the techniques discussed in Appendix B. We divide the rest of the proof into several steps marked as below.

Original gate sequence. Given a gate $U \in \mathrm{SU}(d)$ which we wish to approximate to accuracy ϵ, we first run the usual Solovay-Kitaev algorithm (see Section 1) to obtain a sequence $S_{\epsilon/2}$ of gates whose product $\epsilon/2$-approximates U, using elements from $\mathcal{G}$ *and* their inverses. This sequence contains both elements from the set $R(G)$ (which is closed under inversion), as well as gates U_i and $U_i^\dagger$. All of these are in the gate set $\mathcal{G}$ except the $U_i^\dagger$—and there are only $O\big(\log^{3.97}(1/\epsilon)\big)$ many of these. To prove Theorem 1, it therefore suffices to give a

Solovay-Kitaev algorithm for approximating the $U_i^\dagger$ in terms of a sequence of $O(\text{polylog}(1/\epsilon))$ gates from the set $\mathcal{G}$.

More concretely, assume we show how to ϵ-approximate each $U_i^\dagger$ using $O(\log^c(1/\epsilon))$ gates from $\mathcal{G}$ for some constant $c > 0$. Then we can set $\epsilon' := \frac{\epsilon}{2}/O\big(\log^{3.97}(1/\epsilon)\big)$ and run this algorithm to ϵ'-approximate each $U_i^\dagger$ appearing in the sequence $S_{\epsilon/2}$ produced by the regular Solovay-Kitaev algorithm. If we substitute these approximations of $U_i^\dagger$ back into $S_{\epsilon/2}$, by the triangle inequality the existing error of $\epsilon/2$ in $S_{\epsilon/2}$ will be increased by another $\epsilon/2$ contributed jointly by all $U_i^\dagger$'s. These two contributions together give us the desired ϵ-approximation of U. Note that an ϵ'-approximation of $U_i^\dagger$ requires $O\big(\log^c(1/\epsilon)\big)$ gates.[9] Hence the ϵ-approximation to U in total will use $O\big(\log^{c+3.97}(1/\epsilon)\big)$ gates from $\mathcal{G}$.

Initial approximation of $U_i^\dagger$. Since $\mathcal{G}$ generates a dense subgroup of $\mathrm{SU}(d)$, there exists a finite length ℓ_0 such that length-ℓ_0 sequences of elements of $\mathcal{G}$ are ϵ_0-dense in $\mathrm{SU}(d)$, for a small fixed constant

$$\epsilon_0 := \frac{1}{6|G|(d-1)! + 2|G|^2}. \tag{3}$$

Let us pick among these sequences an initial ϵ_0-approximation of $U_i^\dagger$ and denote it by V. Then

$$\epsilon_0 \geq \|V - U_i^\dagger\| = \|VU_i - I\|, \tag{4}$$

where $\|\cdot\|$ denotes the *operator norm* which is unitarily invariant.

Symmetrization. Now consider the operator f on $\mathrm{SU}(d)$ defined by

$$f(W) := \prod_{g\in G} R(g)WR(g)^\dagger, \tag{5}$$

where the order of the products is taken arbitrarily, as long as the last (rightmost) element of the product corresponds to the identity element $e \in G$. We are interested in the action of f on VU_i. If we denote the difference in eq. (4) by $\mathcal{O} := VU_i - I$ and distribute the product in eq. (5) into several sums (with no $\mathcal{O}$'s, with a single copy of $\mathcal{O}$, two copies of $\mathcal{O}$, etc.), we get

$$f(VU_i) = \prod_{g\in G} R(g)(I+\mathcal{O})R(g)^\dagger \tag{6}$$

$$= I + \sum_{g\in G} R(g)\mathcal{O}R(g)^\dagger + \sum_{\substack{g,g'\in G\\ g<g'}} R(g)\mathcal{O}R(g)^\dagger R(g')\mathcal{O}R(g')^\dagger \tag{7}$$

$$+ \cdots + \prod_{g\in G} R(g)\mathcal{O}R(g)^\dagger, \tag{8}$$

where the order of terms in all products is inherited from eq. (5) and $g < g'$ refers to this order. Note that the number of terms with k copies of $\mathcal{O}$ is $\binom{|G|}{k}$.

If one were to naively apply the triangle inequality to this sum, one would obtain that

$$\|f(VU_i) - I\| \leq |G|\|\mathcal{O}\| + \binom{|G|}{2}\|\mathcal{O}\|^2 + \dots \tag{9}$$

[9] One can easily see that $\log^c\big(\frac{\log^{3.97}(1/\epsilon)}{\epsilon}\big) = O\big(\log^c(1/\epsilon)\big)$ as the additional $\log^{3.97}(1/\epsilon)$ factor only adds lower order $\log\log(1/\epsilon)$ terms.

In other words, one would get that we have moved $f(VU_i)$ further from the identity than we started. To fix this, we will show that the first term of the above is actually much smaller—of order $\|\mathcal{O}\|^2$—and therefore our application of f has moved us closer to the identity. To see this, first note that using representation theory, one can show that the norm of the first-order term in eq. (8) is

$$\left\|\sum_{g\in G} R(g)\mathcal{O}R(g)^\dagger\right\| = \left\||G|\frac{\operatorname{Tr}\mathcal{O}}{d}I\right\| \tag{10}$$

$$= |G|\frac{|\operatorname{Tr}(VU_i - I)|}{d}. \tag{11}$$

In other words, the traceless component of the first order term vanishes. This follows from certain orthogonality relations obeyed by irreps, and is proven in Claim 2 in Appendix A.

Next, we show that the trace of $\mathcal{O} = VU_i - I$ is small compared to its norm, namely

$$|\operatorname{Tr}(VU_i - I)| \le (2^d + d!)\|VU_i - I\|^2. \tag{12}$$

This is proven in Claim 3 in Appendix A, and follows essentially because the Lie algebra of the special unitary group is traceless. Plugging this in to eq. (11), we see that

$$\left\|\sum_{g\in G} R(g)\mathcal{O}R(g)^\dagger\right\| \le |G|\frac{2^d + d!}{d}\|VU_i - I\|^2 \tag{13}$$

$$\le |G|\frac{2^d + d!}{d}\epsilon_0^2 \tag{14}$$

where we used Claim 3 to get the first inequality and eq. (4) to get the second.

Hence, by applying these results and then applying the triangle inequality to eq. (8) we get

$$\|f(VU_i) - I\| \le |G|\frac{2^d + d!}{d}\epsilon_0^2 + \sum_{k=2}^{|G|}\epsilon_0^k\binom{|G|}{k} \tag{15}$$

$$\le \left(|G|\frac{2^d + d!}{d} + \frac{|G|^2}{2} + |G|^2\sum_{k=1}^{|G|-2}\epsilon_0^k|G|^k\right)\epsilon_0^2 \tag{16}$$

$$\le \left(|G|\frac{2^d + d!}{d} + \frac{|G|^2}{2} + \frac{|G|^2}{2}\right)\epsilon_0^2 \tag{17}$$

Where in eq. (16) we used the fact that $\binom{|G|}{2} \le \frac{|G|^2}{2}$ and $\binom{|G|}{k} \le |G|^k$, and in eq. (17) we used the fact that $\epsilon_0 < \frac{1}{2|G|^2}$, so since $|G| > 2$ (as G has an irrep of dimension at least 2), we have that $\epsilon_0|G| \le 1/4$ so the geometric sum converges to a quantity $\le \frac{1}{2}$.

Replacing this with a crude upper bound that $2^d \le 2d!$ for $d > 1$, we get that

$$\|f(VU_i) - I\| \le \big(3|G|(d-1)! + |G|^2\big)\epsilon_0^2 =: \epsilon_1 \tag{18}$$

Since we chose ϵ_0 to be $\frac{1}{2(3|G|(d-1)!+|G|^2)}$ in eq. (3), $\epsilon_1 \le \frac{\epsilon_0}{2}$ – in other words $f(VU_i)$ is closer to the identity than VU_i.

Multiplying $f(VU_i) - I$ in eq. (18) by $U_i^\dagger$ on the right, we have that $f(VU_i)U_i^\dagger$ is an ϵ_1-approximation to $U_i^\dagger$. We chose the identity to come last in the definition of f in eq. (5), so the string of operators $f(VU_i)$ has the form

$$f(VU_i) = R(g_1)VU_iR(g_1)^\dagger R(g_2)VU_iR(g_2)^\dagger \cdots VU_i. \tag{19}$$

Since $U_iU_i^\dagger$ cancels at the end, $f(VU_i)U_i^\dagger$ is an ϵ_1-approximation to $U_i^\dagger$ using only terms from $\mathcal{G}$.

Iterative refinement. To complete the proof, we iterate this construction by considering

$$f^{(k)}(VU_i) := f(f(\cdots f(VU_i))). \tag{20}$$

Note from eq. (18) that $f^{(k)}(VU_i)U_i^\dagger$ is an ϵ_k-approximation to $U_i^\dagger$, where $\epsilon_k \leq (3|G|(d-1)! + |G|^2)\epsilon_{k-1}^2$. The length of the sequence $f^{(k)}$, denoted ℓ_k, obeys $\ell_k = |G|\ell_{k-1} + 2|G|$. Again $f^{(k)}(VU_i)U_i^\dagger$ can be expressed only in terms of elements of $\mathcal{G}$ (since the last U_i in the expansion of $f^{(k)}(VU_i)$ cancels with the rightmost $U_i^\dagger$ as before). One can easily show that these recurrence relations imply that as k grows:

- the approximation error ϵ_k shrinks doubly exponentially: $\epsilon_k \leq \frac{2\epsilon_0}{2^{2^k}}$;
- the length of the sequence ℓ_k grows exponentially: $\ell_k = O(|G|^k\ell_0)$.

Note that this sort of asymptotic behavior occurs simply because $\epsilon_k = O(\epsilon_{k-1}^2)$ while $\ell_k = O(\ell_{k-1})$ (though of course the value of ϵ_0 used in the recurrence may depend on the hidden constant in the big-O notation). This immediately implies that one can approximate $U_i^\dagger$ to accuracy ϵ with merely polylog overhead, as desired. More specifically, such approximation uses

$$O\big(\ell_0 \log^{\log_2|G|}(1/\epsilon)\big) \tag{21}$$

elements of $\mathcal{G}$. By our analysis at the beginning of the proof, this gives a Solovay-Kitaev theorem with an exponent of $\log_2|G|+3.97$ in the polylog, completing the proof of Theorem 1. ◀

We have therefore shown that one can ϵ-approximate any $U_i^\dagger$ using only gates from our gate set $\mathcal{G}$ using merely polylog$(1/\epsilon)$ gates. The exponent of the polylog for approximating each $U_i^\dagger$ is again easily computed to be $O(\log_2|G|)$. So putting this all together, our approximation for the overall unitary U requires

$$O\big(\log^{\log_2|G|}(1/\epsilon)\big) \tag{22}$$

gates from $\mathcal{G}$. Note that the dependence on dimension d and order of the group G is hidden in the big-O notation, which hides a factor of ℓ_0, the length of sequences required to achieve an initial ϵ_0-net of SU(d). By a volume argument $\ell_0 = \Omega(d^2)$ [16]. In fact our choice of ϵ_0 implies that $\ell_0 = \Omega(d^3\log d)$ in our construction.[10]

2.1 Extensions of our theorem

We have shown a Solovay-Kitaev theorem for any gate set $\mathcal{G}$ that contains an irrep of a finite group G, without requiring $\mathcal{G}$ to be inverse-closed. Our result can be easily generalized in two directions.

First, our proof also works if instead of an irrep we have a *projective* irrep. That is, a map $R : G \to \mathrm{SU}(d)$ such that, for any $g_1, g_2 \in G$,

$$R(g_1)R(g_2) = e^{i\theta(g_1,g_2)}R(g_1g_2) \tag{23}$$

[10] Since an ϵ_0-ball occupies $\Theta(\epsilon_0^{d^2})$ volume in SU(d), $\ell_0 = \Omega\big(d^2\log(1/\epsilon_0)\big)$ [16]. Since we set $\epsilon_0 = (2|G|^2 + 6|G|(d-1)!)^{-1}$ in eq. (3), we have that $\ell_0 = \Omega\big(d^3\log d\big)$ since $\log d!$ scales as $O(d\log d)$ by Stirling's formula.

for some collection of phases[11] $\theta(g_1, g_2) \in [0, 2\pi)$. In such case one still has a Solovay-Kitaev theorem for any universal gate set that includes $R(G)$. For instance, the Pauli matrices $\{I, X, Y, Z\}$ form a projective irrep, but not an irrep (though the matrices $\{\pm 1, \pm i\} \cdot \{I, X, Y, Z\}$ do form an irrep). Since the exponent of the logarithm of our version of the Solovay-Kitaev theorem contains $\log_2 |G|$, this generalization improves the exponent (e.g. using the four Pauli matrices instead of the eight-element Pauli group improves the exponent by an additive 2). We give details on why projective irreps suffice in Appendix B.

Second, we note that our proof can be extended to the *special linear group* $\mathrm{SL}(d, \mathbb{C})$ as well. That is, one can also efficiently compile non-singular matrices, so long as a (projective) irrep is present in a gate set that is universal for $\mathrm{SL}(d, \mathbb{C})$. A Solovay-Kitaev Theorem (with inverses) for the special linear group was first shown by Aharnov, Arad, Eban and Landau [2], who used it to prove that additive approximations to the Tutte polynomial are BQP-hard in many regimes. It was also applied by [11] to the problem of classifying quantum gate sets, where it arose naturally because the "postselection gadgets" used in their proof are non-unitary. For a formal description of the non-unitary version of this theorem, please see Appendix C. Since postselection gadgets are often non-unitary [11], we likewise expect this version of the theorem will be more useful for gate classification problems.

3 Open problems

The main unresolved problem left by our work is to prove a generic inverse-free Solovay-Kitaev theorem, which has been a longstanding open problem [16, 23].

▶ **Conjecture** (Inverse-free Solovay-Kitaev theorem). *For any fixed $d \geq 2$, if $\mathcal{G} \subset \mathrm{SU}(d)$ is a finite gate set which densely generates $\mathrm{SU}(d)$, then there is an algorithm which outputs an ϵ-approximation to any $U \in \mathrm{SU}(d)$ using merely $O(\mathrm{polylog}(1/\epsilon))$ elements from $\mathcal{G}$.*

One can easily see that for any universal gate set (possibly without inverses), one can ϵ-approximate arbitrary unitaries with $O(1/\epsilon)$ overhead. This follows from simply running the Solovay-Kitaev theorem with inverses, and then approximating each inverse $W^\dagger$ with W^k for some integer k (which one can do with $O(1/\epsilon)$ overhead as this is simply composing irrational rotations about a single axis). However current approaches seem to be unable to improve this compilation algorithm from $O(1/\epsilon)$ to $\mathrm{polylog}(1/\epsilon)$. As discussed in Section 1.1, current proofs of the Solovay-Kitaev theorem require a special cancellation of error terms in order to convert an ϵ-approximation of some operator into an $O(\epsilon^2)$-approximation. This cancellation of error terms can be achieved by taking group commutators [16] or, as in this work and [29], it can be achieved by averaging over irreps and using the orthogonality of irreps. However, there is no known technique for achieving this sort of error cancellation without having some structure in the gate set.[12]

Additionally, a natural question is whether the value of ϵ_0 can be improved. This would improve the scaling of our result with dimension. In our result (and in the inverse-closed Solovay-Kitaev Theorem) the big-O notation hides a factor of ℓ_0—the length of the initial sequences required to achieve an ϵ_0-net. In our result ϵ_0 scales as $1/d!$, and hence a volume argument implies $\ell_0 = \Omega(d^3 \log d)$. In contrast the (inverse-closed) Solovay-Kitaev theorem

[11] The quantity $e^{i\theta(g_1,g_2)}$ is also known as a *Schur multiplier* of G.

[12] For example, Zhiyenbayev, Akulin, and Mandilara [36] have recently studied an alternative setting where instead of inverses a certain "isotropic" property of the gates is assumed.

merely requires $\epsilon_0 = \Theta(1)$ resulting in $\ell_0 = \Omega(d^2)$ [16]. It is a natural question if one can improve the value of ϵ_0 and therefore improve dimension dependence of our construction.

A somewhat simpler open problem is whether our theorem can be improved by considering particular orders of the group elements in eq. (5). The function $f(U)$ which we iterate when proving Theorem 1 is defined by averaging over the irrep of G in an arbitrary order; our theorem essentially works because if U is ϵ-close to the identity then $f(U)$ is $O(\epsilon^2)$-close to the identity. However, we have found by direct calculation that for the 2-dimensional irrep of S_3, considering particular orders of the group can lead to the $O(\epsilon^2)$ terms cancelling out as well, leaving only $O(\epsilon^3)$ terms. It is an interesting open problem if these additional cancellations can be generalized to other groups. If so, they would improve the $\log_2 |G|$ in the exponent of the logarithm of our result to $\log_k |G|$, where k is the lowest order remaining error term.

Finally, we note one may be able to extend our results to compilation over more general Lie groups, just as Kuperberg extended the inverse-closed Solovay-Kitaev theorem to arbitrary connected Lie groups whose Lie algebra is perfect [23]. We leave this as an open problem.

A Auxiliary claims

▶ **Claim 2.** *If R is a d-dimensional (projective) irrep of some finite group G and M is any $d \times d$ complex matrix then*

$$\sum_{g\in G} R(g) M R(g)^\dagger = |G| \frac{\operatorname{Tr} M}{d} I. \tag{24}$$

Proof. If R and R' are any two irreps of a finite group G, with dimensions d_R and $d_{R'}$ respectively, their matrix entries obey the following orthogonality relations [33]:

$$\frac{d_R}{|G|} \sum_{g\in G} R(g)_{ij} \overline{R'(g)_{kl}} = \delta_{RR'}\delta_{ik}\delta_{jl}, \qquad \forall i,j \in \{1,\dots,d_R\}, \quad \forall k,l \in \{1,\dots,d_{R'}\}. \tag{25}$$

In particular, if $R = R'$ and we write the matrix entries as $R(g)_{ij} = \langle i|R(g)|j\rangle$ then

$$\frac{d}{|G|} \sum_{g\in G} \langle i|R(g)|j\rangle\langle l|R(g)^\dagger|k\rangle = \delta_{ik}\delta_{jl}, \qquad \forall i,j,k,l \in \{1,\dots,d\} \tag{26}$$

where $d := d_R = d_{R'}$. If we multiply both sides by $|i\rangle\langle k|$ and then sum over i and k, we get

$$\frac{d}{|G|} \sum_{g\in G} R(g)|j\rangle\langle l|R(g)^\dagger = I\delta_{jl}, \qquad \forall j,l \in \{1,\dots,d\}. \tag{27}$$

If $M = \sum_{j,l=1}^d m_{jl}|j\rangle\langle l|$ then by linearity,

$$\frac{d}{|G|} \sum_{g\in G} R(g) M R(g)^\dagger = I \sum_{j,l=1}^{d} m_{jl}\delta_{jl} = I \operatorname{Tr} M, \tag{28}$$

which completes the proof. ◀

Another way to see this result is by noticing that the adjoint action of R decomposes as a direct sum of the trivial representation (acting on the 1-dimensional space spanned by the identity matrix) and a (d^2-1)-dimensional representation without any trivial component. This follows from Schur's first lemma. The result then follows by the orthogonality relations obeyed by the irrep decomposition of the adjoint action.

▶ **Claim 3.** *If $M \in \mathrm{SL}(d, \mathbb{C})$ then $|\mathrm{Tr}\, M - d| \leq (2^d + d!)\|M - I\|^2$.*

Proof. Let $A := M - I$ and denote the entries of A by a_{ij} where $i, j = 1, \ldots, d$. We know that $1 = \det M = \det(A + I)$, so expanding in terms of the a_{ij}'s, we have that

$$1 = \sum_{\sigma \in \mathrm{S}_d} \mathrm{sgn}(\sigma) \prod_{i=1}^{d} (a_{i\sigma(i)} + \delta_{i\sigma(i)}). \tag{29}$$

Now let us simply take out the term with $\sigma = \varepsilon$, the identity permutation:

$$1 = \prod_{i=1}^{d} (a_{ii} + 1) + \sum_{\sigma \in \mathrm{S}_d \setminus \{\varepsilon\}} \mathrm{sgn}(\sigma) \prod_{i=1}^{d} (a_{i\sigma(i)} + \delta_{i\sigma(i)}). \tag{30}$$

And now expanding the first term we see

$$1 = 1 + \sum_{i=1}^{d} a_{ii} + \sum_{i \neq j} a_{ii} a_{jj} + \cdots + a_{11} a_{22} \cdots a_{dd} + \sum_{\sigma \in \mathrm{S}_d \setminus \{\varepsilon\}} \mathrm{sgn}(\sigma) \prod_{i=1}^{d} (a_{i\sigma(i)} + \delta_{i\sigma(i)}), \tag{31}$$

which implies

$$-\mathrm{Tr}\, A = \sum_{i \neq j} a_{ii} a_{jj} + \cdots + a_{11} a_{22} \cdots a_{dd} + \sum_{\sigma \in \mathrm{S}_d \setminus \{\varepsilon\}} \mathrm{sgn}(\sigma) \prod_{i=1}^{d} (a_{i\sigma(i)} + \delta_{i\sigma(i)}). \tag{32}$$

Now observe that each of the terms on the right hand side is quadratic in the a_{ij}'s—this is because any non-identity permutation displaces at least two items. Let $c \leq 2^d + d!$ denote the number of the terms present, which is constant in any fixed dimension d. Hence we have that

$$|\mathrm{Tr}\, M - d| = |\mathrm{Tr}\, A| \leq c \max_{i,j} |a_{ij}|^2 \leq c\|A\|^2 = c\|M - I\|^2 \tag{33}$$

where we used $|a_{ij}| \leq \|A\|$ in the last inequality (this follows by choosing the j-th standard basis vector in the definition of the operator norm). ◀

Note that this claim, i.e. that elements ϵ-close to the identity have trace substantially smaller than ϵ, is a reflection of the fact that the Lie algebra of the special linear group is traceless.

B Representations vs projective representations

Throughout our proof of Theorem 1, we assumed that R is an irrep of the group G. Here we show that the same construction works also for a projective irrep of G. In other words, even if $R(g_1)R(g_2) = e^{i\theta(g_1,g_2)} R(g_1 g_2)$ for some phase $\theta(g_1, g_2) \in [0, 2\pi)$, our version of the Solovay-Kitaev theorem still holds. As the Weyl operators merely form a projective representation, this allows our result to strictly generalize that of [29]. Intuitively, such generalization is to be expected since global phases are non-physical in quantum theory. We make this precise below.

Suppose that we have a projective representation R of a finite group G. It is convenient to think of $R(G)$ as a subset of the *projective unitary group* $\mathrm{PU}(d)$ that consists of equivalence classes of elements of $\mathrm{U}(d)$ that differ only by global phase. Note that $\mathrm{PU}(d) = \mathrm{PSU}(d)$, the *special* projective unitary group, since $\overline{\det(U)}U \in \mathrm{SU}(d)$ for any $U \in \mathrm{U}(d)$. Now, consider extending the projective representation R in $\mathrm{PSU}(d)$ into a representation[13] R' in $\mathrm{SU}(d)$. Since

$$\mathrm{PSU}(d) = \mathrm{SU}(d)/\mathbb{Z}_d, \tag{34}$$

i.e. the only difference between projective and non-projective representations are factors of $e^{2\pi i/d}I$, this merely increases the size of the group by an integer multiple k which is a divisor of d. Let us denote this larger group by G'.

Now consider applying our proof of Theorem 1 to R' and G'. The corresponding averaging operator is

$$f'(W) := \prod_{g\in G'} R'(g)WR'(g)^\dagger. \tag{35}$$

Our proof essentially uses two facts:

1. The trace of W is small relative to its distance from the identity (Claim 3).
2. The traceless component of W vanishes to lowest order because from Claim 2 we have that for any tracless $\mathcal{O}$,
$$\sum_{g\in G'} R'(g)\mathcal{O}R'(g)^\dagger = 0. \tag{36}$$

Note that if $g, h \in G'$ are such that $R'(g) = e^{i\theta}R'(h)$ for some $\theta \in \mathbb{R}$, then they contribute identical terms in the above sum, since the global phase factors commute through and cancel out. Since the any projectively equivalent group elements g, h contribute the same quantity to the sum, and G' is simply a (projective) k-fold cover of G, this means that we can rewrite eq. (36) as

$$k\sum_{g\in G} R'(g)\mathcal{O}R'(g)^\dagger = 0, \tag{37}$$

where we have simply summed over one representative from each set of projectively equivalent representatives.

Therefore, if we had instead considered averaging over the projective representation only using the original averaging operator (which involves a factor k fewer products),

$$f(W) := \prod_{g\in G} R(g)WR(g)^\dagger, \tag{38}$$

the corresponding sum in eq. (36) (which is the above sum divided by k) would be 0 as well. Therefore, the cancellation of lowest-order terms for the traceless component of the error—i.e. the second fact listed above—still holds. Furthermore, the first fact is true independent of the group G considered, and is simply a fact about matrices of determinant 1 which are close to the identity. Therefore, the proof of Theorem 1 works exactly as before if R is merely a projective representation.

C Extension to the special linear group

In this appendix we describe how to extend our proof of Theorem 1 to the non-unitary case. Namely, we want to approximate some matrix $M \in \mathrm{SL}(d, \mathbb{C})$, our gate set $\mathcal{G} \subset \mathrm{SL}(d, \mathbb{C})$ is

[13] This is known as a *central extension* of the representation.

dense in $\mathrm{SL}(d,\mathbb{C})$, and it contains a (possibly non-unitary) irrep of a finite group G as well as some additional gates $U_i \in \mathrm{SL}(d,\mathbb{C})$.

Let us argue that an ϵ-approximation of M can be obtained using the same algorithm as in the proof of Theorem 1, but with one minor change. Namely, in the first step of the algorithm one must apply the non-unitary Solovay-Kitaev Theorem (with inverses) of Aharonov, Arad, Eban, and Landau [2] rather than the usual unitary Solovay-Kitaev Theorem (with inverses). As before, the problem therefore reduces to finding an expression for the elements U_i^{-1} in terms of $\mathcal{G}$. Note that no other step of our proof requires any matrices to be unitary! Recall that the heart of the proof was in showing that if VU_i is ϵ-close to I then $f(VU_i)$ is $O(\epsilon^2)$-close to I, where V denotes the initial ϵ_0-approximation of U_i^{-1}. The key facts that we used to show this are:

- Claim 2, which states that the traceless component of VU_i vanishes to first order under the application of f due to the orthogonality of irreps.
- Claim 3, which states that matrices of determinant 1 which are ϵ-close to the identity have trace $O(\epsilon^2)$.

Neither of these depends on the matrices involved being unitary—indeed the Schur orthogonality relations between irreps in eq. (25) also hold for non-unitary irreps. Therefore, our proof implies the following:

▶ **Theorem 4.** *For any fixed $d \geq 2$, suppose $\mathcal{G} \subset \mathrm{SL}(d,\mathbb{C})$ is a finite gate set that contains a (projective) irrep of some finite group G. Let $r > 0$ be any fixed radius, let B_r be the ball of radius r about the identity in $\mathrm{SL}(d,\mathbb{C})$, and suppose that $\mathcal{G}$ densely generates all transformations in B_r. Then there is an algorithm which outputs an ϵ-approximation to any $M \in B_r$ using merely $O(\mathrm{polylog}(1/\epsilon))$ elements from $\mathcal{G}$.*

Other than the replacement of $\mathrm{SU}(d)$ with $\mathrm{SL}(d,\mathbb{C})$, the only thing that differs between this theorem and Theorem 1 is the additional restriction that the matrix M we are approximating is a finite distance from the identity (as is present in the non-unitary Solovay-Kitaev theorem of [2] as well). This restriction arises simply because $\mathrm{SL}(d,\mathbb{C})$ is not compact, and approximating elements very far from the identity requires longer sequences of gates. For instance, it requires more applications of the gate $\left(\begin{smallmatrix} 2 & 0 \\ 0 & 1/2 \end{smallmatrix}\right)$ to reach $\begin{pmatrix} 2^{1000} & 0 \\ 0 & 2^{-1000} \end{pmatrix}$ than it requires to reach $\begin{pmatrix} 2^{2} & 0 \\ 0 & 2^{-2} \end{pmatrix}$. Since points arbitrarily far from the identity require arbitrarily long gate sequences to approximate, one cannot upper bound the length of sequences required to ϵ-approximate arbitrary $M \in \mathrm{SL}(d,\mathbb{C})$ as a function of ϵ only—rather the length would depend on the distance of M to the identity as well. Restricting M's distance to the identity allows one to upper bound the length of the approximating sequence in terms of ϵ only.

References

1 Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 333–342. ACM, 2011. doi:10.1145/1993636.1993682.

2 Dorit Aharonov, Itai Arad, Elad Eban, and Zeph Landau. Polynomial quantum algorithms for additive approximations of the Potts model and other points of the Tutte plane, 2007. arXiv:quant-ph/0702008.

3 Dorit Aharonov and Michael Ben-Or. Fault-tolerant quantum computation with constant error rate. *SIAM Journal on Computing*, 38(4):1207–1282, 2008. doi:10.1137/S0097539799359385.

4 Dorit Aharonov, Vaughan Jones, and Zeph Landau. A polynomial quantum algorithm for approximating the Jones polynomial. *Algorithmica*, 55(3):395–421, Nov 2009. doi:10.1007/s00453-008-9168-0.

5 Johannes Bausch, Toby Cubitt, and Maris Ozols. The complexity of translationally invariant spin chains with low local dimension. *Annales Henri Poincaré*, 18(11):3449–3513, Nov 2017. doi:10.1007/s00023-017-0609-7.

6 Johannes Bausch and Stephen Piddock. The complexity of translationally invariant low-dimensional spin lattices 3D. *Journal of Mathematical Physics*, 58(11):111901, 2017. doi:10.1063/1.5011338.

7 Dominic W. Berry, Andrew M. Childs, and Robin Kothari. Hamiltonian simulation with nearly optimal dependence on all parameters. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 792–809. IEEE, Oct 2015. doi:10.1109/FOCS.2015.54.

8 Alex Bocharov, Martin Roetteler, and Krysta M. Svore. Efficient synthesis of probabilistic quantum circuits with fallback. *Phys. Rev. A*, 91(5):052317, May 2015. doi:10.1103/PhysRevA.91.052317.

9 Alex Bocharov, Martin Roetteler, and Krysta M. Svore. Efficient synthesis of universal repeat-until-success quantum circuits. *Phys. Rev. Lett.*, 114(8):080502, Feb 2015. doi:10.1103/PhysRevLett.114.080502.

10 Adam Bouland, Joseph F. Fitzsimons, and Dax E. Koh. Complexity classification of conjugated Clifford circuits. *Proc. 33rd Computational Complexity Conference (CCC)*, 2018. arXiv:1709.01805.

11 Adam Bouland, Laura Mančinska, and Xue Zhang. Complexity classification of two-qubit commuting Hamiltonians. In Ran Raz, editor, *31st Conference on Computational Complexity (CCC 2016)*, volume 50 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 28:1–28:33, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2016.28.

12 Michael J. Bremner, Richard Jozsa, and Dan J. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 2010. doi:10.1098/rspa.2010.0301.

13 Andrew M. Childs. Fourier analysis in nonabelian groups. Lecture notes at University of Waterloo, 2013. URL: http://www.cs.umd.edu/~amchilds/teaching/w13/l06.pdf.

14 Andrew M. Childs. Lecture notes on quantum algorithms. Lecture notes at University of Maryland, 2017. URL: http://www.cs.umd.edu/~amchilds/qa/qa.pdf.

15 Andrew M. Childs, Debbie Leung, Laura Mančinska, and Maris Ozols. Characterization of universal two-qubit Hamiltonians. *Quantum Information & Computation*, 11(1&2):19–39, Jan 2011. arXiv:1004.1645.

16 Christopher M. Dawson and Michael A. Nielsen. The Solovay-Kitaev algorithm. *Quantum Information & Computation*, 6(1):81–95, Jan 2006. arXiv:quant-ph/0505030.

17 Daniel Gottesman and Sandy Irani. The quantum and classical complexity of translationally invariant tiling and Hamiltonian problems. *Theory of Computing*, 9(2):31–116, 2013. doi:10.4086/toc.2013.v009a002.

18 Aram W. Harrow, Benjamin Recht, and Isaac L. Chuang. Efficient discrete approximations of quantum gates. *Journal of Mathematical Physics*, 43(9):4445–4451, 2002. doi:10.1063/1.1495899.

19 Alexei Yu. Kitaev. Quantum computations: algorithms and error correction. *Russian Mathematical Surveys*, 52(6):1191–1249, 1997. doi:10.1070/RM1997v052n06ABEH002155.

20 Alexei Yu. Kitaev, Alexander Shen, and Mikhail N. Vyalyi. *Classical and Quantum Computation*, volume 47 of *Graduate Studies in Mathematics*. American Mathematical Society, 2002. URL: `https://books.google.com/books?id=qYHTvHPvmG8C`.

21 Vadym Kliuchnikov. Synthesis of unitaries with Clifford+T circuits, 2013. `arXiv:1306.3200`.

22 Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca. Practical approximation of single-qubit unitaries by single-qubit quantum Clifford and T circuits. *IEEE Transactions on Computers*, 65(1):161–172, Jan 2016. `doi:10.1109/TC.2015.2409842`.

23 Greg Kuperberg. How hard is it to approximate the Jones polynomial? *Theory of Computing*, 11(6):183–219, 2015. `doi:10.4086/toc.2015.v011a006`.

24 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010. URL: `https://books.google.com/books?id=-s4DEy7o-a0C`.

25 Māris Ozols. The Solovay-Kitaev theorem. Essay at University of Waterloo, 2009. URL: `http://home.lu.lv/~sd20008/papers/essays/Solovay-Kitaev.pdf`.

26 Adam Paetznick and Krysta M. Svore. Repeat-until-success: Non-deterministic decomposition of single-qubit unitaries. *Quantum Information & Computation*, 14(15&16):1277–1301, 2014. `arXiv:1311.1074`.

27 Ori Parzanchevski and Peter Sarnak. Super-Golden-Gates for PU(2). *Advances in Mathematics*, 327:869–901, 2018. Special volume honoring David Kazhdan. `doi:10.1016/j.aim.2017.06.022`.

28 Neil J. Ross and Peter Selinger. Optimal ancilla-free Clifford+T approximation of Z-rotations. *Quantum Information & Computation*, 16(11&12):0901–0953, 2016. `doi:10.26421/QIC16.11-12`.

29 Imdad S.B. Sardharwalla, Toby S. Cubitt, Aram W. Harrow, and Noah Linden. Universal refocusing of systematic quantum noise, 2016. `arXiv:1602.07963`.

30 Peter Sarnak. Letter to Aaronson and Pollington on the Solvay-Kitaev Theorem and Golden Gates, 2015. URL: `http://publications.ias.edu/sarnak/paper/2637`.

31 Adam Sawicki and Katarzyna Karnas. Universality of single-qudit gates. *Annales Henri Poincaré*, Aug 2017. `doi:10.1007/s00023-017-0604-z`.

32 Peter Selinger. Efficient Clifford+T approximation of single-qubit operators. *Quantum Information & Computation*, 15(1&2):159–180, 2015. `arXiv:1212.6253`.

33 Jean-Pierre Serre. *Linear Representations of Finite Groups*. Springer, 2012. URL: `https://books.google.com/books?id=9mT1BwAAQBAJ`.

34 Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. `doi:10.1137/S0097539795293172`.

35 Péter Pál Varjú. Random walks in compact groups. *Documenta Mathematica*, 18:1137–1175, 2013. `arXiv:1209.1745`.

36 Y. Zhiyenbayev, V. M. Akulin, and A. Mandilara. Quantum compiling with diffusive sets of gates, 2017. `arXiv:1708.08909`.

Two-qubit Stabilizer Circuits with Recovery I: Existence

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

In this paper, we further investigate the many ways of using stabilizer operations to generate a single qubit output from a two-qubit state. In particular, by restricting the input to certain product states, we discover probabilistic operations capable of transforming stabilizer circuit outputs back into stabilizer circuit inputs. These secondary operations are ideally suited for recovery purposes and require only one extra resource input to proceed. As a result of reusing qubits in this manner, we present an alternative to the original state preparation process that can lower the overall costs of executing a two-qubit stabilizer procedure involving non-stabilizer resources.

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.2018.7

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

1 Introduction

There has been significant progress to building quantum computers. We can protect qubits with quantum codes, and we can combat the spread of errors with fault-tolerance; high thresholds approaching 1% [17] is already within reach. Rather, one of the central challenges is in the efficient handling of noise, where it is necessary to strike a delicate balance between quality and cost. Currently many physical qubits are required to achieve this desired level of protection on a logical qubit [10], but this comprises only one part of a larger problem. The fact remains that most fault-tolerant schemes are constrained to a finite number of native operations, so there is a limit to the type of computations that we can perform. This usually consists of stabilizer operations – Clifford group unitaries, Pauli measurements, and ancilla $|0\rangle$ preparation – which are efficiently simulable on classical computers and capable of producing highly entangled states. Unfortunately, stabilizer operations by themselves are not universal, placing a premium on any non-stabilizer resource added to a circuit.

Magic state distillation is one solution addressing this inherent limitation of stabilizer operations [4]. It works as follows: prepare imperfect "magic states," measure certain stabilizer code syndrome operators, then postselect on some target outcome. The process is repeated recursively until the qubits are at a high enough quality to consume: the magic

13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2018).
Editor: Stacey Jeffery; Article No. 7; pp. 7:1–7:15
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

states are injected into quantum circuits to implement quantum gates outside the Clifford group of operations. Despite being quite resource intensive in the early days, numerous proposals over the last few years have progressively increased the efficiency of distilling magic states [3, 7, 8, 12, 19], although the overall format more or less remains the same. Interestingly, stabilizer operations are enough to perform the distillation, which is a testament to their versatility. Then given a supply of non-Clifford gates, we may employ any number of pre-existing synthesis algorithms to approximate unitaries over this basis. Previous work has already succeeded in producing solutions able to generate sequences for single qubit rotations in an optimal fashion [15, 16, 22, 23]. A recent one even suggests a kind of distill-and-synthesis hybrid to reduce resource usage even further: a factor of 3 savings with quadratic error suppression is possible over traditional distill-then-synthesize methods [5, 6].

The creativity that went into designing these distillation protocols is one reason motivating our broader study of stabilizer operations. Other uses include procedures for distilling multiple types of magic qubits [7, 8, 12, 18], as well as implementing phase rotations with low depth circuits. Some notable examples of the latter are contained in [9] and [13], both of which feature the same stabilizer circuit to perform the operation. The differences lie in the pre-computed ancillae injected into the circuit, where Duclos-Cianci and Svore [9] additionally demonstrated how to use the same circuit to create other resource qubits. At any rate, though simple, both displayed the advantages of having a large set of non-homogeneous states at our disposal, and all that is required is a two-qubit stabilizer circuit.

Inspired by the magic state model to universal quantum computation, we consider general two-to-one stabilizer procedures that take a two-qubit state and produce a single qubit output using stabilizer operations only. Our intent is to explore these processes from a different angle, outside the realm of state distillation, and simply examine their behavior on more arbitrary input. And though our problem size is small, we discover some encouraging ideas that are worth pursuing for larger settings. Some limits on distilling two-qubit states are already discussed in [21]. Instead, we refine the implementation details first provided by Reichardt [21] to identify three circuit configurations characterizing all such two-to-one procedures. These three forms suggest that in addition to Pauli measurements and postselection, single qubit Clifford gates and at most one CNOT or SWAP are enough to realize any stabilizer procedure acting on two qubits. When the input set is further confined to certain product states, we discover an interesting connection between stabilizer circuits of the single CNOT variety – "interacting" circuits in our dictionary. That is, there are "recovery circuits" that can recuperate a product state input from a corrupted stabilizer circuit output. Informally our main result (Theorem 12) states the following.

Main Result (informal): *For any interacting two-to-one stabilizer procedure there exist recovery circuits, and all such recovery circuits are equivalent to one-and-another and hence have the same probability of recovery.*

The magic state injection process is one good area for utilizing such a recovery technique. We end the article with a few numerical experiments showcasing the benefits of the derived recovery protocols.

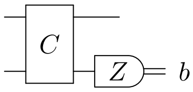


Figure 1 A postselected two-to-one stabilizer circuit (C, b) consists of a stabilizer circuit component C and a postselected bit value b.

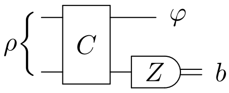


Figure 2 The qubit $\varphi = \Phi_b(C, \rho)$ is the output of a postselected two-to-one stabilizer circuit (C, b) on the two-qubit input ρ.

2 Preliminaries

This section provides an overview of the elementary stabilizer operations and basic concepts. The single qubit Pauli matrices are

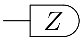
$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{1}$$

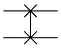
They satisfy not only the identities $X^2 = Y^2 = Z^2 = I$ and $XY = iZ$, but they also form a basis for the space of 2×2 Hermitian matrices. We can expand any single qubit density matrix φ in terms of Pauli matrices using the expression $\varphi = (I + xX + yY + zZ)/2$. If we collect the three previous coefficients, then $(x, y, z) \in \mathbb{R}^3$ is the *Bloch vector* of φ.

An n-qubit stabilizer circuit is limited to certain quantum gates and measurements. It may use elements from the Clifford group $\mathcal{C}(n)$, and it may apply measurements in the Z-basis. The Clifford group is generated by the Controlled-NOT (CNOT), Hadamard (H), and Phase (P) operators:

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

A stabilizer circuit thus contains entirely of CNOT, H, and P gates. For the values of n we are concerned with, $\mathcal{C}(1)$ and $\mathcal{C}(2)$ have sizes 24 and 11520, respectively, modulo global phases. The circuit diagram for a Z-measurement is given by the left image below:





while the right image represents a qubit SWAP. A Clifford circuit is a stabilizer circuit that excludes measurements and implements a Clifford group unitary only.

3 Postselected Two-to-One Stabilizer Circuits

We revisit the study of stabilizer reductions from [21] to derive Lemma 4. Part of the novelty that Lemma 4 brings is the realization of recovery circuits described in the next section. We first introduce some terminology and notation to more concisely capture Reichardt's observations in [21] to present our result.

An n-to-1 stabilizer reduction is a procedure that accepts an n-qubit state and generates a single qubit output using stabilizer operations only. This means all post-measurement activities are also restricted to classical control over stabilizer operations. Reichardt showed that any reduction can be standardized to a particular form: an application of a Clifford unitary on n qubits, followed by a projection of qubits 2 to n onto a computational basis state [21]. Since our focus is on $n = 2$, we have the following definition.

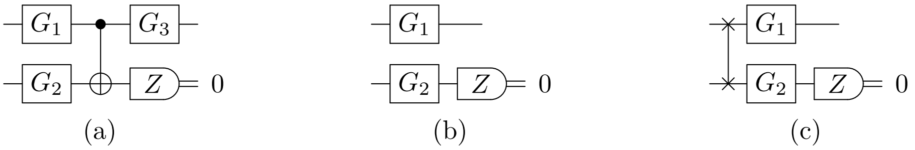


Figure 3 Any stabilizer procedure generating one qubit from two can be described by a postselected circuit (C, b) resembling circuit (a), circuit (b), or circuit (c). The choice of single qubit Clifford gates G_1, G_2, and G_3 depend on C and the postselected measurement b. Circuit (a) is known as an interacting postselected circuit; the precise definition is provided in Section 4.

▶ **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 ρ be a two-qubit state. Then the *probability* Q_b of outcome b on the transformed state $C\rho C^\dagger$ is $Q_b(C, \rho) = \mathrm{Tr}((I \otimes \langle b|)C\rho C^\dagger(I \otimes |b\rangle))$. If $Q_b(C, \rho) > 0$, then the *output* Φ_b of a postselected circuit (C, b) on an input ρ is

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

At times, we may say *run circuit* C, which translates to an application of the unitary C on the input ρ, followed by a Z-measurement on the second qubit. This is often followed by details on what course of action to take conditional on b (or $1 - b$). The term *circuit* C thus references the stabilizer circuit piece only of the postselected circuit, including the measurement at the end. The next definition describes what it means for postselected circuits to produce similar outputs.

▶ **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 ρ, we have the equality

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

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. $Q_{b_1}(C_1, \rho) = Q_{b_2}(C_2, \rho)$. We say two postselected circuits are simply *equivalent*, $(C_1, b_1) \equiv (C_2, b_2)$, if and only if $G = I$ in Equation 4.

We may alter the circuits using $|b_2\rangle = X|1 - b_2\rangle$ in Equation 4 so that both postselect on the same value. As we mentioned before, any two-to-one stabilizer reduction can be achieved through a postselected two-to-one stabilizer circuit. Despite $|\mathcal{C}(2)| = 11520$, the number of actual reductions we need to consider is 30: one for each nontrivial two-qubit Pauli, plus the bit [21]. As such, we can introduce three forms in the following lemma to represent all postselected circuits (C, b). The proof is provided in Appendix A.

▶ **Lemma 4.** *For every postselected two-to-one stabilizer circuit* (C, b)*, there exist single qubit Clifford gates* G_1 *and* G_2 *such that either* $(C, b) \sim (I \otimes G_1, 0)$*, or* $(C, b) \sim ((I \otimes G_1)\mathrm{SWAP}, 0)$*, or* $(C, b) \sim (\mathrm{CNOT}(G_1 \otimes G_2), 0)$.

▶ **Corollary 5.** *If a postselected two-to-one stabilizer circuit* (C, b) *is Clifford equivalent to* $(C', 0)$*, where* $C' = I \otimes G_1$*, or* $C' = (I \otimes G_1)\mathrm{SWAP}$*, or* $C' = \mathrm{CNOT}(G_1 \otimes G_2)$*, and* G_1 *and* G_2 *are single qubit Clifford gates, then* $(C, 1 - b) \sim ((I \otimes X)C', 0)$.

Due to Lemma 4, we have a remarkably much easier time studying postselected circuits. We may substitute (C, b) with another that likely uses fewer gates but behaves in exactly the same way. Because there are many identities on Pauli operators and Clifford gates, G_1 and G_2 are not unique e.g. $((\mathrm{CNOT}(Z \otimes I), 0) \equiv ((Z \otimes I)\mathrm{CNOT}, 0) \sim (\mathrm{CNOT}, 0)$. Of the 30 reductions available, it is easy to see that there are 18 varieties of $(\mathrm{CNOT}(G_1 \otimes G_2), 0)$, and 6 each for $(I \otimes G_1, 0)$ and $((I \otimes G_1)\mathrm{SWAP}, 0)$. If we want to separate the circuits by the stricter kind of equivalence "$\equiv$", the number of classes is multiplied by 24 e.g. $18 \cdot 24 = 432$ for $((G_3 \otimes I)\mathrm{CNOT}(G_1 \otimes G_2), 0)$, since there are $|\mathcal{C}(1)| = 24$ choices of G_3.

4 Recovery Circuits

A quantum circuit involving measurements likely has outcomes that we prefer over others. If we are less than fortunate, convention dictates that we discard the output and rerun the circuit on new input instances until we succeed. This is not much of an issue when the initial overhead is low, but can become problematic otherwise. If the cost associated with state preparation is a barrier to large computations, any method that alleviates this burden is highly desirable. It turns out when ρ is a tensor product state, i.e. $\rho = \varphi \otimes |\psi\rangle\langle\psi|$, we have an alternative: there exist operations capable of reusing an undesirable output to try and recovery φ.

This input requirement means the only circuit configuration of Lemma 4 worth considering is $(\mathrm{CNOT}(G_1 \otimes G_2), 0)$. We can easily see that when $(C, b) \sim (I \otimes G_1, 0)$, the output of (C, b) on $\varphi_1 \otimes \varphi_2$ is essentially φ_1. The output is always an input, and the same is similarly true for all circuits $(C, b) \sim ((I \otimes G_1)\mathrm{SWAP}, 0)$.

▶ **Definition 6** (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 $(C, b) \sim (\mathrm{CNOT}(G_1 \otimes G_2), 0)$. We say circuit C is *interacting* if and only if $(C, 0)$ is interacting.

With that, we define the notion of a recovery circuit. For convenience, we use ψ in place $|\psi\rangle\langle\psi|$ throughout the remainder of our discussion on recovery circuits.

▶ **Definition 7** (recovery circuit). Let (C, b) be an interacting postselected circuit. 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 $\varphi = \Phi_{b'}(C', \Phi_{1-b}(C, \varphi \otimes \psi) \otimes \psi)$.

Notice that an input qubit to (C', b') is the output of $(C, 1-b)$ on $\varphi \otimes \psi$. In this context, if b is more desirable than $1 - b$, then we say circuit C is *successful* upon measuring b on $C\,(\varphi \otimes \psi)\,C^\dagger$. Otherwise circuit C is *unsuccessful*, and the recovery circuit provides a second chance at obtaining the output of (C, b) on $\varphi \otimes \psi$. The presumption is that the implementation of C' is far simpler to pursue than the original method to prepare φ. Our next lemma presents one way on how to design such a recovery circuit to (C, b).

▶ **Lemma 8.** *Every interacting postselected circuit (C, b) has a recovery circuit.*

Proof. Let $(C, b) \sim (\mathrm{CNOT}(G_1 \otimes G), 0)$, where G_1 and G are single qubit Clifford gates. By Corollary 5, we know $(C, 1-b) \sim (\mathrm{CNOT}(G_1 \otimes G), 1)$, which means there is a single qubit Clifford gate G_2 such that $(C, 1-b) \equiv ((G_2^\dagger \otimes I)\mathrm{CNOT}(G_1 \otimes G), 1)$. We shall show that $((G_1^\dagger \otimes I)\mathrm{CNOT}(G_2 \otimes G), 0)$ is a recovery circuit of (C, b). Figure 4 includes reference diagrams to aid comprehension.

If the input to circuit C is $\varphi_1 \otimes \psi$, consider $\varphi_1' \otimes \psi' = G_1 \varphi_1 G_1^\dagger \otimes G \psi G^\dagger$. Let (x_1, y_1, z_1) be the Bloch vector of φ_1' and (x, y, z) be the Bloch vector of $|\psi'\rangle$. For ease of notation, we

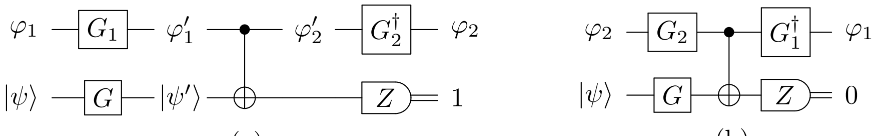


Figure 4 Suppose $(C, 1-b) \equiv ((G_2^\dagger \otimes I)\text{CNOT}(G_1 \otimes G), 1)$. This equivalence allows us to study $(C, 1-b)$ via its substitute in (a) and come up with the recovery circuit in (b). We include intermediate states like φ_1' and $\varphi_2' = G_2\varphi_2 G_2^\dagger$ in (a) to signify stages in the circuit.

define outputs $\varphi_2' = \Phi_1(\text{CNOT}, \varphi_1' \otimes \psi')$ and $\varphi_2 = G_2^\dagger \varphi_2' G_2 = \Phi_{1-b}(C, \varphi_1 \otimes \psi)$. Then the Bloch vector (x_2, y_2, z_2) of φ_2' becomes

$$x_2 = \frac{x_1 x + y_1 y}{1 - z_1 z}, \qquad y_2 = \frac{y_1 x - x_1 y}{1 - z_1 z}, \qquad z_2 = \frac{z_1 - z}{1 - z_1 z}. \tag{5}$$

Now suppose $\varphi_3 = \Phi_0(\text{CNOT}(G_2 \otimes G), \varphi_2 \otimes \psi)$. For postselected circuits with one CNOT, the equations for computing the output's Bloch vector are essentially the same:

$$x_3 = \frac{x_2 x - y_2 y}{1 + z_2 z}, \qquad y_3 = \frac{y_2 x + x_2 y}{1 + z_2 z}, \qquad z_3 = \frac{z_2 + z}{1 + z_2 z}, \tag{6}$$

where (x_3, y_3, z_3) represents the Bloch vector of φ_3. Using $x^2 + y^2 + z^2 = 1$, we can show

$$x_3 = \frac{x_1 x^2 + x y_1 y - x y_1 y + x_1 y^2}{1 - z_1 z + z_1 z - z^2} = x_1. \tag{7}$$

Likewise, $y_3 = y_1$ and $z_3 = z_1$, which means $\varphi_3 = \varphi_1' = G_1 \varphi_1 G_1^\dagger$. The circuit $((G_1^\dagger \otimes I)\text{CNOT}(G_2 \otimes G), 0)$ is therefore a recovery circuit of (C, b). ◀

Between (C, b) and its recovery circuit $((G_1^\dagger \otimes I)\text{CNOT}(G_2 \otimes G), 0)$, there is a relatively straightforward relationship between the probability that circuit C would have been successful and the probability that circuit $C' = (G_1^\dagger \otimes I)\text{CNOT}(G_2 \otimes G)$ will be successful.

▶ **Corollary 9.** *Let $\varphi_1 \otimes \psi$ be a two-qubit state and let $C' = (G_1^\dagger \otimes I)\text{CNOT}(G_2 \otimes G)$ be a two-qubit Clifford unitary such that $(C', 0)$ is a recovery circuit of (C, b). Then*

$$Q_0(C', \Phi_{1-b}(C, \varphi_1 \otimes \psi) \otimes \psi) = \frac{(1 - z^2)/4}{1 - Q_b(C, \varphi_1 \otimes \psi)} \tag{8}$$

where $z = \langle\psi|G^\dagger Z G|\psi\rangle$.

Proof. We assume for simplicity that $C = \text{CNOT}$ and $b = 0$, which implies $G_1 = G_2 = G = I$. Let $z_1 = \text{Tr}(Z\varphi_1)$ and $z = \langle\psi|Z|\psi\rangle$. Also let $\varphi_2 = \Phi_1(C, \varphi_1 \otimes \psi)$. Then

$$Q_1(C, \varphi_1 \otimes \psi) = \frac{1 - z_1 z}{2}, \quad z_2 = \text{Tr}(Z\varphi_2) = \frac{z_1 - z}{1 - z_1 z}. \tag{9}$$

The probability of recovering φ_1 is now clear:

$$Q_0(C', \varphi_2 \otimes \psi) = \frac{1 + z_2 z}{2} = \frac{1 - z_1 z + z_1 z - z^2}{4\left(\frac{1 - z_1 z}{2}\right)} = \frac{(1 - z^2)/4}{1 - Q_0(C, \varphi_1 \otimes \psi)} \tag{10}$$

since the circuits perform a single measurement. ◀

Another implication of the proof to Lemma 8 is that $\Phi_{1-b}(C, \varphi_1 \otimes \psi)$ is always φ_1, up to a single qubit Clifford gate, whenever $|\psi\rangle$ is an eigenstate of X, Y, or Z (a stabilizer qubit). Under these circumstances, the behavior of (C, b) on these types of inputs is actually no different than non-interacting circuits. Hence it does not warrant the use of a circuit $C' = (G_1^\dagger \otimes I)\text{CNOT}(G_2 \otimes G)$ to try and perform a recovery because the qubit is basically φ_1. It is also quite evident by now that there is only one type of recovery circuit, especially given our construction in Lemma 8.

▶ **Lemma 10.** *All recovery circuits are interacting postselected circuits.*

Proof. Let (C, b) be an interacting postselected circuit and suppose (C', b') is a recovery circuit of (C, b). If (C', b') is not an interacting postselected circuit, then $(C', b') \sim (I \otimes G, 0)$ or $(C', b') \sim ((I \otimes G)\text{SWAP}, 0)$, where G is a single qubit Clifford gate. We can easily find a two-qubit state $\varphi \otimes \psi$ such that (C', b') fails to recover φ on the input $\Phi_{1-b}(C, \varphi \otimes \psi) \otimes \psi$. ◀

Lastly, it should not come as a surprise that more than one recovery circuit of (C, b) exists. Even so, we can guarantee that not any one recovery circuit will outperform another.

▶ **Lemma 11.** *Let (C, b) be an interacting postselected circuit, and let $C'' = (G_2^\dagger \otimes I)\text{CNOT}(G_1 \otimes G)$ be a two-qubit Clifford unitary such that $(C, 1-b) \equiv (C'', 1)$. Then (C', b') is a recovery circuit of (C, b) if and only if $(C', b') \equiv ((G_1^\dagger \otimes I)\text{CNOT}(G_2 \otimes G), 0)$.*

Proof. In the reverse direction, equivalence of postselected stabilizer circuits means both produce the exact same output at the same success rate for all two-qubit states ρ. This certainly includes all two-qubit product states $\varphi_2 \otimes \psi$, where φ_2 is the output of $(C, 1-b)$ on another input $\varphi_1 \otimes \psi$.

In the forward direction, Lemmas 14 and 15 in the appendices do most of the job: $(C', b') \sim ((G_1^\dagger \otimes I)\text{CNOT}(G_2 \otimes G), 0)$. We just need to prove equivalence. We look back at the definition of Clifford equivalent postselected circuits, where we must have a single qubit Clifford gate G' such that

$$\begin{aligned}(G' \otimes \langle b'|)C'\rho C'^\dagger(G'^\dagger \otimes |b'\rangle) = \\ (G_1^\dagger \otimes \langle 0|)\text{CNOT}(G_2 \otimes G)\rho(G_2^\dagger \otimes G^\dagger)\text{CNOT}(G_1 \otimes |0\rangle) \qquad (11)\end{aligned}$$

for all two-qubit states ρ. If it is indeed the case that they are strictly Clifford equivalent i.e. $G' \neq I$, then (C', b') cannot have been a recovery circuit of (C, b) because the output from (C', b') on ρ will be rotated by $G'^\dagger$. Thus the two must be equivalent (with "≡"). ◀

From Lemmas 8 and 11, we reach our main result, with Corollary 13 as an immediate consequence to our theorem.

▶ **Theorem 12.** *Every interacting postselected circuit (C, b) has a recovery circuit (C', b'). Moreover, all recovery circuits of (C, b) are equivalent to (C', b').*

▶ **Corollary 13.** *Every recovery circuit (C', b') has its own recovery circuit (C'', b'').*

5 Example Routines Featuring Recovery Circuits

Recovery circuits appear in the literature, where the use cases for our recovery operation seem more pertinent to state injection and implementing non-Clifford operations than to state distillation itself. For instance, the programmable ancilla rotation (PAR) of [13] uses qubits of the type $|\theta\rangle = (|0\rangle + e^{i\theta}|1\rangle)/\sqrt{2}$ and an interacting circuit CNOT to rotate $|q\rangle = \alpha|0\rangle + \beta|1\rangle$

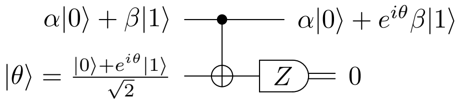

$\alpha|0\rangle + \beta|1\rangle$
$\alpha|0\rangle + e^{i\theta}\beta|1\rangle$
$|\theta\rangle = \frac{|0\rangle+e^{i\theta}|1\rangle}{\sqrt{2}}$
Z
0

Figure 5 Procedure with the postselected circuit $(\mathrm{CNOT}, 0)$ from [13] to rotate $\alpha|0\rangle+\beta|1\rangle$ by θ about the Z-axis.

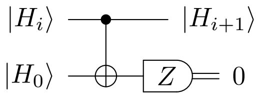


Figure 6 The same circuit $(\mathrm{CNOT}, 0)$ appears in [9] to produce "ladder" qubits $|H_{i+1}\rangle$ from $|H_i\rangle \otimes |H_0\rangle$, where $H|H_0\rangle = |H_0\rangle$.

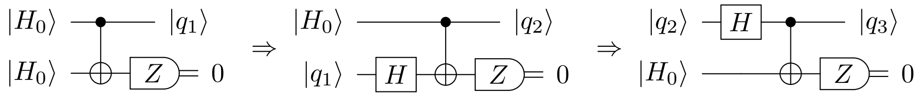


Figure 7 Approach to generate $|q_3\rangle$ with three postselected circuits and four $|H_0\rangle$ states. This qubit appears in [9] (as $|\psi_0^0\rangle$) to help create more diverse "ladder" qubits. If we measure 1 at any of the three steps, then we restart from the first circuit on the left with two new $|H_0\rangle$ copies. Adding recovery for the last two-qubit circuit additionally improves the average $|H_0\rangle$ cost.

about the Z-axis by an angle θ. This is demonstrated in Figure 5. On the chance that the Z-measurement returns 1, then instead of $|q+\theta\rangle = \alpha|0\rangle + e^{i\theta}\beta|1\rangle$, the output becomes $|q-\theta\rangle = \alpha|0\rangle + e^{-i\theta}\beta|1\rangle$, which is $|q\rangle$ rotated by $-\theta$. Jones et. al [13] suggest pairing $|q-\theta\rangle$ with $|2\theta\rangle$ as a direct line to $|q+\theta\rangle$, but we can alternatively break this down into two smaller steps if $|\theta\rangle$ are the only states available. We first run the CNOT circuit on $|q-\theta\rangle \otimes |\theta\rangle$. If we measure 0, then we recover $|q\rangle$, and we proceed with rerunning circuit CNOT on $|q\rangle \otimes |\theta\rangle$.

The method in [9] is similar. It uses the same interacting circuit with a single CNOT to obtain "ladder" qubit states of the kind

$$|H_i\rangle = \cos(\theta_i)\,|0\rangle + \sin(\theta_i)\,|1\rangle,\ \cot(\theta_i) = \cot^{i+1}(\pi/8) \tag{12}$$

for $i \geq 0$. The production starts by supplying two copies of the magic state $H|H_0\rangle = |H_0\rangle$ to the circuit, as seen in Figure 6. Each time we gain a new state $|H_i\rangle$, we reuse the qubit to try and create the next $|H_{i+1}\rangle$. If the attempt fails, then the output of $(\mathrm{CNOT}, 1)$ on $|H_i\rangle \otimes |H_0\rangle$ is $|H_{i-1}\rangle$. Given that the recovery circuit of $(\mathrm{CNOT}, 0)$ is itself, the method to recover $|H_i\rangle$ from $|H_{i-1}\rangle \otimes |H_0\rangle$ is no different than the procedure to create it.

Another example is provided in Figure 7. Here, we show our recovery technique improves the average magic $|H_0\rangle$ cost to produce

$$|q_3\rangle = \cos(\phi_3)|0\rangle + \sin(\phi_3)|1\rangle, \quad \cos(2\phi_3) = \frac{6+5\sqrt{2}}{6+6\sqrt{2}}, \quad 2\phi_3 \approx 0.4456. \tag{13}$$

This qubit participates in the same ladder routine of [9] to generate more varied ladder states. Duclos-Cianci and Svore's method [9] leads to an average cost 12.5 $|H_0\rangle$ states, but we find $|q_3\rangle$ is also obtainable following the procedure in Figure 7. As such, we may consider incorporating a recovery step at one or two places to try and optimize our magic state usage. Simulations of the process in Figure 7 without recovery report an average 10.04 $|H_0\rangle$ qubits, but adding recovery for the final stabilizer circuit brings the number down slightly to 9.45. Although the reduction is small, the Section 6 experiments suggest the potential is greater when the relative cost increases between $|q_2\rangle$ and $|H_0\rangle$.

In general, if we start with the two-qubit state $\varphi \otimes \psi$, then φ is allowed to be mixed, and it can even be part of a larger entangled system. As a quick demonstration, suppose we have the situation as illustrated in the left circuit of Figure 8. Let (C', b') be a recovery circuit of (C, b) and let

$$U\rho U^\dagger = \frac{1}{2^n}\left(\mathbf{P}_I \otimes I + \mathbf{P}_X \otimes X + \mathbf{P}_Y \otimes Y + \mathbf{P}_Z \otimes Z\right) \tag{14}$$

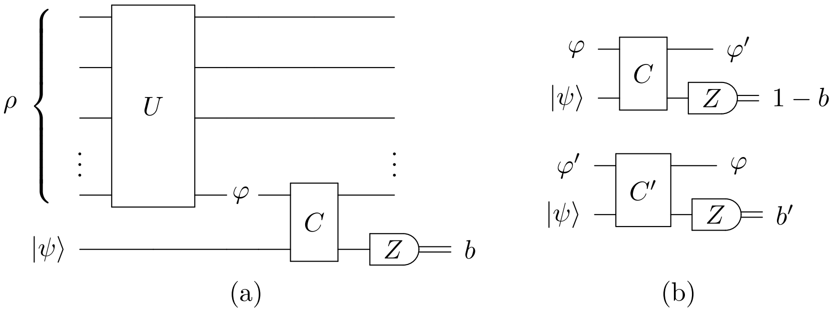


Figure 8 Recovery circuits are also applicable when one of the qubits is entangled with another system. In (a), we trace out all but the n-th qubit of $U\rho U^\dagger$ to get $\varphi \otimes \psi$ as input to circuit C. If we measure $1-b$ as pictured in the top circuit of (b), then we execute circuit C' on $\varphi' \otimes \psi$ to try and recover φ. We succeed with the recovery if we measure b'.

where $\mathbf{P}_L$ are Pauli operator sums on the first $n-1$ qubits. While the proof to Lemma 8 is generalizable to include the unused portions $\mathbf{P}_L$ of the entangled state, the math is simpler and works out the same if we trace out the first $n-1$ qubits, keeping only the last qubit $\varphi = \mathrm{Tr}_{1,n-1}\left(U\rho U^\dagger\right)$ that we need for the two-qubit circuit. If we are unlucky, then qubit n becomes $\varphi' = \Phi_{1-b}(C, \varphi \otimes \psi)$, but we can try to regain φ by executing circuit C' on $\varphi' \otimes \psi$. If the recovery is successful, then we have another opportunity at the output $\Phi_b(C, \varphi \otimes \psi)$. In all likelihood, this is a less lengthy process than preparing another ρ and running the circuit of U again; by some estimates, a synthesis of U over a universal gate set may require an exponential number of gates [11]. This is a stark contrast to C', which uses one CNOT with possibly a couple more single qubit Clifford gates.

6 Experimentation with Recovery Circuits

Consider a two-qubit Clifford unitary C_1 and a two-qubit state $\varphi \otimes \psi$. Suppose we have a target outcome of b_1; the intent is to produce output $\Phi_{b_1}(C_1, \varphi \otimes \psi)$. Then by Corollary 13, we can define a depth k protocol to be a procedure on $k-1$ postselected circuits (C_1, b_1), ..., (C_{k-1}, b_{k-1}) such that (C_{i+1}, b_{i+1}) is the recovery circuit of (C_i, b_i). We start by running circuit C_1 on $\varphi \otimes \psi$. If circuit C_1 is successful i.e. we measure b_1, then no recovery attempts are necessary and we declare success. Otherwise, we enlist circuit C_2 to try and obtain φ. More generally, if circuit C_i is successful, then we recover an input qubit to circuit C_{i-1}; if not, we run circuit C_{i+1} to recover an input qubit to circuit C_i.

The value of k represents a stopping point in our protocol: when circuit C_{k-1} is unsuccessful, we declare failure, discard the output, and restart with a new copy $\varphi \otimes \psi$ to circuit C_1. Thus this process on $k-1$ circuits is nothing more than a classical random walk on $k+1$ integers $\{0, \ldots, k\}$, where the walk begins at location 1, a step onto 0 signifies success, and a step onto k means failure. The success probability of circuit C_i is the probability of a left step from i to $i-1$ and is determined recursively by Equation 8 in Corollary 9. A step in either direction consumes one $|\psi\rangle$.

We conduct simulations of this process to obtain a better idea for N_k, the expected number of $|\psi\rangle$ resources needed to create one $\Phi_{b_1}(C_1, \varphi \otimes \psi)$ with our depth k protocol. Let d be the cost to prepare a single instance of φ relative to the cost of $|\psi\rangle$. Then the cost of one execution or trial is the same as d plus the number of $|\psi\rangle$ qubits used before halting, regardless of declaring success or fail. The costs from all trials are averaged to obtain N_k.

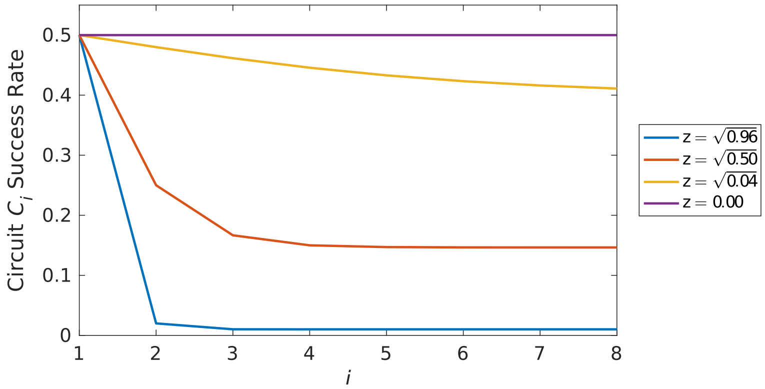


Figure 9 The success probability between circuit C_i and circuit C_{i+1} defined recursively in Corollary 9 drops more dramatically as z moves closer to 1. This leads to a greater expected cost N_k of our protocol since the recovery is less likely to succeed relative to other choices of z. On the other end of the spectrum, the success probability of each circuit C_i is uniform when $z = 0$.

We compare this against the expected cost without recovery ($k = 2$), which is

$$N_2 = \frac{d+1}{Q_{b_1}(C_1, \varphi \otimes \psi)}. \tag{15}$$

We assume for the sake of simplicity that $(C_1, b_1) = (\text{CNOT}, 0)$, which means $(C_2, b_2) = (\text{CNOT}, 0)$, and so forth for the other $k - 3$ recovery circuits.

We further assume that $Q_0(\text{CNOT}, \varphi \otimes \psi) = 1/2$. Since we fix the first success probability, N_k is dependent on the parameter $z = \langle\psi|Z|\psi\rangle$ that appears in the recovery success rate Equation 8. Technically, we need a different φ with each choice of $|\psi\rangle$ to maintain $Q_0(\text{CNOT}, \varphi \otimes \psi) = 1/2$ and the same output $\Phi_0(\text{CNOT}, \varphi \otimes \psi)$. Usually different φ means different costs d, but we will ignore this momentarily and assume the preparation overhead d for each φ is the same for the purposes of a broader comparison of N_k across different $|\psi\rangle$ qubits. In the first set of experiments, we include only one recovery circuit ($k = 3$). The following table summarizes the expected costs for four samples of z obtained over the course of 100000 trials:

d	N_2	$N_3\colon z = \sqrt{0.96}$	$N_3\colon z = \sqrt{0.50}$	$N_3\colon z = \sqrt{0.04}$	$N_3\colon z = 0$
10^{-1}	2.2	3.20	3.18	3.15	3.15
10^0	4.0	4.99	4.75	4.51	4.50
10^1	22	22.7	20.5	18.2	18.0
10^2	202	200.4	177.9	155.1	157.7
10^3	2002	1988.9	1750.7	1521.9	1498.7
10^4	20002	19816.4	17488.0	15215.4	14998.7

The first row with $d = 0.1$ should be interpreted as φ being cheaper to prepare than $|\psi\rangle$. We clearly see an improvement when factoring in recovery in the face of large relative preparation overhead between φ and $|\psi\rangle$. We also see a trend of lower costs as z grows smaller, when $|\psi\rangle$ is moving closer to the XY-plane in the Bloch sphere. This is due to the differences in the recovery success rate at circuit C_2, which are 0.02, 0.25, 0.48, and 0.5, respectively.

In the second batch of experiments, we maintain $d = 1000$ but vary the number of circuits parameterized by k. Again, $Q_0(\text{CNOT}, \varphi \otimes \psi) = 1/2$ and we run 100000 trials. Data for N_k is compiled together in the table below, starting with $k = 3$:

k	$N_k: z = \sqrt{0.96}$	$N_k: z = \sqrt{0.50}$	$N_k: z = \sqrt{0.04}$	$N_k: z = 0$
3	1981.7	1753.2	1522.9	1501.6
4	1982.9	1720.5	1372.2	1336.9
5	1982.4	1716.5	1302.9	1255.2
6	1987.5	1710.9	1266.6	1206.2
7	1982.5	1715.3	1246.7	1174.7
10	1991.7	1717.0	1221.5	1120.8
20	2002.5	1727.3	1220.2	1072.9
30	2006.3	1734.6	1231.4	1064.5
40	2023.5	1743.7	1240.8	1066.3

Observe that the value of N_k continues to lower noticeably for some of the $|\psi\rangle$ cases as more circuits are added before increasing again. This behavior is no surprise since at some point, the penalty to sustain the recovery process will exceed the overhead of repeating the computation. If we look at the success probabilities for the first eight circuits of the protocol for each of the four z samples in Figure 9, we also see the success rates decrease to some lower boundary as i increases, with the exception of when $z = 0$. The drop in probabilities from circuit C_1 to circuit C_3 is quite significant when z is close to 1 (and $1 - z^2$ is small), so the chance of recovery at circuit C_3 is only slightly larger than 0. This explains why there is no apparent change in N_k between one recovery circuit ($k = 3$) versus two ($k = 4$) for the case $z = \sqrt{0.96}$. The ideal situation is to know beforehand how many circuits to include to minimize resource usage.

7 Conclusion

We have shown two-qubit stabilizer circuits require nothing more than a few Clifford gates to perform a job. These simplifications shed light into the complementary nature between interacting circuits. Despite measurements generally being irreversible, we find an exception when handling a two-qubit product state $\varphi \otimes \psi$. That is, we can use $|\psi\rangle$ in conjunction with a specific circuit to salvage the expensive resource qubit φ. What direct effects the recovery operation will have on larger, more complex distillation schemes is unclear. At the moment, we are only able to recognize a small number of applications that involve injecting a non-stabilizer resource state into a computation.

To better gauge the utility of recovery circuits, one direction we may pursue is a more detailed and thorough examination of the depth k protocol in Section 6. In particular, there is an optimal number of circuits to employ that uses the fewest number of resources in expectation on each invocation. As we saw earlier, the behavior of our protocol is akin to that of a (possibly non-uniform) random walk. This modeling of probabilistic circuits is nothing new (see [1, 9, 13]). One matter we need to keep in mind is the costs of attaining qubits φ and $|\psi\rangle$. The amount of work that went into preparing φ should exceed that of $|\psi\rangle$ in order for the recovery to be cost effective, which stems from the fact that we need a copy of $|\psi\rangle$ to operate each circuit. The random walk techniques in [14] should also prove useful for gathering a more precise cost estimate.

Since our two-qubit setting is appropriate for only a limited number of scenarios, a natural follow-up is whether something resembling recovery circuits can easily be extended to larger stabilizer circuits. This question has been answered to an extent for the Clifford+T gate set in [1, 2, 20], where we can treat $|\psi\rangle = HP^\dagger|H_0\rangle$ to perform a non-Clifford $\pi/4$ phase rotation T. The goal in [1, 2, 20] uses a multiqubit circuit of Clifford+T gates to approximate an arbitrary single qubit unitary U up to some error ϵ. If the measurements are unfavorable, then there is a backup operation that either returns the qubits to the initial state, or directly tries to approximate U using a secondary circuit. It is worth investigating whether there exist conditions that enable larger stabilizer circuits to exhibit the recovery feature we demonstrated here on general $|\psi\rangle$ resources.

References

1 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.

2 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.

3 Sergey Bravyi and Jeongwan Haah. Magic-state distillation with low overhead. *Phys. Rev. A*, 86:052329, Nov 2012. doi:10.1103/PhysRevA.86.052329.

4 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.

5 Earl T. Campbell and Mark Howard. Unified framework for magic state distillation and multiqubit gate synthesis with reduced resource cost. *Phys. Rev. A*, 95:022316, Feb 2017. doi:10.1103/PhysRevA.95.022316.

6 Earl T. Campbell and Mark Howard. Unifying Gate Synthesis and Magic State Distillation. *Phys. Rev. Lett.*, 118:060501, Feb 2017. doi:10.1103/PhysRevLett.118.060501.

7 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.

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 Brett Giles and Peter Selinger. Exact synthesis of multiqubit Clifford+T circuits. *Phys. Rev. A*, 87:032332, Mar 2013. doi:10.1103/PhysRevA.87.032332.

12 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.

13 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.

14 J.G. Kemény and J.L. Snell. *Finite markov chains.* University series in undergraduate mathematics. Springer-Verlag New York, 1976.

15 Vadym Kliuchnikov, Dmitri Maslov, and Michele Mosca. Asymptotically Optimal Approximation of Single Qubit Unitaries by Clifford and T Circuits Using a Constant Number of Ancillary Qubits. *Phys. Rev. Lett.*, 110:190502, May 2013. doi:10.1103/PhysRevLett.110.190502.

16 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`.

17 E. Knill. Quantum computing with realistically noisy devices. *Nature*, 434, Mar 2005. `doi:10.1038/nature03350`.

18 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`.

19 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`.

20 Adam Paetznick and Krysta M. Svore. Repeat-Until-Success: Non-determistic decomposition of single-qubit unitaries. *Quantum Info. Comput.*, 14(15-16):1277–1301, Nov 2014. URL: `http://arxiv.org/abs/1311.1074`.

21 Ben Reichardt. Quantum universality by state distillation. *Quantum Information and Computation*, 9:1030–1052, 2009. URL: `http://arxiv.org/abs/quant-ph/0608085v2`.

22 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`.

23 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`.

A Proof of Lemma 4

Similar to a single qubit, a two-qubit density matrix ρ can be expressed as a real combination of two-qubit Pauli operators $\sigma_{jk} = \sigma_j \otimes \sigma_k$, where $\sigma_0 = I$, $\sigma_1 = X$, $\sigma_2 = Y$, and $\sigma_3 = Z$ e.g. $\sigma_{13} = X \otimes Z$. We omit the tensor product and use σ_{jk} for notation reasons. We define $\mathcal{P}_\pm = \{\pm\sigma_{jk} \mid j \neq 0 \text{ and } k \neq 0\}$ to be a set of nontrivial two-qubit Pauli operators.

To prove Lemma 4, we start by rewriting Equation 4 in Definition 3 as

$$C_1\Pi_1\rho\Pi_1C_1^\dagger = (G \otimes I)C_2\Pi_2\rho\Pi_2C_2^\dagger(G^\dagger \otimes I) \tag{16}$$

where $\Pi_1 = C_1^\dagger(I \otimes |b_1\rangle\langle b_1|)C_1$ and $\Pi_2 = C_2^\dagger(I \otimes |b_2\rangle\langle b_2|)C_2$ are projection operators. Reichardt [21] showed that Equation 16 holds for some single qubit Clifford G on all states ρ if $\Pi_1 = \Pi_2$. In our two-qubit case, there are only 30 cases of $\Pi_1 = \Pi_2$. We make some refinements here to make the ideas in [21] a little more digestible in our notation.

▶ **Lemma 14.** *Let (C_1, b_1) and (C_2, b_2) be postselected two-to-one stabilizer circuits. If $\Pi = C_1^\dagger(I \otimes |b_1\rangle\langle b_1|)C_1 = C_2^\dagger(I \otimes |b_2\rangle\langle b_2|)C_2$, then $(C_1, b_1) \sim (C_2, b_2)$.*

Proof. Note that $2(I \otimes |b_j\rangle\langle b_j|) = \sigma_{00} + (-1)^{b_j}\sigma_{03}$. Let $2\Pi = \sigma_{00} + \lambda_{03}$, where $\lambda_{03} \in \mathcal{P}_\pm$, and let λ_{10}, λ_{20}, $\lambda_{30} \in \mathcal{P}_\pm$ be two-qubit Pauli operators such that $[\lambda_{03}, \lambda_{10}] = [\lambda_{03}, \lambda_{20}] = [\lambda_{03}, \lambda_{30}] = 0$ and $i\lambda_{30} = \lambda_{10}\lambda_{20}$. Let ρ be a two-qubit state. Then

$$\Pi\rho\Pi = \frac{1}{8}\left(w\sigma_{00} + w\lambda_{03} + x\lambda_{10} + x\lambda_{13} + y\lambda_{20} + y\lambda_{23} + z\lambda_{30} + z\lambda_{33}\right) \tag{17}$$

where $\lambda_{k3} = \lambda_{03}\lambda_{k0}$ and $x = \mathrm{Tr}((\lambda_{10} + \lambda_{13})\rho)$. The coefficients w, y, z are determined similarly with $\sigma_{00} + \lambda_{03}$, $\lambda_{20} + \lambda_{23}$, and $\lambda_{30} + \lambda_{33}$, respectively. Our starting condition

$C_j\lambda_{03}C_j^\dagger = (-1)^{b_j}\sigma_{03}$ implies

$$\begin{aligned} C_j\lambda_{10}C_j^\dagger, C_j\lambda_{20}C_j^\dagger \in \{\, & \sigma_{10}, (-1)^{b_j}\sigma_{13}, -\sigma_{10}, (-1)^{b_j+1}\sigma_{13}, \\ & \sigma_{20}, (-1)^{b_j}\sigma_{23}, -\sigma_{20}, (-1)^{b_j+1}\sigma_{23}, \\ & \sigma_{30}, (-1)^{b_j}\sigma_{33}, -\sigma_{30}, (-1)^{b_j+1}\sigma_{33} \,\}. \end{aligned} \tag{18}$$

This means there are single qubit Clifford gates G_j to permute the operators in a way that

$$(G_j \otimes I)C_j\lambda_{10}C_j^\dagger(G_j^\dagger \otimes I) \in \{\, \sigma_{10}, (-1)^{b_j}\sigma_{13} \,\} \tag{19}$$

$$(G_j \otimes I)C_j\lambda_{20}C_j^\dagger(G_j^\dagger \otimes I) \in \{\, \sigma_{20}, (-1)^{b_j}\sigma_{23} \,\}. \tag{20}$$

The value of $(G_j \otimes I)C_j\lambda_{30}C_j^\dagger(G_j^\dagger \otimes I)$ is fixed given the other two. Our unnormalized post-measurement states $\rho_j' = (G_j \otimes I)C_j\Pi\rho\Pi C_j^\dagger(G_j^\dagger \otimes I)$ are now

$$\rho_j' = \frac{1}{4}(wI + xX + yY + zZ) \otimes |b_j\rangle\langle b_j|. \tag{21}$$

The first qubit of ρ_1' and ρ_2' are the same after G_1 and G_2. Therefore $(C_1, b_1) \sim (C_2, b_2)$. ◀

We now have the tools to prove Lemma 4. Note that a Clifford equivalence $(C_1, b_1) \sim (C_2, b_2)$ is invariant with respect to Clifford circuits that execute prior to circuits C_1 and C_2 i.e. $(C_1, b_1) \sim (C_2, b_2)$ if and only if $(C_1U, b_1) \sim (C_2U, b_2)$ for any Clifford unitary U.

Proof. We partition the 15 Pauli operators σ_{jk} into the following sets:

$$\mathcal{P}_A = \{\sigma_{jk} \mid j,k \in \{1,2,3\}\}, \qquad \mathcal{P}_B = \{\sigma_{01}, \sigma_{02}, \sigma_{03}\}, \qquad \mathcal{P}_C = \{\sigma_{10}, \sigma_{20}, \sigma_{30}\}. \tag{22}$$

We look at σ_{33} first. Suppose there is a bit b' such that $C\sigma_{33}C^\dagger = (-1)^{b'}\sigma_{03}$. For readability, set $C' = \text{CNOT}$. Knowing $C'\sigma_{33}C'^\dagger = \sigma_{03}$, we obtain $(C, b) \sim (\text{CNOT}, b + b' \bmod 2)$ from Lemma 14. For the remaining $\sigma_{jk} \in \mathcal{P}_A$, suppose $C\sigma_{jk}C^\dagger = \pm\sigma_{03}$. Choose single qubit Clifford gates G_1 and G_2 such that $(G_1 \otimes G_2)\sigma_{jk}(G_1^\dagger \otimes G_2^\dagger) = \sigma_{33}$. Define $C'' = C(G_1^\dagger \otimes G_2^\dagger)$. Then $C''\sigma_{33}C''^\dagger = (-1)^{b'}\sigma_{03}$ for some b'. The rest follows from previous arguments to conclude $(C''(G_1 \otimes G_2), b) = (C, b) \sim (\text{CNOT}(G_1 \otimes G_2), b + b' \bmod 2)$.

For the operator $\sigma_{03} \in \mathcal{P}_B$, assume $C\sigma_{03}C^\dagger = (-1)^{b'}\sigma_{03}$. Then $(C, b) \sim (\sigma_{00}, b{+}b' \bmod 2)$. Coverage of the other five from $\mathcal{P}_B$ and $\mathcal{P}_C$ is similar to the above.

To finish, suppose $(C, b) \sim (I \otimes G, b + b' \bmod 2)$, where G is a single qubit Clifford gate. If $b + b' \bmod 2 = 1$, then $(C, b) \sim (I \otimes G, 1) \equiv (I \otimes XG, 0)$. The same applies when $(C, b) \sim ((I \otimes G)\text{SWAP}, 1)$. If $(C, b) \sim (\text{CNOT}(G_1 \otimes G_2), 1)$, then we include $(I \otimes X)\text{CNOT}(G_1 \otimes G_2) = \text{CNOT}(G_1 \otimes XG_2)$. The other case $b + b' \bmod 2 = 0$ follows directly from Lemma 14. ◀

B Additional Material on Recovery Circuits

We may use the following to help us determine when two recovery circuits are Clifford equivalent. In particular, it dispels concerns that there may be two recovery circuits where one has a better chance of succeeding than the other. We use the same notation for two-qubit Paulis σ_{jk} and $\mathcal{P}_\pm$ as in Appendix A.

▶ **Lemma 15.** *Let (C_1, b_1) be a recovery circuit of an interacting postselected circuit (C, b). If (C_2, b_2) is also a recovery circuit of (C, b), then $C_1^\dagger(I \otimes |b_1\rangle\langle b_1|)C_1 = C_2^\dagger(I \otimes |b_2\rangle\langle b_2|)C_2$.*

Proof. It is easier to prove the contrapositive. Specifically, we show the recovery from (C_2, b_2) will fail on one particular pair of qubits φ_2 and $|\psi\rangle$, although many exist that are equally as good. Suppose $\Pi_2 = C_2^\dagger(I \otimes |b_2\rangle\langle b_2|)C_2$. Let $2\Pi_2 = \sigma_{00} + \lambda_{03}$, where $\lambda_{03} \in \{\pm\sigma_{jk} \mid j,k \in \{1,2,3\}\}$, and let λ_{30} and λ_{33} be two-qubit Pauli operators from $\mathcal{P}_\pm$ such that $[\lambda_{03}, \lambda_{30}] = 0$ and $\lambda_{03} = \lambda_{30}\lambda_{33}$. The qubits φ_2 and $|\psi\rangle$ we choose shall have Bloch vectors

$$\varphi_2\colon\ (x_2, y_2, z_2) = \left(\sqrt{\frac{2}{17}}, \sqrt{\frac{5}{17}}, \sqrt{\frac{10}{17}}\right), \qquad |\psi\rangle\colon\ (x, y, z) = \left(\sqrt{\frac{1}{11}}, \sqrt{\frac{3}{11}}, \sqrt{\frac{7}{11}}\right). \tag{23}$$

Let φ_1 be a qubit so that $\varphi_2 = \Phi_{1-b}(C, \varphi_1 \otimes \psi)$. Let $\varphi_1' = \Phi_{b_2}(C_2, \varphi_2 \otimes \psi)$.

To prove the recovery by (C_2, b_2) will fail, we merely need to verify that the Bloch vectors from all 18 choices of λ_{03} are different, which implies $\varphi_1' \neq \varphi_1$ whenever $C_1^\dagger(I \otimes |b_1\rangle\langle b_1|)C_1 \neq \Pi_2$. We track the coefficients $a_{jk} = \mathrm{Tr}(\lambda_{jk}(\varphi_2 \otimes \psi))$. Then

$$\mathrm{Tr}\left(\Pi_2\left(\varphi_2 \otimes \psi\right)\Pi_2\right) = \frac{1 + a_{03}}{2}, \quad \mathrm{Tr}\left(\lambda_{30}\Pi_2\left(\varphi_2 \otimes \psi\right)\Pi_2\right) = \frac{a_{30} + a_{33}}{2}, \tag{24}$$

yielding $v = (a_{30} + a_{33})/(1 + a_{03})$ as a Bloch vector component of φ_1'. The most convenient choices for λ_{30} and λ_{33} are tensor products with the identity e.g. $\lambda_{03} = -\sigma_{33}$, $\lambda_{30} = \sigma_{30}$, $\lambda_{33} = -\sigma_{03}$, and $\lambda_{03} = \sigma_{11}$, $\lambda_{30} = \sigma_{10}$, $\lambda_{33} = \sigma_{01}$, which means that $a_{03} = a_{30}a_{33}$. If we look at the coefficients from the first example with $\lambda_{03} = -\sigma_{33}$, then $a_{30} = z_2$ and $a_{33} = -z$. We get the following components for each of the positive possibilities for λ_{03}:

λ_{03}	a_{03}	λ_{30}	a_{30}	λ_{33}	a_{33}	v
σ_{11}	x_2x	σ_{10}	x_2	σ_{01}	x	0.5841
σ_{12}	x_2y	σ_{10}	x_2	σ_{02}	y	0.7338
σ_{13}	x_2z	σ_{10}	x_2	σ_{03}	z	0.8957
σ_{21}	y_2y	σ_{20}	y_2	σ_{01}	x	0.7252
σ_{22}	y_2y	σ_{20}	y_2	σ_{02}	y	0.8296
σ_{23}	y_2z	σ_{20}	y_2	σ_{03}	z	0.9354
σ_{31}	z_2x	σ_{30}	z_2	σ_{01}	x	0.8678
σ_{32}	z_2y	σ_{30}	z_2	σ_{02}	y	0.9205
σ_{33}	z_2z	σ_{30}	z_2	σ_{03}	z	0.9708

and the following for each of the negative possibilities for λ_{03}:

λ_{03}	a_{03}	λ_{30}	a_{30}	λ_{33}	a_{33}	v
$-\sigma_{11}$	$-x_2x$	σ_{10}	x_2	$-\sigma_{01}$	$-x$	0.0463
$-\sigma_{12}$	$-x_2y$	σ_{10}	x_2	$-\sigma_{02}$	$-y$	−0.2183
$-\sigma_{13}$	$-x_2z$	σ_{10}	x_2	$-\sigma_{03}$	$-z$	−0.6260
$-\sigma_{21}$	$-y_2y$	σ_{20}	y_2	$-\sigma_{01}$	$-x$	0.2879
$-\sigma_{22}$	$-y_2y$	σ_{20}	y_2	$-\sigma_{02}$	$-y$	0.0280
$-\sigma_{23}$	$-y_2z$	σ_{20}	y_2	$-\sigma_{03}$	$-z$	−0.4501
$-\sigma_{31}$	$-z_2x$	σ_{30}	z_2	$-\sigma_{01}$	$-x$	0.6055
$-\sigma_{32}$	$-z_2y$	σ_{30}	z_2	$-\sigma_{02}$	$-y$	0.4083
$-\sigma_{33}$	$-z_2z$	σ_{30}	z_2	$-\sigma_{03}$	$-z$	−0.0792

Neither are any of the values v the same if we multiple each one by -1, which may come about from an application of a single qubit Pauli on the output. Thus our statement holds. ◀

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

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
LIPICS 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

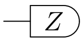
The Pauli group consists of n-qubit Pauli operators on the four matrices

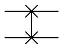
$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{1}$$

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

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

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 ρ 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* Φ_b of a postselected circuit (C, b) on an input ρ is

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

The expression *run circuit* C shall mean an application of unitary C on the initial state ρ, 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 ρ, 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 ρ, we have the equality

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

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].

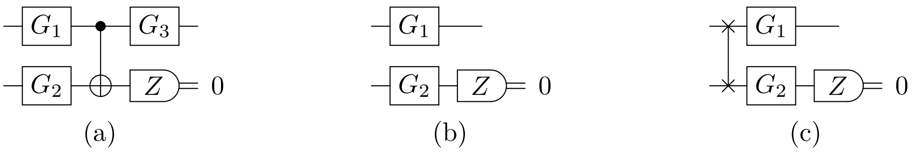


Figure 1 Any stabilizer procedure generating a single qubit output from a two-qubit input can be implemented by a postselected stabilizer circuit (C, b) taking on one of the three forms above. The exact single qubit Clifford gates G_1, G_2, and G_3 depend on C and b and are not unique.

2.3 Recovery Circuits

If ρ is the product state $\varphi \otimes |\psi\rangle\langle\psi|$, then only postselected circuits of the kind $(C, b) \sim (\mathrm{CNOT}(G_1 \otimes G_2), 0)$ qualify for a recovery circuit. For convenience, we use ψ 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 $(C, b) \sim (\mathrm{CNOT}(G_1 \otimes G_2), 0)$. 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 φ. There is also a relatively simple construction to acquire a recovery circuit. If $(C, b) \sim (\mathrm{CNOT}(G_1 \otimes G), 0)$ for single qubit Clifford gates G_1 and G, then there is a third Clifford gate G_2 satisfying $(C, 1 - b) \equiv ((G_2^\dagger \otimes I)\mathrm{CNOT}(G_1 \otimes G), 1)$. We may then use this to design a recovery circuit $(C', 0)$ of (C, b), where $C' = (G_1^\dagger \otimes I)\mathrm{CNOT}(G_2 \otimes G)$ [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 φ_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 $z = \langle\psi|G^\dagger Z G|\psi\rangle$. 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 $\Phi_{1-b}(C, \varphi_1 \otimes \psi) = G\varphi_1 G^\dagger$ whenever $|\psi\rangle$ is a stabilizer qubit [21]. Since the output is essentially φ_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

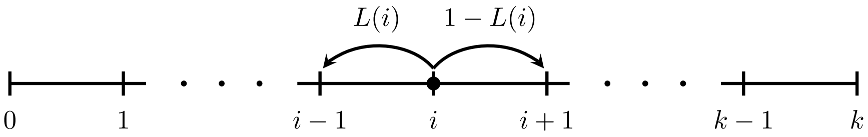


Figure 2 The behavior of a depth k protocol corresponds to a random walk on integers $\{0, \dots, k\}$ and starts at position 1. The random walk ends upon reaching 0 or k, with 0 representing success and k representing failure. The transition from i to $i-1$ is the success probability of the i-th circuit 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 φ_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 φ_1 and we repeat step 2. Otherwise we get the output φ_3 of $(C_2, 1)$ on $\varphi_2 \otimes \psi$.
4. Repeat step 3 as necessary for other circuits C_i. That is, let φ_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 φ_{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 φ_0 while reducing the number of times we rerun the prior computation on new copies of φ_1. As pointed out by the simulations in [21], we expect the protocol is more useful when φ_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 φ_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*

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

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

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

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

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

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

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

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

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 $\varphi_i' = G_i \varphi_i G_i^\dagger$. From our G_{i+1} selection, this means

$$\varphi_{i+1}' = \Phi_1(\text{CNOT}, \varphi_i' \otimes \psi') = G_{i+1}\varphi_{i+1}G_{i+1}^\dagger \tag{11}$$

$$\varphi_i' = \Phi_0(\text{CNOT}, \varphi_{i+1}' \otimes \psi'). \tag{12}$$

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 φ_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*

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

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

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

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

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

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 Block vector (x, y, z) of $G|\psi\rangle$, we may interpret λ 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 $L(1)$, …, $L(k-1)$, generated entirely by a recurrence relation $L(i)$ defined on two real numbers which we call λ and γ. The depth k only serves to indicate a stopping point when generating that sequence, so our protocol is basically controlled by three parameters (λ, γ, 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, λ is the *reduced XY-overlap* of resource qubit $|\psi\rangle$, and γ is the *starting success probability* $Q_0(C_1, \varphi_1 \otimes \psi)$. However, if we want to treat λ and γ 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 λ and γ that are necessary to define a difference equation adhering to Lemma 8.

▶ **Definition 9** (probability specification and boundary)**.** Given real numbers (λ, γ), let

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

Then (λ, γ) 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 (α, β) are the *boundaries* of the probability specification.

▶ **Definition 10** (intermediate functions and rde)**.** Let (λ, γ) be a probability specification and let (α, β) be its boundaries. The following are the *intermediate* functions of (λ, γ):

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

and the following is a *rational difference equation (rde)* on (λ, γ):

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

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 (λ, γ), the boundaries α and β 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 φ_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 γ 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 φ_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 (λ, γ). Then the success probability p and expected demand s of a protocol set to (λ, γ) and depth k are*

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

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 (λ, γ, k) behaves quite differently when $\lambda = 1/4$ versus the more general (λ, γ) 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 (λ, γ) and depth k is*

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

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

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

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 (λ, γ) 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_{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 (λ, γ) and startup costs d. The examples provide a convincing argument to assume $N(k)$ has

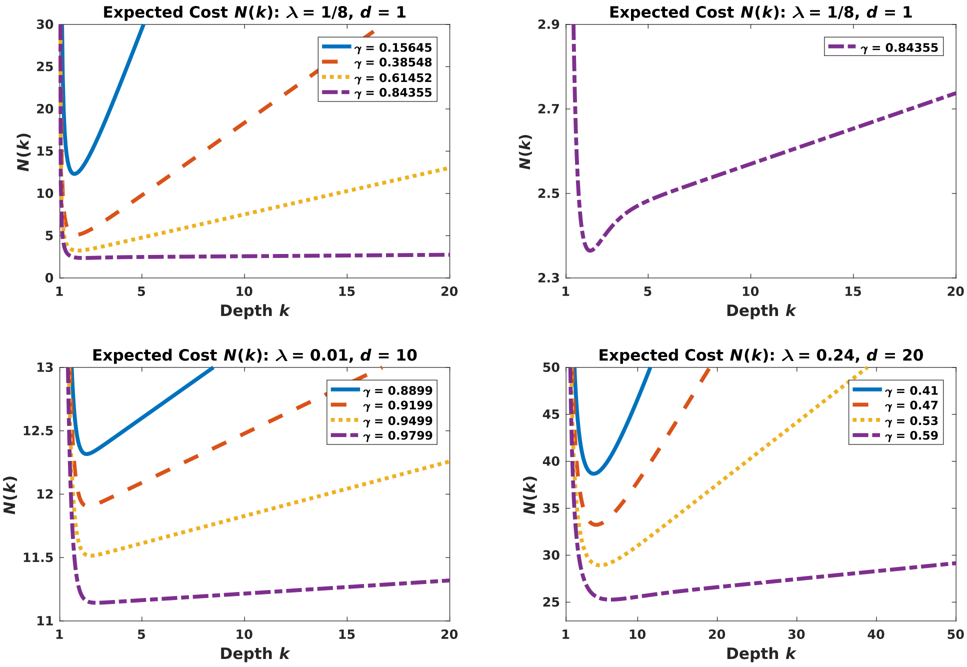


Figure 3 This figure contains plots of the expected cost $N(k)$ for three choices of the reduced XY-overlap $\lambda \in (0, 1/4)$ and varying starting probabilities γ. Although the curve of $\gamma = 0.84355$ for $\lambda = 1/8$ appear to reach a constant, the close-up in the top right graph suggests 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_{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_{opt} for a protocol set to a restricted probability specification (λ, γ). By the end, we propose that k_{opt} scales logarithmically with respect to the startup cost d. Let (α, β) be the boundaries of (λ, γ). Then the first derivative in its entirety is

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

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

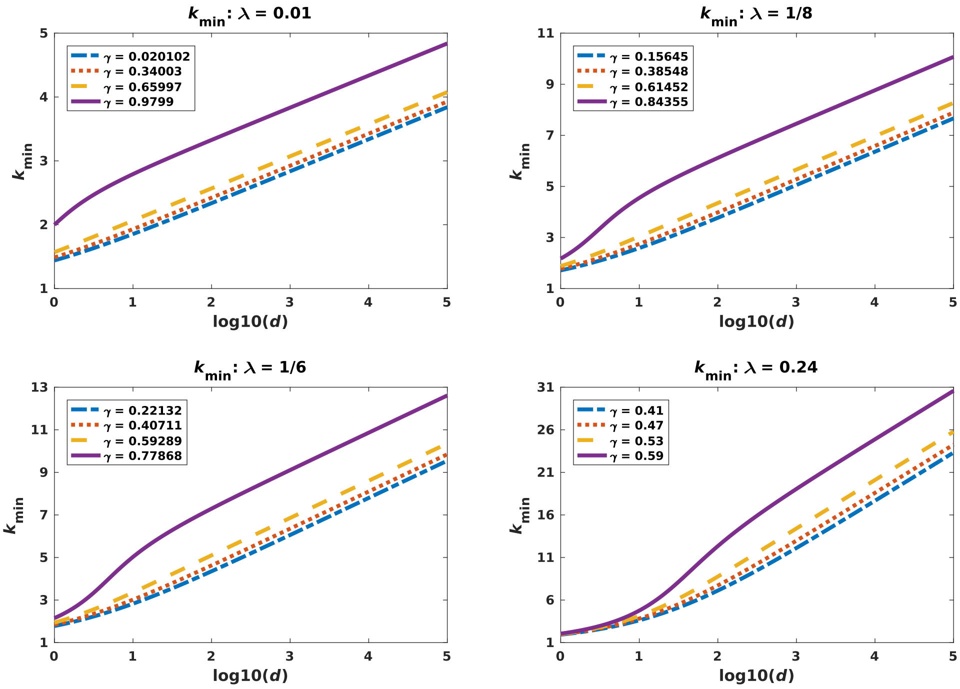


Figure 4 The data for $k_{\min}$ suggests a protocol set to a restricted probability specification should use $O(\log d)$ circuits to keep costs to a minimum, where d is the startup cost.

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

$$\lim_{k\to\infty} N'(k) = \lim_{k\to\infty} \frac{\left(\alpha + (\beta/\alpha)^{k-1}\beta - \left(1 + (\beta/\alpha)^{k-1}\right)\gamma\right)}{(\alpha-\beta)\left(1-(\beta/\alpha)^{k-1}\right)\gamma} = \frac{\alpha-\gamma}{(\alpha-\beta)\,\gamma} > 0 \tag{23}$$

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

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

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

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

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

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

Making the substitution

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

turns Equation 26 into

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

where

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

The solution t to Equation 28 indicates that

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

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

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

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 $k_{\mathrm{opt}} = O(\log d)$. Equation 31 is a good starting point to begin a search for the exact value of k_{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 (λ, γ). Then the optimal depth is $k_{\mathrm{opt}} = \min(\lceil 1+\sqrt{d}\rceil, \lfloor 1+\sqrt{d}\rfloor)$ when $\lambda = 1/4$ and $O(\log d)$ when (λ, γ) 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 $N(k_{\mathrm{opt}})$. We look at $N(2)/N(k_{\mathrm{opt}})$ under the assumptions of Theorem 16.

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

$$\lim_{d\to\infty} \frac{N(2)}{N(k_{\mathrm{opt}})} \le 2. \tag{32}$$

Proof. We consider a restricted probability specification (λ, γ) first. Let (α, β) 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

$$\frac{N(2)}{N(k)} = \frac{(d+1)\, A_1(k-1)\, (A_1(1))^2}{dA_1(k)\, (A_1(1))^2 + (k-1)\, B_2(k-1)A_1(1) + (\gamma - 2\lambda)\, A_1(k-1)}. \tag{33}$$

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

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

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:

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

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

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

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

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

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

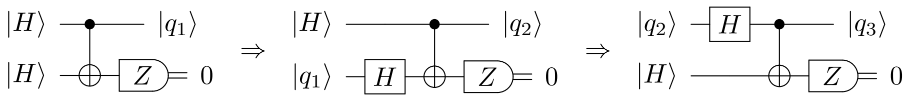


Figure 5 Approach to generate $|q_3\rangle$ with three postselected circuits and four $|H\rangle$ states as seen in [21]. Adding recovery for the last two-qubit circuit lowers the average $|H\rangle$ usage.

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 φ_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

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

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 φ_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_{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 φ_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 (λ, γ) 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 (α, β) of (λ, γ). 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

$$\begin{aligned}
B_1(i)A_1(i+1) &= A_1(i+1)A_1(i+1) - \gamma A_1(i+1)A_1(i) && (39)\\
&= A_2(2i+2) - 2\lambda^{i+1} - \gamma A_2(2i+1) + \gamma\lambda^i && (40)\\
&= B_2(2i+1) - 2\lambda^{i+1} + \gamma\lambda^i && (41)\\
B_2(2i+1)A_1(1) &= A_2(2i+2)A_1(1) - \gamma A_2(2i+1)A_1(1) && (42)\\
&= A_1(2i+3) - \lambda A_1(2i+1) - \gamma A_1(2i+2) + \gamma\lambda A_1(2i) && (43)\\
&= B_1(2i+2) - \lambda B_1(2i) && (44)
\end{aligned}$$

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

$$\begin{aligned}
B_1(j-i)A_1(i) &= A_1(j-i+1)A_1(i) - \gamma A_1(j-i)A_1(i) && (45)\\
&= A_2(j+1) - \lambda^i A_2(j-2i+1) - \gamma A_2(j) + \gamma\lambda^i A_2(j-2i) && (46)\\
&= B_2(j) - \lambda^i B_2(j-2i) && (47)
\end{aligned}$$

while (vii) is based on (iv):

$$\begin{aligned}
\lambda^i A_2(j-2i)A_1(1) + A_2(j-i+1)A_1(i) &= \alpha^i\beta^i\left(A_1(j-2i+1) - \alpha\beta A_1(j-2i-1)\right)\\
&\quad + A_1(j+1) - \alpha^i\beta^i A_1(j-2i+1) && (48)\\
&= A_1(j+1) - \alpha^{i+1}\beta^{i+1}A_1(j-2i-1) && (49)\\
&= A_2(j-i)A_1(i+1). && (50)
\end{aligned}$$

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 $\{0, \ldots, k\}$, 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 $\{0, \ldots, k\}$. 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

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

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

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

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 $\{0, \ldots, k\}$ 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

$$\begin{array}{c} \begin{array}{cc} \overbrace{\qquad}^{2} & \overbrace{\qquad}^{k-1} \end{array} \\ \left[\begin{array}{c|c} I & O \\ \hline U & V \end{array}\right] \begin{array}{l} \}\,2 \\ \}\,k-1 \end{array} \end{array} \tag{53}$$

where O contains all zeroes and I is the 2×2 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

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

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,

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

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 $S(i) = (E\vec{\mathbf{1}})_i$, where $\vec{\mathbf{1}}$ is a column vector of ones. If $S(i)$ is an expectation in the number of steps taken, then the variance is $(2E - I)E\vec{\mathbf{1}} - \mathrm{Sq}(E\vec{\mathbf{1}})$, where $\mathrm{Sq}(E\vec{\mathbf{1}})$ squares each entry of $E\vec{\mathbf{1}}$. 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:

$$F(i) = F(i-1) - L(i)R(i-1)F(i-2),\ F(0) = 1,\ F(-1) = 0 \tag{56}$$

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

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

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

then Gaussian elimination eventually yields

$$E = \begin{bmatrix} \dfrac{\overline{F}(2,4)F(0)}{\overline{F}(1,4)} & \dfrac{R(1)\overline{F}(3,4)F(0)}{\overline{F}(1,4)} & \dfrac{R(2)R(1)\overline{F}(4,4)F(0)}{\overline{F}(1,4)} \\ \dfrac{L(2)\overline{F}(3,4)F(0)}{\overline{F}(1,4)} & \dfrac{\overline{F}(3,4)F(1)}{\overline{F}(1,4)} & \dfrac{R(2)\overline{F}(4,4)F(1)}{\overline{F}(1,4)} \\ \dfrac{L(3)L(2)\overline{F}(4,4)F(0)}{\overline{F}(1,4)} & \dfrac{L(3)\overline{F}(4,4)F(1)}{\overline{F}(1,4)} & \dfrac{\overline{F}(4,4)F(2)}{\overline{F}(1,4)} \end{bmatrix} \tag{59}$$

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 $\overline{F}(i,k)$ 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 α, β such that $\alpha + \beta = 1$. Then*

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

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

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

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

$$= \sum_{j=0}^{i}\alpha^{i-j}\beta^{j}. \tag{63}$$

For the second identity,

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

which finishes the proof. ◀

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

$$F(i)=F(i-1)-\alpha\beta\,F(i-2),\;\; F(0)=1,\;\; F(-1)=0 \tag{65}$$

$$\overline{F}(i,k)=F(i+1,k)-\alpha\beta\,\overline{F}(i+2,k),\;\; \overline{F}(k,k)=1,\;\; \overline{F}(k+1,k)=0 \tag{66}$$

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

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

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

$$=F(k-(i+1))-\alpha\beta F(k-(i+2)) \tag{68}$$

$$=F(k-i-1)-\alpha\beta F(k-i-2)=F(k-i). \tag{69}$$

This completes the proof. ◀

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 (λ,γ). If $L(i)$ determines the left step probabilities of a random walk over $\{0,\ldots,k\}$, then the following describes the entries of the fundamental matrix E:*

$$E_{i,j}=\begin{cases}\dfrac{\lambda^{i-j}A_1(k-i)A_1(j)B_1(j-1)}{A_1(1)A_1(k)B_1(i-1)} & \textit{if } i\geq j\\[2ex] \dfrac{A_1(k-j)A_1(i)B_1(j-1)}{A_1(1)A_1(k)B_1(i-1)} & \textit{otherwise}\end{cases} \tag{70}$$

where $A_1(i)$ and $B_1(i)$ are intermediate functions of (λ,γ).

Proof. Let (α,β) be the boundaries of (λ,γ). 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

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

The pattern in Equation 59 suggests

$$E_{i,j}=\begin{cases}\dfrac{\overline{F}(j+1,k)F(i-1)}{\overline{F}(1,k)}\displaystyle\prod_{l=i}^{j-1}R(l) & \text{if } i<j\\[2ex] \dfrac{\overline{F}(i+1,k)F(i-1)}{\overline{F}(1,k)} & \text{if } i=j\\[2ex] \dfrac{\overline{F}(i+1,k)F(j-1)}{\overline{F}(1,k)}\displaystyle\prod_{l=j+1}^{i}L(l) & \text{if } i>j.\end{cases} \tag{72}$$

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

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

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

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

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*

$$E_{i,j} = \begin{cases} \dfrac{2\,(k-i)\,j}{k} & \textit{if } i\geq j \\ \dfrac{2\,(k-j)\,i}{k} & \textit{otherwise.}\end{cases} \tag{75}$$

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

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

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 (λ,γ). Then for all integers $i\geq 0$,*

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

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

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

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

$$= A_2(2)-2\lambda = A_1(1)A_1(1). \tag{80}$$

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

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

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

as a result of Lemma 18(v). ◀

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

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

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

$$J_2(1) = B_1(k-1-1)A_1(1) + B_2(k-1) - B_2(k-1) \tag{84}$$

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

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

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

$$J_2(i) = B_1(k-i-1)A_1(i) + J_2(i-1) \tag{87}$$

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

By Lemma 18(vi), we end up with

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

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

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

due to Lemma 18(viii). ◀

We are ready to solve Equations 51 and 52.

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

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

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

where $A_1(i)$ and $B_j(i)$ are intermediate functions of (λ,γ).

Proof. More formally,

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

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

$$\begin{aligned} A_1(i)B_2(i) &= A_1(i)A_2(i+1) - \gamma A_1(i)A_2(i) &(94)\\ &= A_1(2i+1) - \gamma A_1(2i) - \lambda^i A_1(1) &(95)\\ &= B_1(2i) - \lambda^i A_1(1) &(96)\end{aligned}$$

and Lemma 18, we arrive at

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

Then combining it with

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

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,

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

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 $\{0, \ldots, k\}$.*

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

$$\begin{aligned} S(i) &= \sum_{j=1}^{i} \frac{2\,(k-i)\,j}{k} + \sum_{j=i+1}^{k-1} \frac{2\,(k-j)\,i}{k} &(100)\\ &= \frac{2\,(k-i)}{k}\frac{(i+1)\,i}{2} + \frac{2i}{k}\frac{(k-i)\,(k-i-1)}{2} &(101)\\ &= \frac{(i+1+k-i-1)\,(k-i)\,i}{k} = ki - i^2. &(102)\end{aligned}$$

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

Efficient Population Transfer via Non-Ergodic Extended States in Quantum Spin Glass

Kostyantyn Kechedzhi[1]
QuAIL and USRA, NASA Ames Research Center, Moffett Field, CA 94035, USA
Google Inc., Venice, CA 90291, USA
kostyantyn.kechedzhi@nasa.gov

Vadim Smelyanskiy
Google Inc., Venice, CA 90291, USA
smelyan@google.com

Jarrod R. McClean
Google Inc., Venice, CA 90291, USA
jmcclean@google.com

Vasil S. Denchev
Google Inc., Venice, CA 90291, USA
denchev@google.com

Masoud Mohseni
Google Inc., Venice, CA 90291, USA
mohseni@google.com

Sergei Isakov
Google Inc., Venice, CA 90291, USA
iserge@google.com

Sergio Boixo
Google Inc., Venice, CA 90291, USA
boixo@google.com

Boris Altshuler
Physics Department, Columbia University,
538 West 120th Street, New York, New York 10027 , USA
bla@phys.columbia.edu

Hartmut Neven
Google Inc., Venice, CA 90291, USA
neven@google.com

Abstract

Quantum tunneling has been proposed as a physical mechanism for solving binary optimization problems on a quantum computer because it provides an alternative to simulated annealing by directly connecting deep local minima of the energy landscape separated by large Hamming distances. However, classical simulations using Quantum Monte Carlo (QMC) were found to

[1] K.K. acknowledges support by NASA Academic Mission Services, contract number NNA16BD14C. This research is based upon work supported in part by the AFRL Information Directorate under grant F4HBKC4162G001 and the Office of the Director of National Intelligence (ODNI) and the Intelligence Advanced Research Projects Activity (IARPA), via IAA 145483. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of ODNI, IARPA, AFRL, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purpose notwithstanding any copyright annotation thereon.

13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2018).
Editor: Stacey Jeffery; Article No. 9; pp. 9:1–9:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

efficiently simulate tunneling transitions away from local minima if the tunneling is effectively dominated by a single path. We analyze a new computational role of coherent multi-qubit tunneling that gives rise to bands of non-ergodic extended (NEE) quantum states each formed by a superposition of a large number of deep local minima with similar energies. NEE provide a coherent pathway for population transfer (PT) between computational states with similar energies. In this regime, PT cannot be efficiently simulated by QMC. PT can serve as a new quantum subroutine for quantum search, quantum parallel tempering and reverse annealing optimization algorithms. We study PT resulting from quantum evolution under a transverse field of an n-spin system that encodes the energy function $E(z)$ of an optimization problem over the set of bit configurations z. Transverse field is rapidly switched on in the beginning of algorithm, kept constant for sufficiently long time and switched off at the end. Given an energy function of a binary optimization problem and an initial bit-string with atypically low energy, PT protocol searches for other bitstrings at energies within a narrow window around the initial one. We provide an analytical solution for PT in a simple yet nontrivial model: M randomly chosen marked bit-strings are assigned energies $E(z)$ within a narrow strip $[-n-W/2, n+W/2]$, while the rest of the states are assigned energy 0. The PT starts at a marked state and ends up in a superposition of L marked states inside the narrow energy window whose width is smaller than W. The best known classical algorithm for finding another marked state is the exhaustive search. We find that the scaling of a typical PT runtime with n and L is the same as that in the multi-target Grover's quantum search algorithm, except for a factor that is equal to $\exp(n/(2B^2))$ for finite transverse field $B \gg 1$. Unlike the Hamiltonians used in analog quantum unstructured search algorithms known so far, the model we consider is non-integrable and the transverse field delocalizes the marked states. As a result, our PT protocol is not exponentially sensitive in n to the weight of the driver Hamiltonian and may be initialized with a computational basis state. We develop the microscopic theory of PT by constructing a down-folded dense Hamiltonian acting in the space of marked states of dimension M. It belongs to the class of preferred basis Levy matrices (PBLM) with heavy-tailed distribution of the off-diagonal matrix elements. Under certain conditions, the band of the marked states splits into minibands of non-ergodic delocalized states. We obtain an explicit form of the heavy-tailed distribution of PT times by solving cavity equations for the ensemble of down-folded Hamiltonians. We study numerically the PT subroutine as a part of quantum parallel tempering algorithm for a number of examples of binary optimization problems on fully connected graphs.

2012 ACM Subject Classification Theory of computation → Quantum computation theory, Theory of computation → Discrete optimization, Theory of computation → Machine learning theory

Keywords and phrases Quantum algorithms, Discrete optimization, Quantum spin glass, Machine learning

Digital Object Identifier 10.4230/LIPIcs.TQC.2018.9

1 Introduction

Analog quantum enhanced search and optimization algorithms could, at the very least, provide a stop-gap solution for quantum applications prior to fault-tolerant universal quantum computation [34]. Typically the classical cost function in binary optimization problems is encoded in a n-qubit Hamiltonian $H_{\mathrm{cl}} = \sum_z \mathcal{E}_z |z\rangle\langle z|$ diagonal in the computational basis $\{|z\rangle\}$. The energy landscape $\{\mathcal{E}_z\}$ of a typical hard optimization problem is characterized by a large number of spurious local minima. While close in energy, they can be separated by large Hamming distances. This landscape gives rise to an interesting computational primitive: given an initial bit-string with sufficiently low energy, we are to produce other bit-strings within a certain narrow range of energies $\Delta\mathcal{E}_{\rm cl}$ in the vicinity of the initial state. In general, this is a hard computational task. For instance, given a solution of a SAT problem finding another solution of a similar quality is generally as hard as finding the initial solution. Inspired by previous work in analogue quantum computing [22, 16, 8], we propose the following quantum population transfer (PT) protocol: given an initial computational state $|z_0\rangle$ with classical energy $\mathcal{E}_{z_0}$, we evolve with the Hamiltonian

$$H = H_{\rm cl} + H_D, \quad H_D = -B_\perp \sum_{j=0}^{n} \sigma_j^x, \tag{1}$$

where H_D is the driver Hamiltonian, without fine-tuning the evolution time or the strength of the transverse field $B_\perp$. Finally we measure in the computational basis and check the energy $\mathcal{E}_z$ of the outcome $|z\rangle$ if $z \neq z_0$. Practically, analog implementation requires rapid (diabatic) ramp on/off of the transverse field at the beginning/end of the protocol.

The distribution $\rho(E_\gamma)$ of the eigenvalues of H, where $H\,|\psi_\gamma\rangle = E_\gamma\,|\psi_\gamma\rangle$, is typically well localized around the mean classical energy, with an exponentially decaying tail reaching towards the low energy states. The initial state $|z_0\rangle$ is located at a deep local minimum of the classical landspace $\{\mathcal{E}_z\}$, at the tail the distribution $\rho(E_\gamma)$. The non-diagonal matrix elements $-B_\perp$ give rise to hopping between states separated by one bit-flip. Matrices with diagonal disorder and hoping between neighbors correspond to the Anderson model introduced in the context of transport and localization in disordered media [5]. The transverse field $B_\perp$ couples the local minima via perturbation theory in a high order given by the Hamming distance between them. In this model (1), as well as in the original Anderson model, there exist bands of localized and extended states separated in energy by a so-called “mobility edge”. It was demonstrated in Ref. [4] that localization is responsible for the failure of Quantum Annealing to find a solution of the constraint satisfaction problem (although, the detailed analysis of this effect is still lacking [24, 23]). Nonetheless, extended states could provide a mechanism for population transfer away from the initial state. In spin-systems with transverse field the existence of a so-called “mobility edge” at the tail of the distribution $\rho(E_\gamma)$, separating in the energy spectrum localized and delocalized eigenstates of H, has been recently studied in Refs. [37, 27, 31].

The population transfer corresponds to the formation of a “conduction band” that could be understood from the following arguments. We express the probability of a transition from $|z_0\rangle$ to $|z\rangle$,

$$P(t, z|z_0) = \left| \sum_\gamma \langle z|\psi_\gamma\rangle \, \langle\psi_\gamma|z_0\rangle \, e^{-iE_\gamma t} \right|^2 , \tag{2}$$

in terms of the eigenstates $\{|\psi_\gamma\rangle\}$ and eigenvalues $\{E_\gamma\}$ of the system Hamiltonian H. In the delocalized phase the state $|z_0\rangle$ has a sizable overlap with a large set of eigenstates of size Ω with energies within some range $|E_\gamma - E_{\gamma'}| \sim \Delta E$. These are the eigenstates that dominate the sum in Eq. (2). The eigenstates in this set posses an important property extensively studied in the theory of transport in disordered systems [26, 3, 2]. They have largely overlapping supports over a support set $\mathscr{S}$ of bit-strings. This implies, from Eq. (2), that after a typical population transfer time $t_{\rm PT} \propto 1/\Delta E$ and for any initial state $z_0 \in \mathscr{S}$ the population is spread over the entire set $\mathscr{S}$, that is $P(t_{\rm PT}, z_1|z_0) \sim 1/|\mathscr{S}|$ for all $z_1 \in \mathscr{S}$. The conduction band is formed by the eigenstates within a spectrum width ΔE associated with the bit-strings in $\mathscr{S}$. From the point of view of condensed matter physics, the eigenstates that overlap with $|z_0\rangle$ are non-ergodic. Nevertheless, they form a conduction miniband with energies below the quantum spin glass transition at the tail of $\rho(E_\gamma)$ [27, 2].

The formation of the conduction band explained above is the physical mechanism that we intend to exploit in the PT protocol to solve the computational primitive defined above. It is well established that simulating unitary time quantum dynamics in the delocalized phase to approximate $P(t_{\mathrm{PT}}, z_1|z_0)$ can not be done efficiently by known numerical techniques, such as quantum Monte-Carlo or tensor network methods, due to the coherent many-body nature of this transport phenomena. In addition to tunneling the transverse field $B_\perp$ gives rise to shifts in the classical energies $\mathcal{E}_{\mathrm{cl}}$ distributed over the width $\Delta\mathcal{E}_{\mathrm{cl}}$. This limits how narrow the target window of classical energies in the primitive can be.

More generally, the PT protocol can provide a useful primitive to explore energy landscapes on the way to lower energy states for optimization, reverse annealing [33] and quantum machine learning [10, 7]. The output of PT z can be used as an input of a classical optimization heuristic such as simulated annealing or parallel tempering in a "hybrid" optimization algorithm [32] where quantum and classical steps can be used sequentially to gain the complementary advantages of both [11]. A quantum advantage of the population transfer primitive would imply an advantage of such quantum parallel tempering over similar classical algorithms.

We propose a theoretical approach to analyze this problem with detailed analysis presented in [36]. Here we provide the results. Our approach exploits the existence of two relevant energy scales. The first scale is the width of the non-ergodic conduction miniband ΔE. The second scale is the typical change in classical energy $\mathcal{E}_{\mathrm{cl}}$ corresponding to one spin flip. Because of the large Hamming distances separating states in the support of the conduction band $\mathscr{S}$, the effecitive coupling elements that couple them correspond to high order in perturbation theory in $B_\perp$, and therefore $|\mathcal{E}_{\mathrm{cl}}| \gg \Delta E$. The dynamics within the miniband is described by the effective downfolded Hamiltonian

$$\mathscr{H} = \sum_{j=1}^{M} \varepsilon_j \, |j\rangle \langle j| + \sum_{j,k=1}^{M} V_{jk} \, |j\rangle \langle k| \ . \tag{3}$$

The sum is over a size M of the subset of computational basis states $|z_j\rangle$ sufficiently wide such that it contains the support set $\mathscr{S}$. The ε_j's are appropriately renormalized energies of the Hamiltonian H_{cl}. The non-diagonal matrix elements V_{jk} correspond to the sum over all elementary spin-flip processes that begin in state $|j\rangle$, proceed through virtual states separated by energies at least $\mathcal{E}$ from the miniband, and return back to the miniband only at the last step, at the state $|k\rangle$. We emphasize that in general V_{jk} takes into account all loops where the process returns back to the same virtual state without visiting the miniband.

In this paper we apply this analytical framework to "impurity band" model which demonstrates a quantum spin glass behavior yet allows analytical description of the quantum dynamics in the course of the PT protocol. In the second part of the paper we present numerical analysis of a set of more practical models defined by 2-local Hamiltonians. The impurity band model is defined by the Hamiltonian,

$$H = H_{\mathrm{cl}} - B_\perp \sum_{j=0}^{n} \sigma_j^x, \quad H_{\mathrm{cl}} = \sum_{j=1}^{M} \mathcal{E}(z_j) \, |z_j\rangle \langle z_j| \tag{4}$$

where the n-bit-strings $\{z_j\}_{j=1}^{M}$ are chosen uniformly at random from all bit-strings of length n, there are $M \gg 1$ marked states $|z_i\rangle$, with energies $\mathcal{E}_{z_j} = -n + \varepsilon_j$. The ε_j's are independently distributed around 0 with a narrow width $W \ll 1$ to be discussed below. All other states have energy 0 and are separated by a large gap $\sim n$ from the very narrow band of marked states. The typical distance between marked states is $n/2$. If M is exponentially large in n the typical distance $d_{\min}$ to the nearest marked state is much smaller than $n/2$ but remains extensive $d_{\min} = \mathcal{O}(n)$. As such, each marked state $|z_j\rangle$ is a deep local minimum of $\mathcal{E}(z)$ coupled to other marked states via transverse field induced multiqubit tunneling with amplitude decreasing exponentially with Hamming distance d.

We obtain an explicit analytical form for the statistical properties of the PT dynamics in the above model (4) by deriving in the form of Eq. (3) an effective down-folded Hamiltonian in the energy strip associated with the PT [36]

$$\mathscr{H}_{ij} = \delta_{ij}\epsilon_j + (1-\delta_{ij})\mathcal{V}_{ij}\sqrt{2}\sin\phi(d_{ij})\ . \tag{5}$$

Here the diagonal elements ϵ_j are given by the marked state energies counted off from the center of the impurity band shifted due to the effect of the transverse field $\sim B_\perp^2$. Their PDF is assumed to be exponentially bounded with some width W.

Explicit analytical form of the off-diagonal elements is obtained using WKB approach. In Eq. (5) $\phi(d) \equiv \phi(E^{(0)}, d)$ is a WKB phase that describes the oscillation of the matrix elements with the Hamming distance. The tunneling amplitude $\mathcal{V}_{ij}$ equals

$$\mathcal{V}_{ij} \equiv V(d_{ij}), \quad V(d) = \sqrt{A(d/n, B_\perp)}\ \frac{n^{5/4}\, e^{-n\theta(B_\perp)}}{\sqrt{\binom{n}{d}}}\ , \tag{6}$$

where $i \neq j$ and the coefficient $A(\rho, B_\perp)$ is a smooth function of its arguments. The function $\theta(B_\perp)$ is given in [36]. Below we use its asymptotical form in the limit $B_\perp \gg 1$,

$$\theta \simeq \frac{1}{4B_\perp^2} + \frac{1}{24B_\perp^4} + \frac{1}{60B_\perp^6} + \dots . \tag{7}$$

In this limit $\theta \ll 1$. We shall refer to $\mathscr{H}$ in (5) as the Impurity Band (IB) Hamiltonian.

The typical matrix element corresponds to tunneling at distance $n/2$ given by,

$$V_{\text{typ}} \sim n^2 2^{-n/2} e^{-n/(4B_\perp^2)}. \tag{8}$$

At the same time the typical distance from a marked state to its nearest neighbor is extensive $d_{\min} = \mathcal{O}(n)$ which corresponds to a matrix element, see Eq. (6), exponentially larger than the typical value V_{typ}. Therefore there is a hierarchy of off-diagonal matrix elements of $\mathscr{H}_{ij}$. The off diagonal matrix elements of random realizations of $\mathscr{H}$ are described by a heavy-tailed probability density function [14, 30]. Such random matrices are called Levi matrices.

The PDF of the rescaled squared amplitudes $w_{ij} = V^2(d_{ij})/V_{\text{typ}}^2$ can be obtained in the explicit form [36],

$$\text{PDF}(w) = \frac{1}{w^2\sqrt{\pi \log w}}, \quad w \in [1, \infty). \tag{9}$$

The particular form of scaling is the direct consequence of the fact that our problem has no "structure": the tunneling matrix elements depend only on Hamming distance and marked states are chosen at random.

The key difference of the ensemble of matrices $\mathscr{H}_{ij}$ from Levi matrices studied in the literature [38, 30, 29, 14] is that the dispersion, W, of the diagonal matrix elements is much larger than the typical magnitude of the off-diagonal elements V_{typ}. Therefore $\mathscr{H}_{ij}$ can be called preferred basis Levi matrices (PBLM).

We note that the existence of heavy tails in the PDF of the off-diagonal matrix elements of the down-folded Hamiltonian $\mathscr{H}$ is due to the infinite dimension of the Hilbert space of the original problem (1) for $n \to \infty$. This happens because the exponential decay of the matrix elements with the Hamming distance d is compensated by the exponential growth of the number of states at the distance d from a given state. We expect that this PBLM structure is a generic feature of the effective Hamiltonians for PT at the tail of the density of states in quantum spin glass problems.

The competition between the exponential decrease of the matrix element and the increase of the number of neighbors at distance d can result in eigenstates $|\psi_\beta\rangle$ of H associated with the impurity band becoming delocalized over a large subset of marked states $\mathscr{S}_\beta$ with size $1 \ll |\mathscr{S}_\beta| \propto M^\alpha$ and $0 < \alpha \leq 1$. For $\alpha = 0$ the eigenstate $|\psi_\beta\rangle$ is localized, for $\alpha = 1$ the eigenstate is delocalized in the entire space of marked states. For $0 < \alpha < 1$ the eigenstate can be considered "non-ergodic" and its support set $\mathscr{S}_\beta$ is sparse in the space of the marked states. The PBLM matrices support non-ergodic delocalized states when the width W is much bigger than the largest off-diagonal matrix element in a typical row of $\mathscr{H}_{ij}$ and much smaller than the largest off-diagonal element in a matrix

$$V_{\rm typ} M^{1/2} \ll W \ll V_{\rm typ} M \ . \tag{10}$$

For smaller dispersion $W \lesssim V_{\rm typ} M^{1/2}$ the matrix eigenstates are ergodic while for $W \gtrsim V_{\rm typ} M$ the eigenstates are localized. This non-ergodic regime is a distinct feature of the PBLM and is absent in Levi matrices. Such phase diagram resembles the one in the Rosenzweig-Porter (RP) model [26, 15]. The difference of RP from PBLM is that the statistics of the off-diagonal matrix elements in the RP ensemble are Gaussian [35] rather than polynomial (9). In this paper we focus on PT transfer within the non-ergodic delocalized phase, which is more likely to generalize to other models. We note that the localized phase does not support population transfer.

In the delocalized phase eigenstates with largely overlapping supports $\bigcap_\beta \mathscr{S}_\beta \approx \mathscr{S}(z_j)$ form narrow mini-bands. The mini-band width Γ may be interpreted as the inverse scrambling time and determines the width of the plateau in the Fourier-transform of the typical transition probability $\tilde{P}(\omega, z|z_j)$ [26].[2] In other words, the significant PT of $P(t, z|z_j)$ from the initial marked state $|z_j\rangle \in \mathscr{S}$ into other states of the same miniband $\mathscr{S}$ occurs over time $t_{\rm PT} \sim 1/\Gamma$.

Because of the PBLM structure of the Hamiltonian $\mathscr{H}$ one can expect that the runtime of the PT protocol $t_{\rm PT}$ will have a heavy-tailed PDF whose form is of practical interest. It is closely related to the PDF of the miniband widths $\Gamma \sim 1/t_{\rm PT}$. We obtained the PDF(Γ) using the cavity method for random symmetric matrices [1, 14, 9, 38].

Previously cavity equations were solved only in their linearized form, i.e., near the localization transition. We were able to solve the fully nonlinear cavity equations in the delocalized non-ergodic phase [36]. We obtained boundaries of the non-ergodic phase analytically in terms of the ratio of $W/V_{\rm typ}$ and the form of the PDF $\mathscr{P}(\Gamma)$ inside the phase. It is given by the alpha-stable Levi distribution [19, 14] with the tail index 1, see Fig. 1

$$\mathscr{P}(\Sigma'') = \frac{1}{C} L_1^{1,1} \left(\frac{\Sigma'' - \Sigma''_{\rm typ}}{C} \right) \ , \tag{11}$$

$$\Sigma''_{\rm typ} = \mu_\Omega \Sigma''_*, \quad C = \sigma_\Omega \Sigma''_* \ . \tag{12}$$

[2] The same plateau width characterizes the frequency dependence of the eigenfunction overlap correlation coefficient $K(\omega) = M \sum_{j=1}^{M} \sum_{\beta,\beta'} |\langle j|\psi_\beta\rangle|^2 |\langle j|\psi_{\beta'}\rangle|^2 \delta(\omega - E_\beta + E_{\beta'})$ [26].

Here $\Sigma''_{\rm typ}$ is a shift of the distribution and C its scale parameter (characteristic width) and we introduced a notation $\Sigma''_* = \pi V_{\rm typ}^2/(W/M)$.

$$\mu_\Omega \simeq \frac{1}{\sigma_\Omega} + \frac{2\sigma_\Omega(1-\gamma_{\rm Euler})}{\pi} \,. \tag{13}$$

$$\sigma_\Omega = \sqrt{\frac{\pi}{4\log\Omega}} \,. \tag{14}$$

Here Ω is the number of states in the miniband. This number $\Omega = (\pi M V_{\rm typ}/W)^2$ is a square function of the ratio of the typical tunneling matrix element $V_{\rm typ}$ to the level separation W/M.

We introduce the scaling of the width of the distribution of ϵ_m with the matrix size M,

$$W = \lambda M^{\gamma/2} V_{\rm typ} \,, \tag{15}$$

where γ is a real non-negative parameter that controls the scaling of the ratio of the typical diagonal to off-diagonal matrix element $V_{\rm typ}$ given in Eq. (8), and λ is an auxiliary constant of order one. With this scaling ansatz we get [36],

$$\Omega = \left(\frac{\pi}{\lambda}\right)^2 M^{2-\gamma} \,. \tag{16}$$

Using the above scaling ansatz (15) and the expressions for σ_Ω (14) and μ_Ω (13) we obtain,

$$\Sigma''_{\rm typ} \simeq \frac{2\pi^{1/2}}{\lambda} V_{\rm typ} M^{1-\gamma/2} (\log\Omega)^{1/2} \,, \tag{17}$$

$$C \simeq \frac{\pi^{3/2}}{2\lambda} V_{\rm typ} M^{1-\gamma/2} (\log\Omega)^{-1/2} \,. \tag{18}$$

The most probable value of the miniband width is $\Gamma_{\rm typ} = V_{\rm typ}(\pi\Omega\log\Omega/4)^{1/2}$, and its characteristic dispersion $\pi\Gamma_{\rm typ}/(4\log\Omega)$. In a non-ergodic delocalized phase $M \gg \Omega \gg 1$ and the typical PT time $t_{\rm PT} \sim 1/\Gamma_{\rm typ}$ obeys the condition

$$(M\log M)^{-1/2} \ll t_{\rm PT} V_{\rm typ} \sim (\Omega\log\Omega)^{-1/2} \ll 1 \,. \tag{19}$$

2 Complexity of the Population Transfer protocol

Starting at $t=0$ from a marked state $|z_j\rangle$ the probability for the population to be transferred to other marked states is $1-\psi^2(z_j,t)$. At the initial stage the survival probability $\psi^2(z_j,t)$ decays exponentially with the mean decay time $1/\Gamma_j = 1/(2\Sigma''_j)$.

The initial marked state $|z_j\rangle$ decays into the eigenstates $|\psi_\beta\rangle$ of the IB Hamiltonian $\mathscr{H}$ with typical energies E_β inside the narrow interval corresponding to the miniband associated with $|z_j\rangle$. It has width Σ''_j and is centered around $\mathscr{H}_{jj} = \epsilon_j$. Typical classical energies ϵ of the bit-strings measured at the end of PT protocol will obey the probability distribution $\mathcal{P}(\epsilon - \epsilon_j - \Sigma'_j)$ where Σ'_j is the self-energy shift of the marked state ϵ_j and $\mathcal{P}$ is the rescaled Cauchy distribution shown in Fig. 1 which reads [36],

$$\mathcal{P}(\Sigma') = \frac{1}{\pi}\frac{\Sigma'_{\rm typ}}{\left(\Sigma'_{\rm typ}\right)^2 + (\Sigma')^2}, \quad \Sigma'_{\rm typ} = \Sigma''_*\sqrt{\frac{4\log M}{\pi}}. \tag{20}$$

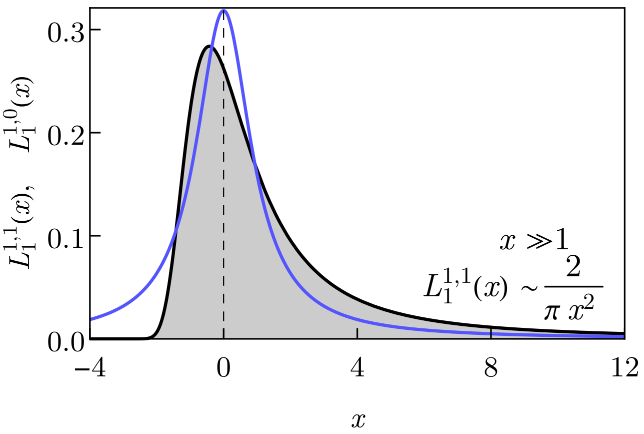


Figure 1 Black solid line shows the plot of the Levi alpha-stable distribution $L_\alpha^{C,\beta}(x)$ [14] with tail index $\alpha = 1$, asymmetry parameter $\beta = 1$ and unit scale parameter $C = 1$. Inset shows asymptotic behavior of the distribution at large positive x. At $-x \gg 1$ the function decays steeply as a double exponential, $\log L_1^{1,1}(x) \propto -\exp(-\frac{\pi}{2}x)$. Blue line shows the Cauchy distribution $L_1^{1,0}(x) = \frac{1}{\pi(1+x^2)}$. We follow here the definition introduced in [14] and used in subsequent papers on Levi matrices in physics literature. In mathematical literature [39] a different definition is usually used, corresponding to $f(x; \alpha, \beta, C^{1/\alpha}, 0) = L_\alpha^{C,\beta}(x)$.

The success of PT protocol is to find a bit-string distinct from z_j at a time t with energy inside the window $\Delta\mathcal{E}_{\rm cl}$ around ϵ_j. The PT success time therefore equals

$$t^j_{\rm PT} = \frac{1}{2\Sigma''_j p_{\Delta\mathcal{E}}}, \quad p_{\Delta\mathcal{E}} = \int_0^{\Delta\mathcal{E}_{\rm cl}} \mathcal{P}\left(\epsilon - \Sigma'_j - \frac{\Delta\mathcal{E}_{\rm cl}}{2}\right) d\epsilon \,.$$

Here $p_{\Delta\mathcal{E}}$ is the probability of detecting a bit-string inside the target window $\Delta\mathcal{E}_{\rm cl}$ under the condition that initial state has decayed. Assume that the PT window is as wide as the typical miniband width, $\Delta\mathcal{E}_{\rm cl} = \Sigma''_{\rm typ}$. In this case $p_{\Delta\mathcal{E}}$ differs from 1 only by a constant factor that does not depend on M. Therefore after a sufficiently long time a solution, a bit-string inside the PT window, is detected with finite probability. Because the initial state $|z_j\rangle$ is picked at random the typical time to success of PT $t_{\rm PT} \sim 1/\Sigma''_{\rm typ}$ corresponds to the inverse typical width of the miniband. All of the states in a miniband are populated at (roughly) the same time $t_{\rm PT}$ because the transition rate to a subset of states on a distance d away from $|z_j\rangle$ depends on d very weakly. This is a result of a cancellation between the combinatorial number $\binom{n}{d}$ of states (and hence decay channels) at distance d from a given marked state and the dependence of the matrix element squared on d, see (6).

We characterize the PT by the relation between the typical success time of PT $t_{\rm PT}$ and the number of states Ω over which the population is spread during PT

$$t_{\rm PT} \sim \frac{1}{V_{\rm typ}\sqrt{\Omega \log \Omega}} \sim \left(\frac{2^n}{n\Omega \log \Omega}\right)^{1/2} e^{2\theta n} \,, \tag{21}$$

where we set $\Delta\mathcal{E}_{\rm cl} \sim \Sigma''_*$ (see discussion above). The time $t_{\rm G}$ for the Grover algorithm for unstructured quantum search to find Ω items in a database of the size 2^n is $t_{\rm G} \sim (2^n/\Omega)^{1/2}$. PT time $t_{\rm PT}$ scales worse than Grover time $t_{\rm G}$ by an additional exponential factor $e^{2\theta n} \simeq e^{\frac{n}{2B_\perp^2}}$ (7). At large transverse fields $1 \ll B_\perp = \mathcal{O}(n^0)$ the scaling exponent is small $2\theta \ll 1$.

3 Comparison with the analogue Grover search

Inspired by the Hamiltonian version of Grover algorithm proposed in [18] we consider the PT protocol in the IB model $H_{\rm cl}$ starting from the ground state of H_D which is a fully symmetric state $|S\rangle = 2^{-n/2}\sum_{j=1}^{n}|z\rangle$ in a computational basis. This protocol can be implemented by adjusting the value of transverse field $B_\perp \approx 1$ so that the ground state energy of the driver is set near the center of the IB. Then we can replace the full driver with the projector on its ground state, $H_D \to -nB_\perp |S\rangle\langle S|$. The quantum evolution is described by the Hamiltonian

$$H_{\rm G} = -nB_\perp |S\rangle\langle S| + \sum_{j=1}^{M}\mathcal{E}(z_j)\,|z_j\rangle\langle z_j| \ , \tag{22}$$

with the initial condition $|\psi(0)\rangle = |S\rangle$. In the case where all impurity energies are equal to each other, $\{\mathcal{E}(z_j) = -n\}_{j=1}^{M}$, and $B_\perp = 1$ the Hamiltonian $H_{\rm G}$ is a generalization of the analog version of Grover search [18] for the case of M target states. The system performs Rabi oscillations between the initial state $|S\rangle$ and the state which is an equal superposition of all marked (solution) states. Time to solution is the half-period of the oscillations, the "Grover time" $t_{\rm G}$

$$t_{\rm G} = \frac{\pi}{2nB_\perp}\sqrt{\frac{2^n}{M}} \ . \tag{23}$$

Hamiltonian versions of Grover search with transverse field driver whose ground state were tuned at resonance with that of the solution state were considered in [17, 12].

We assume as before that marked state energies take distinct values $\mathcal{E}(z_j) = -n + \epsilon_j$ randomly distributed over some narrow range W. We investigate the effect of systematic error in the Grover diffusion operator [20]. In the Hamiltonian formulation [18] this corresponds to the deviation from unity of the parameter $B_\perp$ that controls the weight of the driver in (23). We will define the driver error ϵ_0 by,

$$B_\perp = 1 - \frac{\epsilon_0}{n} \ . \tag{24}$$

Assuming that $N \gg M$ one can instead of the state $|S\rangle$ consider the decay of the state $|0\rangle \equiv \frac{1}{\sqrt{N-M}}\sum_{j=M+1}^{N}|z_j\rangle$, where $j \in [1, M]$ corresponds to marked states. We use (24) and omit constant terms and small corrections $\mathcal{O}(M/N)$ in H_G. The non-zero matrix elements $H_G^{ij} = \langle i|\,H_G\,|j\rangle$ in this subspace $\mathscr{S}$ have the form

$$H_G^{jj} = \epsilon_j, \quad H_G^{j0} = -(1-\delta_{j0})V, \quad V = n2^{-n/2} \ , \tag{25}$$

where $j \in [0, M]$ and $H_G^{j0} = H_G^{0j}$. On a time scale $t \ll 1/\delta\epsilon = M/W$ much smaller than the inverse spacing of the energies ϵ_j the quantum evolution with initial condition $|\psi(0)\rangle = |0\rangle$ corresponds to the decay of the discrete state with energy ϵ_0 into the continuum [28] with the finite spectral width W [25]. We assume that $\epsilon_0 \gg W$ while the spread of the marked state energies $W \lesssim V\sqrt{M}$, so that absent driver errors, PT time would follow a Grover-like scaling law $t \sim 1/(V\sqrt{M})$.

The state $|0\rangle$ is coupled non-resonantly to a continuum with narrow bandwidth. The expression for the population transfer to the marked states can be obtained from the time-dependent perturbation theory in the parameter ϵ_0/W

$$\sum_{m=1}^{M}|\psi_m(t)|^2 = \frac{2MV^2}{\epsilon_0^2}\left(1 - \cos(\epsilon_0 t)\frac{\sin(Wt/2)}{Wt/2}\right) \ .$$

Maximum transfer occurs at the time $t_0 = \pi/\epsilon_0$ with the total transferred probability $p_0 = 4MV^2/\epsilon_0^2$. Typical time $t_{\rm PT} \simeq t_0/p_0$ to achieve the successful population transfer to marked states involves repeating the experiment $1/p_0$ times

$$t_{\rm PT} = \frac{1}{\Gamma_0}\frac{\pi^2\epsilon_0}{W}\ , \tag{26}$$

where $\Gamma_0 = 2\pi V^2/(W/M)$ and the first factor in r.h.s gives the typical transfer time in absence of driver errors. Errors increase the transfer time by a large factor ϵ_0/W.

For the maximum possible bandwidth W when nearly all states are populated, $W \sim \Gamma_0{\sim}V\sqrt{M}$, the time of population transfer (26) is

$$t_{\rm PT} \sim t_{\rm G}\,(t_{\rm G}\epsilon_0) \quad (\epsilon_0 \gg t_{\rm G}^{-1} \sim V\sqrt{M})\ . \tag{27}$$

As expected, when the driver error exceeds inverse Grover time $1/t_{\rm G}$ the performance of analogue Grover algorithms (22) degrades relative to $t_{\rm G}$. This is a direct consequence of the fact that the quantum evolution begins from fully symmetric state which is a ground state of the driver Hamiltonian whose energy is tuned at resonance with the marked states. In this case the transverse field Hamiltonian driver effectively corresponds to the projector (22). Because the ground state is not degenerate, the resonance region is exponentially narrow ($\sim 2^{-n/2}\sqrt{M}$). This results in the exponential sensitivity of the Grover algorithm performance to the value of driver weight. This critical behavior was studied in the work on quantum spatial search [13] for the case of one marked state.

In contrast, in the PT protocol considered in this paper there was no need to fine-tune the value of $B_\perp$ other than making it large, $B_\perp \gg 1$. This happened because the effective coupling between the marked states described by the down-folded Hamiltonian $\mathscr{H}$ (5) was not due to any one particular eigenstate of the driver (such as the state $|S\rangle$ for the Grover case). Instead this coupling was formed due to an exponentially large (in n) number of non-resonant, virtual transitions between the marked states and highly exited states of the transverse field Hamiltonian H_D. This resulted in a significant improvement in robustness for the proposed PT relative to the analogue Grover algorithm.

4 Numerical simulations of population transfer

We illustrate by numerical simulations the potential of the PT subroutine for hybrid quantum-classical search and optimization algorithms. We consider a model defined by the following 2-local Hamiltonian,

$$\mathcal{H} = \mathcal{H}_{\rm cl} + \Gamma\sum_{i,j}^{n}\left(\delta_{ij}(|h_i|+1)\sigma_i^x + (|J_{ij}|+1)\sigma_i^x\sigma_j^x\right),\ \ \mathcal{H}_{\rm cl} = \sum_{i=1}^{n} h_i\sigma_i^z + \sum_{i,j}^{n} J_{ij}\sigma_i^z\sigma_j^z, \tag{28}$$

where δ_{ij} is the Kronecker symbol, $J_{ij} \in [-1,1]$ and $h_i \in [-1,1]$ are uniformly distributed random numbers with finite 6-bit precision. A subset of $n/2$ bonds (i,j) are chosen to be dimers, non-overlapping pairs of spins with strong ferromagnetic interactions $J_{ij} = -4$. Note that we are using "matched" driver whose strength scales with the total longitudinal field acting on a given qubit.

We simulate of $n = 25$ qubit system. A single bit flip steepest descent (SD) algorithm starting from all possible bistrings identifies all local minima of a given realization of the model $\mathcal{H}_{\rm cl}$ in Eq. (28). Optimization of simulated annealing parameters for this type of

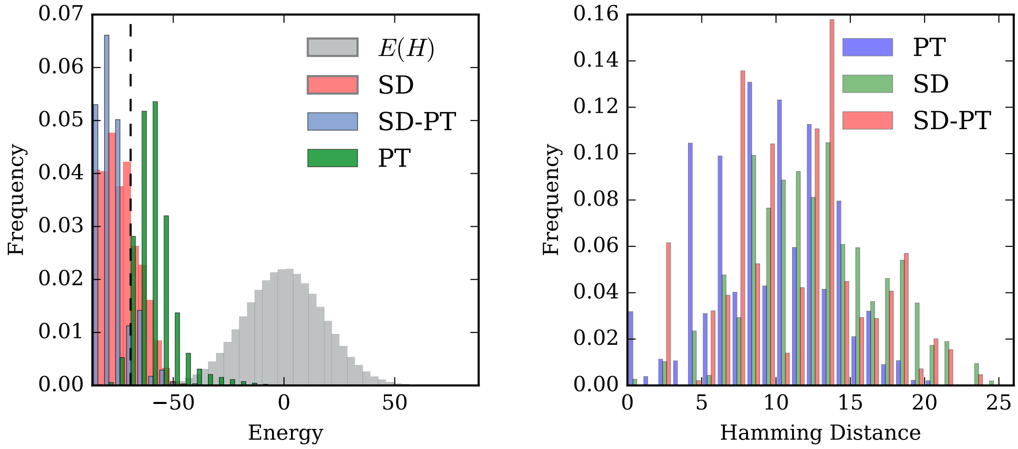


Figure 2 *Left panel* shows a histogram of normalized weights of classical energies in $H_{cl}^{(2)}$. $E(H)$ is the density of states in the original Hamiltonian spectrum. The classical steepest descent (SD) distribution shows the probability of ending up in a local minimum following a steepest descent run performed greedily by single spin flips. PT histogram shows the weight of classical energies in the output wave function following a population transfer (PT) run relative to the initial state energy (dotted black line). The SD-PT run is the distribution of local minimum starting from a state measured from the PT state. We note that the SD and SD-PT distributions are plotted as adjacent histograms to increase visibility of the data, but depicted bins are actually overlapping and not alternating. *Right panel* shows the distribution of Hamming distances from the initial state for states that fall within 1 standard deviation of the energy from the peak of the population transfer (PT) distribution for several methods.

instances suggested that SD (low temperature limit) is near optimal and therefore can serve as a proxy for hardness of finding a given bitstring, see histogram in the left panel of Fig. 2. Starting from a low-energy local minimum we perform a population transfer, Hamiltonian evolution with a fixed strength of the driver strength $\Gamma = 0.2$ for sufficiently long time. The quantum evolution is simulated using Trotter decomposition with 300 steps. The evolution time is chosen sufficiently long for the population transfer to approximately saturate. The histogram of weights of classical energies in the output wave function, Fig. 2, shows significant weight remaining in the low energy region of the spectrum in the vicinity of the initial bitstring energy. The PT output wave function has support on bitstrings separated from the initial state by large Hamming distances, see right panel in Fig. 2. Moreover, repeated sampling from the PT output produces bitstrings separated by large Hamming distances from each other, see left panel in Fig. 3. Bitstrings sampled from the PT output wave function can be used as starting states for SD which finds some of the low energy minima with higher probability than SD initialized with uniformly random bitstrings, see right panel in Fig. 3. Therefore PT protocol provides a quantum coherent pathway between low energy states that is complementary to SD and simulated annealing and therefore PT could potentially serve as useful subroutine in hybrid algorithms such as quantum parallel tempering.

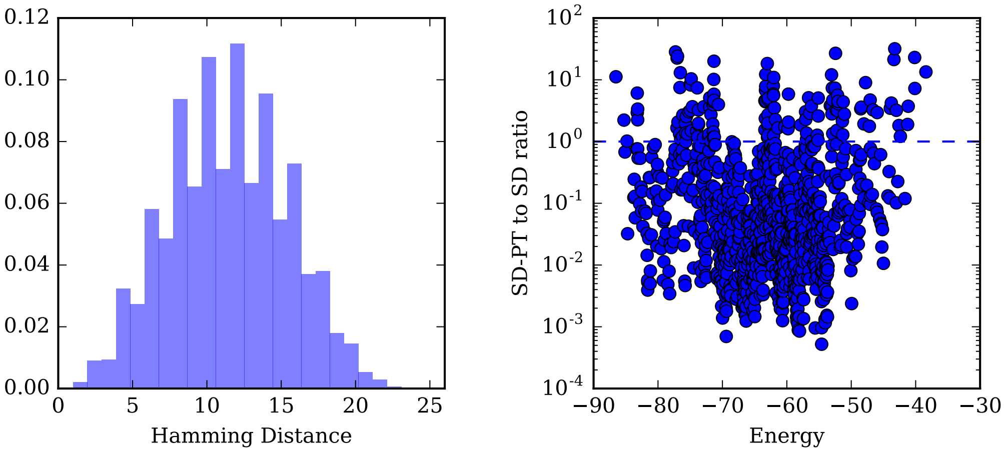


Figure 3 *Left panel* shows the histogram of Hamming distance between all pairs of states within 1 standard deviation of the energy from the peak of the population transfer (PT) distribution. Hamming distances are weighted by their joint probability within the PT distribution. The 0 hamming distance self contributions are excluded. The even-odd pattern that is observed results from the dimer structure of the Hamiltonian and the corresponding match driver. *Right panel* plots every local minimum with respect to single spin flips in the spectrum, which is concentrated to the low energy sector of the total Hamiltonian. For each local minimum we depict the ratio of the probability of ending up in that local minimum when starting from the PT distribution against a uniformly random initial state. We see that a significant fraction of states are enriched in the PT case, including the global minimum.

5 Conclusion

We analyze the computational role of coherent multiqubit tunneling that gives rise to bands of nonergodic delocalized quantum states as a coherent pathway for population transfer (PT) between computational states with close energies. In this regime PT cannot be efficiently simulated by QMC.

We solve this problem using the following quantum population transfer (PT) protocol: prepare the system in a computational state $|z_j\rangle$ with classical energy $\mathcal{E}(z_j)$, then evolve it with the transverse-field quantum spin Hamiltonian. Classical energies $\mathcal{E}(z)$ are encoded in the problem Hamiltonian diagonal in the basis of states $|z\rangle$. The key difference between PT protocol and QA [22, 16, 8] or analogue quantum search Hamiltonians [18, 13] is that the transverse field is kept constant throughout the algorithm and is not fine-tuned to any particular value. At the final moment of PT we projectively measure in the computational basis and check if the outcome z is a “solution”, i.e., $z \neq z_j$, and the energy $\mathcal{E}(z)$ is inside the window $\Delta\mathcal{E}_{\rm cl}$.

In this paper we analyzed PT dynamics in Impurity Band (IB) model with a “bimodal” energy function: $\mathcal{E}(z) = 0$ for all states except for M “marked” states $|z_j\rangle$ picked at random with energies forming a narrow band of the width W separated by a large gap $\mathcal{O}(n)$ from the rest of the states. This landscape is similar to that in analogue Grover search [18, 17] with multiple target states and a distribution of oracle values for the targets. The best known classical algorithm for finding another marked state has cost $O(2^n/M)$.

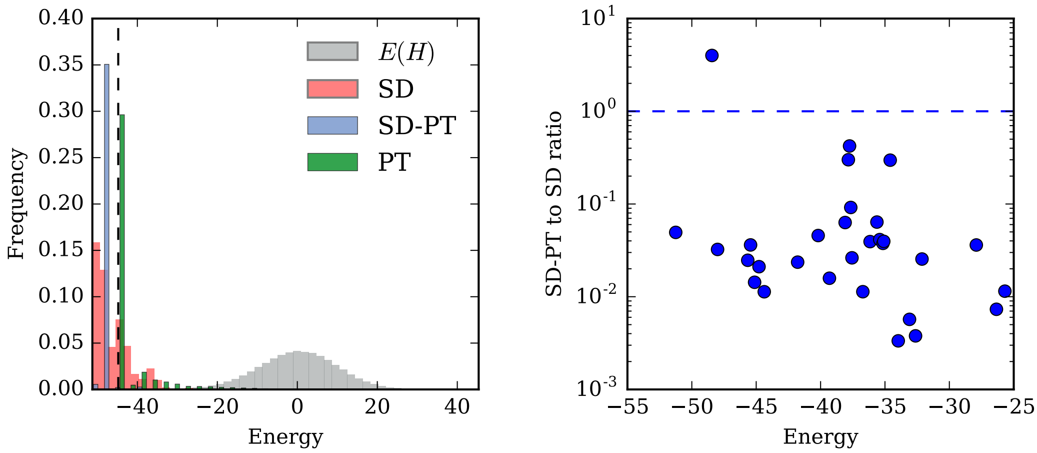


Figure 4 For comparison we include the analog of Figs. 2 and 3 (for $n = 22$) for a model without dimers, where SD performs better.

The PT dynamics is described by the down-folded $M \times M$ Hamiltonian $\mathscr{H}$ that is dense in the space of the marked states $|z_j\rangle$. The distribution of matrix elements $\mathscr{H}_{ij}$ has a heavy tail decaying as a cubic power for $V(d) \gg V_{\mathrm{typ}}$. This is a remarkable result of the competition between the very steep decay of the off-diagonal tunneling matrix element with the Hamming distance d, and the steep increase in the number of marked states $M_d \propto \binom{n}{d}$ at distance d. We emphasize that such polynomial tail in the distribution of matrix elements is only possible either in infinite dimension or in presence of long-range interactions (e.g, dipolar glass).

The dispersion of the diagonal elements $\mathscr{H}_{jj} = \mathcal{E}(z_j)$ is expected to be large, $W \sim V_{\mathrm{typ}} M^{\gamma/2} \gg V_{\mathrm{typ}}$ with $\gamma \in [1, 2]$. In the range $1 < \gamma < 2$ there exist minibands of non-ergodic delocalized eigenstates of $\mathscr{H}$. Their width is proportional to $1/t_{\mathrm{PT}} \ll W$. Each miniband associated with a support set $\mathscr{S}$ over the marked states.

The distribution of miniband widths Γ obeys alpha-stable Levi law with tail index 1. The typical value of Γ and its characteristic variance exceeds the typical matrix element of $\mathscr{H}$ by a factor $\Omega^{1/2}$ where $\Omega = (MV_{\mathrm{typ}}/W)^2$ is a size of the support set in a typical miniband.

We demonstrate that quantum PT finds another state within a target window of energies Ω in time $t_{\mathrm{PT}} \propto 2^{n/2}\Omega^{-1/2}\exp(n/(2B_\perp^2))$. The scaling exponent of t_{PT} with n differs from that in Grover's algorithm by a factor $\propto B_\perp^{-2}$, which can be made small with large transverse fields $n \gg B_\perp^2 \gg 1$.

Crucial distinctions between this case and the Hamiltonian in the analogue version of Grover's algorithm [18] for the case of multiple target states are the non-integrability of our model, and the delocalized nature of the eigenstates within the energy band W. Furthermore, analogue Grover's algorithm for multiple targets is exponentially sensitive in n to the weight of the driver Hamiltonian, and cannot be initialized with a computational basis state.

The quantum spin Hamiltonian in (1) can be described using n-body, infinite-range interactions. However, it shares key properties with the infinite range spin-glass models involving only p-body interactions at finite values of p >2 that could be implemented on a quantum computer with polynomial in n resources. The evidence of non-ergodic extended states were recently uncover numerically in the low-energy part of the spectrum of the quantum transverse field p-spin model [6].

Similar to the model (1) the low-energy part of the spectrum of transverse field p-spin model is characterized by the proliferation of statistically independent deep local minima separated by large, O(n), number of spin flips. Model of this type are characterized by a RSB-1 type of spin glass transition and were studied in the context of quantum annealing [21]. They represent perhaps the next step to study PT algorithms.

Finally, we analyzed numerically the PT initiated at a low energy local minimum of a 2-local spin glass model and observed that sampling PT output together with subsequent application of classical steepest descent allows exploring the energy landscape in a way complimentary to steepest descent and simulated annealing. This suggest possible use of PT as a subroutine for hybrid quantum-classical algorithms for search and optimization such as quantum paralleled tempering.

References

1 Ragi Abou-Chacra, DJ Thouless, and PW Anderson. A selfconsistent theory of localization. *Journal of Physics C: Solid State Physics*, 6(10):1734, 1973.

2 BL Altshuler, E Cuevas, LB Ioffe, and VE Kravtsov. Nonergodic phases in strongly disordered random regular graphs. *Physical Review Letters*, 117(15):156601, 2016.

3 BL Altshuler, LB Ioffe, and VE Kravtsov. Multifractal states in self-consistent theory of localization: analytical solution. *arXiv preprint arXiv:1610.00758*, 2016.

4 Boris Altshuler, Hari Krovi, and Jérémie Roland. Anderson localization makes adiabatic quantum optimization fail. *Proceedings of the National Academy of Sciences*, 107(28):12446–12450, 2010.

5 Philip W Anderson. Absence of diffusion in certain random lattices. *Physical review*, 109(5):1492, 1958.

6 C. L. Baldwin, C. R. Laumann, A. Pal, and A. Scardicchio. Clustering of nonergodic eigenstates in quantum spin glasses. *Phys. Rev. Lett.*, 118:127201, Mar 2017. `doi:10.1103/PhysRevLett.118.127201`.

7 Jacob Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549(7671):195, 2017.

8 J Brooke, D Bitko, G Aeppli, et al. Quantum annealing of a disordered magnet. *Science*, 284(5415):779–781, 1999.

9 Zdzisław Burda, Jerzy Jurkiewicz, Maciej A Nowak, Gabor Papp, and Ismail Zahed. Free random lévy and wigner-lévy matrices. *Physical Review E*, 75(5):051126, 2007.

10 Giuseppe Carleo and Matthias Troyer. Solving the quantum many-body problem with artificial neural networks. *Science*, 355(6325):602, 2017.

11 Nicholas Chancellor. Modernizing quantum annealing using local searches. *New Journal of Physics*, 19(2):023024, 2017.

12 Andrew M Childs, Enrico Deotto, Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Andrew J Landahl. Quantum search by measurement. *Physical Review A*, 66(3):032314, 2002.

13 Andrew M Childs and Jeffrey Goldstone. Spatial search by quantum walk. *Physical Review A*, 70(2):022314, 2004.

14 Pierre Cizeau and Jean-Philippe Bouchaud. Theory of lévy matrices. *Physical Review E*, 50(3):1810, 1994.

15 Davide Facoetti, Pierpaolo Vivo, and Giulio Biroli. From non-ergodic eigenvectors to local resolvent statistics and back: A random matrix perspective. *EPL (Europhysics Letters)*, 115(4):47003, 2016.

16 Edward Farhi, Jeffrey Goldstone, Sam Gutmann, Joshua Lapan, Andrew Lundgren, and Daniel Preda. A quantum adiabatic evolution algorithm applied to random instances of an np-complete problem. *Science*, 292(5516):472–475, 2001.

17 Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*, 2000.

18 Edward Farhi and Sam Gutmann. Analog analogue of a digital quantum computation. *Physical Review A*, 57(4):2403, 1998.

19 BV Gnedenko and AN Kolmogorov. Limit distributions for sums of independent. *Am. J. Math.*, 105, 1954.

20 Lov K Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79(2):325, 1997.

21 Thomas Jörg, Florent Krzakala, Jorge Kurchan, and A. C. Maggs. Simple glass models and their quantum annealing. *Phys. Rev. Lett.*, 101:147204, Oct 2008. `doi:10.1103/PhysRevLett.101.147204`.

22 Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Physical Review E*, 58(5):5355, 1998.

23 Sergey Knysh. Zero-temperature quantum annealing bottlenecks in the spin-glass phase. *Nature communications*, 7:12370, 2016.

24 Sergey Knysh and Vadim Smelyanskiy. On the relevance of avoided crossings away from quantum critical point to the complexity of quantum adiabatic algorithm. *arXiv preprint arXiv:1005.3011*, 2010.

25 Eugene Kogan. On the analytic structure of green's function for the fano-anderson model. *arXiv preprint quant-ph/0611043*, 2006.

26 V E Kravtsov, I M Khaymovich, E Cuevas, and M Amini. A random matrix model with localization and ergodic transitions. *New Journal of Physics*, 17(12):122002, 2015. URL: `http://stacks.iop.org/1367-2630/17/i=12/a=122002`.

27 Christopher R Laumann, A Pal, and A Scardicchio. Many-body mobility edge in a mean-field quantum spin glass. *Physical Review Letters*, 113(20):200405, 2014.

28 Gerald D Mahan. *Many-particle physics*. Springer Science & Business Media, 2013.

29 Fernando Lucas Metz, Izaak Neri, and Désiré Bollé. Localization transition in symmetric random matrices. *Physical Review E*, 82(3):031135, 2010.

30 Cécile Monthus. Localization transition in random lévy matrices: multifractality of eigenvectors in the localized phase and at criticality. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(9):093304, 2016.

31 Gianni Mossi and Antonello Scardicchio. Many body localization transition in quantum spin glasses on the bethe lattice. *arXiv preprint arXiv:1703.03678*, 2017.

32 Hartmut Neven. Enhancing simulated annealing with quantum annealing, December 2015. US Patent Application Publication, PCT/US2016/068400. URL: `https://patents.google.com/patent/WO2017117016A1`.

33 M. Ohkuwa, H. Nishimori, and D. A. Lidar. Reverse annealing for the fully connected p-spin model. *ArXiv e-prints*, 2018. `arXiv:1806.02542`.

34 J. Preskill. Quantum Computing in the NISQ era and beyond. *ArXiv e-prints*, 2018. `arXiv:1801.00862`.

35 Norbert Rosenzweig and Charles E Porter. " repulsion of energy levels" in complex atomic spectra. *Physical Review*, 120(5):1698, 1960.

36 V. N. Smelyanskiy, K. Kechedzhi, S. Boixo, S. V. Isakov, H. Neven, and B. Altshuler. Non-ergodic delocalized states for efficient population transfer within a narrow band of the energy landscape. *arXiv preprint arXiv:1802.09542*, 2018. `arXiv:1802.09542`.

37 Vadim N Smelyanskiy, Udo v Toussaint, and Dogan A Timucin. Dynamics of quantum adiabatic evolution algorithm for number partitioning. *arXiv preprint quant-ph/0202155*, 2002.

38 Elena Tarquini, Giulio Biroli, and Marco Tarzia. Level statistics and localization transitions of levy matrices. *Physical Review Letters*, 116(1):010601, 2016.

39 Johannes Voit. *The statistical mechanics of financial markets*. Springer Science & Business Media, 2013.

Quantum Network Code for Multiple-Unicast Network with Quantum Invertible Linear Operations

Seunghoan Song
Graduate School of Mathematics, Nagoya University, Nagoya, Japan
m17021a@math.nagoya-u.ac.jp

Masahito Hayashi
Graduate School of Mathematics, Nagoya University, Nagoya, Japan
Centre for Quantum Technologies, National University of Singapore, Singapore, Singapore
Shenzhen Institute for Quantum Science and Engineering, Southern University of Science and Technology, Shenzhen, China
masahito@math.nagoya-u.ac.jp

Abstract

This paper considers the communication over a quantum multiple-unicast network where r sender-receiver pairs communicate independent quantum states. We concretely construct a quantum network code for the quantum multiple-unicast network as a generalization of the code [Song and Hayashi, arxiv:1801.03306, 2018] for the quantum unicast network. When the given node operations are restricted to invertible linear operations between bit basis states and the rates of transmissions and interferences are restricted, our code certainly transmits a quantum state for each sender-receiver pair by n-use of the network asymptotically, which guarantees no information leakage to the other users. Our code is implemented only by the coding operation in the senders and receivers and employs no classical communication and no manipulation of the node operations. Several networks that our code can be applied are also given.

2012 ACM Subject Classification Hardware → Quantum communication and cryptography

Keywords and phrases Quantum network code, Multiple-unicast quantum network, Quantum invertible linear operation

Digital Object Identifier 10.4230/LIPIcs.TQC.2018.10

Acknowledgements SS is supported by Rotary Yoneyama Memorial Master Course Scholarship (YM). This work was supported in part by a JSPS Grant-in-Aids for Scientific Research (A) No.17H01280 and for Scientific Research (B) No.16KT0017, and Kayamori Foundation of Information Science Advancement.

1 Introduction

When we transmit information via network, it is useful to make codes by reflecting the network structure. Such type of coding is called network coding and was initiated by Ahlswede et al. [1]. This topic has been extensively researched by many researchers. Network coding employs computation-and-forward in intermediate nodes instead of the naive routing method in traditional network communication. For the quantum network, the paper [5] started the discussion of the quantum network coding, and many papers [2, 9–12] have advanced the study of quantum network coding.

In the network coding, unicast network is the most basic network model that the entire network is used by a sender and a receiver. As one of the remarkable achievements of network

13th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2018).
Editor: Stacey Jeffery; Article No. 10; pp. 10:1–10:20
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

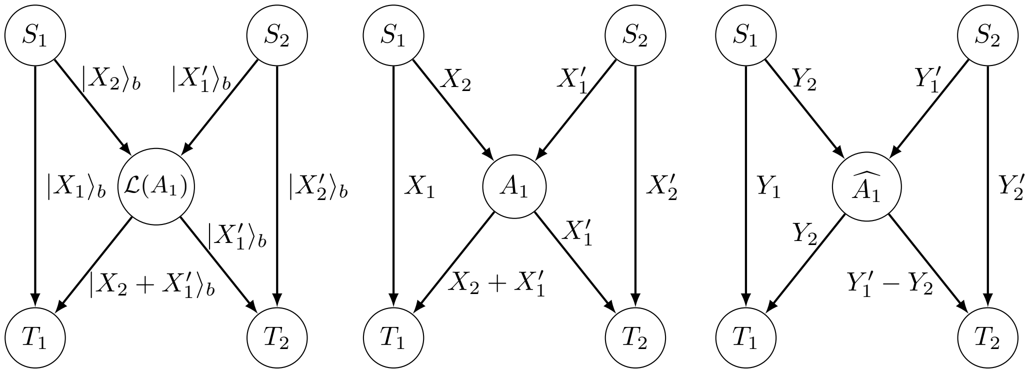
coding for the unicast network, on the classical linear network with malicious adversaries, the papers [6, 7] proposed codes that implement the classical communication by asymptotic n-use of the network. In [6, 7], when the transmission rate m in absence of attacks is at least the maximum rate a of attack (i.e., $a < m$), the codes in [6, 7] implement the rate $m - a$ communication asymptotically. As a quantum generalization of the codes in [6, 7], the paper [14] constructed a quantum network code that transmits a quantum state correctly and secretly by asymptotic n-use of the network. Similarly to [6, 7], when the transmission rate m without attacks is at least twice of the maximum number a of the attacked edges (i.e., $2a < m$), the code in [14] implements the rate $m-2a$ quantum communication asymptotically.

However, since a network is used by several users in general, it is needed to treat the network model with multiple users instead of the unicast network. For this purpose, the multiple-unicast network has been researched, in which disjoint r sender-receiver pairs $(S_1, T_1), \ldots, (S_r, T_r)$ communicate over a network. The paper [8] studied a quantum network code for the multiple-unicast network. The code in [8] transmits a state successfully for each use of the network. However, [8] has a limitation that the code should manipulate the node operations in the network and therefore the code depends on the network structure. In addition, the code in [8] requires the free use of the classical communication.

This paper proposes a quantum network code for the multiple-unicast network which is a generalization of the unicast quantum network code in [14] and overcomes the shortcomings of the multiple-unicast quantum network code in [8]. In the same way as [14], the given node operations are invertible linear with respect to the bit basis states, which is called *quantum invertible linear operations* described in Section 2, our code requires the asymptotic n-use of the network for the correct transmission of the state, and the encoding and decoding operations are performed on the input and output quantum systems of the n-use of the network, respectively. On the other hand, differently from [8], our code can be implemented without any manipulation of the network operations and any classical communication. Moreover, our code makes no information leakage asymptotically from a sender S_i to the receivers other than T_i because the correctness of the transmitted state guarantees no information leakage [13].

To discuss the achievable rate by our code, we consider the situation that the input states of all senders are the bit basis states. Then, our network can be considered as a classical network, called *bit classical network*, because a bit basis state is transformed to another bit basis state by our quantum node operations. In the bit classical network, we assume that when the inputs of the senders other than S_i are to zero, *the transmission rate* from S_i to T_i is m_i, which is the same as the number of outgoing edges of S_i and incoming edges of T_i. Also, when we define *the interference rate* by the rate of the transmitted information to T_i from the senders other than S_i, we assume that the interference rate to T_i is at most a_i in the bit classical network. In the same way, in case that the input states of all senders are set to the phase basis states (defined in Section 2), we call the network as *phase classical network.* In the phase classical network, we also assume that the transmission rate from S_i to T_i is m_i when the inputs of the senders other than S_i are zero. Also, the interference rate to T_i is at most a'_i in the phase classical network. Under these constraints, if $a_i + a'_i < m_i$, our code achieves the rate $m_i - a_i - a'_i$ quantum communication from S_i to T_i asymptotically.

To help the understanding of the rates described above, we explain the achievable transmission rate from S_1 to T_1 in the network in Fig. 1. The bit and the phase classical networks (Fig. 1b and Fig. 1c) are determined from the quantum network (Fig. 1a) (see Section 2). When $X'_1 = X'_2 = Y'_1 = Y'_2 = 0$, the transmission rates from S_1 to T_1 are 2 for both networks, i.e., $m_1 = 2$, which is also the number of outgoing edges of S_1 and incoming



(a) Quantum network **(b)** Bit classical network **(c)** Phase classical network

Figure 1 Toy example of a multiple-unicast network. In quantum network (a), $|\cdot\rangle_b$ denote bit basis states and $\mathcal{L}(A_1)$ is the network operation (see Section 2). The network (b) and (c) is the bit and phase classical networks of the quantum network (a).

edges of T_1. Also, the interference rates from S_2 to T_1 are 1 and 0 for the bit and the phase classical networks, respectively. On this network, if our code from S_1 to T_1 with the rates $(m_1, a_1, a'_1) = (2, 1, 0)$ is constructed, the conditions $a_1 \geq 1$, $a'_1 \geq 0$ and $a_1 + a'_1 < m_1$ are satisfied, and therefore our code implements the rate $m_1 - a_1 - a'_1 = 1$ quantum transmission from S_1 to T_1 asymptotically.

In the practical sense, our code can cope with the node malfunctions in the following case: on the multiple-unicast network with quantum invertible linear operations, the network operations are well-determined so that there is no interference between all sender-receiver pairs, but three broken nodes apply quantum invertible linear operations different from the determined ones. Moreover, let the transmission rate m_1 without interferences from S_1 to T_1 be 100 and the number of outgoing edges of the three broken nodes be 4. In this case, $3 \times 4 = 12$ outgoing edges of the three broken nodes transmit the unexpected information which implies the bit (phase) interference rate is at most 12. Therefore, by our code with $m_1 = 100$ and $a_1, a'_1 > 12$, the sender S_1 can transmit quantum states to the receiver T_1 correctly with the rate $100 - a_1 - a'_1 < 76$ by asymptotically many uses of the network.

The remaining of this paper is organized as follows. Section 2 introduces the formal description of the quantum multiple-unicast network with quantum invertible linear operations. Section 3 gives the main results of this paper. Based on the preliminaries in Section 4, Section 5 concretely constructs our code with the free use of negligible rate shared randomness. The encoder and decoder of our code is given in this section. Section 6 analyzes the correctness of the code in Section 5. Then, Section 7 constructs our code without the assumption of shared randomness by attaching the secret and correctable communication protocol [15] to the code given in Section 5, which proves the main result given in Section 3. Section 8 gives several examples of the network that our code can be applied. Section 9 is the conclusion of this paper.

2 Quantum Network with Invertible Linear Operations

Our code is designed on the quantum network which is a generalization of a classical multiple-unicast network. In this section, we first introduce the multiple-unicast network with classical invertible linear operations and generalize this network as a network with quantum invertible linear operations. The node operations introduced in this section are identical to the operations in [14, Section II].

2.1 Classical Network with Invertible Linear Operations

First, we describe the multiple-unicast network with classical invertible linear operations. The network topology is given as a directed Graph $G = (V, E)$. The r senders and r receivers are given as r source nodes $S_1, \dots, S_r$ and r terminal nodes $T_1, \dots, T_r$. The sender S_i has m_i outgoing edges and the receiver T_i has m_i incoming edges. Define $m := m_1 + \cdots + m_r$. The intermediate nodes are numbered from 1 to c $(= |V| - 2r)$ accordingly to the order of the transmission. The intermediate node numbered t has the same number k_t of incoming and outgoing edges where $1 \leq k_t \leq m$.

Next, we describe the transmission and the operations on this network. Each edge sends an element of the finite field $\mathbb{F}_q$ where q is a power of a prime number p. The t-th node operation is described as an invertible linear operation A_t from the information on k_t incoming edges to that of k_t outgoing edges. Since node operations are invertible linear, the entire network operation is written as $K = A_c \cdots A_1 \in \mathbb{F}_q^{m \times m}$. For the network operation K, we introduce the following notation:

$$K := \begin{bmatrix} K_{1,1} & K_{1,2} & \cdots & K_{1,r} \\ K_{2,1} & K_{2,2} & \cdots & K_{2,r} \\ \vdots & & \ddots & \vdots \\ K_{r,1} & K_{r,2} & \cdots & K_{r,r} \end{bmatrix}, \quad K_{i,j} \in \mathbb{F}_q^{m_i \times m_j}.$$

Then, $K_{i,j}$ is the network operation from S_i to T_j. We assume $\operatorname{rank} K_{i,i} = m_i$ which means the information from S_i to T_i is completely transmitted if there is no interference.

When the network inputs by senders $S_1, \dots, S_r$ are $x_1 \in \mathbb{F}_q^{m_1}, \dots, x_r \in \mathbb{F}_q^{m_r}$, the output $y_i \in \mathbb{F}_q^{m_i}$ at the receiver T_i $(i = 1, \dots, r)$ is written as

$$\begin{aligned} y_i &= \sum_{j=1}^{r} K_{i,j} x_j = K_{i,i} x_i + K_{i^c} z_{i^c}, \qquad (1) \\ K_{i^c} &:= [K_{i,1} \ \cdots \ K_{i,i-1} \ K_{i,i+1} \ \cdots \ K_{i,r}] \in \mathbb{F}_q^{m_i \times (m - m_i)}, \\ z_{i^c} &:= [x_1^{\mathrm{T}} \ \cdots \ x_{i-1}^{\mathrm{T}} \ x_{i+1}^{\mathrm{T}} \ \cdots \ x_r^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{F}_q^{m - m_i}. \end{aligned}$$

The second term $K_{i^c} z_{i^c}$ of (1) is called the interference to T_i, and $\operatorname{rank} K_{i^c}$ is called the rate of the interference to T_i.

Consider the n-use of the above network. When the inputs by senders $S_1, ..., S_r$ are $X_1 \in \mathbb{F}_q^{m_1 \times n}, \dots, X_r \in \mathbb{F}_q^{m_r \times n}$, the output $Y_i \in \mathbb{F}_q^{m_i \times n}$ at the receiver T_i $(i = 1, \dots, r)$ is

$$\begin{aligned} Y_i &= \sum_{j=1}^{r} K_{i,j} X_j = K_{i,i} X_i + K_{i^c} Z_{i^c}, \\ Z_{i^c} &:= [X_1^{\mathrm{T}} \ \cdots \ X_{i-1}^{\mathrm{T}} \ X_{i+1}^{\mathrm{T}} \ \cdots \ X_r^{\mathrm{T}}]^{\mathrm{T}} \in \mathbb{F}_q^{(m - m_i) \times n}. \end{aligned}$$

2.2 Quantum Network with Invertible Linear Operations

We generalize the multiple-unicast network with classical invertible linear operations to the network with quantum invertible linear operations. In this quantum network, the network topology is the same graph $G = (V, E)$. Each edge transmits a quantum system $\mathcal{H}$ which is q-dimensional Hilbert space spanned by the bit basis $\{|x\rangle_b\}_{x \in \mathbb{F}_q}$. In n-use of the network, we treat the quantum system $\mathcal{H}^{\otimes m_i \times n}$ spanned by the bit basis $\{|X\rangle_b\}_{X \in \mathbb{F}_q^{m_i \times n}}$. The sender S_i sends a quantum system $\mathcal{H}_{S_i} := \mathcal{H}^{\otimes m_i \times n}$ and the receiver T_i receives a quantum system $\mathcal{H}_{T_i} := \mathcal{H}^{\otimes m_i \times n}$

To describe the quantum node operation, we define the following quantum operations.

▶ **Definition 2.1** (Quantum Invertible Linear Operation). For invertible matrices $A \in \mathbb{F}_q^{m\times m}$ and $B \in \mathbb{F}_q^{n\times n}$, two unitaries $\mathcal{L}(A)$ and $\mathcal{R}(B)$ are defined for any $X \in \mathbb{F}_q^{m\times n}$ as

$$\mathcal{L}(A)|X\rangle_b := |AX\rangle_b, \quad \mathcal{R}(B)|X\rangle_b := |XB\rangle_b.$$

The operations $\mathcal{L}(A)$ and $\mathcal{R}(B)$ are called *quantum invertible linear operations.*

The t-th node operation is given as $\mathcal{L}(A_t)$ and it is called quantum invertible linear operation. The entire network operation is written as the unitary $\mathcal{L}(K) = \mathcal{L}(A_c \cdots A_1) = \mathcal{L}(A_c)\cdots\mathcal{L}(A_1)$. When a state ρ on $\mathcal{H}_{S_1} \otimes \cdots \otimes \mathcal{H}_{S_r}$ is transmitted by senders $S_1, \ldots, S_r$, the network output σ_{T_i} at $\mathcal{H}_{T_i}$ is written as

$$\sigma_{T_i} := \operatorname*{Tr}_{T_1,\ldots,T_{i-1},T_{i+1},\ldots,T_r} \mathcal{L}(K)\rho\mathcal{L}(K)^\dagger,$$

where $\mathrm{Tr}_{T_1,\ldots,T_{i-1},T_{i+1},\ldots,T_r}$ is the partial trace on the system $\mathcal{H}_{T_1} \otimes \ldots \otimes \mathcal{H}_{T_{i-1}} \otimes \mathcal{H}_{T_{i+1}} \otimes \ldots \otimes \mathcal{H}_{T_r}$.

When the input state on the network is $|M\rangle_b$ on $\mathcal{H}_{S_1} \otimes \cdots \otimes \mathcal{H}_{S_r}$, this quantum network can be considered as the classical network in Subsection 2.1. In the same way as the classical network, we assume $\operatorname{rank} K_{i,i} = m_i$ which means S_i transmits any bit basis states completely to T_i if the input states on source nodes S_j $(j \neq i)$ are zero bit basis states. Similarly, $\operatorname{rank} K_{i^c}$ is called the rate of the bit interference to T_i.

We can discuss the interference similarly on the phase basis $\{|z\rangle_p\}_{z\in\mathbb{F}_q}$ defined in [3, Section 8.1.2] by

$$|z\rangle_p := \frac{1}{\sqrt{q}} \sum_{x\in\mathbb{F}_q} \omega^{-\operatorname{tr} xz}|x\rangle_b,$$

where $\omega := \exp\frac{2\pi i}{p}$ and $\operatorname{tr} y := \operatorname{Tr} M_y$ $(y \in \mathbb{F}_q)$ with the multiplication map $M_y : x \mapsto yx$ identifying the finite field $\mathbb{F}_q$ with the vector space $\mathbb{F}_p^t$. For the analysis of the phase basis interference, we give Lemma 2.2 which explains how node operations $\mathcal{L}(A_t)$ are applied to the phase basis states.

▶ **Lemma 2.2** ([14, Appendix A]). *Let $A \in \mathbb{F}_q^{m\times m}$ and $B \in \mathbb{F}_q^{n\times n}$ be invertible matrices. For any $M \in \mathbb{F}_q^{m\times n}$, we have*

$$\mathcal{L}(A)|M\rangle_p = |(A^{\mathrm{T}})^{-1}M\rangle_p, \quad \mathcal{R}(B)|M\rangle_p = |M(B^{\mathrm{T}})^{-1}\rangle_p.$$

For notational convenience, we denote $\widehat{A} := (A^{\mathrm{T}})^{-1}$. When the input state is a phase basis state $|M\rangle_p$ on $\mathcal{H}_{S_1} \otimes \cdots \otimes \mathcal{H}_{S_r}$, the network operation $\mathcal{L}(K)$ is applied by $\mathcal{L}(K)|M\rangle_p = |\widehat{K}M\rangle_p$. In this case, this quantum network can also be considered as a classical network with network operation $\widehat{K} = \widehat{A_c}\cdots\widehat{A_1}$. Then, $\widehat{K}_{i,j}$ is defined from $\widehat{K}$ in the same way as $K_{i,j}$.

$$\widehat{K} := \begin{bmatrix} \widehat{K}_{1,1} & \widehat{K}_{1,2} & \cdots & \widehat{K}_{1,r} \\ \widehat{K}_{2,1} & \widehat{K}_{2,2} & \cdots & \widehat{K}_{2,r} \\ \vdots & \ddots & & \vdots \\ \widehat{K}_{r,1} & \widehat{K}_{r,2} & \cdots & \widehat{K}_{r,r} \end{bmatrix}, \quad \widehat{K}_{i,j} \in \mathbb{F}_q^{m_i\times m_j},$$
$$\widehat{K}_{i^c} := [\widehat{K}_{i,1} \ \cdots \ \widehat{K}_{i,i-1} \ \widehat{K}_{i,i+1} \ \cdots \ \widehat{K}_{i,r}].$$

Similarly to the condition $\operatorname{rank} K_{i,i} = m_i$, we also assume $\operatorname{rank} \widehat{K}_{i,i} = m_i$. We also call $\operatorname{rank} \widehat{K}_{i^c}$ the rate of phase interference to T_i. The transmission rates from S_i to T_i are summarized in Table 1.

Table 1 Definitions of Information Rates.

Rate	Meaning
$m_i = \operatorname{rank} K_{i,i} = \operatorname{rank} \widehat{K}_{i,i}$	Bit (phase) transmission rates from S_i to T_i without interference
$\operatorname{rank} K_{i^c}$	Rate of interference to T_i
$\operatorname{rank} \widehat{K}_{i^c}$	Rate of phase interference to T_i
a_i	Maximum rate of bit interference to T_i
a'_i	Maximum rate of phase interference to T_i

3 Main Results

In this section, we propose two main theorems of this paper. The two theorems state the existence of our code with and without negligible rate shared randomness, respectively. The codes stated in the theorems are concretely constructed in Section 5 and 7, respectively. The theorems are stated with respect to the completely mixed state ρ_{mix} and the *entanglement fidelity* $F_e^2(\rho, \kappa) := \langle x|\kappa \otimes \iota_R(|x\rangle\langle x|)|x\rangle$ for the quantum channel κ and a purification $|x\rangle$ of the state ρ.

▶ **Theorem 3.1.** *Consider the transmission from the sender S_i to the receiver T_i over a quantum multiple-unicast network with quantum invertible linear operations given in Section 2. Let m_i be the bit and phase transmission rates from S_i to T_i without interferences ($m_i = \operatorname{rank} K_{i,i} = \operatorname{rank} \widehat{K}_{i,i}$), and a_i, a'_i be the upper bounds of the bit and phase interferences, respectively ($\operatorname{rank} K_{i^c} \leq a_i$, $\operatorname{rank} \widehat{K}_{i^c} \leq a'_i$). When the condition $a_i + a'_i < m_i$ holds and the sender S_i and receiver T_i can share the randomness whose rate is negligible in comparison with the block-length n, there exists a quantum network code whose rate is $m_i - a_i - a'_i$ and the entanglement fidelity $F_e^2(\rho_{mix}, \kappa_i)$ satisfies $n(1 - F_e^2(\rho_{mix}, \kappa_i)) \to 0$ where κ_i is the quantum code protocol from sender S_i to receiver T_i.*

Section 5 constructs the code stated in Theorem 3.1 and Section 6 shows that this code has the performance in Theorem 3.1. Note that this code does not depend on the detailed network structure, but depends only on the information rates m_i, a_i and a'_i. As explained in [14, Section III], our code has no information leakage from the condition $n(1 - F_e^2(\rho_{mix}, \kappa_i)) \to 0$.

Although Theorem 3.1 assumed the free use of the negligible rate shared randomness, it is possible to design the code of the same performance without negligible rate shared randomness as follows. The paper [15] gives the secret and correctable classical network communication protocol for the classical network with malicious attacks, when the transmission rate is more than the sum of the rate of attacks and the rate of information leakage. By applying the protocol in [15] to our quantum network with bit basis states, the negligible rate shared randomness can be generated. By this method, we have the following Theorem 3.2 and the details are explained in Section 7.

▶ **Theorem 3.2.** *Consider the transmission from the sender S_i to the receiver T_i over a quantum multiple-unicast network with quantum invertible linear operations given in Section 2. Let m_i be the bit and phase transmission rates from S_i to T_i without interferences ($m_i = \operatorname{rank} K_{i,i} = \operatorname{rank} \widehat{K}_{i,i}$), and a_i, a'_i be the upper bounds of the bit and phase interferences, respectively ($\operatorname{rank} K_{i^c} \leq a_i$, $\operatorname{rank} \widehat{K}_{i^c} \leq a'_i$). When $a_i + a'_i < m_i$, there exists a quantum network code whose rate is $m_i - a_i - a'_i$ and the entanglement fidelity $F_e^2(\rho_{mix}, \kappa_i)$ satisfies $n(1 - F_e^2(\rho_{mix}, \kappa_i)) \to 0$ where κ_i is the quantum code protocol from sender S_i to receiver T_i.*

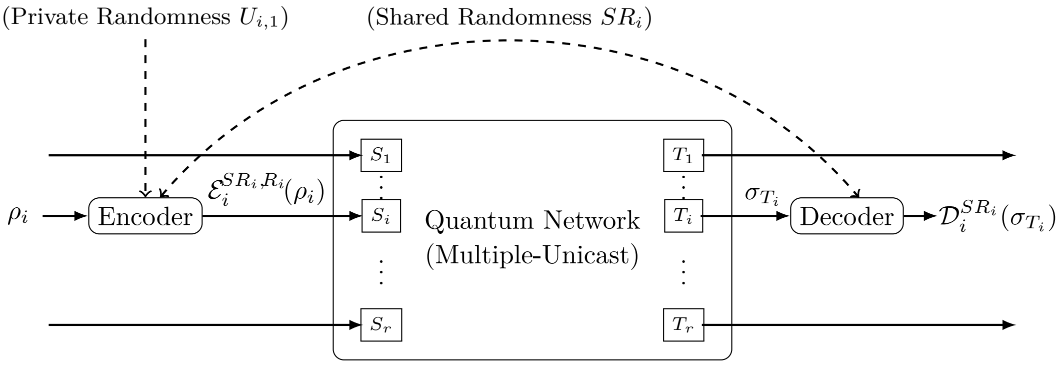

(Private Randomness $U_{i,1}$)
(Shared Randomness SR_i)
ρ_i
Encoder
$\mathcal{E}_i^{SR_i,R_i}(\rho_i)$
S_1
S_i
S_r
Quantum Network
(Multiple-Unicast)
T_1
T_i
T_r
σ_{T_i}
Decoder
$\mathcal{D}_i^{SR_i}(\sigma_{T_i})$

Figure 2 Overview of code protocol from a sender S_i to a receiver T_i. States ρ_i and $\mathcal{D}_i^{SR_i}(\sigma_{T_i})$ are in code space $\mathcal{H}'_{\text{code}}$.

4 Preliminaries for Code Construction

Before code construction, we prepare the extended quantum system, notations, and CSS code used in our code.

4.1 Extended Quantum System

Although the unit quantum system for the network transmission is $\mathcal{H}$, our code is constructed based on the extended quantum system $\mathcal{H}'$ described below.

First, dependently on the block-length n, we choose a power $q' := q^\alpha$ to satisfy $n \cdot (n')^{m_i}/(q')^{m_i-\max\{a_i,a_i\}} \to 0$ (e.g. $q' = O(n^{1+(\max\{a_i,a_i'\}+2)/(m_i-\max\{a_i,a_i'\})})$) where $n' := n/\alpha$. Let $\mathbb{F}_{q'}$ be the α-dimensional field extension of $\mathbb{F}_q$. Similarly, let $\mathcal{H}' := \mathcal{H}^{\otimes\alpha}$ be the quantum system spanned by $\{|x\rangle_b\}_{x\in\mathbb{F}_{q'}}$. Then, the n-use of the network over $\mathcal{H}$ can be considered as the n'-use of the network over $\mathcal{H}'$. The quantum invertible linear operations (Definition 2.1) can also be defined for invertible matrices $A' \in \mathbb{F}_{q'}^{m\times m}$ and $B' \in \mathbb{F}_{q'}^{n\times n}$ as

$$\mathcal{L}'(A)|X\rangle_b = |AX\rangle_b, \;\; \mathcal{R}'(B)|X\rangle_b = |XB\rangle_b, \quad \text{for any } X \in \mathbb{F}_{q'}^{m\times n}.$$

4.2 Notations for Quantum Systems and States in Our Code

We introduce notations used in our code. By the n-use of the network, the sender S_i transmits the system $\mathcal{H}_{S_i} = \mathcal{H}^{\otimes m_i\times n}$ and the receiver T_i receives the system $\mathcal{H}_{T_i} = \mathcal{H}^{\otimes m_i\times n}$, which are identical to $\mathcal{H}'^{\otimes m_i\times n'}$. We partition the quantum system $\mathcal{H}'^{\otimes m_i\times n'}$ as $\mathcal{H}'_{\mathcal{A}}\otimes\mathcal{H}'_{\mathcal{B}}\otimes\mathcal{H}'_{\mathcal{C}} := \mathcal{H}'^{\otimes m_i\times m_i}\otimes\mathcal{H}'^{\otimes m_i\times m_i}\otimes\mathcal{H}'^{\otimes m_i\times(n'-2m_i)}$. Furthermore, we partition the systems $\mathcal{H}'_{\mathcal{A}}, \mathcal{H}'_{\mathcal{B}}, \mathcal{H}'_{\mathcal{C}}$ by

$$\begin{aligned}
\mathcal{H}'_{\mathcal{A}} &= \mathcal{H}'_{\mathcal{A}1}\otimes\mathcal{H}'_{\mathcal{A}2}\otimes\mathcal{H}'_{\mathcal{A}3} := \mathcal{H}'^{\otimes a_i\times m_i}\otimes\mathcal{H}'^{\otimes(m_i-a_i-a_i')\times m_i}\otimes\mathcal{H}'^{\otimes a_i'\times m_i},\\
\mathcal{H}'_{\mathcal{B}} &= \mathcal{H}'_{\mathcal{B}1}\otimes\mathcal{H}'_{\mathcal{B}2}\otimes\mathcal{H}'_{\mathcal{B}3} := \mathcal{H}'^{\otimes a_i\times m_i}\otimes\mathcal{H}'^{\otimes(m_i-a_i-a_i')\times m_i}\otimes\mathcal{H}'^{\otimes a_i'\times m_i},\\
\mathcal{H}'_{\mathcal{C}} &= \mathcal{H}'_{\mathcal{C}1}\otimes\mathcal{H}'_{\mathcal{C}2}\otimes\mathcal{H}'_{\mathcal{C}3} := \mathcal{H}'^{\otimes a_i\times(n'-2m_i)}\otimes\mathcal{H}'^{\otimes(m_i-a_i-a_i')\times(n'-2m_i)}\otimes\mathcal{H}'^{\otimes a_i'\times(n'-2m_i)}.
\end{aligned}$$

For states $|\phi\rangle \in \mathcal{H}'_{\mathcal{A}1}, |\psi\rangle \in \mathcal{H}'_{\mathcal{A}2}$, and $|\varphi\rangle \in \mathcal{H}'_{\mathcal{A}3}$, the tensor product state in $\mathcal{H}'_{\mathcal{A}}$ is denoted as

$$\begin{bmatrix} |\phi\rangle \\ |\psi\rangle \\ |\varphi\rangle \end{bmatrix} := |\phi\rangle \otimes |\psi\rangle \otimes |\varphi\rangle \in \mathcal{H}'_{\mathcal{A}} . \tag{2}$$

The bit or phase basis state of $(X, Y, Z) \in \mathbb{F}_{q'}^{a_i \times m_i} \times \mathbb{F}_{q'}^{(m_i - a_i - a'_i) \times m_i} \times \mathbb{F}_{q'}^{a'_i \times m_i}$ is denoted as

$$\left| \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \right\rangle_b := \begin{bmatrix} |X\rangle_b \\ |Y\rangle_b \\ |Z\rangle_b \end{bmatrix}, \quad \left| \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \right\rangle_p := \begin{bmatrix} |X\rangle_p \\ |Y\rangle_p \\ |Z\rangle_p \end{bmatrix}. \tag{3}$$

We also introduce notations for the states in $\mathcal{H}'_{\mathcal{B}}$ and $\mathcal{H}'_{\mathcal{C}}$ in the same way as (2) and (3). In the following, we denote the $k \times l$ zero matrix as $\mathbf{0}_{k,l}$.

4.3 CSS Code in Our Code

In our code construction, we use the CSS code defined in this subsection which is similarly defined from [14, Subsection IV-B]. Define two classical codes $C_1, C_2 \subset \mathbb{F}_{q'}^{m_i \times (n' - 2m_i)}$ which satisfy $C_1 \supset C_2^{\perp}$ as

$$C_1 := \left\{ \begin{bmatrix} \mathbf{0}_{a_i, n'-2m_i} \\ X_2 \\ X_3 \end{bmatrix} \in \mathbb{F}_{q'}^{m_i \times (n'-2m_i)} \middle| X_2 \in \mathbb{F}_{q'}^{(m_i - a_i - a'_i) \times (n'-2m_i)}, X_3 \in \mathbb{F}_{q'}^{a'_i \times (n'-2m_i)} \right\},$$

$$C_2 := \left\{ \begin{bmatrix} X_1 \\ X_2 \\ \mathbf{0}_{a'_i, n'-2m_i} \end{bmatrix} \in \mathbb{F}_{q'}^{m_i \times (n'-2m_i)} \middle| X_1 \in \mathbb{F}_{q'}^{a_i \times (n'-2m_i)}, X_2 \in \mathbb{F}_{q'}^{(m_i - a_i - a'_i) \times (n'-2m_i)} \right\}.$$

For any $[M_1] \in C_1 / C_2^{\perp}$ where $M_1 \in \mathbb{F}_{q'}^{(m_i - a_i - a'_i) \times (n'-2m_i)}$, define the quantum state $|[M_1]\rangle_b \in \mathcal{H}_{\mathcal{C}}$ by

$$|[M_1]\rangle_b := \frac{1}{\sqrt{|C_2^{\perp}|}} \sum_{Y \in C_2^{\perp}} \left| \begin{bmatrix} \mathbf{0}_{a_i, n'-2m_i} \\ M_1 \\ \mathbf{0}_{a'_i, n'-2m_i} \end{bmatrix} + Y \right\rangle_b = \begin{bmatrix} |\mathbf{0}_{a_i, n'-2m_i}\rangle_b \\ |M_1\rangle_b \\ |\mathbf{0}_{a'_i, n'-2m_i}\rangle_p \end{bmatrix}.$$

With the above definitions, the code space is given as $\mathcal{H}'_{\text{code}} := \mathcal{H}'_{\mathcal{C}2} = \mathcal{H}'^{\otimes (m_i - a_i - a'_i) \times (n'-2m_i)}$ and a pure state $|\phi\rangle \in \mathcal{H}'_{\text{code}}$ is encoded as a superposition of the states $|[M_1]\rangle_b$ in this CSS code by

$$\begin{bmatrix} |\mathbf{0}_{a_i, n'-2m_i}\rangle_b \\ |\phi\rangle \\ |\mathbf{0}_{a'_i, n'-2m_i}\rangle_p \end{bmatrix} \in \mathcal{H}_{\mathcal{C}} .$$

5 Code Construction with Negligible Rate Shared Randomness

In this section, we construct our code that allows a sender S_i to transmit a state ρ_i on $\mathcal{H}'_{\text{code}} = \mathcal{H}'^{\otimes (m_i - a_i - a'_i) \times (n'-2m_i)}$ correctly to a receiver T_i by n-use of the network when the encoder and decoder share the negligible rate random variable $SR_i := (R_i, V_i)$.

The encoder and decoder are defined depending on the private randomness $U_{i,1}$ owned by encoder and the randomness SR_i shared between the encoder and decoder. These random variables are uniformly chosen from the values or matrices satisfying the following respective conditions: the variable $R_i := (R_{i,1}, R_{i,2}) \in \mathbb{F}_{q'}^{(m_i-a_i)\times m_i} \times \mathbb{F}_{q'}^{(m_i-a'_i)\times m_i}$ satisfies $\operatorname{rank} R_{i,1} = m_i - a_i$ and $\operatorname{rank} R_{i,2} = m_i - a'_i$, the random variable $V_i := (V_{i,1}, \ldots, V_{i,4m_i})$ consists of $4m_i$ values $V_{i,1}, \ldots, V_{i,4m_i} \in \mathbb{F}_{q'}^{4m_i}$ and the random variable $U_{i,1} \in \mathbb{F}_{q'}^{m_i\times m_i}$ satisfies $\operatorname{rank} U_{i,1} = m_i$.

Next, we construct the encoder $\mathcal{E}_i^{SR_i,U_{i,1}}$ and decoder $\mathcal{D}_i^{SR_i}$. Depending on SR_i and $U_{i,1}$, the encoder $\mathcal{E}_i^{SR_i,U_{i,1}}$ of the sender S_i is defined as an isometry channel from $\mathcal{H}'_{\text{code}}$ to $\mathcal{H}_{S_i} = \mathcal{H}'^{\otimes m_i\times n'}$. Depending on SR_i, the decoder $\mathcal{D}_i^{SR_i}$ of the receiver T_i is defined as a TP-CP map from $\mathcal{H}_{T_i} = \mathcal{H}'^{\otimes m_i\times n'}$ to $\mathcal{H}'_{\text{code}}$. Note that the randomness SR_i is shared between the encoder and the decoder. Because SR_i consists of $\alpha m_i(2m_i - a_i - a'_i + 4)$ elements of $\mathbb{F}_q$, the size of the shared randomness SR_i is sublinear with respect to n (i.e., negligible).

5.1 Encoder $\mathcal{E}_i^{SR_i,U_{i,1}}$ of the sender S_i

The encoder $\mathcal{E}_i^{SR_i,U_{i,1}}$ consists of three steps. In the following, we describe the encoding of the state $|\phi\rangle$ in $\mathcal{H}'_{\text{code}}$.

Step E1 The isometry map $U_{i,0}^{R_i}$ encodes the state $|\phi\rangle$ with the CSS code defined in Subsection 4.3 and the quantum systems $\mathcal{H}'_{\mathcal{A}}$ and $\mathcal{H}'_{\mathcal{B}}$ as

$$|\phi_1\rangle := U_{i,0}^{R_i}|\phi\rangle = \left|\begin{bmatrix}\mathbf{0}_{a_i,m_i}\\ R_{i,1}\end{bmatrix}\right\rangle_b \otimes \left|\begin{bmatrix}R_{i,2}\\ \mathbf{0}_{a'_i,m_i}\end{bmatrix}\right\rangle_p \otimes \begin{bmatrix}|\mathbf{0}_{a_i,m_i}\rangle_b\\ |\phi\rangle\\ |\mathbf{0}_{a'_i,m_i}\rangle_p\end{bmatrix} \in \mathcal{H}'_{\mathcal{A}} \otimes \mathcal{H}'_{\mathcal{B}} \otimes \mathcal{H}'_{\mathcal{C}} = \mathcal{H}_{S_i}\,.$$

Step E2 By quantum invertible linear operation $\mathcal{L}'(U_{i,1})$, the encoder maps $|\phi_1\rangle$ to $|\phi_2\rangle := \mathcal{L}'(U_{i,1})|\phi_1\rangle$.

Step E3 From random variable $V_i = (V_{i,1}, \ldots, V_{i,4m_i})$, define matrices $Q_{i,1;j,k} := (V_{i,k})^j$, $Q_{i,2;j,k} := (V_{i,m_i+k})^j$ for $1 \le j \le n' - 2m_i$, $1 \le k \le m_i$, and $Q_{i,3;j,k} := (V_{i,2m_i+k})^j$, $Q_{i,4;j,k} := (V_{i,3m_i+k})^j$ for $1 \le j, k \le m_i$. With these matrices, define the matrix $U_{i,2}^{V_i} \in \mathbb{F}_{q'}^{n'\times n'}$ as

$$U_{i,2}^{V_i} := \begin{bmatrix} I_{m_i} & \mathbf{0}_{m_i,m_i} & \mathbf{0}_{m_i,n'-2m_i}\\ Q_{i,3}^{\mathrm{T}}Q_{i,4} & I_{m_i} & \mathbf{0}_{m_i,n'-2m_i}\\ \mathbf{0}_{n'-2m_i,m_i} & \mathbf{0}_{n'-2m_i,m_i} & I_{n'-2m_i}\end{bmatrix} \cdot \begin{bmatrix} I_{m_i} & \mathbf{0}_{m_i,m_i} & \mathbf{0}_{m_i,n'-2m_i}\\ \mathbf{0}_{m_i,m_i} & I_{m_i} & Q_{i,2}^{\mathrm{T}}\\ \mathbf{0}_{n'-2m_i,m_i} & \mathbf{0}_{n'-2m_i,m_i} & I_{n'-2m_i}\end{bmatrix}$$
$$\cdot \begin{bmatrix} I_{m_i} & \mathbf{0}_{m_i,m_i} & \mathbf{0}_{m_i,n'-2m_i}\\ \mathbf{0}_{m_i,m_i} & I_{m_i} & \mathbf{0}_{m_i,n'-2m_i}\\ Q_{i,1} & \mathbf{0}_{n'-2m_i,m_i} & I_{n'-2m_i}\end{bmatrix},$$

where I_d is the identity matrix of size d. By quantum invertible linear operation $\mathcal{R}'(U_{i,2}^{V_i})$, the encoder maps $|\phi_2\rangle$ to $\mathcal{R}'(U_{i,2}^{V_i})|\phi_2\rangle$.

By above three steps, the encoder $\mathcal{E}_i^{SR_i,U_{i,1}}$ is described as an isometry map

$$\mathcal{E}_i^{SR_i,U_{i,1}} : |\phi\rangle \mapsto \mathcal{R}'(U_{i,2}^{V_i})\mathcal{L}'(U_{i,1})U_{i,0}^{R_i}|\phi\rangle \in \mathcal{H}_{S_i}\,.$$

5.2 Decoder $\mathcal{D}_i^{SR_i}$ of the receiver T_i

Decoder $\mathcal{D}_i^{SR_i}$ consists of two steps. In the following, we describe the decoding of the state $|\psi\rangle \in \mathcal{H}_{T_i}$.

Step D1 Since $(U_{i,2}^{V_i})^{-1}$ can be constructed from shared randomness V_i by

$$(U_{i,2}^{V_i})^{-1} = \begin{bmatrix} I_{m_i} & \mathbf{0}_{m_i,m_i} & \mathbf{0}_{m_i,n'-2m_i} \\ \mathbf{0}_{m_i,m_i} & I_{m_i} & \mathbf{0}_{m_i,n'-2m_i} \\ -Q_{i,1} & \mathbf{0}_{n'-2m_i,m_i} & I_{n'-2m_i} \end{bmatrix} \cdot \begin{bmatrix} I_{m_i} & \mathbf{0}_{m_i,m_i} & \mathbf{0}_{m_i,n'-2m_i} \\ \mathbf{0}_{m_i,m_i} & I_{m_i} & -Q_{i,2}^{\mathrm{T}} \\ \mathbf{0}_{n'-2m_i,m_i} & \mathbf{0}_{n'-2m_i,m_i} & I_{n'-2m_i} \end{bmatrix}$$
$$\cdot \begin{bmatrix} I_{m_i} & \mathbf{0}_{m_i,m_i} & \mathbf{0}_{m_i,n'-2m_i} \\ -Q_{i,3}^{\mathrm{T}}Q_{i,4} & I_{m_i} & \mathbf{0}_{m_i,n'-2m_i} \\ \mathbf{0}_{n'-2m_i,m_i} & \mathbf{0}_{n'-2m_i,m_i} & I_{n'-2m_i} \end{bmatrix},$$

the decoder applies the reverse operation $\mathcal{R}'(U_{i,2}^{V_i})^\dagger = \mathcal{R}'((U_{i,2}^{V_i})^{-1})$ of Step E3 as $|\psi_1\rangle := \mathcal{R}'(U_{i,2}^{V_i})^\dagger|\psi\rangle$.

Step D2 Perform the bit and phase basis measurements on $\mathcal{H}'_{\mathcal{A}}$ and $\mathcal{H}'_{\mathcal{B}}$, respectively, and let $O_{i,1}, O_{i,2} \in \mathbb{F}_{q'}^{m_i \times m_i}$ be the respective measurement outcomes. By Gaussian elimination, find invertible matrices $D_{i,1}^{R_{i,1},O_{i,1}}, D_{i,2}^{R_{i,2},O_{i,2}} \in \mathbb{F}_{q'}^{m_i \times m_i}$ satisfying

$$P_{\mathcal{W}_{i,1}} D_{i,1}^{R_{i,1},O_{i,1}} O_{i,1} = \begin{bmatrix} \mathbf{0}_{a_i,m_i} \\ \\ R_{i,1} \end{bmatrix}, \quad P_{\mathcal{W}_{i,2}} D_{i,2}^{R_{i,2},O_{i,2}} O_{i,2} = \begin{bmatrix} R_{i,2} \\ \\ \mathbf{0}_{a'_i,m_i} \end{bmatrix}. \tag{4}$$

where $P_{\mathcal{W}}$ is the projection from $\mathbb{F}_{q'}^{m_i}$ to the subspace $\mathcal{W}$, the subspace $\mathcal{W}_{i,1}$ consists of the vectors whose 1-st, …, a_i-th elements are zero and the subspace $\mathcal{W}_{i,2}$ consists of the vectors whose $(m_i - a'_i + 1)$-st, …, m_i-th elements are zero. The case of non-existence of $D_{i,1}^{R_{i,1},O_{i,1}}$ nor $D_{i,2}^{R_{i,2},O_{i,2}}$ means decoding failure, which implies that the decoder performs no more operations. Also, when $D_{i,1}^{R_{i,1},O_{i,1}}$ and $D_{i,2}^{R_{i,2},O_{i,2}}$ are not determined uniquely, the decoder chooses $D_{i,1}^{R_{i,1},O_{i,1}}$ and $D_{i,2}^{R_{i,2},O_{i,2}}$ deterministically depending on $O_{i,1}, R_{i,1}$ and $O_{i,2}, R_{i,2}$, respectively.

Based on $D_{i,1}^{R_{i,1},O_{i,1}}$ and $D_{i,2}^{R_{i,2},O_{i,2}}$ found by (4), the decoder applies $\mathcal{L}'(D_{i,1}^{R_{i,1},O_{i,1}})$ and $\mathcal{L}'(\widehat{D_{i,2}^{R_{i,2},O_{i,2}}})$ consecutively to $|\psi_1\rangle$, and the resultant state on $\mathcal{H}_{\mathrm{code}}$ is the output of Step D2. Then, Step D2 is written as the following TP-CP map $D_i^{R_i}$:

$$D_i^{R_i}(|\psi_1\rangle\langle\psi_1|) := \operatorname*{Tr}_{\mathcal{C}1,\mathcal{C}3} \sum_{O_{i,1},O_{i,2} \in \mathbb{F}_{q'}^{m_i \times m_i}} U_D^{R_i,O_{i,1},O_{i,2}} \sigma_{O_{i,1},O_{i,2}} (U_D^{R_i,O_{i,1},O_{i,2}})^\dagger,$$

where the matrices $U_D^{R_i,O_{i,1},O_{i,2}}$ and $\sigma_{O_{i,1},O_{i,2}}$ are defined as

$$U_D^{R_i,O_{i,1},O_{i,2}} := \mathcal{L}'(\widehat{D_{i,2}^{R_{i,2},O_{i,2}}})\mathcal{L}'(D_{i,1}^{R_{i,1},O_{i,1}}),$$
$$\sigma_{O_{i,1},O_{i,2}} := \operatorname*{Tr}_{\mathcal{A},\mathcal{B}} |\psi_1\rangle\langle\psi_1|(|O_{i,1}\rangle_{bb}\langle O_{i,1}| \otimes |O_{i,2}\rangle_{pp}\langle O_{i,2}| \otimes I_{\mathcal{C}}),$$

with the identity operator $I_{\mathcal{C}}$ on $\mathcal{H}_{\mathcal{C}}$.

By above two steps, the decoder $\mathcal{D}_i^{SR_i}$ is described as

$$\mathcal{D}_i^{SR_i}(|\psi\rangle\langle\psi|) := D_i^{R_i}\Big(\mathcal{R}'(U_{i,2}^{V_i})^\dagger|\psi\rangle\langle\psi|\mathcal{R}'(U_{i,2}^{V_i})\Big).$$

Since the size of the shared randomness SR_i is sublinear with respect to n, our code is implemented with negligible rate shared randomness.

6 Correctness of Our Code

In this section, we confirm that our code correctly transmits the state from the sender S_i to the receiver T_i. As is mentioned in Section 3, we show the condition $n(1-F_e^2(\rho_{mix},\kappa_i)) \to 0$ which implies the correctness of our code.

First, we describe the quantum code protocol κ_i from S_i to T_i, which is an integration of the encoding, transmission, and decoding. The encoding and decoding in κ_i is given by the probabilistic mixture of the code in Section 5 depending on the uniformly chosen random variables SR_i and $U_{i,1}$. Then, the code protocol κ_i is written as, for the state ρ_i on $\mathcal{H}'_{\mathrm{code}}$,

$$\kappa_i(\rho_i) := \sum_{SR_i,U_{i,1}} \frac{1}{N}\mathcal{D}_i^{SR_i}\Big(\operatorname*{Tr}_{T_1,\ldots,T_{i-1},T_{i+1},\ldots,T_r} \mathcal{L}(K)\Big(\mathcal{E}_i^{SR_i,U_{i,1}}(\rho_i)\otimes\rho_{i^c}\Big)\mathcal{L}(K)^\dagger\Big),$$

where ρ_{i^c} is the state in $\mathcal{H}_{S_1}\otimes\cdots\otimes\mathcal{H}_{S_{i-1}}\otimes\mathcal{H}_{S_{i+1}}\otimes\cdots\otimes\mathcal{H}_{S_r}$ of senders other than S_i, and $N := q'^{4m_i} + |\{U_{i,1}\in\mathbb{F}_{q'}^{m_i\times m_i} \,|\, \operatorname{rank} U_{i,1} = m_i\}| + |\{R_{i,1}\in\mathbb{F}_{q'}^{(m_i-a_i)\times m_i} \,|\, \operatorname{rank} R_{i,1} = m_i-a_i\}| + |\{R_{i,2}\in\mathbb{F}_{q'}^{(m_i-a'_i)\times m_i} \,|\, \operatorname{rank} R_{i,1} = m_i-a'_i\}|$.

As explained in [14, Section IV], $1-F_e^2(\rho_{mix},\kappa_i)$ is upper bounded by the sum of the bit error probability and the phase error probability. The bit error probability is the probability that a bit basis state $|X\rangle_b\in\mathcal{H}'_{\mathrm{code}}$ is sent but the bit basis measurement outcome on the decoder output is not X. In the similar way, the phase error probability is defined for the phase basis. We will show in Subsections 6.2 and 6.3 that the bit and phase error probabilities are upper bounded by $O\Big(\max\Big\{\frac{1}{q'},\frac{(n')^{m_i}}{(q')^{m_i-a_i}}\Big\}\Big)$ and $O\Big(\max\Big\{\frac{1}{q'},\frac{(n')^{m_i}}{(q')^{m_i-a'_i}}\Big\}\Big)$, respectively. Therefore, we have

$$n(1-F_e^2(\rho_{mix},\kappa_i)) \le nO\Big(\max\Big\{\frac{1}{q'},\frac{(n')^{m_i}}{(q')^{m_i-\max\{a_i,a'_i\}}}\Big\}\Big). \tag{5}$$

Since q' is taken in Section 4 to satisfy $\frac{n\cdot(n')^{m_i}}{(q')^{m_i-\max\{a_i,a'_i\}}}\to 0$, the RHS of (5) converges to 0 and therefore $n(1-F_e^2(\rho_{mix},\kappa_i))\to 0$. This completes the proof of Theorem 3.1.

6.1 Notation and Lemmas for Bit and Phase Error Probabilities

In this subsection, we prepare a notation and lemmas for proving the upper bounds of the bit and phase error probabilities. The upper bounds of these probabilities are shown separately in Subsections 6.2 and 6.3.

We introduce the notation $X := (X^{\mathcal{A}},X^{\mathcal{B}},X^{\mathcal{C}}) \in \mathbb{F}_{q'}^{k\times m_i}\times\mathbb{F}_{q'}^{k\times m_i}\times\mathbb{F}_{q'}^{k\times(n'-2m_i)}$ for $X\in\mathbb{F}_{q'}^{k\times n'}$ with arbitrary positive integer k. Also, we prepare the following lemmas.

▶ **Lemma 6.1.** *For integers $d_0\ge d_1+d_2$, let $\mathcal{W}_1\subset\mathbb{F}_{q'}^{d_0}$ be a d_1-dimensional subspace and $\mathcal{W}_2\subset\mathbb{F}_{q'}^{d_0}$ be a d_2-dimensional subspace. Assume the following three conditions.*
(Γ1) $\mathcal{W}_1\cap\mathcal{W}_2 = \{\mathbf{0}_{d_0,1}\}$.

(Γ2) *Let $\bar{m} \geq d_1 + d_2$. The vectors $x_1, \ldots, x_{\bar{m}} \in \mathcal{W}_1$ and $y_1, \ldots, y_{\bar{m}} \in \mathcal{W}_2$ satisfy*

$$\mathrm{span}((x_1, y_1), \ldots, (x_{\bar{m}}, y_{\bar{m}})) = \mathcal{W}_1 \oplus \mathcal{W}_2.$$

(Γ3) *Let $W_1' \subset \mathbb{F}_{q'}^{d_0}$ be a d_1-dimensional subspace and $r_1, \ldots, r_{\bar{m}} \in \mathcal{W}_1'$. There exists an invertible linear map $A : \mathcal{W}_1' \to \mathcal{W}_1$ which maps*

$$[x_1, \ldots, x_{\bar{m}}] = A[r_1, \ldots, r_{\bar{m}}].$$

Then, the following two statements hold.

(Δ1) *There exists invertible linear map $D : \mathbb{F}_{q'}^{d_0} \to \mathbb{F}_{q'}^{d_0}$ that*

$$P_{\mathcal{W}_1'} D[(x_1, y_1), \ldots, (x_{\bar{m}}, y_{\bar{m}})] = A^{-1}[x_1, \ldots, x_{\bar{m}}] = [r_1, \ldots, r_{\bar{m}}]. \tag{6}$$

(Δ2) *For the above linear map D, it holds for any $x \in \mathcal{W}_1$ and $y \in \mathcal{W}_2$ that*

$$P_{\mathcal{W}_1'} D(x, y) = A^{-1}x. \tag{7}$$

Proof. First, we show the item (Δ1). Let $\mathcal{W}_3$ be a subspace of $\mathbb{F}_{q'}^{d_0}$ that satisfies $\mathcal{W}_1 \oplus \mathcal{W}_2 \oplus \mathcal{W}_3 = \mathbb{F}_{q'}^{d_0}$. If D is defined as $D|_{\mathcal{W}_1} = A^{-1}$ and $D|_{\mathcal{W}_2 \oplus \mathcal{W}_3}(\mathcal{W}_2 \oplus \mathcal{W}_3) = \mathcal{W}_1'^{\perp}$, we obtain (6), i.e., (Δ1) from

$$P_{\mathcal{W}_1'} D((x_i, y_i)) = P_{\mathcal{W}_1'}(D|_{\mathcal{W}_1}(x_i) + D|_{\mathcal{W}_2 \oplus \mathcal{W}_3}(y_i)) = A^{-1}x_i = r_i.$$

Next, we show that the item (Δ2). Since arbitrary $(x, y) \in \mathcal{W}_1 \oplus \mathcal{W}_2$ is spanned by $(x_1, y_1), \ldots, (x_{\bar{m}}, y_{\bar{m}})$, Eq. (6) implies (7), which yields (Δ2). ◀

▶ **Lemma 6.2** ([14, Lemma 7.1]). *For integers $d_a \geq d_b + d_c$, fix a d_b-dimensional subspace $\mathcal{W} \subset \mathbb{F}_{q'}^{d_a}$, and randomly choose a d_c-dimensional subspace $\mathcal{R} \subset \mathbb{F}_{q'}^{d_a}$ with the uniform distribution. Then, we have*

$$\Pr[\mathcal{W} \cap \mathcal{R} = \{\mathbf{0}_{d_a,1}\}] = 1 - O({q'}^{d_b + d_c - d_a - 1}).$$

▶ **Lemma 6.3.** *For $d \geq d'$,*

$$\Pr\Big[\mathrm{rank}[t_1, \ldots, t_d] = d' \;\Big|\; t_1, \ldots, t_d \in \mathbb{F}_{q'}^{d'}\Big] \geq 1 - O\Big(\frac{1}{q'}\Big).$$

Proof. From $d \geq d'$, we have

$$\Pr\Big[\mathrm{rank}[t_1, \ldots, t_d] = d' \;\Big|\; t_1, \ldots, t_d \in \mathbb{F}_{q'}^{d'}\Big] \geq \Pr\Big[\mathrm{rank}[t_1, \ldots, t_{d'}] = d' \;\Big|\; t_1, \ldots, t_{d'} \in \mathbb{F}_{q'}^{d'}\Big]. \tag{8}$$

On the other hand, the RHS of (8) is equivalent to the probability to choose d' independent vectors in $\mathbb{F}_{q'}^{d'}$:

$$\Pr\Big[\mathrm{rank}[t_1, \ldots, t_{d'}] = d' \;\Big|\; t_1, \ldots, t_{d'} \in \mathbb{F}_{q'}^{d'}\Big] = \frac{{q'}^{d'}}{{q'}^{d'}} \cdot \frac{{q'}^{d'} - q'}{{q'}^{d'}} \cdots \frac{{q'}^{d'} - {q'}^{d'-1}}{{q'}^{d'}} = 1 - O\Big(\frac{1}{q'}\Big).$$

By combining the above inequality and equality, we have the lemma. ◀

▶ **Lemma 6.4** ([14, Lemmas 7.2 and 7.4]). *For the random matrix $U_{i,2}^{V_i}$ defined in Step E3, we have*

$$\max_{\mathbf{0}_{n',1} \neq x \in \mathbb{F}_{q'}^{n'}} \Pr[x^{\mathrm{T}}((U_{i,2}^{V_i})^{-1})^{\mathcal{A}} = \mathbf{0}_{1,m_i}] \leq \Big(\frac{n' - 2m_i}{q'}\Big)^{m_i},$$

$$\max_{\mathbf{0}_{n',1} \neq x \in \mathbb{F}_{q'}^{n'}} \Pr[x^{\mathrm{T}}((\widehat{U_{i,2}^{V_i}})^{-1})^{\mathcal{B}} = \mathbf{0}_{1,m_i}] \leq \Big(\frac{n' - 2m_i}{q'}\Big)^{m_i}.$$

6.2 Bit Error Probability

In this subsection, we show that when arbitrary bit basis state $|M\rangle_b \in \mathcal{H}'_{\text{code}}$ is the input state of the sender S_i, the original message M is correctly recovered with probability at least $1 - O\Big(\max\Big\{\frac{1}{q'}, \frac{(n')^{m_i}}{(q')^{m_i - a_i}}\Big\}\Big)$.

Step 1: We derive a necessary condition for correct decoding of the original message M in bit basis. When arbitrary bit basis state $|M\rangle_b \in \mathcal{H}'_{\text{code}}$ is the input state of the sender S_i, the encoded state is written as

$$\mathcal{E}_i^{SR_i,R_i}(|M\rangle_b) = \sum_{\bar{E}_1 \in \mathbb{F}_{q'}^{m_i\times m_i}, \bar{E}_2 \in \mathbb{F}_{q'}^{a'_i \times (n'-2m_i)}} \left| U_{i,1} \begin{bmatrix} \mathbf{0}_{a_i,m_i} & & \mathbf{0}_{a_i,n'-2m_i} \\ R_{i,1} & \bar{E}_1 & M \\ & & \bar{E}_2 \end{bmatrix} U_{i,2}^{V_i} \right\rangle_b ,$$

where we ignore the normalizing factors and phase factors.

Note that bit state measurement on network output system $\mathcal{H}_{T_i} = \mathcal{H}'^{\otimes m_i \times n'_i}$ commutes with the decoding operation $\mathcal{D}_i^{SR_i}$ on $\mathcal{H}_{T_i}$. Therefore, in the analysis of the bit error probability, we take the method to perform bit state measurement to $\mathcal{H}_{T_i}$ first, and then apply the decoding operation corresponding to $\mathcal{D}_i^{SR_i}$, instead of decoding first and performing bit state measurement.

By performing the bit basis measurement to the network output $\sigma_{T_i} = \kappa_i(|M\rangle_{bb}\langle M|)$, we have the following measurement outcome Y:

$$Y = K_{i,i}U_{i,1} \begin{bmatrix} \mathbf{0}_{a_i,m_i} & & \mathbf{0}_{a_i,n'-2m_i} \\ R_{i,1} & \bar{E}_1 & M \\ & & \bar{E}_2 \end{bmatrix} U_{i,2}^{V_i} + K_{i^c}Z,$$

where $\bar{E}_1 \in \mathbb{F}_{q'}^{m_i\times m_i}$, $\bar{E}_2 \in \mathbb{F}_{q'}^{a'_i \times (n'-2m_i)}$ and $Z \in \mathbb{F}_{q'}^{(m-m_i)\times n'}$. By Step D1, Y is decoded to

$$\bar{Y} = Y(U_{i,2}^{V_i})^{-1} = K_{i,i}U_{i,1} \begin{bmatrix} \mathbf{0}_{a_i,m_i} & & \mathbf{0}_{a_i,n'-2m_i} \\ R_{i,1} & \bar{E}_1 & M \\ & & \bar{E}_2 \end{bmatrix} + K_{i^c}Z(U_{i,2}^{V_i})^{-1}.$$

The measurement outcome $O_{i,1}$ in Step D2 is

$$O_{i,1} = \bar{Y}^{\mathcal{A}} = K_{i,i}U_{i,1} \begin{bmatrix} \mathbf{0}_{a_i,m_i} \\ R_{i,1} \end{bmatrix} + (K_{i^c}Z(U_{i,2}^{V_i})^{-1})^{\mathcal{A}}.$$

Since the decoder knows $O_{i,1}$ and $R_{i,1}$, the matrix $D_{i,1}^{R_{i,1},O_{i,1}}$ is found by Gaussian elimination to the left equation of (4) which is written as

$$P_{\mathcal{W}_{i,1}} D_{i,1}^{R_{i,1},O_{i,1}} O_{i,1} = P_{\mathcal{W}_{i,1}} D_{i,1}^{R_{i,1},O_{i,1}} \left(K_{i,i}U_{i,1} \begin{bmatrix} \mathbf{0}_{a_i,m_i} \\ R_{i,1} \end{bmatrix} + (K_{i^c}Z(U_{i,2}^{V_i})^{-1})^{\mathcal{A}} \right) = \begin{bmatrix} \mathbf{0}_{a_i,m_i} \\ R_{i,1} \end{bmatrix}. \quad (9)$$

Therefore, if the matrix $D_{i,1}^{R_{i,1},O_{i,1}}$ derived in (9) satisfies the following equation

$$P_{\mathcal{W}_{i,1}} D_{i,1}^{R_{i,1},O_{i,1}} \bar{Y}^{\mathcal{C}} = P_{\mathcal{W}_{i,1}} D_{i,1}^{R_{i,1},O_{i,1}} \left(K_{i,i}U_{i,1} \begin{bmatrix} \mathbf{0}_{a_i,n'-2m_i} \\ M \\ \bar{E}_2 \end{bmatrix} + (K_{i^c}Z(U_{i,2}^{V_i})^{-1})^{\mathcal{C}} \right) = \begin{bmatrix} \mathbf{0}_{a_i,n'-2m_i} \\ M \\ \bar{E}_2 \end{bmatrix}, \quad (10)$$

the original message M is correctly recovered.

Step 2: In the next step, we show that the conditions (Γ1), (Γ2) and (Γ3) of Lemma 6.1 in the following case imply Eq. (10);

$$\mathcal{W}_1 := \operatorname{col}\left(K_{i,i}U_{i,1} \begin{bmatrix} \mathbf{0}_{a_i,m_i} \\ R_{i,1} \end{bmatrix} \right), \quad \mathcal{W}_2 := \operatorname{col}\big(K_{i^c}Z(U_{i,2}^{V_i})^{-1}\big), \quad \mathcal{W}_1' := \mathcal{W}_{i,1}, \quad \bar{m} := m_i,$$

$$[x_1,\ldots,x_{\bar{m}}] := K_{i,i}U_{i,1}\begin{bmatrix} \mathbf{0}_{a_i,m_i} \\ R_{i,1} \end{bmatrix}, \quad [y_1,\ldots,y_{\bar{m}}] := (K_{i^c}Z(U_{i,2}^{V_i})^{-1})^{\mathcal{A}},$$

$$[r_1,\ldots,r_{\bar{m}}] := \begin{bmatrix} \mathbf{0}_{a_i,m_i} \\ R_{i,1} \end{bmatrix}, \quad A := (K_{i,i}U_{i,1})|_{\mathcal{W}_1'}, \quad (d_0,d_1,d_2) := (m_i, m_i - a_i, \operatorname{rank} K_{i^c}Z),$$

where $\operatorname{col}(T)$ of the matrix T is the column space of T and $\mathcal{W}_{i,1}$ is defined in Step D2 of Subsection 5.2.

Applying Lemma 6.1, we show that Eq. (10) holds if the conditions (Γ1), (Γ2) and (Γ3) are satisfied. Assume that (Γ1), (Γ2) and (Γ3) are satisfied. Then, the condition (Δ1) holds which implies the existence of $D_{i,1}^{R_{i,1},O_{i,1}}$ in (9). Moreover, (Δ2) implies that for any $r \in \mathcal{W}_1', y \in \mathcal{W}_2$ and $x = K_{i,i}U_{i,1}r \in \mathcal{W}_1$, it holds

$$P_{\mathcal{W}_1'} D_{i,1}^{R_{i,1},O_{i,1}}(x+y) = A^{-1}x = \big((K_{i,i}U_{i,1})|_{\mathcal{W}_1'}\big)^{-1}(K_{i,i}U_{i,1}r) = r,$$

and this yields (10).

Step 3: In the third step, we show that the relations (Γ1), (Γ2) and (Γ3) hold at least with probability $1 - O\Big(\max\Big\{\frac{1}{q'}, \frac{(n')^{m_i}}{(q')^{m_i-a_i}}\Big\}\Big)$, which proves the desired statement by combining the conclusion of Steps 1 and 2.

Step 3-1: In this substep, we show that the probability satisfying (Γ1), (Γ2) and (Γ3) is obtained by

$$\Pr[(\Gamma1)\cap(\Gamma2)\cap(\Gamma3)] = \Pr[(\Gamma1)]\cdot\Pr[(\Gamma2')]\cdot\Pr[(\Gamma2)|(\Gamma2')\cap(\Gamma1)], \tag{11}$$

where the condition (Γ2′) is given as
(Γ2′) $\operatorname{rank} K_{i^c}Z((U_{i,2}^{V_i})^{-1})^{\mathcal{A}} = \operatorname{rank} K_{i^c}Z$.
Eq. (11) is derived by the following reductions:

$$\begin{aligned} &\Pr[(\Gamma1)\cap(\Gamma2)\cap(\Gamma3)] \overset{(a)}{=} \Pr[(\Gamma1)\cap(\Gamma2)] \overset{(b)}{=} \Pr[(\Gamma1)]\cdot\Pr[(\Gamma2)|(\Gamma1)] \\ \overset{(c)}{=} &\Pr[(\Gamma1)]\cdot\Pr[(\Gamma2)\cap(\Gamma2')|(\Gamma1)] \overset{(d)}{=} \Pr[(\Gamma1)]\cdot\Pr[(\Gamma2')|(\Gamma1)]\cdot\Pr[(\Gamma2)|(\Gamma2')\cap(\Gamma1)] \\ \overset{(e)}{=} &\Pr[(\Gamma1)]\cdot\Pr[(\Gamma2')]\cdot\Pr[(\Gamma2)|(\Gamma2')\cap(\Gamma1)]. \end{aligned}$$

The equality (a) follows from the fact that (Γ3) is always satisfied for A defined in Step 2, and (b) and (d) are trivial. (c) is obtained because (Γ2′) is a necessary condition for (Γ2). Since $\operatorname{span}(y_1,\ldots,y_{\bar{m}}) = \mathcal{W}_2$ is a necessary condition for (Γ2) in Lemma 6.1, the condition (Γ2′) is also necessary for (Γ2) from

$$\begin{aligned} \operatorname{rank} K_{i^c}Z((U_{i,2}^{V_i})^{-1})^{\mathcal{A}} &= \operatorname{rank}(K_{i^c}Z(U_{i,2}^{V_i})^{-1})^{\mathcal{A}} = \operatorname{dimspan}(y_1,\ldots,y_{\bar{m}}) \\ &= \dim\mathcal{W}_2 = \operatorname{rank} K_{i^c}Z(U_{i,2}^{V_i})^{-1} = \operatorname{rank} K_{i^c}Z. \end{aligned}$$

The equality (e) follows from the fact that (Γ1) and (Γ2′) are independent, which will be shown by $\Pr[(\Gamma1)|(\Gamma2')] = \Pr[(\Gamma1)]$ in Step 3-2.

Step 3-2: In this step, we prove $\Pr[(\Gamma 1)] \geq 1 - O(1/q')$ and $\Pr[(\Gamma 1)|(\Gamma 2')] = \Pr[(\Gamma 1)]$. Fix $R_{i,1}$ and $U_{i,2}^{V_i}$. Then, $\mathcal{W}_1$ is randomly chosen d_1-dimensional subspace under uniform distribution and $\mathcal{W}_2$ is fixed d_2-dimensional subspace. Therefore, Lemma 6.2 can be applied with $(d_a, d_b, d_c, \mathcal{W}) := (d_0, d_2, d_1, \mathcal{W}_2)$ and $\Pr[(\Gamma 1)] = 1 - O({q'}^{d_2+d_1-d_0-1}) \geq 1 - O(1/q')$. On the other hand, since $\Pr[(\Gamma 1)]$ does not depend on $U_{i,2}^{V_i}$ but $\Pr[(\Gamma 2)]$ depends only on $U_{i,2}^{V_i}$, we have $\Pr[(\Gamma 1)|(\Gamma 2')] = \Pr[(\Gamma 1)]$.

Step 3-3: In this step, we show $\Pr[(\Gamma 2')] \geq 1 - \frac{{n'}^{m_i}}{{q'}^{m_i - a_i}}$. The condition $(\Gamma 2')$ holds if and only if $x^{\mathrm{T}} K_{i^c} Z ((U_{i,2}^{V_i})^{-1})^{\mathcal{A}} \neq \mathbf{0}_{1,m_i}$ for any vector $x \in \mathbb{F}_{q'}^{m_i}$ satisfying $x^{\mathrm{T}} K_{i^c} Z \neq \mathbf{0}_{1,n'}$ (considering K_{i^c}, Z and $((U_{i,2}^{V_i})^{-1})^{\mathcal{A}}$ as linear maps on row vector spaces, this is equivalent that $((U_{i,2}^{V_i})^{-1})^{\mathcal{A}}$ has trivial kernel $\{\mathbf{0}_{1,n'}\}$ for the image of $K_{i^c} Z$). Therefore, by applying Lemma 6.4 for all distinct $x^{\mathrm{T}} K_{i^c} Z$ which is not zero vector, we have

$$\Pr[(\Gamma 2')] \geq 1 - {q'}^{\operatorname{rank} K_{i^c} Z} \left(\frac{n' - 2m_i}{q'} \right)^{m_i} \geq 1 - {q'}^{a_i} \left(\frac{n' - 2m_i}{q'} \right)^{m_i} \geq 1 - \frac{{n'}^{m_i}}{{q'}^{m_i - a_i}}.$$

Step 3-4: Now we evaluate the probability $\Pr[(\Gamma 2)|(\Gamma 2') \cap (\Gamma 1)] \geq 1 - O(1/{q'}^{-1})$. Fix the random variable $U_{i,2}^{V_i}$ so that $(\Gamma 2')$ holds in the following. Define matrices $T_x = [x_{i(1)}, \ldots, x_{i(d_1+d_2)}]$, $T_y = [y_{i(1)}, \ldots, y_{i(d_1+d_2)}]$ and $T = T_x + T_y \in \mathbb{F}_{q'}^{d_0 \times (d_1+d_2)}$ where $i : \{1, \ldots, d_1 + d_2\} \to \{1, \ldots, \bar{m}\}$ is an injective index function such that $y_{i(1)}, \ldots, y_{i(d_2)}$ are linearly independent i.e., $\operatorname{rank} T_y = d_2$. Then, we have

$$\begin{aligned}
&\Pr\Big[(\Gamma 2)|(\Gamma 2') \cap (\Gamma 1)\Big] \geq \Pr[\operatorname{span}\big((x_{i(1)}, y_{i(1)}), \ldots, (x_{i(d_1+d_2)}, y_{i(d_1+d_2)})\big) = \mathcal{W}_1 \oplus \mathcal{W}_2 \,|\, (\Gamma 2') \cap (\Gamma 1)] \\
&\overset{(a)}{=} \Pr\Big[\operatorname{rank} T = d_1 + d_2 \mid (\Gamma 2') \cap (\Gamma 1)\Big] = \Pr\Big[\ker T = \{\mathbf{0}_{d_1+d_2,1}\} \mid (\Gamma 2') \cap (\Gamma 1)\Big] \\
&\overset{(b)}{=} \Pr\Big[\ker T_x \cap \ker T_y = \{\mathbf{0}_{d_1+d_2,1}\} \mid (\Gamma 2') \cap (\Gamma 1)\Big],
\end{aligned}$$

where (a) follows from $\operatorname{span}\big((x_{i(1)}, y_{i(1)}), \ldots, (x_{i(d_1+d_2)}, y_{i(d_1+d_2)})\big) \subset \mathcal{W}_1 \oplus \mathcal{W}_2$, and (b) follows from the condition $(\Gamma 1)$. Since $\operatorname{rank} T_x \leq d_1$ follows from its definition and the dimension of $\ker T_y$ is d_1, the condition $\operatorname{rank} T_x = d_1$ is a necessary condition for $\ker T_x \cap \ker T_y = \{\mathbf{0}_{d_1+d_2,1}\}$. Therefore, we have

$$\begin{aligned}
&\Pr[\ker T_x \cap \ker T_y = \{\mathbf{0}_{d_1+d_2,1}\} \mid (\Gamma 2') \cap (\Gamma 1)] \\
&= \Pr[\ker T_x \cap \ker T_y \mid \operatorname{rank} T_x = d_1 \cap (\Gamma 2') \cap (\Gamma 1)] \cdot \Pr[\operatorname{rank} T_x = d_1 \mid (\Gamma 2') \cap (\Gamma 1)]. \quad (12)
\end{aligned}$$

By applying Lemma 6.2 for $(d_a, d_b, d_c, \mathcal{W}) := (d_1 + d_2, d_1 = \dim \ker T_y, d_2 = \dim \ker T_x, \ker T_y)$, the first multiplicand of (12) equals to $1 - O(1/{q'}^{-1})$. From $\Pr[\operatorname{rank} T_x = d_1 \mid (\Gamma 2') \cap (\Gamma 1)] \geq \Pr\left[\operatorname{rank}[t_1, \ldots, t_{d_1+d_2}] = d_1 \mid t_1, \ldots, t_{d_1+d_2} \in \mathbb{F}_{q'}^{d_1}\right]$ and Lemma 6.3, the second multiplicand of (12) is greater than or equal to $1 - O(1/{q'}^{-1})$. Therefore, $\Pr[(\Gamma 2)|(\Gamma 2') \cap (\Gamma 1)] \geq 1 - O(1/{q'}^{-1})$.

In summary, we obtain

$$\begin{aligned}
&\Pr[(\Gamma 1) \cap (\Gamma 2) \cap (\Gamma 3)] = \Pr[(\Gamma 1)] \cdot \Pr[(\Gamma 2')] \cdot \Pr[(\Gamma 2)|(\Gamma 2') \cap (\Gamma 1)] \\
&\geq \left(1 - O\left(\frac{1}{q'}\right)\right) \cdot \left(1 - \frac{{n'}^{m_i}}{{q'}^{m_i - a_i}}\right) \cdot \left(1 - O\left(\frac{1}{q'}\right)\right) = 1 - O\left(\max\left\{\frac{1}{q'}, \frac{(n')^{m_i}}{(q')^{m_i - a_i}}\right\}\right).
\end{aligned}$$

6.3 Phase Error Probability

In this subsection, we show that the original message M' in the phase basis is correctly recovered with probability at least $1 - O\left(\max\left\{\frac{1}{q'}, \frac{(n')^{m_i}}{(q')^{m_i - a_i'}}\right\}\right)$.

Step 1: We derive a necessary condition for correct decoding of the original message M' in phase basis. For the analysis of the phase error probability, we apply the same discussion as the bit error probability. When a phase basis state $|M'\rangle_p \in \mathcal{H}'_{\text{code}}$ is the input state of sender S_i, the encoded state is written as

$$\mathcal{E}_i^{SR_i,R_i}(|M'\rangle_p) = \sum_{\bar{E}'_1 \in \mathbb{F}_{q'}^{m_i \times m_i}, \bar{E}'_2 \in \mathbb{F}_{q'}^{a_i \times (n'-2m_i)}} \left| \widehat{U_{i,1}} \begin{bmatrix} & & \bar{E}'_2 \\ \bar{E}'_1 & R_{i,2} & M' \\ & \mathbf{0}_{a'_i,m_i} & \mathbf{0}_{a'_i,n'-2m_i} \end{bmatrix} \widehat{U_{i,2}^{V_i}} \right\rangle_p ,$$

where we ignore normalizing factors and phase factors.

Since phase basis measurement and decoding operation $\mathcal{D}_i^{SR_i}$ commutes, we first apply phase basis measurement, and then decode the measurement outcome for the analysis of the phase error probability. Then, the phase basis measurement outcome Y' on the network output of T_i is written as

$$Y' = \widehat{K}_{i,i}\widehat{U_{i,1}} \begin{bmatrix} & & \bar{E}'_2 \\ \bar{E}'_1 & R_{i,2} & M' \\ & \mathbf{0}_{a'_i,m_i} & \mathbf{0}_{a'_i,n'-2m_i} \end{bmatrix} \widehat{U_{i,2}^{V_i}} + \widehat{K}_{i^c} Z,$$

where $\bar{E}'_1 \in \mathbb{F}_{q'}^{m_i \times m_i}$, $\bar{E}'_2 \in \mathbb{F}_{q'}^{a_i \times (n'-2m_i)}$ and $Z \in \mathbb{F}_{q'}^{(m-m_i)\times n'}$. By Step D1, Y' is decoded to

$$\bar{Y}' = Y'(\widehat{U_{i,2}^{V_i}})^{-1} = \widehat{K}_{i,i}\widehat{U_{i,1}} \begin{bmatrix} & & \bar{E}'_2 \\ \bar{E}'_1 & R_{i,2} & M' \\ & \mathbf{0}_{a'_i,m_i} & \mathbf{0}_{a'_i,n'-2m_i} \end{bmatrix} + \widehat{K}_{i^c} Z(\widehat{U_{i,2}^{V_i}})^{-1}.$$

By Step D2, the measurement outcome $O_{i,2}$ is given as $O_{i,2} = \bar{Y}'^{\mathcal{B}} = \widehat{K}_{i,i}\widehat{U_{i,1}} \begin{bmatrix} R_{i,2} \\ \mathbf{0}_{a'_i,m_i} \end{bmatrix} + (\widehat{K}_{i^c} Z(\widehat{U_{i,2}^{V_i}})^{-1})^{\mathcal{B}}$, and $D_{i,2}^{R_{i,2},O_{i,2}}$ is found by Gaussian elimination to the right equation of (4) which is written as

$$P_{\mathcal{W}_{i,2}} D_{i,2}^{R_{i,2},O_{i,2}} O_{i,2} = P_{\mathcal{W}_{i,2}} D_{i,2}^{R_{i,2},O_{i,2}} \left(\widehat{K}_{i,i}\widehat{U_{i,1}} \begin{bmatrix} R_{i,2} \\ \mathbf{0}_{a'_i,m_i} \end{bmatrix} + (\widehat{K}_{i^c} Z(\widehat{U_{i,2}^{V_i}})^{-1})^{\mathcal{B}} \right) = \begin{bmatrix} R_{i,2} \\ \mathbf{0}_{a'_i,m_i} \end{bmatrix}. \tag{13}$$

Thus, the correct estimate of M' is obtained when the following relation holds for $D_{i,2}^{R_{i,2},O_{i,2}}$ derived in (13):

$$P_{\mathcal{W}_{i,2}} D_{i,2}^{R_{i,2},O_{i,2}} \bar{Y}'^{\mathcal{C}} = P_{\mathcal{W}_{i,2}} D_{i,2}^{R_{i,2},O_{i,2}} \left(\widehat{K}_{i,i}\widehat{U_{i,1}} \begin{bmatrix} \bar{E}'_2 \\ M' \\ \mathbf{0}_{a'_i,n'-2m_i} \end{bmatrix} + (\widehat{K}_{i^c} Z(\widehat{U_{i,2}^{V_i}})^{-1})^{\mathcal{C}} \right) = \begin{bmatrix} \bar{E}'_2 \\ M' \\ \mathbf{0}_{a'_i,n'-2m_i} \end{bmatrix}. \tag{14}$$

Step 2: In the next step, we show that the equation (14) holds with probability at least $1 - O\left(\max\left\{ \frac{1}{q'}, \frac{(n')^{m_i}}{(q')^{m_i - a'_i}} \right\} \right)$, which shows the desired statement by combining Step 1.

In the same way as Subsection 6.2, the conditions (Γ1), (Γ2) and (Γ3) of Lemma 6.1 in the following case imply Eq. (14);

$$\mathcal{W}_1 := \mathrm{col}\left(\widehat{K}_{i,i}\widehat{U_{i,1}}\begin{bmatrix} R_{i,2} \\ \mathbf{0}_{a'_i,m_i}\end{bmatrix}\right), \quad \mathcal{W}_2 := \mathrm{col}\Big(\widehat{K}_{i^c}Z(\widehat{U^{V_i}_{i,2}})^{-1}\Big), \quad \mathcal{W}'_1 := \mathcal{W}_{i,2}, \quad \bar{m} := m_i,$$

$$[x_1,\dots,x_{\bar{m}}] := \widehat{K}_{i,i}\widehat{U_{i,1}}\begin{bmatrix} R_{i,2} \\ \mathbf{0}_{a'_i,m_i}\end{bmatrix}, \quad [y_1,\dots,y_{\bar{m}}] := (\widehat{K}_{i^c}Z(\widehat{U^{V_i}_{i,2}})^{-1})^{\mathcal{B}},$$

$$[r_1,\dots,r_{\bar{m}}] := \begin{bmatrix} R_{i,2} \\ \mathbf{0}_{a'_i,m_i}\end{bmatrix}, \quad A := (\widehat{K}_{i,i}\widehat{U_{i,1}})|_{\mathcal{W}'_1}, \quad (d_0,d_1,d_2) := (m_i, m_i - a'_i, \mathrm{rank}\,\widehat{K}_{i^c}Z),$$

where $\mathcal{W}_{i,2}$ is defined in Step D2 of Subsection 5.2. Also, in the same way, the conditions (Γ1), (Γ2) and (Γ3) holds with probability at least $1 - O\left(\max\left\{\frac{1}{q'}, \frac{(n')^{m_i}}{(q')^{m_i - a'_i}}\right\}\right)$.

7 Code Construction Without Free Classical Communication

We show that our code in Theorem 3.1 can be implemented without the assumption of negligible rate shared randomness. The paper [15] shows the following Proposition 7.1 by constructing a secret and correctable classical communication protocol for the classical unicast linear network. Due to the relation between the phase error and the information leakage in the bit basis [4, Lemma 5.9], we find that the dimension of leaked information is a'_i in the information transmission from the sender S_i to the receiver T_i. We apply Proposition 7.1 to the sender-receiver pair (S_i, T_i) with $c_1 := a_i$ and $c_2 := a'_i$. Therefore, the protocol of Proposition 7.1 can be implemented in our multiple-unicast network by preparing the input state of S_i in the bit basis. By attaching Proposition 7.1 to our code in the above method, we can implement our code satisfying Theorem 3.2.

▶ **Proposition 7.1** ([15, Theorem 1]). *Let q_1 be the size of the finite field which is the information unit of the network edges. We assume the inequality $c_1 + c_2 < c_0$ for the classical network where c_0 is the transmission rate from the sender S to the receiver T, c_1 is the rate of noise injection, and c_2 is the rate of information leakage. Define $q_2 := q_1^{c_0}$. Then, there exists a k-bit transmission protocol of block-length $n_1 := c_0(c_0 - c_2 + 1)k$ over $\mathbb{F}_{q_2}$ such that $P_{err} \leq kc_0/q_2$ and $I(M;E) = 0$, where P_{err} is the error probability and $I(M;E)$ is the mutual information between the message $M \in \mathbb{F}_2^k$ and the leaked information E.*

The proof of Theorem 3.2 takes a similar method to the proof of [14, Theorem 3.2].

Proof of Theorem 3.2. To construct the code satisfying the conditions of Theorem 3.2, we generate the shared randomness SR_i by Proposition 7.1 and then apply the code in Section 5. To apply Proposition 7.1 in our quantum network, we prepare the input state as a bit basis state. Given a block-length n, we take $q_1 = q^{\beta}$ such that $\beta = \lfloor \frac{2\log_2 \log_2 n}{m_i \log_2 q} \rfloor$ i.e., $q_2/(\log n)^2 = q_1^{m_i}/(\log n)^2 \to 1$, and $q' = q^{\alpha}$ such that $\alpha = \lfloor \frac{(m_i+2)\log_2 n}{\log_2 q} \rfloor$ i.e., $q'/n^{m_i+2} \to 1$.

First, by the protocol of Proposition 7.1 with $(c_0, c_1, c_2) := (m_i, a_i, a'_i)$, the sender S_i and the receiver T_i share the randomness SR_i. Since SR_i consists of $m_i(2m_i - a_i - a'_i + 4)$ elements of $\mathbb{F}_{q'}$, the number of bits to be shared is

$$\begin{aligned} k = \lceil m_i(2m_i - a_i - a'_i + 4)\log_2 q' \rceil &= \left\lceil m_i(2m_i - a_i - a'_i + 4)\left\lfloor \frac{(m_i+2)\log_2 n}{\log_2 q} \right\rfloor \log_2 q \right\rceil \\ &\leq \lceil m_i(m_i+2)(2m_i - a_i - a'_i + 4)\log_2 n \rceil. \end{aligned}$$

The error probability is $P_{err} \leq (m_i/q_1^{m_i}) \cdot \lceil m_i(m_i+2)(2m_i-a_i-a_i'+4)\log_2 n \rceil = O\left(\frac{\log_2 n}{(\log_2 n)^2}\right) \to 0$, and the block-length over $\mathbb{F}_q$ is

$$n_1 = m_i(m_i-a_i'+1)k\beta \leq m_i(m_i-a_i'+1) \cdot \lceil m_i(m_i+2)(2m_i-a_i-a_i'+4)\log_2 n \rceil \cdot \left\lfloor \frac{2\log_2 \log_2 n}{m_i \log_2 q} \right\rfloor,$$

which implies $n_1/n \to 0$. Therefore, the sharing protocol is implemented with negligible rate uses of the network.

Next, we apply the code in Section 5 with the extended field of size q' and $n_2 := n - n_1$ uses of the network. The relation $n_2/n = (n-n_1)/n \to 1$ holds and therefore the field size q' satisfies $n_2 \cdot (n_2')^{m_i}/(q')^{m_i - \max\{a_i, a_i\}} \to 0$ where $n_2' := n_2/\alpha$. Thus, this code implements the code in Theorem 3.2. ◀

8 Examples of Network

In this section, we give several network examples that our code can be applied.

First, as the most trivial case, if $\operatorname{rank} K_{i,i} = m_i$ and any distinct sender-receiver pairs do not interfere with each other, i.e, $K_{i,j}$ $(i \neq j)$ are zero matrices, the network operation K is a block matrix. This is the case where each pair independently communicates. In this case, our code is implemented with the rate m_i.

8.1 Simple Network in Fig. 1

In the network in Fig. 1, the network and node operations are described as

$$K = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad \widehat{K} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad A_1 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}.$$

When we consider the transmission from S_1 to T_1, the rates of bit and phase interferences are

$$\operatorname{rank} K_{1^c} = \operatorname{rank} \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} = 1, \quad \operatorname{rank} \widehat{K}_{1^c} = \operatorname{rank} \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = 0.$$

In this network, by constructing our code with $(m_1, a_1, a_1') := (2, 1, 0)$, our coding protocol transmits the state of rate $m_1 - a_1 - a_1' = 1$ asymptotically from S_1 to T_1.

8.2 Network with Bit Interference from One Sender

As a generalization of the network in Fig. 1, consider the case where the network consists of two sender-receiver pairs, and there is no bit interference from the sender S_1 to receiver T_2. The network operation of this network can be described by $\mathcal{L}(K)$ with

$$K = \begin{bmatrix} K_{1,1} & K_{1,2} \\ \mathbf{0}_{m_2,m_1} & K_{2,2} \end{bmatrix}, \quad \widehat{K} = \begin{bmatrix} (K_{1,1}^{\mathrm{T}})^{-1} & \mathbf{0}_{m_1,m_2} \\ -(K_{2,2}^{\mathrm{T}})^{-1} K_{1,2}^{\mathrm{T}} (K_{1,1}^{\mathrm{T}})^{-1} & (K_{2,2}^{\mathrm{T}})^{-1} \end{bmatrix}.$$

In this network, there is no phase interference from the sender S_2 to receiver T_1, and the other two rates $\operatorname{rank} K_{1,2}$ and $\operatorname{rank}(K_{2,2}^{\mathrm{T}})^{-1} K_{1,2}^{\mathrm{T}} (K_{1,1}^{\mathrm{T}})^{-1}$ coincide from $\operatorname{rank} K_{1,2} = \operatorname{rank} K_{1,2}^{\mathrm{T}} = \operatorname{rank}(K_{2,2}^{\mathrm{T}})^{-1} K_{1,2}^{\mathrm{T}} (K_{1,1}^{\mathrm{T}})^{-1}$. Therefore, by implementing our code with a_i, a_i' $(i = 1, 2)$ satisfying $\operatorname{rank} K_{1,2} \leq a_1, a_2' < m_i$ and $a_1' = a_2 := 0$, each sender-receiver pair can transmit the states.

Moreover, we generalize the above network for arbitrary r sender-receiver pairs where the interferences are generated only from one sender S_1. In this network, the network operation is given by the unitary operator $\mathcal{L}(K)$ with K defined as follows:

$$K = \begin{bmatrix} K_{1,1} & K_{1,2} & K_{1,3} & \cdots & K_{1,r} \\ \mathbf{0}_{m_2,m_1} & K_{2,2} & \mathbf{0}_{m_2,m_3} & \cdots & \mathbf{0}_{m_2,m_r} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{m_r,m_1} & \mathbf{0}_{m_r,m_2} & \mathbf{0}_{m_r,m_3} & \cdots & K_{r,r} \end{bmatrix},$$

$$\widehat{K} = \begin{bmatrix} (K_{1,1}^{\mathrm{T}})^{-1} & \mathbf{0}_{m_1,m_2} & \mathbf{0}_{m_1,m_3} & \cdots & \mathbf{0}_{m_1,m_r} \\ -(K_{2,2}^{\mathrm{T}})^{-1}K_{1,2}^{\mathrm{T}}(K_{1,1}^{\mathrm{T}})^{-1} & (K_{2,2}^{\mathrm{T}})^{-1} & \mathbf{0}_{m_2,m_3} & \cdots & \mathbf{0}_{m_1,m_r} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -(K_{r,r}^{\mathrm{T}})^{-1}K_{1,r}^{\mathrm{T}}(K_{1,1}^{\mathrm{T}})^{-1} & \mathbf{0}_{m_r,m_2} & \mathbf{0}_{m_r,m_3} & \cdots & (K_{r,r}^{\mathrm{T}})^{-1} \end{bmatrix},$$

where the ranks of $m_i \times m_i$ matrices $K_{i,i}$ are m_i, resepctively. In this network, if a_i, a_i' $(i = 1, \ldots, r)$ are set to $a_1 \geq \operatorname{rank}[K_{1,2} \;\; K_{1,3} \;\; \cdots \;\; K_{1,r}]$, $a_i' \geq \operatorname{rank} K_{1,i}$ $(i = 2, \ldots, r)$, and $a_1' = a_2 = a_3 = \cdots = a_r \geq 0$ and the condition $a_i + a_i' < m_i$ holds, the sender S_i can send to the receiver T_i the rate $m_i - a_i - a_i'$ state asymptotically by our code.

8.3 Network with Two Way Bit Interferences

In this subsection, we consider the code implementation over the network described as follows: The size q is 3, there exists two pairs (S_1, T_1) and (S_2, T_2) in the network, S_1, S_2, T_1, T_2 are connected to three edges, and the network operation is given by $\mathcal{L}(K)$ of

$$K = \begin{bmatrix} K_{1,1} & K_{1,2} \\ K_{2,1} & K_{2,2} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \quad \widehat{K} = \begin{bmatrix} 2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ -2 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

Then, differently from the previous examples, there are bit interferences both from S_1 to T_2 and from S_2 to T_1 because $K_{1,2}$ and $K_{2,1}$ are not zero matrix.

In the above network, we construct our code for S_1 to T_1 with $(m_1, a_1, a_1') := (3, 1, 1)$. Then, our code implements the rate $m_i - a_i - a_i' = 3 - 1 - 1 = 1$ quantum communication asymptotically from the relations

$$\operatorname{rank} K_{11} = \operatorname{rank} \widehat{K}_{11} = m_1 = 3, \quad \operatorname{rank} K_{1^c} = \operatorname{rank} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = 1, \quad \operatorname{rank} \widehat{K}_{1^c} = \operatorname{rank} \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = 1.$$

9 Conclusion

In this paper, we have proposed a quantum network code for the multiple-unicast network with quantum invertible linear operations. As the constraints of information rates, we assumed that the bit and phase transmission rates from S_i to T_i without interference are m_i $(m_i = \operatorname{rank} K_{i,i} = \operatorname{rank} \widehat{K}_{i,i})$, the upper bounds of the bit and phase interferences are a_i, a_i', respectively $(\operatorname{rank} K_{i^c} \leq a_i, \operatorname{rank} \widehat{K}_{i^c} \leq a_i')$, and $a_i + a_i' < m_i$ holds. Under these constraints, our code achieves the rate $m_i - a_i - a_i'$ quantum communication by asymptotic n-use of the network. The negligible rate shared randomness plays a crucial role in our code, and it is realized by attaching the protocol in [15].

Our code can be applied for the multiple-unicast network with the malicious adversary. When the eavesdropper attacks at most a_i'' edges connected with the sender S_i and the receiver T_i, if $a_i + a_i' + 2a_i'' < m_i$ holds, our code implements the rate $m_i - a_i - a_i' - 2a_i''$ quantum communications asymptotically. This fact can be shown by integrating the methods in this paper and [14].

References

1 Rudolf Ahlswede, Ning Cai, S-YR Li, and Raymond W Yeung. Network information flow. *IEEE Transactions on information theory*, 46(4):1204–1216, 2000.

2 Masahito Hayashi. Prior entanglement between senders enables perfect quantum network coding with modification. *physical review A*, 76(4):040301, 2007.

3 Masahito Hayashi. *Group Representation for Quantum Theory.* Springer, 2017.

4 Masahito Hayashi. *Group Theoretic Approach to Quantum Information.* Springer, 2017.

5 Masahito Hayashi, Kazuo Iwama, Harumichi Nishimura, Rudy Raymond, and Shigeru Yamashita. Quantum network coding. In *Annual Symposium on Theoretical Aspects of Computer Science*, pages 610–621. Springer, 2007.

6 Masahito Hayashi, Masaki Owari, Go Kato, and Ning Cai. Secrecy and robustness for active attack in secure network coding. In *Proceedings on 2017 IEEE International Symposium on Information Theory (ISIT)*, pages 1172–1176, 2017.

7 S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, M. Medard, and M. Effros. Resilient network coding in the presence of byzantine adversaries. *IEEE Transactions on Information Theory*, 54(6):2596–2603, June 2008.

8 Go Kato, Masaki Owari, and Masahito Hayashi. Single-shot secure quantum network coding for general multiple unicast network with free public communication. In *International Conference on Information Theoretic Security*, pages 166–187. Springer, 2017.

9 Hirotada Kobayashi, François Le Gall, Harumichi Nishimura, and Martin Rötteler. General scheme for perfect quantum network coding with free classical communication. In *International Colloquium on Automata, Languages, and Programming*, pages 622–633. Springer, 2009.

10 Hirotada Kobayashi, François Le Gall, Harumichi Nishimura, and Martin Rötteler. Perfect quantum network communication protocol based on classical network coding. In *Proceedings of 2010 IEEE International Symposium on Information Theory (ISIT)*, pages 2686–2690, 2010.

11 Hirotada Kobayashi, François Le Gall, Harumichi Nishimura, and Martin Rötteler. Constructing quantum network coding schemes from classical nonlinear protocols. In *Proceedings of 2011 IEEE International Symposium on Information Theory (ISIT)*, pages 109–113, 2011.

12 Debbie Leung, Jonathan Oppenheim, and Andreas Winter. Quantum network communication—the butterfly and beyond. *IEEE Transactions on Information Theory*, 56(7):3478–3490, 2010.

13 Benjamin Schumacher. Sending entanglement through noisy quantum channels. *Physical Review A*, 54(4):2614, 1996.

14 Seunghoan Song and Masahito Hayashi. Secure quantum network code without classical communication. *arXiv:1801.03306*, 2018.

15 Hongyi Yao, Danilo Silva, Sidharth Jaggi, and Michael Langberg. Network codes resilient to jamming and eavesdropping. *IEEE/ACM Transactions on Networking*, 22(6):1978–1987, 2014.