



This is a repository copy of *A theory of single-shot error correction for adversarial noise*.

White Rose Research Online URL for this paper:
<http://eprints.whiterose.ac.uk/142719/>

Version: Accepted Version

Article:

Campbell, E. (2019) A theory of single-shot error correction for adversarial noise. *Quantum Science and Technology*, 4 (2). 025006. ISSN 2058-9565

<https://doi.org/10.1088/2058-9565/aafc8f>

© 2019 IOP Publishing Ltd. This is an author produced version of a paper subsequently published in *Quantum Science and Technology*. Uploaded in accordance with the publisher's self-archiving policy, under the terms of the Creative Commons Attribution-NonCommercial Licence (<http://creativecommons.org/licenses/by-nc/3.0/>).

Reuse

This article is distributed under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs (CC BY-NC-ND) licence. This licence only allows you to download this work and share it with others as long as you credit the authors, but you can't change the article in any way or use it commercially. More information and the full terms of the licence here: <https://creativecommons.org/licenses/>

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



eprints@whiterose.ac.uk
<https://eprints.whiterose.ac.uk/>

A theory of single-shot error correction for adversarial noise

Earl T. Campbell

Department of Physics & Astronomy, University of Sheffield, Sheffield, S3 7RH, United Kingdom.

Single-shot error correction is a technique for correcting physical errors using only a single round of noisy check measurements, such that any residual noise affects a small number of qubits. We propose a general theory of single-shot error correction and establish a sufficient condition called good soundness of the code's measurement checks. Good code soundness in topological (or LDPC) codes is shown to entail a macroscopic energy barrier for the associated Hamiltonian. Consequently, 2D topological codes with local checks can not have good soundness. In tension with this, we also show that for any code a specific choice of measurement checks does exist that provides good soundness. In other words, every code can perform single-shot error correction but the required checks may be nonlocal and act on many qubits. If we desire codes with both good soundness and simple measurement checks (the LDPC property) then careful constructions are needed. Finally, we use a double application of the homological product to construct quantum LDPC codes with single-shot error correcting capabilities. Our double homological product codes exploit redundancy in measurements checks through a process we call metachecking.

In the simplest model of quantum error correction, noise affecting qubits is corrected under the assumption that measurements are performed perfectly. In reality, measurement results will be unreliable. The standard tactic for combating measurement noise is to repeat the measurements and build a timeline of measurement data. Error correction software can then attempt to infer the most likely explanation of the observed measurement results. The number of measurement rounds required will typically grow with the code size. Recently, single-shot error correction was proposed by Bombin as a radically different solution to measurement noise [1]. In single-shot error correction, no repeated measurements are needed. Benefits include faster error correction and an inherent resilience to temporally correlated noise [2]. However, few codes are known to support single-shot error correction. This idea was proposed in the setting of topological codes in three or four spatial dimensions, such as the three dimensional gauge colour code [3] and four dimensional toric code [4]. Very recently, it has been reported that quantum expander codes also allow for single-shot error correction [5]. Quantum data-syndrome codes are also closely related to single-shot codes [6, 7]. The development and implementation of decoding algorithms for single-shot error correction is also limited with only a few examples [8, 9]. So far progress has been focused on specific examples and one of our goals here is to lay down a common framework within which single-shot error correction can be understood and analysed.

In the idealised setting of perfect measurements, error correction will return the system back into the code-space, either with or without a logical error. A quantum code is parametrised by its distance d where perfect measurements can always detect noise on fewer than d qubits. Consequently, noise on any $(d - 1)/2$ qubits can be successfully corrected, even if the damaged qubits are chosen by an adversary who is attempting to corrupt the quantum information. Here we consider adversarial noise in the single-shot setting. We allow for physical qubit errors and measurement errors to appear in any pattern but af-

fecting a limited number of qubits and measurements. Given corrupt measurement data, error correction may not even return the system to the code-space, but will leave some residual qubit error. Single-shot error correction aims to control the size of this residual error. Central to achieving this is the notion of soundness [10, 11]. Loosely, a code has good soundness if small measurement syndromes can be produced by small qubit errors. Good soundness is closely related to local testability of codes [10, 11] and energy barriers in self-correcting quantum memories [12–14]. It is clear that the 4D toric codes have good soundness properties. We shall also show that good soundness entails the existence of a macroscopic energy barrier [12, 15] and consequently 2D topological codes cannot possess good soundness properties. However, we also show that given any quantum code we can adapt the check measurements to ensure good soundness, though in the process any topological and or low-density parity check (LDPC) properties will be lost. This leads to the surprising insight that any quantum error correction code can perform single-shot error correction, provided we are content with error correction measurements involving a large number of qubits. The interesting challenge is then to find codes that combine good soundness with LDPC properties.

The second part of this work provides techniques for constructing quantum codes with good single-shot correcting capabilities. Our approach is to use a double application of the homological, or hypergraph, product. The hypergraph product was first used by Tillich and Zémor [16] to show that any two classical codes can be combined to make a new quantum code. Unlike the standard CSS construction, no special relationship between the two codes is required by the hypergraph product. If the original classical codes are good LDPC codes (constant rate and linear distance) then the hypergraph product produces a quantum LDPC code with a good rate (the number of logical qubits k scaling as a constant fraction of the number of physical qubits n) and distance scaling as $\Theta(\sqrt{n})$; becoming the first quantum LDPC code to

achieve such parameters. Subsequently, Leverrier, Tillich and Zémor proposed quantum expander codes that result from the hypergraph product of two expander graph codes, which are a specific family of good LDPC codes. The expansion properties of these codes enabled them to devise an efficient decoder correcting adversarial errors affecting upto $O(\sqrt{n})$ qubits. Later it was shown that the decoder could correct $O(n)$ random errors with high probability [17] and support single-shot error correction [5]. Furthermore, maximum likelihood decoding has been investigated for quantum expander codes providing both analytical lower bounds [18] and numerical estimates [19].

Our approach here has overlap in the formal techniques but is more widely applicable since it does not depend on the strong assumption that the initial classical code is an expander graph code. The hypergraph product is closely related to the homological product used in the study of algebraic topology. Bravyi and Hastings [20] used the homological product to construct codes with a linear distance and good rate; though they were not strictly low-density parity check codes. Audoux and Couvreur studied repeated application of the homological product [21].

We will use two applications of the homological product to design single-shot codes from any classical code. Two applications of the homological product generates a structure that in homology theory would be described as length 4. Sometimes this length-4 algebraic structure can be embedded within a geometrically local 4-dimensional manifold and the resulting quantum code would be a 4-dimensional topological code. Given a family of LDPC classical codes, our construction gives a family of LDPC quantum codes with good soundness, successfully combining these two desirable properties. However, our approach is inherently algebraic, providing many codes with no natural spatial topology, unless the original classical codes are topological. From the perspective of practical implementations, a topological code of modest dimension may seem preferable. However, topological codes are constrained by trade-off bounds on the achievable code parameters [22, 23] and so non-topological codes can be much more efficient.

We begin by reviewing the key concepts (Sec. I) before giving a more technical statement of the main results (Sec. II). We prove sufficient conditions for single-shot error correction in Sec. III. We discuss the relationship between soundness and energy barriers in Sec. IV. We show how measurement checks can be redefined for any code to provide good soundness in Sec. V. We give a general overview of how homology theory can be used to describe quantum codes in Sec. VI. This establishes the technical groundwork for Sec. VII where we give code constructions that meet our criteria using a double application of the homological product. We conclude with a discussion of the remaining open problems and the limitations of considering adversarial noise rather than stochastic noise.

I. KEY CONCEPTS

The preliminary material covered in this section draws from the work of Bombin [1, 2] and was influenced by Brueckmann’s thesis [24], though our presentation is less topological and has some new ideas.

A. Stabiliser codes

An n qubit error correcting code storing k logical qubits can be represented by a projector Π onto the codespace. Stabiliser codes are an important class where Π can be described in terms of the code stabiliser \mathcal{S} . That is, \mathcal{S} is an abelian subgroup of the Pauli group such that for all $S \in \mathcal{S}$ we have $S\Pi = \Pi S = \Pi$. To perform error correction we measure some set of checks $\mathcal{M} \subset \mathcal{S}$ that generate \mathcal{S} under multiplication. We require that \mathcal{M} suffices to generate the whole stabiliser of the codespace but we allow for the possibility of \mathcal{M} being overcomplete. We define the weight $\text{wt}(\cdot)$ of a Pauli operator P as the number of qubits on which P acts nontrivially (the identity is the only trivial Pauli). Given a family of check sets \mathcal{M}_n with index n , which we will call a *check family*, we find there is a corresponding code family Π_n . For a given code family, there may be many different choices of check family, so many statements are more precisely defined with respect to check families. For instance, we have a notion of low-density parity check (LDPC) and we say a check family is LDPC if there exists a constant C such that for every n

1. For all $S \in \mathcal{M}_n$ we have $\text{wt}(S) \leq C$;
2. For every physical qubit in the code, there are no more than C checks in \mathcal{M}_n that act non-trivially on that qubit.

It is crucial that the constant C is the same for every member of the family. One practical consequence is that for codes with an LDPC check family, the complexity of measuring checks does not increase with the code size. Crudely, one can say a code family is LDPC if there exists at least one corresponding LDPC check family. Note that topological code families are always LDPC.

Also important is the code distance d_Q . We use the subscript Q to distinguish this from the single-shot distance (denoted d_{ss}) that we define later. The distance d_Q is simply the minimum $\text{wt}(P)$ over all P such that $P\Pi = \Pi P$ but $P \notin \mathcal{S}$. It is useful to also define the min-weight wt_{\min} of a Pauli operator, which is

$$\text{wt}_{\min}(P) := \{\text{wt}(PS) : S \in \mathcal{S}\}. \quad (1)$$

To summarise, an $[[n, k, d_Q]]$ code has parameters n (number of physical qubits), k (number of logical qubits) and d_Q (qubit code distance).

The measurement syndrome is the result of measuring $\mathcal{M} = (M_1, M_2, \dots, M_m)$. Given a physical Pauli error E

we can denote $\sigma(E)$ as the syndrome due to E assuming perfect measurements. We use the convention that $\sigma(E)$ is a binary column vector with elements

$$[\sigma(E)]_i = \begin{cases} 1 & \text{if } EM_i = -M_iE \\ 0 & \text{if } EM_i = M_iE \end{cases} \quad (2)$$

We will be interested in the weight of the syndrome and always use $|\dots|$ to denote the Hamming weight of binary vectors. The Hamming weight is the number of nonzero elements.

B. Single-shot error correction

A decoder is an algorithm that takes a measurement syndrome $s \in \mathbb{Z}_2^m$ and outputs a recovery Pauli operator E_{rec} . We model measurement errors as introducing an additional syndrome vector u so that we physically observe syndrome $s = \sigma(E) + u$ where E is the physical error. Good decoder design would ensure that given s the recovery is such that residual error $E_{\text{rec}}E$ has low min-weight. We propose the following definition

Definition 1 (Single-shot error correction) *Let p and q be integers and $f : \mathbb{Z} \rightarrow \mathbb{R}$ be some function with $f(0) = 0$. We say a check set is (p, q, f) single-shot if there exists a decoder such that for all u and E such that*

1. $|u| < p$; and
2. $f(2|u|) + \text{wt}(E) < q$

the decoder takes syndrome $s = \sigma(E) + u$ and outputs recovery operation E_{rec} such that $\text{wt}_{\min}(E_{\text{rec}} \cdot E) \leq f(2|u|)$.

This captures all instances of single-shot error correction known to the author. We are interested in good cases where p and q are large and f is in some sense small. A very bad case is when $p = 1$ so that no measurement errors ($|u| < 1$) can be tolerated. A more rigorous notion of good single-shot properties requires us to consider not just a single instance but an infinite check-family.

Definition 2 (Good single-shot families) *Consider an infinite check family \mathcal{M}_n of n -qubit codes. We say the family is a good single-shot family if each \mathcal{M}_n is (p, q, f) single-shot where*

1. p and q grow with n such that $p, q \geq an^b$ for some positive constants a, b . That is, $p, q \in \Omega(n^b)$ with $b > 0$;
2. and $f(x)$ is some polynomial that is monotonically increasing with x and independent of n .

We need p and q to grow so that we can tolerate more errors as the code size grows. We want f to be independent of n so that the residual errors remain contained.

Single-shot error correction is defined for a single round but it is informative to see what the consequences are

for N rounds of error correction. We use a label $\tau \in \{1, \dots, N\}$ for the round number. On round τ , we denote u_τ for the measurement errors and E_τ for the new physical errors. We must combine E_τ with the residual error from the previous round $R_{\tau-1}$ to obtain the total error $E_\tau R_{\tau-1}$. For the τ^{th} round to satisfy the conditions in Def. 1 we need that $|u_\tau| < p$ and

$$f(2|u_\tau|) + \text{wt}(E_\tau R_{\tau-1}) < q. \quad (3)$$

Assuming similar conditions were satisfied on the previous round, we may upper bound $\text{wt}(R_{\tau-1})$ using Def. 1 and have

$$f(2|u_\tau|) + f(2|u_{\tau-1}|) + \text{wt}(E_\tau) < q. \quad (4)$$

Therefore, provided the measurement errors and new physical errors are small for every round, the residual error will be kept under control over many rounds and not grow in size.

The above definition of single-shot error correction is difficult to analyse since it contains the clause ‘‘if there exists a decoder’’ and there are many possible decoders. Therefore, we also consider a complementary concept called soundness which will be shown to entail single-shot error correction. Roughly, this extra property is that for low weight syndromes there exists a low weight physical error producing the syndrome. More formally,

Definition 3 (Soundness) *Let t be an integer and $f : \mathbb{Z} \rightarrow \mathbb{R}$ be some function called the soundness function with $f(0) = 0$. Given some set of Pauli checks \mathcal{M} , we say it is (t, f) -sound if for all Pauli errors E with $|\sigma(E)| = x < t$, it follows that there exists an E^* with $\sigma(E^*) = \sigma(E)$ such that $\text{wt}(E^*) \leq f(x)$.*

The phrase soundness comes from the literature on locally testable codes [10, 11]. In particular, the above definition is similar to Def 14 of Ref. [10] though this earlier work did not allow for the $|\sigma(E)| < t$ clause.

Again, good soundness would mean ‘‘small’’ f . More rigorously, we define the following notion of goodness

Definition 4 (Good soundness) *Consider an infinite check family \mathcal{M}_n . We say the family has good soundness if each \mathcal{M}_n is (t, f) -sound where:*

1. t grows with n such that $t \geq an^b$ for some positive constants a, b . That is, $t \in \Omega(n^b)$ with $b > 0$;
2. and $f(x)$ is some polynomial that is monotonically increasing with x and independent of n .

The intuition behind f being a polynomial is that we are formalising an algebraic version of an area or volume law that is encountered in topological codes. For instance, in the classical 2D Ising model we know that the area within a boundary follows a quadratic scaling (you may wish to look ahead to Fig. 2b3). Ultimately, f will govern the size of residual errors after performing single-shot error correction, so we do not want it to grow with the number

of qubits. In contrast, t captures the scale at which this boundary law breaks down and so it must grow with the code size to enable single-shot error correction of larger errors as the code grows.

It is clear that not all check families have good soundness. For 2D toric codes with the standard choice of checks, an error violating only 2 checks can be of arbitrarily large size.

C. Energy barriers

Energy barriers play an important role in the design of passive quantum memories [13, 14]. While passive quantum memories are a distinct topic from active single-shot error correction, the two topics are intertwined. Earlier work [10] has commented on the relationship between soundness and energy barriers, though they used a more restrictive notion of soundness. For a stabiliser code with checks \mathcal{M} we define a Hamiltonian

$$H = - \sum_{S \in \mathcal{M}} S. \quad (5)$$

We are interested in walks of quantum states $W = \{\psi_0, \psi_1, \psi_2, \dots, \psi_L\}$ that fulfil

1. groundstates: ψ_0 and ψ_L are groundstates of H ;
2. orthogonality: ψ_0 and ψ_L are orthogonal;
3. local errors: for every $j \in [1, L]$ there exists a single-qubit Pauli P_j such that $|\psi_j\rangle = P_j|\psi_{j-1}\rangle$.

For every such walk we associate an energy penalty

$$ep(W) = \max_{\psi_j \in W} \langle \psi_j | H | \psi_j \rangle - E_{gs}, \quad (6)$$

where E_{gs} is the ground state energy. The energy barrier of check set \mathcal{M} and associated Hamiltonian is then the minimum $ep(W)$ over all walks W satisfying the above conditions. Less formally, the energy barrier is the minimum energy required to go from one ground state to another.

Every quantum code will have some size energy barrier. We are really interested in the scaling with code size. Given an infinite check family \mathcal{M}_n of n -qubit codes, if the energy barrier scales as $\Omega(n^c)$ for some positive constant c , then we say the family has a *macroscopic* energy barrier.

D. Measurement redundancy and single-shot distance

We have allowed for some redundancy so that checks \mathcal{M} may be overcomplete. This is pivotal for us to capture the single-shot properties of the 4D toric codes since they are only known to exhibit good soundness when an overcomplete set of checks are used. We quantify the amount

of redundancy in a measurement scheme as the ratio between the number of measurements performed and the minimum number required to generate the stabiliser of the code and use v to denote this ratio. Good soundness can always be achieved by allowing v to grow with n by simply repeating the same measurements. Rather, the most interesting cases are check families where v is no more than a small constant factor. There may also be interesting intermediate cases where v grows but slowly (e.g. sublinearly), though a constant factor is more desirable and is what we prove later in our constructions. Since topological codes can use redundancy to achieve good soundness, it is reasonable to ask whether redundancy is necessary for good soundness? We will see later that redundancy is not always essential for good soundness (see Thm. 3 and Sec. V). However, it seems that redundancy does play an important role when one attempts to marry good soundness with LDPC properties.

Check redundancy provides consistency conditions that one can inspect for evidence of measurement errors. These are checks on checks and we call them metachecks. They do not represent a physical measurement but classical postprocessing on the measurement outcomes. It is essentially a classical error correcting code that can be represented by a parity check matrix H . Given a binary string s representing the outcome of syndrome measurements, we say Hs is the metacheck syndrome, where Hs is evaluated modulo 2. If there are no measurement errors then $s = \sigma(E)$ where E is the physical error. Recall that we model measurement errors as introducing an additional error u so that $s = \sigma(E) + u$. Since the metachecks are intended to look for measurement errors, we require that $H\sigma(E) = 0$ for all E . It follows that the metasynndrome $Hs = H(\sigma(E) + u) = Hu$ depends only on the measurement error u . There will always exist a maximal set of metachecks H_{\max} such that $H_{\max}s = 0$ if and only if there exists an error E such that $s = \sigma(E)$. However, we are flexible and allow for H to contain fewer checks than H_{\max} , so that not all check redundancies are metachecked. While it might seem odd to not use the maximum information present, this occurs naturally in some local decoders for topological codes where local metachecks are used but non-local metachecks are ignored by the decoder (see for instance the discussion on “energy-barrier limited decoding” in Ref. [9]). Given a non-maximal set of meta-checks, there are syndromes s that pass all metachecks ($Hs = 0$) and yet there is no error E satisfying $s = \sigma(E)$. This motivates the following definition.

Definition 5 (Single-shot distance) *For a code with checks \mathcal{M} and metacheck matrix H we define the single-shot distance as*

$$d_{ss} = \min\{|s| : Hs = 0, s \notin \text{im}(\sigma)\}. \quad (7)$$

We use the convention that $d_{ss} = \infty$ if for all s there exists some E such that $s = \sigma(E)$.

Here, $\text{im}(\sigma)$ is the image of map σ , which is the set of s such that $s = \sigma(E)$ for some E . A equivalent definition

is that d_{ss} is the minimal weight s such that $Hs = 0$ but $H_{\max}s \neq 0$. The single-shot distance relates to how many measurement errors can be tolerated before a failure occurs that we call a metacheck failure. In a metacheck failure, the syndrome has no explanation in terms of qubit errors.

We remark that for any \mathcal{M} we can always choose $H = H_{\max}$ and then d_{ss} is infinite. However, sometimes a finite single-shot distance may be preferred to ensure that the metacheck decoding process can be implemented using a local decoder [9]. For a code with metachecks we extend the notation $[[n, k, d_Q]]$ to $[[n, k, d_Q, d_{ss}]]$.

II. SUMMARY OF RESULTS

Here we prove the following:

Theorem 1 (Single-shot success) *Consider a quantum error correcting code with parameters $[[n, k, d_Q, d_{ss}]]$ and check set that is (t, f) -sound. It is also (p, q, f) single-shot where*

$$p = \frac{1}{2} \min[d_{ss}, t] \quad (8)$$

$$q = d_Q/2. \quad (9)$$

For the above bounds to be useful, the code must have a soundness function f that is fairly gentle (e.g. some polynomial). The proof is mostly linear algebra and is given in Sec. III.

Our second result is an observation on the connection between soundness and energy barriers.

Theorem 2 *Any LDPC check family with good soundness and code distance d_Q growing as $\Omega(n^c)$ for some constant $0 < c$ will also have a macroscopic energy barrier.*

This is proved in Sec. IV. We remark that Aharonov and Eldar made a similar observation [10] though using a much stronger notion of soundness. Since Bravyi and Terhal proved that no 2D topological code can have a macroscopic energy barrier [25], it follows immediately that

Corollary 1 *Any 2D topological check family with code distance d_Q growing as $\Omega(n^c)$ for some constant $0 < c$ will not have good soundness.*

We thank Michael Beverland for pointing out that this corollary follows directly from Thm. 2 and the Bravyi and Terhal result.

Next, we show that

Theorem 3 *For any n -qubit quantum error correcting code we can find a set of checks generating the code stabiliser (without any redundancy) such that these checks are $(\infty, f(x) = x)$ -sound.*

The proof is elementary and given in Sec. V. While this is a simple result, it carries important implications for our understanding of soundness. It shows that any code family can be bestowed with good soundness by appropriate choice of checks, but in the process the LDPC property may be lost. Therefore, the interesting question is for which code families we can find checks that are simultaneously LDPC and of good soundness.

Our last main result is a recipe for quantum codes with the required properties. We show that

Theorem 4 (Construction of single-shot codes)

Given a classical error correcting code with parameters $[n, k, d]$ we can construct a quantum error correcting code with parameters $[[n_Q, k^4, d_Q \geq d, d_{ss} = \infty]]$ where

$$n_Q = n^4 + 4n^2(n - k)^2 + (n - k)^4. \quad (10)$$

Furthermore, the resulting checks are (d, f) -sound and also $(\frac{d}{2}, \frac{d}{2}, f)$ single-shot, with $f(x) = x^3/4$ or better. The check redundancy is bounded $v < 2$. Given a classical LDPC check family, this construction gives a quantum LDPC check family. Given a classical check family where $d \in \Omega(n^a)$ we have a good single shot family.

We remark that the distance bound $d_Q \geq d$ and soundness properties are loosely bounded. Indeed, very recently Zeng and Pryadko [31] considered the same code family and showed that $d = d^2$.

Before giving the proof of Thm. 4, we establish how the mathematics of homology theory and chain complexes can be used to define quantum codes with metachecks. As such, we provide a pedagogical interlude in Sec. VI that introduces this correspondence. The proof is then given in Sec. VII and uses the homological product on chain complexes. Where possible we have converted abstract homological proofs into linear algebra. The constructions of Thm. 4 will emerge as a simple, special case of the techniques explored in Sec. VII, and we will see that codes with finite single-shot distance are also possible. An important metric is the encoding rate, the number of logical qubits per physical qubit k_Q/n_Q . The expressions for the inverse rate are neater to write

$$\begin{aligned} \frac{n_Q}{k_Q} &= \frac{n^4 + 4n^2(n - k)^2 + (n - k)^4}{k^4} \quad (11) \\ &= 6 \left(\frac{n}{k}\right)^4 - 12 \left(\frac{n}{k}\right)^3 + 10 \left(\frac{n}{k}\right)^2 - 4 \left(\frac{n}{k}\right) + 1. \end{aligned}$$

From this, it is clear that for any family of classical codes with constant rate $n/k \leq A$, will yield a family of quantum codes with constant rate $n_Q/k_Q \leq A_Q \sim O(A^4)$.

III. CONDITIONS FOR SUCCESSFUL SINGLE-SHOT ERROR CORRECTION

This section proves that soundness leads to single shot error correction as stated in Thm. 1. Our analysis will use a minimum weight decoder defined as follows:

Definition 6 (MW single-shot error decoding)

Given measurement outcomes $s = \sigma(E) + u$, a minimum weight decoder performs the following 2 steps

1. Syndrome decode: find s_{rec} with minimal $|s_{rec}|$ such that $s + s_{rec}$ passes all metachecks (so $H(s + s_{rec}) = 0$);
2. Qubit decode: find E_{rec} with minimal $\text{wt}(E_{rec})$ such that $\sigma(E_{rec}) = s + s_{rec}$;

We call $R = E \cdot E_{rec}$ the residual error.

This is the most common notion of weight minimisation and for instance was suggested by Bombin [1]. Other decoders may correct more errors or may be more efficient to implement. However, the minimum weight decoder is especially useful in the following analysis.

Note that it is not possible to always find solutions to the above problem. For instance, one may find a minimising s_{rec} but then there is no E_{rec} satisfying the second condition. We call such an event a metacheck failure, but we do have the following guarantee

Lemma 1 (Meta-check success) *We can find a solution to MW single-shot decoding provided that $|u| < d_{ss}/2$.*

The proof is essentially the same as standard proofs for correcting adversarial qubit errors. Metacheck failures correspond to cases where there exists a minimal weight s_{rec} where $H(s + s_{rec}) = 0$ but there is no physical Pauli error E such that $\sigma(E) = s + s_{rec}$. Note that whenever we use “+” between two binary vectors it should be read as addition modulo 2. First, we note that $H(s + s_{rec}) = H(\sigma(E) + u + s_{rec})$ and using $H\sigma(E) = 0$ we get that s_{rec} must satisfy $H(u + s_{rec}) = 0$. Since, $s_{rec} = u$ would satisfy this requirement and s_{rec} is minimum weight, we infer that $|s_{rec}| \leq |u|$. Using the triangle inequality we get $|s_{rec} + u| \leq 2|u| < d_{ss}$. By the definition of single-shot distance, it follows that there exists a physical error E' such that $\sigma(E') = s_{rec} + u$. Using the syndrome relation $\sigma(E \cdot E') = \sigma(E) + \sigma(E')$ we obtain

$$\sigma(E \cdot E') = s + u + s_{rec} + u = s + s_{rec}. \quad (12)$$

Therefore, there is always a physical error (e.g. $E_{rec} = E \cdot E'$) consistent with the repaired syndrome $s + s_{rec}$ and the lemma is proved.

The above proof shows that the code can tolerate up to $d_{ss}/2 - 1$ adversarial measurement errors and provide a solution to single-shot decoding. However, the story is not finished since even if a metacheck failure does not occur, a conventional logical failure might yet occur. Therefore, next we address the question of how we can ensure the residual error $R = E_{rec} \cdot E$ has bounded size. From $\sigma(E_{rec}) = s + s_{rec}$ we deduce $\sigma(R) = u + s_{rec}$ and so

$$|\sigma(R)| \leq 2|u| < d_{ss} \quad (13)$$

This prompts the question, given a small syndrome (consistent with metachecks) does there even exist a small

weight physical error generating this syndrome! Indeed, this is not always the case; unless the code has nice soundness properties. Using our notion of soundness we can prove the following

Lemma 2 (An upper bound on residual error)

Consider a quantum error correcting code with parameters $[[n, k, d_Q, d_{ss}]]$ that is (t, f) -sound. Given measurement error u and physical error E . If

1. $|u| < d_{ss}/2$: the measurement error is small enough to ensure no metacheck failures;
2. $|u| < t/2$: the measurement error is small enough to use soundness properties;
3. $f(2|u|) + \text{wt}(E) < d_Q/2$: the combined errors are sufficiently small;

It follows that a solution to MW single-shot decoding will yield a residual error $R = E \cdot E_{rec}$ with $\text{wt}_{\min}(R) \leq f(2|u|)$.

We know from above (Eq. 13) that the residual error R satisfies $|\sigma(R)| \leq 2|u| < d_{ss}$. By using the definition of (t, f) -soundness, we know that provided $2|u| < t$ there exists an R^* such that $\sigma(R) = \sigma(R^*)$ and $\text{wt}(R^*) \leq f(2|u|)$. It remains to show that $S = RR^*$ is a stabiliser of the code as this would entail that $\text{wt}_{\min}(R) \leq \text{wt}(R^*) \leq f(2|u|)$. Clearly, $\sigma(RR^*) = \sigma(S) = 0$ so S is either a stabiliser or a nontrivial logical operator. It can only be a nontrivial logical operator if $d_Q \leq \text{wt}(RR^*)$. The rest of the proof shows that we instead have $\text{wt}(RR^*) < d_Q$ and so S is a stabiliser. We start with

$$R \cdot R^* = E \cdot E_{rec} \cdot R^*, \quad (14)$$

and

$$\text{wt}(R \cdot R^*) = \text{wt}(E \cdot E_{rec} \cdot R^*). \quad (15)$$

Using the triangle inequality

$$\text{wt}(R \cdot R^*) \leq \text{wt}(E_{rec}) + \text{wt}(E \cdot R^*). \quad (16)$$

Since, E_{rec} is a minimum weight solution, we can assume that $\text{wt}(E_{rec}) \leq \text{wt}(E \cdot R^*)$, and hence

$$\text{wt}(R \cdot R^*) \leq 2\text{wt}(E \cdot R^*) \leq 2\text{wt}(E) + 2\text{wt}(R^*). \quad (17)$$

Using again that $\text{wt}(R^*) \leq f(2|u|)$ we obtain

$$\text{wt}(R \cdot R^*) \leq 2(f(2|u|) + \text{wt}(E)). \quad (18)$$

We are interested in when the LHS is upper bounded by d_Q , which follows from the RHS being upper bounded by d_Q , which is precisely the third condition of the lemma. Therefore, $\text{wt}(R \cdot R^*) < d_Q$ and consequently $R = S \cdot R^*$. This proves the lemma, and Thm. 1 follows by simply rephrasing the lemma into the language of Def. 1.

IV. SOUNDNESS AND ENERGY BARRIERS

Here we discuss the relationship between the concept of code soundness and energy barriers in physical systems, resulting in a proof of Thm. 2. The reader ought to ensure familiarity with the introductory material in subsections IB and IC. Aharonov and Eldar remarked in Ref. [10] that codes with good soundness lead to large energy barriers, though they were interested in a strictly stronger definition of soundness.

A key lemma is the following

Lemma 3 *Consider a $[[n, k, d_Q]]$ quantum code with checks \mathcal{M} that is (t, f) -sound and where all qubits are involved in no more than C checks. It follows that the energy barrier is at least $f^{-1}(w)$ where $w = \min[(t-1)/C, (d_Q-1)/2]$ and f^{-1} is the inverse of the soundness function.*

For any walk of states $\{\psi_0, \psi_1, \psi_2, \dots, \psi_L\}$ we have a sequence of Pauli operators $\{\mathbb{1}, E_1, E_2, \dots, E_L\}$, so that $|\psi_j\rangle = E_j|\psi_0\rangle$ and $E_j E_{j-1}^\dagger = E_j E_{j-1} = P_j$ is a one qubit Pauli error (the local error condition). For every E_j in the sequence we consider the reduced weight

$$\text{wt}_{\text{red}}(E) := \min_V \{\text{wt}(EV) : V \in \mathcal{P}, \sigma(V) = 0\}, \quad (19)$$

where the minimisation is over all Pauli V with trivial syndrome. Note that reduced weight is slightly different from min-weight since the minimisation is over a bigger group than the code stabiliser. Herein we use V_j to denote Pauli operators that achieve the above minimisation. Since $\sigma(V_j) = 0$ every V_j is either a stabiliser or a nontrivial logical operator. By the groundstates and orthogonality property, it follows that $V_0 = \mathbb{1}$ and $V_L = E_L$. So the sequence starts with a stabiliser and ends with a nontrivial logical operator. Therefore, there must exist a j^* such that V_{j^*} is a stabiliser and V_{j^*+1} is a nontrivial logical operator. Therefore, $V_{j^*} V_{j^*+1}$ must also be a nontrivial logical operator and so

$$d_Q \leq \text{wt}(V_{j^*} V_{j^*+1}). \quad (20)$$

Furthermore, we have

$$\begin{aligned} \text{wt}(V_{j^*} V_{j^*+1}) &= \text{wt}(V_{j^*} V_{j^*+1} E_{j^*}^\dagger E_{j^*+1}^\dagger E_{j^*+1}^\dagger) \\ &= \text{wt}(V_{j^*} E_{j^*} V_{j^*+1} E_{j^*+1} E_{j^*+1}), \end{aligned} \quad (21)$$

and using the triangle inequality twice we have

$$\begin{aligned} \text{wt}(V_{j^*} V_{j^*+1}) &\leq \text{wt}(V_{j^*} E_{j^*}) + \text{wt}(V_{j^*+1} E_{j^*+1}) \\ &\quad + \text{wt}(E_{j^*} E_{j^*+1}) \\ &= \text{wt}_{\text{red}}(E_{j^*}) + \text{wt}_{\text{red}}(E_{j^*+1}) + 1. \end{aligned} \quad (22)$$

We have used $\text{wt}_{\text{red}}(E_j) = \text{wt}(V_j E_j)$ on the first two terms and the local errors condition on the last term. Combining this with Eq. (20), leads to

$$d_Q \leq 2\max[\text{wt}_{\text{red}}(E_{j^*}), \text{wt}_{\text{red}}(E_{j^*+1})] + 1, \quad (23)$$

and so

$$\frac{d_Q - 1}{2} \leq \max[\text{wt}_{\text{red}}(E_{j^*}), \text{wt}_{\text{red}}(E_{j^*+1})]. \quad (24)$$

Consider the sequence of reduced weights $\{\text{wt}_{\text{red}}(E_0), \text{wt}_{\text{red}}(E_1), \dots, \text{wt}_{\text{red}}(E_n)\}$. The sequence starts and ends with zero and at some point must reach $(d_Q - 1)/2$ or higher. Furthermore, the local error condition entails that $|\text{wt}_{\text{red}}(E_{j+1}) - \text{wt}_{\text{red}}(E_j)|$ is either 0 or 1 and so the sequence of reduced weights must include every integer from 0 to $(d_Q - 1)/2$. Therefore, we can set w equal to $\min[t/C, (d_Q - 1)/2]$ and there must exist an E_j with $\text{wt}_{\text{red}}(E_j) = w$. Next, we consider the syndrome $\sigma(E_j)$ and note that $\sigma(E_j) = \sigma(E_j V_j)$ where $\text{wt}(E_j V_j) = \text{wt}_{\text{red}}(E_j)$. The LDPC condition of the code ensures that for any E we have $|\sigma(E)| \leq C \text{wt}(E)$. Therefore, for the E_j with $\text{wt}_{\text{red}}(E_j) = w$ we have $|\sigma(E_j)| \leq Cw$. Since $w \leq (t-1)/C$ we have $|\sigma(E_j)| \leq t-1 < t$ and the soundness property can be deployed to conclude that $f^{-1}(w) \leq |\sigma(E_j)|$. Since this holds for every possible walk, $f^{-1}(w)$ gives a lower on the energy barrier and we have proved Lem. 3.

From Lem. 3 we can quickly obtain a proof of Thm. 2. We consider an infinite family of $[[n, k, d_Q]]$ codes with an LDPC check family \mathcal{M} with good soundness. That is, the codes are (t_n, f) -sound such that: the soundness function $f \in O(x^a)$ is independent of n ; and t_n grows as $\Omega(n^b)$ for some constants a and b . We further assume that the code distance d_Q grows as $\Omega(n^c)$ for some constant c . Since $d_Q \in \Omega(n^c)$ and $t \in \Omega(n^b)$, we can choose $w = \min[t/C, (d_Q - 1)/2] \in \Omega(n^{\min\{c, b\}})$ in Lem. 3. It follows that the energy barrier scales as $\Omega(n^{\min\{c, b\}/a})$ since $f \in O(x^a)$ and so $f^{-1} \in \Omega(x^{1/a})$. Therefore this check family has a macroscopic energy barrier. Notice that soundness is not the only ingredient in the proof, the LDPC condition is also crucial. It is unclear whether a similar result can be shown without the LDPC condition.

We remark that the converse statement would be that any LDPC check family with a macroscopic energy barrier has good soundness. We have neither proof nor counterexample and so the status of this converse statement remains open.

Bravyi and Terhal proved that no 2D topological stabiliser codes have a macroscopic energy barrier [25]. Therefore, such codes cannot have good soundness as we stated in corollary 1. This is nearly a statement that single-shot error correction is impossible in 2D topological stabiliser codes and we believe this to be the case. Though one must be cautious as we have shown good soundness to be a sufficient condition for single-shot error correction but not a necessary one. Clearly, if a code does not have good soundness then minimum weight decoding (in the sense of Def. 6) can lead to large weight residual error. However, if one deviates from the minimum weight decoding strategy then the picture becomes less clear. For instance, one strategy might be that when the minimum weight solution is high weight, we do not at-

tempt to return the system to the codespace but instead apply a partial recovery. For instance, if we observe two far apart checks with “-1” outcomes in the 2D toric code, then we could apply a partial recovery that reduces the distance between these checks. Indeed, there are cellular automata decoders for the 2D toric code that behave just like this [9, 26–28]. These fail to qualify as single-shot decoders in the usual sense as they rely on the syndrome history (partially stored in a cellular automata). But they highlight that single-shot error correction might be possible using an imaginative decoder approach based on partial recoveries.

V. GOOD SOUNDNESS FOR ALL CODES

It is common to conflate a quantum error correction code with a set of checks \mathcal{M} that generate the stabiliser. But there are many choices of checks for any given code. Crucially, the soundness properties depend on the set of checks. Here we prove Thm. 3, which roughly states that for any code we can find a check set with good soundness properties. The proof follows from the following lemma.

Lemma 4 *Given an $[[n, k, d_Q]]$ quantum error correction code with stabiliser \mathcal{S} there exists a minimal set of generators $\mathcal{M} = \{M_1, M_2, \dots, M_{n-k}\}$ and associated Pauli errors $\mathcal{E} = \{E_1, E_2, \dots, E_{n-k}\}$ such that: (1) $[M_i, E_j] \neq 0$ if and only if $i = j$; and (2) every E_j acts non-trivially on only a single qubit and so $\text{wt}(E_j) = 1$.*

We first consider the consequence of this lemma. Given such a set of checks, it follows that if s is a syndrome unit vector (so $|s| = 1$) with a 1 entry in the j^{th} location, then $s = \sigma(E_j)$ (recall Eq. (2)). More generally, s can be written as a sum of $|s|$ unit vectors and therefore $s = \sigma(E)$ where

$$E = \prod_{j:s_j=1} E_j. \quad (25)$$

Since $\text{wt}(E_j) = 1$ we have $\text{wt}(E) \leq |s|$ (with more work one can prove equality). Therefore, the checks are (t, f) -sound with $t = \infty$ and $f(x) = x$ since: the argument holds for any weight syndrome, and so the value of t is unbounded; and the weight of the physical error is no more than the weight of the syndrome, so we have $f(x) = x$.

The proof of Lem. 4 is essentially a step in the proof Lem. 2 of Ref. [29]. In Ref. [29], it is shown that upto to qubit labelling and local Clifford unitaries, the generators M_j can be brought into a diagonalised form inspired by the graph state formalism. In this form, M_j acts on the j^{th} qubit with Pauli X . On all others qubits with labels 1 through to $n - k$, the operator M_j acts as either Pauli Z or the identity. Therefore, Pauli Z acting on qubit j anticommutes with generator M_j and commutes with all other generators. Accounting for local Cliffords and original qubit labelling, the required E_j may act on a

different qubit and may be different from Pauli Z , but it will be a single qubit Pauli. This completes the proof.

The soundness properties proven above are extremely strong. This leads to the counter-intuitive result that single-shot error correction is possible for any code and without any check redundancy. The price to pay is that one must use a certain set of checks such as the diagonalised form above. As such, if the checks are initially low weight (e.g. part of an LDPC check family) then this property may be lost as the diagonalisation process leads to high weight checks. Indeed, we can prove the following strong limitation on diagonalisation methods.

Claim 1 *Consider a family of codes with checks in the diagonalised form used in the proof of Lem. 4. Assume also the diagonalised check family is LDPC, such that in every code no qubit is acted on by more than C checks. It follows that the distance is bounded $d_Q \leq C + 1$ for all codes in the family.*

We prove this by constructing an explicit error F that is not in the code stabiliser but $\sigma(F) = 0$ and $\text{wt}(F) \leq C + 1$. First, we let P be some single qubit Pauli ($\text{wt}(P) = 1$) acting on a qubit with label exceeding $n - k$. By the LDPC property $|\sigma(P)| \leq C$. Furthermore, following previous arguments there exists an E acting on the first $n - k$ qubits such that $\sigma(E) = \sigma(P)$ and $\text{wt}(E) \leq |\sigma(P)|$. Combined $\text{wt}(E) \leq |\sigma(P)|$ and $|\sigma(P)| \leq C$ entail $\text{wt}(E) \leq C$. Setting $F = EP$, we have that

$$\sigma(F) = \sigma(E) + \sigma(P) = 2\sigma(E) = 0 \quad (26)$$

and

$$\text{wt}(F) \leq \text{wt}(E) + \text{wt}(P) \leq C + 1. \quad (27)$$

Lastly, we need to show that F is not an element of the stabiliser. First we note that $F \neq \mathbb{1}$ since E and P act on disjoint sets of qubits. Next, let us assume to the contrary that F is a non-trivial element of the stabiliser. Then there is some non-empty set $J \subseteq \{1, \dots, n - k\}$ such that

$$F = \prod_{j \in J} M_j. \quad (28)$$

Following the argument in the proof of Lem. 4, let us assume that each M_j acts with Pauli X on the j^{th} qubit. But all $M_{k \neq j}$ act on the j^{th} qubit with either Pauli Z or the identity. Therefore, for every $j \in J$ we have that F acts on the j^{th} qubit with either X or Y . Since J is non-empty there is at least one qubit with index between 1 and $n - k$ such that F acts as either X or Y . However, $F = EP$ where E acts on the first $n - k$ qubits with either Z or $\mathbb{1}$. Since P acts on one of the last k qubits, we see that F can not be a stabiliser and must instead be a non-trivial logical operator.

The LDPC property is highly desirable and so too is growing code distance. Therefore, we need an alternative route to good soundness.

VI. TANNER GRAPHS, CHAIN COMPLEXES AND HOMOLOGY THEORY

From here on we specialise to codes with checks \mathcal{M} that can be partitioned into checks in the Z and X Pauli basis. For such codes, we describe quantum codes in a graphical language that extends on the classical use of Tanner graphs. We will explain the correspondence between the graphical representation and a linear algebra description in terms of concepts from algebraic topology.

Several example graphs are given in Fig. 1. In every case, the graph breaks up into $D + 1$ -partitions and we will refer to D as the length of the graph. Each partition comes with a set of vertices C_j . We use a binary matrix δ_j to describe the adjacency between vertices in C_j and C_{j+1} . Specifically, matrix δ_j has a “1” in entry (a, b) if and only if the b^{th} vertex in C_j is connected to the a^{th} vertex in C_{j+1} . Therefore, δ_0 is the well-known parity check matrix of a classical code. Furthermore, δ_0 is the parity check matrix for bit-flip (X) errors in a quantum code. Using superscript T for transpose, the matrix δ_{-1}^T is the parity check matrix for phase-flip (Z) errors in a quantum code.

We conflate thinking of C_j as a set of vertices and also as a binary vector space $\mathbb{Z}_2^{n_j}$ where n_j denotes the number of vertices in C_j . A unit vector \hat{u} has only a single entry with value 1 and identifies single vertex in C_j . Therefore, given a pair of unit vectors $\hat{u} \in C_j$ and $\hat{v} \in C_{j+1}$, we have $\hat{v}^T \delta_j \hat{u} = 1$ if and only if the corresponding vertices are connected. Therefore, given a unit vector $\hat{u} \in C_1$ identifying a measurement (or check) for bit-flip errors, the vector $\delta_0^T \hat{u}$ identifies the (qu)bits involved in that check. We use the notation

$$X[u] := \otimes_j X_j^{u_j}, \quad (29)$$

$$Z[v] := \otimes_j Z_j^{v_j}, \quad (30)$$

where u and v are binary vectors. The graph should be read as not just defining a code but also the measurement scheme. So for every unit vector \hat{u} in C_1 , the graphical formalism stipulates that we measure the operator $Z[\delta_0^T \hat{u}]$. So in our earlier notation $Z[\delta_0^T \hat{u}]$ would be a member of \mathcal{M} and is a stabiliser of the code. Since the stabiliser is a group, we have that $Z[\delta_0^T u]$ is a stabiliser for any vector $u \in C_1$. Similarly, $X[\delta_{-1} v]$ is a stabiliser of the code for every $v \in C_{-1}$. Operators $X[\delta_{-1} v]$ and $Z[\delta_0^T u]$ will commute if and only if $(\delta_0^T u)^T \delta_{-1} v = u^T \delta_0 \delta_{-1} v = 0$ where all such equations should be read using addition modulo 2. Since we need all such operators to commute, we require that $\delta_0 \delta_{-1} = 0$. Conversely, if $X[e]$ with $e \in C_0$ is an error, the vector $\delta_0 e$ is the Z -measurement syndrome assuming ideal measurements.

In homology theory, this whole structure is called a chain complex and the operators δ_j are called boundary maps provided the relation $\delta_{j+1} \delta_j = 0$ holds for all j . Therefore, given a homological chain complex the commutation relations are automatically satisfied since $\delta_0 \delta_{-1} = 0$. Remarkably, requiring $\delta_{j+1} \delta_j = 0$ not only

gives us the required commutation relations but also ensures that the metachecks are suitably defined. We will show this formally. Consider a physical error $X[e]$. It will generate Z -syndrome $\delta_0 e$ assuming no measurement errors. Since there are no measurement errors, the meta-syndrome $x = \delta_1 \delta_0 e$ ought to be the all zero vector, which is ensured if $\delta_1 \delta_0 = 0$.

Let us connect this back to the notation used in the first part of this paper. The check set is

$$\mathcal{M} = (Z[\delta_0^T \hat{u}_1], \dots, Z[\delta_0^T \hat{u}_{n_1}], X[\delta_{-1} \hat{v}_1], \dots, X[\delta_{-1} \hat{v}_{n_{-1}}]) \quad (31)$$

where \hat{u}_j and \hat{v}_j are unit vectors with the unit in the j^{th} location. Any Pauli error can be expressed as $E = X[e]Z[f]$ for some vectors e and f . The syndrome of this Pauli is then the combination of the Z and X syndromes, so that

$$\sigma(X[e]Z[f]) = \begin{pmatrix} \delta_0 e \\ \delta_{-1}^T f \end{pmatrix}. \quad (32)$$

Furthermore, the whole metasynndrome matrix has block matrix form

$$H = \begin{pmatrix} \delta_1 & 0 \\ 0 & \delta_{-2}^T \end{pmatrix}. \quad (33)$$

From this we see that the condition required earlier (that $H\sigma(E) = 0$ for all Pauli E) follows from the fundamental property of chain complexes, specifically $\delta_1 \delta_0 = 0$ and $\delta_{-2}^T \delta_{-1}^T = 0$.

Next, we study some parameters of chain complexes. We use n_j to denote the number of vertices in C_j , and equivalently the dimension of the associated vector space $\mathbb{Z}_2^{n_j}$. The matrix δ_j will have n_j columns and n_{j+1} rows. An important parameter is the j^{th} Betti number, which we denote k_j . For our purposes, it suffices to define

$$k_j := \text{nullity}(\delta_j) - \text{rank}(\delta_{j-1}). \quad (34)$$

Here, nullity is the dimension of the kernel, denoted $\ker(\delta_j)$, which is the space of vectors u such that $\delta_j u = 0$. The rank is the number of linearly independent rows in a matrix. Alternatively, the rank is equal to the dimension of the image, denoted $\text{im}(\delta_{j-1})$, which is the space of vectors v such that there exists a u satisfying $v = \delta_{j-1} u$. Those familiar with homology theory may prefer to think of k_j as the dimension of the j^{th} homology group $\mathcal{H}_j = \ker(\delta_j) / \text{im}(\delta_{j-1})$. This counts the number of different homology classes at a particular level of the chain complex. Let c be an element of C_j . If $c \in \ker(\delta_j)$ then we say c is a cycle. However, for any $c \in \text{im}(\delta_{j-1})$ it immediately follows from $\delta_j \delta_{j-1} = 0$ that also $c \in \ker(\delta_j)$ and such a cycle is said to be a trivial cycle. On the other hand, if $c \in \ker(\delta_j)$ but $c \notin \text{im}(\delta_{j-1})$ then c is a non-trivial cycle. If any non-trivial cycles exist then $k_j > 0$, and the value of k_j counts the number of different non-trivial cycles (factoring out homological equivalence). Note that for k_j with the lowest value of j in the chain complex, the matrix δ_{j-1} is not defined and

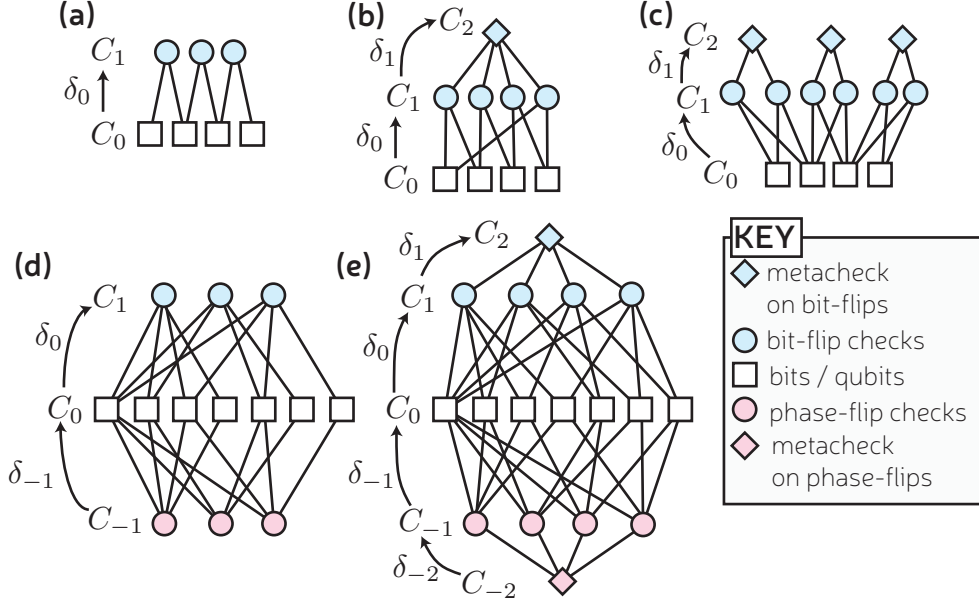


FIG. 1: A graphical representation of some example classical and quantum error correcting codes, including scheme for parity check measurements and metachecks. (a) the 4 bit classical repetition code; (b) the 4 bit classical repetition code with an additional check and corresponding metachecks; (c) the 4 bit classical repetition code with repeated checks and corresponding metachecks; (d) the 7-qubit Steane code; (e) the 7-qubit Steane code with additional checks and corresponding metachecks. The symbol δ_j is a matrix describing the connectivity between vertices in set C_j and C_{j+1} . It can also be considered as a linear map known as the boundary map in homology theory.

so Eq. (34) should be read with δ_{j-1} substituted by the zero matrix. Similarly, for the largest possible j value we must take δ_j as the zero matrix.

One can similarly look at the cohomologies

$$k_j^T := \text{nullity}(\delta_{j-1}^T) - \text{rank}(\delta_j^T). \quad (35)$$

Poincaré duality entails that $k_j^T = k_j$ and for completeness we give a simple proof in App. A using only linear algebra. For quantum codes, k_0 is important as it gives the number of logical qubits encoded by the code. It is useful for us to also to consider k_j for other values of j . For instance, in a code with metachecks, k_1 is the number of classes of syndromes x such that they pass all the metachecks ($\delta_1 x = 0$) but there does not exist an explanation in terms of qubit errors ($\nexists e$ such that $x = \delta_0 e$).

In the context of error correction, we are interested not just in the number of non-trivial cycles, but also their minimum distance. As such, we define

$$d_j := \min\{|c| : c \in \ker(\delta_j), c \notin \text{im}(\delta_{j-1})\}, \quad (36)$$

$$d_j^T := \min\{|c| : c \in \ker(\delta_j^T), c \notin \text{im}(\delta_{j+1}^T)\},$$

where $|c| := \sum_j c_j$ is the Hamming weight. We use the convention that $d_j = \infty$ whenever $k_j = 0$ and similarly $d_j^T = \infty$ whenever $k_{j+1}^T = 0$. We know of no simple relationship between d_j and d_j^T . This is enough for us to define the usual parameters of the corresponding $[[n, k, d_Q]]$ quantum code as $n = n_0$, $k = k_0$ and $d_Q = \min[d_0, d_{-1}^T]$.

However, we also introduce a new parameter that we call the single-shot distance as follows.

Definition 7 (Single-shot distance) *Given a length-4 chain complex we define the single-shot distance as $d_{ss} := \min[d_1, d_{-2}^T]$ where d_1 and d_{-2}^T are special cases of Eq. (36).*

The single-shot distance relates to how many measurement errors can be tolerated before a failure occurs that we call a metacheck failure. In a metacheck failure, the syndrome has no explanation in terms of qubit errors. See Fig. 2b4 for an example of metacheck failure in the 2D Ising model with periodic boundary conditions.

Let us review different ways we can use this formalism. Consider a length-1 chain complex $C_0 \rightarrow_{\delta_0} C_1$. We can consider the vertices in the zeroth level as bits and the first level as parity checks. Thus a length-1 chain complex can be regarded as a classical code. Consider a length-2 chain complex $C_{-1} \rightarrow_{\delta_{-1}} C_0 \rightarrow_{\delta_0} C_1$. This could represent either a quantum code (without any metachecks) or alternatively a classical code equipped with metachecks. In the classical case, our convention is to increment all the indices by one to have $C_0 \rightarrow_{\delta_0} C_1 \rightarrow_{\delta_1} C_2$. We choose this convention such that C_0 always labels the physical bits or qubits. In Fig. 2a1 and Fig. 2b1 we show two graphs representing length-2 chain complexes. The graphs are identical except in Fig. 2a1 it represents a quantum code and in Fig. 2b1 it represents a classical code with metachecks.

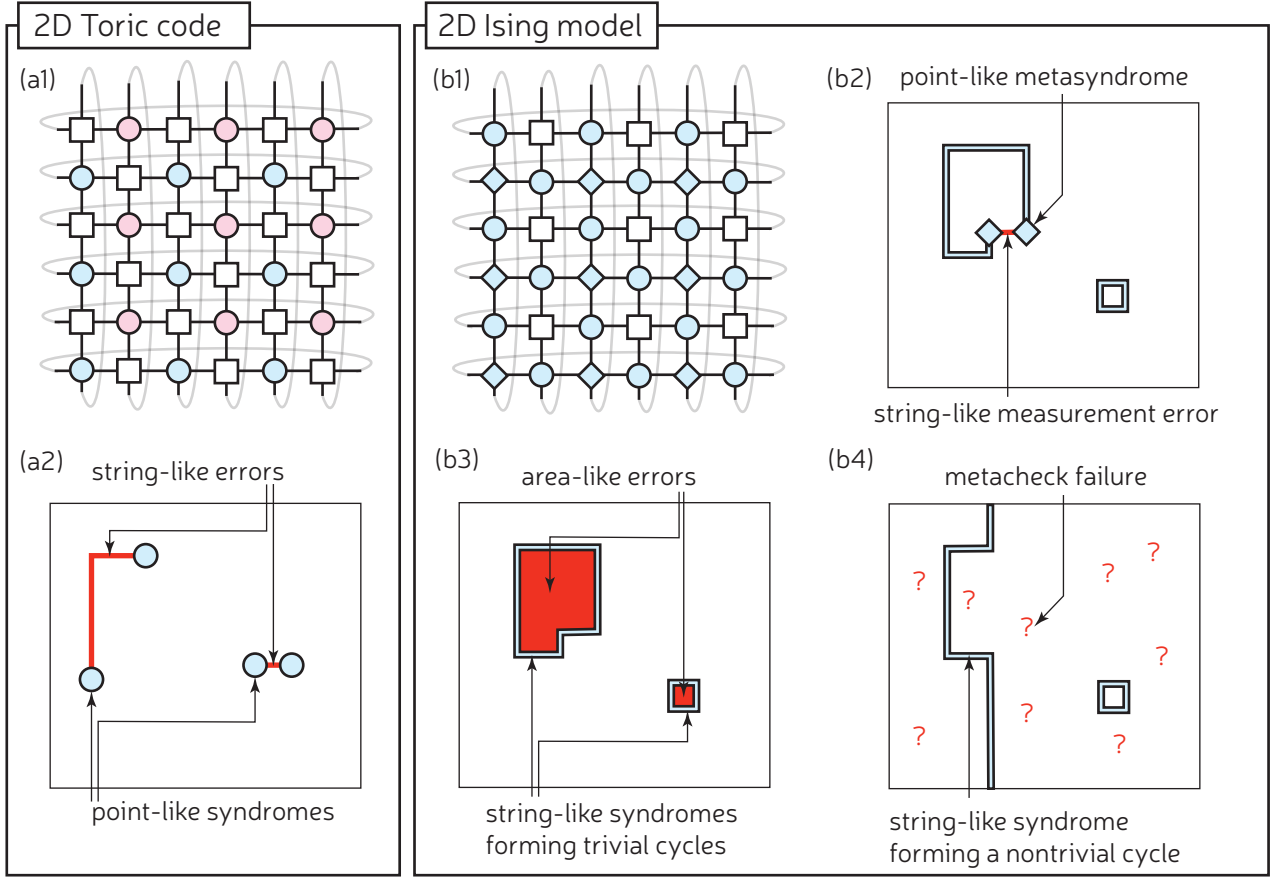


FIG. 2: In (a) we illustrate the 2D toric code. Part (a1) describes the toric code using the vertex labelling from Fig. 1 with grey curved lines highlighting the periodic boundary conditions of the torus. Part (a2) shows the relationship between error and syndromes. Notice that a weight 2 syndrome (two endpoints) could require an arbitrarily long string to produce the syndrome. Therefore, the code does not have good soundness. In (b) we illustrate the 2D Ising model as a classical error correction code. Part (b1) again uses the vertex labelling from Fig. 1. Notice that (b1) represents the same graph as (a1) but with the different types of vertex changing role. Part (b2) shows a measurement error that is detected by metachecks. Part (b3) shows a measurement syndrome that passes all metachecks (i.e. it would be the corrected syndrome of (b2)). The red region shows an error pattern that generates the syndrome. Notice that the size of the physical error scales at most quadratically with the size of the syndrome. Therefore, the code does have good soundness. Part (b4) show a metacheck failure. There is a syndrome that spans the code and forms a non-trivial cycle. Due to periodic boundary conditions there is no error region with this syndrome as its boundary.

Given a length-4 chain complex, the additional layers of homology describe metachecks on the X and Z checks. Note that the additional layers of the chain complex have no direct effect on the code parameters.

We could also consider length-3 chain complexes with metachecks on either X and Z checks. It is also plausible that a length-3 chain complex could support single-shot error correction of both error types by using a form of gauge fixing such as proposed in 3D colour codes [1]. However, we will not explore this here.

We also need to translate the notion of soundness into the language of chain complexes

Definition 8 (Soundness of maps) *Let t be an integer and $f : \mathbb{Z} \rightarrow \mathbb{R}$ be some function called the soundness function. Given a linear map δ , we say it is (t, f) -sound*

if for all r such that $|\delta r| < t$, it follows that:

$$x = |\delta r| \implies \min\{|r'| : \delta r' = \delta r\} \leq f(x). \quad (37)$$

Furthermore, we say a quantum error correcting code is (t, f) -sound if the above holds for both δ_0 and δ_{-1}^T . For a classical error correcting code this is required for just δ_0 .

This is less general than the earlier Def. 3 since the above only applies to CCS codes whereas our earlier definition was valid for any stabiliser code. However, it should be clear that any CCS code satisfying Def. 8 will also satisfy Def. 3. We saw earlier that 2D topological codes cannot have good soundness and we illustrate this in Fig. 2a. Whereas, for the 4D toric code, with an appropriate choice of checks, geometric arguments show that low weight syndromes can always be generated by small

weight errors. To visualise this, it is easier to instead think of the 2D Ising model as a classical error correcting code. In such a code, syndrome cycles have a weight equal to their perimeter and the error generating the syndrome has weight equal to the area (see Fig. 2b3). The area of a 2D region can be no more than $x^2/8$ of the perimeter length x and so the Ising model has a quadratic soundness function. Therefore, it can be helpful to think of soundness as describing the geometric area law relationship between syndromes and errors, albeit in purely algebraic terms.

Check redundancy provides consistency conditions that one can inspect for evidence of measurement errors. These checks on checks are illustrated in Fig. 1 using diamonds. We call these metachecks. They do not represent a physical measurement but classical postprocessing on the measurement outcomes. That is, for a given metacheck node we calculate the parity of all the checks it is connected to. If this parity is odd, a measurement error must have occurred on one of the adjacent nodes. Recall that we quantify the amount of redundancy in a measurement scheme as the ratio between the number of measurements performed (which equals $n_1 + n_{-1}$) and the minimum number required to generate the stabiliser of the code (which equals $n_0 - k_0$). We use v to denote this ratio, so that

$$v = \frac{n_1 + n_{-1}}{n_0 - k_0}, \quad (38)$$

with $v = 1$ indicating no redundancy. In Fig. 1 we give examples of codes with such redundancy (Fig. 1b, Fig. 1c and Fig. 1c). We are interested in check families where v is no more than a small constant factor.

VII. CONSTRUCTING SINGLE-SHOT CODES

Here we show how the homological product can be used to construct new codes supporting single-shot error correction. This will culminate in a proof of Thm. 4 though the techniques allow for a broader range of constructions, including codes where the single-shot distance is finite.

A. A single application constructions

As a warm-up, we begin by considering a single application of the homological product. Our approach is to take a length-1 chain complex (e.g. a conventional classical code) and use the homological, or hypergraph, product to build a length-2 chain complex with the desired properties. In general, one could take two different input classical codes and combine them together using these techniques, but for simplicity we take both input codes to be the same. Furthermore, there are a few different notions of the homological product. For instance, Bravyi and Hastings use a simplified variant that they call the single sector homological product, whereas we will use a

more standard textbook variant that Bravyi and Hastings would call a multi sector homological product [20]. Furthermore, there is some freedom in the notation and we use a convention such that the homological product in this section is manifestly equivalent to the hypergraph product of Tillich and Zemor [16].

Given a chain complex $C_0 \rightarrow_{\delta_0} C_1$ we can define a new chain complex $\tilde{C}_{-1} \rightarrow_{\tilde{\delta}_{-1}} \tilde{C}_0 \rightarrow_{\tilde{\delta}_0} \tilde{C}_1$ of the form

$$C_0 \otimes C_1 \rightarrow_{\tilde{\delta}_{-1}} (C_0 \otimes C_0) \oplus (C_1 \otimes C_1) \rightarrow_{\tilde{\delta}_0} C_1 \otimes C_0. \quad (39)$$

The notation \otimes represents the tensor product. For example, if $a \in C_0$ and $b \in C_1$ then $a \otimes b \in C_0 \otimes C_1$, and the space $C_0 \otimes C_1$ further contains any linear combinations of such vectors. The symbol \oplus represents a direct product. For instance, vectors in $(C_0 \otimes C_0) \oplus (C_1 \otimes C_1)$ can be written as $w = u \oplus v$ where $u \in (C_0 \otimes C_0)$ and $v \in (C_1 \otimes C_1)$. All vectors should be read as column vectors and so the direct product of vectors can also be read as stacking these vectors

$$u \oplus v = \begin{pmatrix} u \\ v \end{pmatrix}. \quad (40)$$

We will use the weight identities $|u \otimes v| = |u| \cdot |v|$ and $|u \oplus v| = |u| + |v|$. The boundary map $\tilde{\delta}_{-1}$ is defined such that for product vectors $a \otimes b \in C_0 \otimes C_1$, we have

$$\tilde{\delta}_{-1}(a \otimes b) = (a \otimes (\delta_0^T b)) \oplus ((\delta_0 a) \otimes b), \quad (41)$$

and it extends linearly to non-product vectors. This is often more concisely denoted as $\tilde{\delta}_{-1} = (\mathbb{1} \otimes \delta_0^T) \oplus (\delta_0 \otimes \mathbb{1})$. The boundary map $\tilde{\delta}_0$ is defined such that for product vectors $a \otimes b \in C_0 \otimes C_0$ and $c \otimes d \in C_1 \otimes C_1$, we have

$$\tilde{\delta}_0((a \otimes b) \oplus (c \otimes d)) = ((\delta_0 a) \otimes b) + (c \otimes (\delta_0^T d)), \quad (42)$$

and again extending linearly to non-product vectors. Both the new boundary maps can also be represented in block matrix form

$$\tilde{\delta}_{-1} = \begin{pmatrix} \mathbb{1} \otimes \delta_0^T \\ \delta_0 \otimes \mathbb{1} \end{pmatrix}, \quad (43)$$

$$\tilde{\delta}_0 = \begin{pmatrix} \delta_0 \otimes \mathbb{1} & \mathbb{1} \otimes \delta_0^T \end{pmatrix}.$$

From here it is easy to verify that they satisfy the requirement that $\tilde{\delta}_0 \tilde{\delta}_{-1} = 2(\delta_0 \otimes \delta_0^T) = 0$, where we have used that all mathematics is being performed modulo 2. These matrices fully characterise the new chain complex and from them we can find graphs of the sort shown in Fig. 1. We give a graphical overview in Fig. 3.

Now we discuss the parameters of this new structure, with some of these results obtained in Ref. [16]. Simple dimension counting tells us that the new chain complex has

$$\begin{aligned} \tilde{n}_{-1} &= n_0 n_1, \\ \tilde{n}_0 &= n_0^2 + n_1^2, \\ \tilde{n}_1 &= n_0 n_1. \end{aligned} \quad (44)$$

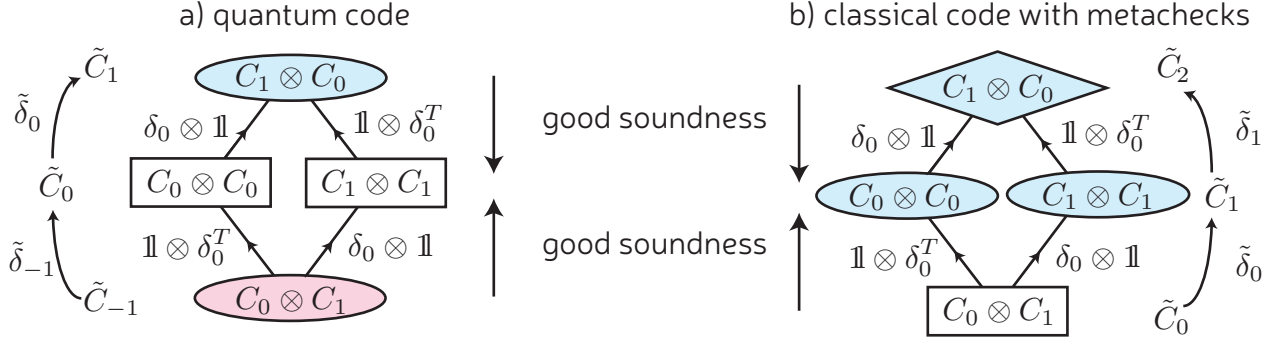


FIG. 3: An overview of a single application of the homological product to generate a length-2 chain complex from a length-1 chain complex (that can be viewed as a classical code). In (a) we label the chain-complex under the assumption that it defines a quantum code, and where the subscripts are consistent with the main text. In (b) we label the chain-complex under the assumption that it defines a classical code. In order that \tilde{C}_0 denotes the bits, we have incremented all the new subscripts by 1. Throughout we use rectangles to show a collection of bit/qubit vertices; we use ovals to show a collection of checks; and diamonds to show a collection of metachecks.

The dimension of the homological classes is more involved, but a well known result from homology theory (the Künneth formula [20, 30]) tells us that

$$\begin{aligned}\tilde{k}_{-1} &= k_0 k_1, \\ \tilde{k}_0 &= k_0^2 + k_1^2, \\ \tilde{k}_1 &= k_1 k_0.\end{aligned}\quad (45)$$

The distance of the code is trickier yet again to prove and is not a standard quantity in homology theory. Nevertheless, one can show that

$$\tilde{d}_{-1} = d_0 d_0^T, \quad (46)$$

$$\tilde{d}_0^T = d_0 d_0^T, \quad (47)$$

$$\tilde{d}_0 \geq \min(d_0, d_0^T), \quad (48)$$

$$\tilde{d}_{-1}^T \geq \min(d_0, d_0^T). \quad (49)$$

We provide proofs in App. C for Eq. (46) and Eq. (47). The results of Eq. (48) and Eq. (49) were shown by Tillich and Zemor [16] but we give an independent proof in the homological formalism in App. C.

Here we instead focus on the following lemma

Lemma 5 (First soundness lemma) *Let $C_0 \rightarrow_{\delta_0} C_1$ be a chain complex. Applying the above homological product we obtain a new chain complex where the map $\tilde{\delta}_0^T$ is (t, f) -sound and $\tilde{\delta}_{-1}$ is (t, f) -sound with $f(x) = x^2/4$ and $t = \min(d_0, d_0^T)$.*

We make no assumptions about the soundness properties of the original chain complex but find this emerges due to the nature of the homological product. However, if one knows that the original chain complex is sound, one could prove a stronger soundness result (with f growing slower than $x^2/4$) for the new chain complex. We prove this lemma in App. D and next discuss its implications.

Using the above homological product, we can construct a quantum code with parameters $[[\tilde{n}_0, \tilde{k}_0, d_Q]]$

where $d_Q = \min[\tilde{d}_0^T, \tilde{d}_0]$. These codes will not necessarily support single-shot error correction because the soundness property in Lem. 5 is not the property required by Thm. 1, which requires that $\tilde{\delta}_0$ and $\tilde{\delta}_{-1}^T$ have good soundness properties.

Why prove Lem. 5 if it does not directly provide quantum codes with single-shot capabilities? First, in the next section we will make a second application of the homological product and Lem. 5 will be used, and so it is a stepping stone result. Second, Lem. 5 is highly instructive as it gives a way to construct classical codes that exhibit single-shot error correction. Let us explore this second point further. A classical code with metachecks needs three layers of structure (recall Fig. 1) and our convention is that the subscript 0 in C_0 always denotes the bits or qubits. So for a classical code with metachecks, we want a chain complex of the form $\tilde{C}_0 \rightarrow_{\tilde{\delta}_0} \tilde{C}_1 \rightarrow_{\tilde{\delta}_1} \tilde{C}_2$. We can use the chain complex generated by the homological product by simply increasing all the subscripts by 1. With these incremented subscripts, Lem. 5 tells us that $\tilde{\delta}_0$ is (d_0^T, f) -sound with $f(x) = x^2/4$. It is easy to get lost in subscripts, so we emphasize that the important feature is that soundness runs in the direction from bits/qubits to checks. This is illustrated in Fig. 3 where it clearly runs the correct way for the classical code but not the quantum code. For instance, the 2D toric code and 2D Ising code can both be obtained by applying the homological product to a classical repetition code, but only the 2D Ising code exhibits good soundness (recall Fig. 2).

Next, we comment on the redundancy of the new quantum code.

Claim 2 (Updated redundancy) *Let $C_0 \rightarrow_{\delta_0} C_1$ be a chain complex associated with an $[[n, k, d]]$ classical code with check redundancy $v = n_1/(n_0 - k_0)$. Applying the above homological product we obtain a new chain complex*

and associated quantum code with check redundancy

$$\tilde{v} = v \frac{n}{v(n-k) + k} < 2v. \quad (50)$$

Notice that if $v = 1$ then $\tilde{v} = 1$.

To prove this, we begin with the definition of redundancy and apply Eqs. (44) and Eqs. (45)

$$\tilde{v} = \frac{\tilde{n}_1 + \tilde{n}_{-1}}{\tilde{n}_0 - \tilde{k}_0} \quad (51)$$

$$= \frac{2n_0n_1}{n_0^2 + n_1^2 - k_0^2 - k_1^2} \quad (52)$$

$$= \frac{2n_0n_1}{(n_0 - k_0)(n_0 + k_0) + (n_1 - k_1)(n_1 + k_1)}. \quad (53)$$

Using that for a length-1 chain complex $n_1 - k_1 = n_0 - k_0$ and the definition of v , we find

$$\begin{aligned} \tilde{v} &= \frac{2n_0n_1}{(n_0 - k_0)(n_0 + k_0 + n_1 + k_1)} \\ &= 2v \frac{n_0}{n_0 + k_0 + n_1 + k_1}. \end{aligned} \quad (54)$$

Since the fraction is clearly less than 1, we have that $\tilde{v} < 2v$. Furthermore, using $n_1 - k_1 = n_0 - k_0$ to eliminate k_1 and $v = n_1/(n_0 - k_0)$ to eliminate n_1 , we obtain

$$\tilde{v} = v \frac{n_0}{v(n_0 - k_0) + k_0}, \quad (55)$$

and the identification $n = n_0$ and $k = k_0$ gives the final expression for \tilde{v} .

We conclude this section by considering a simple application of the above homological product. Given a classical $[n, k, d]$ code, we can associate many different length-1 chain complexes, depending on whether there is redundancy in the check operators. However, for any code there always exists a minimal chain complex where there is no redundancy ($v = 1$). For such a minimal chain complex, we have $n_1 = n - k$, $k_1 = 0$ and $d_0^T = \infty$. This is useful as it allows us to make statements that depend only on well known code properties.

Corollary 2 (Quantum code constructions)

Consider a classical $[n, k, d]$ code. Applying the above homological product to the minimal chain complex of this code, we obtain a $[[2n(n-k) + k^2, k^2, d]]$ quantum code with no check redundancy.

B. A second application of the homological product

For a quantum error correcting code with metachecks we need a length-4 chain complex, which can be constructed by applying the homological product to a length-2 chain complex. We use breve ornaments over symbols in this section to identify matrices, variables and vector spaces associated with the length-4 chain complex, as follows

$$\check{C}_{-2} \rightarrow_{\check{\delta}_{-2}} \check{C}_{-1} \rightarrow_{\check{\delta}_{-1}} \check{C}_0 \rightarrow_{\check{\delta}_0} \check{C}_1 \rightarrow_{\check{\delta}_1} \check{C}_2. \quad (56)$$

The homological product between a pair of 2-dimensional chain complexes will generate a length-4 chain complex according to the general rule that

$$\check{C}_m = \bigoplus_{i-j=m} \check{C}_i \otimes \check{C}_j. \quad (57)$$

The boundary maps are illustrated in Fig. 4 and can be written as block matrices as follows

$$\check{\delta}_{-2} = \begin{pmatrix} \mathbb{1} \otimes \check{\delta}_0^T \\ \check{\delta}_{-1} \otimes \mathbb{1} \end{pmatrix}, \quad (58)$$

$$\check{\delta}_{-1} = \begin{pmatrix} \mathbb{1} \otimes \check{\delta}_{-1}^T & 0 \\ \check{\delta}_{-1} \otimes \mathbb{1} & \mathbb{1} \otimes \check{\delta}_0^T \\ 0 & \check{\delta}_0 \otimes \mathbb{1} \end{pmatrix}, \quad (59)$$

$$\check{\delta}_0 = \begin{pmatrix} \check{\delta}_{-1} \otimes \mathbb{1} & \mathbb{1} \otimes \check{\delta}_{-1}^T & 0 \\ 0 & \check{\delta}_0 \otimes \mathbb{1} & \mathbb{1} \otimes \check{\delta}_0^T \end{pmatrix}, \quad (60)$$

$$\check{\delta}_1 = (\check{\delta}_0 \otimes \mathbb{1} \quad \mathbb{1} \otimes \check{\delta}_{-1}^T). \quad (61)$$

One can verify that $\check{\delta}_{j+1}\check{\delta}_j = 0$ for all j follows from the same condition on the $\check{\delta}$ matrices. As before, one obtains the relations

$$\check{n}_m = \sum_{i-j=m} \check{n}_i \check{n}_j, \quad (62)$$

$$\check{k}_m = \sum_{i-j=m} \check{k}_i \check{k}_j,$$

where the first is simple dimension counting and the second line follows from the Künneth formula.

The distances are lower bounded as follows

$$\check{d}_0, \check{d}_{-1}^T \geq \min[\check{d}_{-1}, \max[\check{d}_0, \check{d}_{-1}^T], \check{d}_0^T], \quad (63)$$

$$\check{d}_1, \check{d}_{-2}^T \geq \min[\check{d}_0, \check{d}_{-1}^T],$$

which we prove in App. E. Note that the distance will often be significantly larger than these lower bounds. Our main technical goal is to prove the following soundness result.

Lemma 6 (Second soundness lemma) Let

$\check{C}_{-1} \rightarrow_{\check{\delta}_{-1}} \check{C}_0 \rightarrow_{\check{\delta}_0} \check{C}_1$ be a chain complex such that $\check{\delta}_0^T$ is (t, f) -sound and $\check{\delta}_{-1}$ is (t, f) -sound with $f(x) = x^2/4$. Applying the above homological product we obtain a new length-4 chain complex (as in Eq. 56) where the map $\check{\delta}_0$ is (t, g) -sound and $\check{\delta}_{-1}^T$ is (t, g) -sound with soundness function $g(x) = x^3/4$.

We show the direction of the resulting soundness in Fig. 4 and this should be contrasted with the direction of the soundness arrows in Fig. 3. We will only prove the results for $\check{\delta}_0$ with the proof for $\check{\delta}_{-1}^T$ being essentially identical.

Let us first discuss how the problem can be divided into three subproblems. Let $s \in \text{im}(\check{\delta}_0)$ so there must exist at least one $r \in \check{C}_0$ such that $\check{\delta}_0 r = s$. We divide r into components

$$r = \begin{pmatrix} r_a \\ r_b \\ r_c \end{pmatrix}, \quad (64)$$

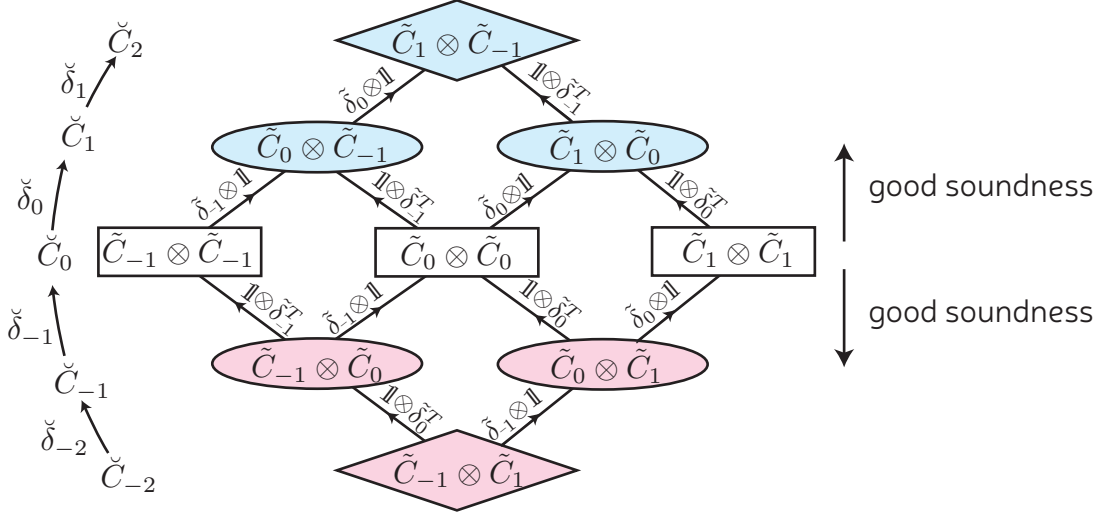


FIG. 4: An overview of the second application of the homological product to generate a length-4 chain complex from a two dimensional chain complex (that can be viewed as a quantum code).

and consider two distinct images

$$s_L = (\tilde{\delta}_{-1} \otimes \mathbb{1})r_a + (\mathbb{1} \otimes \tilde{\delta}_{-1}^T)r_b, \quad (65)$$

$$s_R = (\mathbb{1} \otimes \tilde{\delta}_0^T)r_c + (\tilde{\delta}_0 \otimes \mathbb{1})r_b, \quad (66)$$

where

$$s = \check{\delta}_0 r = \begin{pmatrix} s_L \\ s_R \end{pmatrix}. \quad (67)$$

One always has the weight relations $|r| = |r_a| + |r_b| + |r_c|$ and $|s| = |s_L| + |s_R|$.

For a syndrome that passes all metachecks we have that

$$\check{\delta}_1 s = (\tilde{\delta}_0 \otimes \mathbb{1})s_L + (\mathbb{1} \otimes \tilde{\delta}_{-1}^T)s_R = 0, \quad (68)$$

which entails that

$$m := (\tilde{\delta}_0 \otimes \mathbb{1})s_L = (\mathbb{1} \otimes \tilde{\delta}_{-1}^T)s_R, \quad (69)$$

where we have defined this new quantity to be m . Given a physical error pattern r that generates the syndrome (as in Eqs. (65)-(66)) the metachecks are always passed and one finds that

$$m = (\tilde{\delta}_0 \otimes \tilde{\delta}_{-1}^T)r_b. \quad (70)$$

It is interesting that this depends only on the r_b component of r . We can first try to find low weight r_b that solves Eq. (70). This leads to the following partial solution to the problem

Lemma 7 (Partial soundness result) *Let $\tilde{C}_{-1} \rightarrow_{\tilde{\delta}_{-1}} \tilde{C}_0 \rightarrow_{\tilde{\delta}_0} \tilde{C}_1$ be a chain complex. Applying the above homological product we obtain a new length-4 chain complex (as in Eq. 56) with the following property. For any $s \in \text{im}(\check{\delta}_0)$ there exists an r_b with the following properties*

1. *correctness:* $(\tilde{\delta}_0 \otimes \tilde{\delta}_{-1}^T)r_b = m = (\tilde{\delta}_0 \otimes \mathbb{1})s_L = (\mathbb{1} \otimes \tilde{\delta}_{-1}^T)s_R$;
2. *low weight:* $|r_b| \leq |s_L| \cdot |s_R|$;
3. *small s_L remainder:* $s_L - (\mathbb{1} \otimes \tilde{\delta}_{-1}^T)r_b = \sum_i \alpha_i \otimes \hat{a}_i$ where \hat{a}_i are unit vectors and $\alpha_i \in \ker \tilde{\delta}_0$. There are at most $|s_L|$ nonzero α_i and these are bounded in size $|\alpha_i| \leq |s_L|$;
4. *small s_R remainder:* $s_R - (\tilde{\delta}_0 \otimes \mathbb{1})r_b = \sum_i \hat{b}_i \otimes \beta_i$ where \hat{b}_i are unit vectors and $\beta_i \in \ker \tilde{\delta}_{-1}^T$. There are at most $|s_R|$ nonzero β_i and these are bounded in size $|\beta_i| \leq |s_R|$.

The proof has a similar flavour to the earlier soundness result and is deferred until App. F. Notice that the lemma does not require any soundness of the initial chain complex. Next, we want to find low-weight r_a and r_c such that they provide the remaining elements of the syndrome as follows

$$(\tilde{\delta}_{-1} \otimes \mathbb{1})r_a = s_L - (\mathbb{1} \otimes \tilde{\delta}_{-1}^T)r_b, \quad (71)$$

$$(\mathbb{1} \otimes \tilde{\delta}_0^T)r_c = s_R - (\tilde{\delta}_0 \otimes \mathbb{1})r_b. \quad (72)$$

Fortunately, Lem. 7 ensures that these remainder syndromes are “small” in the defined sense. We may next use the following observation

Claim 3 (Inheritance of soundness) *If $\tilde{\delta}_{-1}$ is (t, f) -sound then $\tilde{\delta}_{-1} \otimes \mathbb{1}$ is also sound in the following strong sense. Let $q \in \text{im}(\tilde{\delta}_{-1} \otimes \mathbb{1})$ with decomposition $q = \sum_i \alpha_i \otimes \hat{a}_i$ such that $|\alpha_i| < t$ then there exists an r_a such that $(\tilde{\delta}_{-1} \otimes \mathbb{1})r_a = q$ and $|r_a| \leq \sum_i f(|\alpha_i|)$. A similar result holds when we interchange the order of tensor products and consider $\tilde{\delta}_0^T$.*

The proof is fairly straightforward. Since $|\alpha_i| < t$ for all i and by assumption $\tilde{\delta}_{-1}$ is (t, f) -sound, there must exist γ_i such that $\tilde{\delta}_{-1}\gamma_i = \alpha_i$ and $|\gamma_i| \leq f(|\alpha_i|)$. By linearity, there exists $r_a = \sum_i \gamma_i \otimes \hat{a}_i$ such that $(\tilde{\delta}_{-1} \otimes \mathbb{1})r_a = q$ and $|r_a| \leq \sum_i |\gamma_i| \leq \sum_i f(|\alpha_i|)$.

Next, we put these pieces together. Combining Lem. 7 and Claim. 3 together with the assumption that $|s| < t$ one immediately obtains that there exist r_a and r_c solving Eq. (71) with weights upper bounded by

$$|r_a| \leq |s_L|f(|s_L|) \quad (73)$$

$$|r_c| \leq |s_R|f(|s_R|) \quad (74)$$

Therefore, we have the total weight

$$|r| \leq |s_L|f(|s_L|) + |s_L| \cdot |s_R| + |s_R|f(|s_R|). \quad (75)$$

We take $f(x) = x^2/4$ as stated in Thm. 6, which leads to

$$|r| \leq \frac{1}{4}|s_L|^3 + |s_L| \cdot |s_R| + \frac{1}{4}|s_R|^3 \quad (76)$$

$$\leq \frac{1}{4}(|s_L| + |s_R|)^3 \quad (77)$$

$$= \frac{1}{4}|s|^3. \quad (78)$$

Therefore, we have proven (t, g) -sound of $\check{\delta}_0$ with $g(x) = x^3/4$. This completes the proof that Thm. 6 follows from Lem. 7.

Next, we comment on the check redundancy of these codes

Claim 4 (Updated redundancy part 2) *Consider a length-2 chain complex and associated quantum code with check redundancy \tilde{v} . Applying the above homological product we obtain a length-4 chain complex and new quantum code with check redundancy $\check{v} < 2\tilde{v}$.*

To prove this we recall the definition of redundancy and then use Eqs. (62) to obtain

$$\check{v} = \frac{\check{n}_1 + \check{n}_{-1}}{\check{n}_0 - \check{k}_0} \quad (79)$$

$$= \frac{2\tilde{n}_0(\tilde{n}_1 + \tilde{n}_{-1})}{(\tilde{n}_{-1}^2 + \tilde{n}_0^2 + \tilde{n}_1^2) - (\tilde{k}_{-1}^2 + \tilde{k}_0^2 + \tilde{k}_1^2)}. \quad (80)$$

Since $\tilde{n}_j \geq \tilde{k}_j$ for all j , the denominator is greater than $\tilde{n}_0^2 - \tilde{k}_0^2$, which itself can be factorised as $(\tilde{n}_0 - \tilde{k}_0)(\tilde{n}_0 + \tilde{k}_0)$ and so

$$\check{v} \leq \frac{2\tilde{n}_0(\tilde{n}_1 + \tilde{n}_{-1})}{(\tilde{n}_0 - \tilde{k}_0)(\tilde{n}_0 + \tilde{k}_0)}, \quad (81)$$

$$= 2 \left(\frac{\tilde{n}_1 + \tilde{n}_{-1}}{\tilde{n}_0 - \tilde{k}_0} \right) \left(\frac{\tilde{n}_0}{\tilde{n}_0 + \tilde{k}_0} \right), \quad (82)$$

$$= 2\tilde{v} \left(\frac{\tilde{n}_0}{\tilde{n}_0 + \tilde{k}_0} \right), \quad (83)$$

Last, we use the loose bound that the fraction is less than 1 to conclude that $\check{v} \leq 2\tilde{v}$ as claimed.

C. Combining homological products

Here we combine the results of the preceding two subsections. Parameters carrying a breve are first expressed in term of parameters carrying a tilde, and then the tilde parameters are replaced with unornamented parameters.

$$\check{n}_0 = \check{n}_1^2 + \check{n}_0^2 + \check{n}_{-1}^2 = (n_0^2 + n_1^2)^2 + 2n_0^2n_1^2, \quad (84)$$

$$\check{n}_1 = \check{n}_{-1} = \check{n}_0(\check{n}_1 + \check{n}_{-1}) = 2(n_0^2 + n_1^2)n_0n_1,$$

$$\check{k}_0 = \check{k}_1^2 + \check{k}_0^2 + \check{k}_{-1}^2 = (k_0^2 + k_1^2)^2 + 2k_0^2k_1^2,$$

$$\check{k}_1 = \check{k}_{-1} = \check{k}_0(\check{k}_1 + \check{k}_{-1}) = 2(k_0^2 + k_1^2)k_0k_1,$$

$$\check{d}_0 = \check{d}_{-1}^T \geq \min[d_0, d_0^T],$$

$$\check{d}_1 = \check{d}_{-1}^T \geq \min[d_0, d_0^T].$$

Furthermore, by combining Claim. 2 and Claim. 4 we obtain an upper bound on the check redundancy

$$\check{v} < 2\tilde{v} = 2v \frac{n}{v(n-k) + k}, \quad (85)$$

where v is the check redundancy of the $[n, k, d]$ classical code associated with the initial length-1 chain complex.

The simplest case is when we use a minimal chain complex representing the initial $[n, k, d]$ classical code. Then $v = 1$, $k_1 = 0$ and $n_1 = n - k$ and the above equations simplify to

$$\check{n}_0 = n^4 + 4n^2(n-k)^2 + (n-k)^4, \quad (86)$$

$$\check{n}_1 = \check{n}_{-1} = 2n(n-k)(n^2 + (n-k)^2),$$

$$\check{k}_0 = k^4,$$

$$\check{k}_1 = \check{k}_{-1} = 0$$

$$\check{v} < 2.$$

$$\check{d}_0 = \check{d}_{-1}^T \geq d,$$

We also know that $\check{d}_1 = \check{d}_{-1}^T = \infty$ as a consequence of $\check{k}_1 = \check{k}_{-1} = 0$. We make the following identifications: \check{n}_0 gives the number of physical qubits n_Q ; \check{k}_0 is the number of logical qubits k_Q ; \check{d}_0 and \check{d}_{-1}^T give the qubit error distance d_Q ; and \check{d}_1 and \check{d}_{-2}^T give the single-shot distance d_{ss} . This proves Thm. 4. We remark that in the final stages of this research, Zeng and Pryadko posted a preprint [31] that shows that the distance is much better than suggested by our bounds, in particular $\check{d}_0 = \check{d}_{-1}^T = d^2$.

In Table I we provide some concrete examples. These are the smallest examples since they use very small initial classical codes. Though the resulting quantum code is much larger. The first three examples correspond to 4D toric codes with cubic tiling either with closed boundary conditions (examples 1 and 2) or periodic boundary conditions (example 3). The last example corresponds to no previous codes that we know of. We have deliberately chosen codes that have low check weight as these will be the most experimentally feasible. Our constructions

δ	Input classical code					Double homological product code						
	parameters n k d	max. check weight	redundancy v	parameters n_Q k_Q d_Q d_{ss}	max. check weight	mean check weight	redundancy \check{v}					
$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$	3 1 3	2	1	241 1 9 ∞	6	4.87179	1.3					
$\begin{pmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$	4 1 4	2	1	913 1 16 ∞	6	5.18	1.31579					
$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$	3 1 3	2	1.5	486 6 9 3	6	6	1.33884					
$\begin{pmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$	6 2 4	3	1	3856 16 16 ∞	8	5.48077	1.3					

TABLE I: Some example small classical codes used to generate a quantum code with good soundness through a double application of the homological product. Many of the parameters come directly from equations in the main text. The mean check weight and redundancy are calculated exactly by constructing the explicit parity check matrices. Our table uses the improved distance d_Q results of Zeng and Pryadko [31].

could potentially be slightly improved using a generalisation of the hypergraph improvements analogous to use of rotated toric lattices [32].

VIII. DISCUSSION & CONCLUSIONS

This is a paper of two halves. The first half was conceptual and gave a presentation of single-shot error correction. We found an intimate connection between single-shot error correction and a property called good soundness. We saw that good soundness in LDPC codes entails a macroscopic energy barrier, which further confirms a relationship between passive quantum memories and single-shot error correction. However, our results leave open whether there exist any codes with a macroscopic energy barrier that lack good soundness. Michael Beverland suggested in discussion that it would be interesting to look at whether Haah’s cubic code [33, 34] has good soundness. The Haah cubic code is notable because it does have a macroscopic energy barrier but is not a good passive quantum memory at all scales due to entropic effects. Also curious is the role of metachecks and redundancy. We saw that good soundness can be achieved by any code without any check redundancy, but the proof used a diagonalised form of the stabiliser generators that typically destroys any LDPC properties.

The second half of this paper was more technical and focused on specific code constructions capable of providing both good soundness and LDPC properties. It has long been known that homology theory provides a natural mathematical framework for CCS codes, but we saw that homology theory is especially useful when metachecks (checks on measurements) are added to the picture. It is well known that for topological codes the energy bar-

rier and single-shot error correction are intimately related to the dimensionality of the code. We abstract away the topological structure and instead work with algebraic homological structure. While these codes no longer have a dimensionality in the geometric sense, we saw that using the homological product can imbue codes with a sort of effective dimensionality. More precisely, a double application of the homological product resulted in single-shot properties similar to 4-dimensional topological codes. Many readers will feel more comfortable with topological codes because of the conceptual and visual crutches they provide. However, topological codes are significantly limited in terms of the code parameters they can achieve due to trade-off bounds [22, 23]. So by freeing ourselves from the constraints of topological codes and pursuing their more abstract cousins, we can seemingly benefit from many of the advantages of high-dimensional topological codes (e.g. single-shot error correction) but with improved code parameters. This prompts the question what other topological code properties might hold for homological product codes. We know that 3D and 4D topological codes can support transversal non-Clifford gates [35–41], which suggests that a similar property might hold for suitably defined homological product codes.

Our code constructions married good soundness and LDPC properties, through the use of check redundancy and associated metachecks. But do any codes exist without check redundancy that are useful for single-shot error correction? A related question is whether our soundness properties are necessary conditions for single-shot error correction as well as being sufficient conditions. While finishing this research, work on quantum expander codes [5] has shown that they can perform single-shot error correction without any check redundancy. Initially,

we speculated (in an early preprint) that the quantum error codes will have good soundness, but Leverrier has shown (in private correspondence) that they do not have this property! Therefore, there is more work to be done on this topic to find a code property more permissive than soundness that encompasses all of our codes and also the quantum expander codes.

The main limitation of this work is that we restrict our attention to adversarial noise. Stochastic noise models instead distribute errors according to some probability distribution and assign a non-zero probability to every error configuration. If the probability of a high weight error is low, then we can still leverage proofs from the adversarial noise setting. However, in an independent noise model where each qubit is affected with probability p , a code with n qubits will typically suffer around pn errors. For all known quantum LDPC code families, the distance scales sublinearly, and so there is some scale at which the code is likely to suffer an error considerable larger than the code distance. Nevertheless, one is often able to prove the existence of an error correcting threshold. The crucial point is that even though some errors of weight pn might not be correctable, these represent a small fraction of all weight pn errors and so happen with small probability. At this point, proof techniques diverge. We can prove that this works for concatenated codes, topological codes and low-density parity check codes [42]. As such, while there is a single theoretical framework for adver-

sarial noise, there is no single theory for stochastic noise in all settings. The situation is likely the same in the setting of single-shot error correction. The pioneering work of Bombin demonstrated that three dimensional colour codes can perform single-shot error correction against a stochastic noise model [1], and so in this sense our results are strictly weaker. On the other hand, our approach is strictly more general as it applies to a broad range of codes, including many new code constructions such as those presented here. It is then natural to wonder what are sufficient and necessary conditions for single-shot error correction to work against stochastic noise? It is reasonable to conjecture that any concatenated or LDPC codes that meets our criteria for adversarial noise will also perform single-shot error correction against stochastic noise.

Acknowledgements.- This work was supported by the EPSRC (EP/M024261/1) and the QCDA project which has received funding from the QuantERA ERA-NET Co-fund in Quantum Technologies implemented within the European Union's Horizon 2020 Programme. I would like to thank Nicolas Delfosse for his tutorial on hypergraph product codes during the FTQT 2016 workshop at the Centro de Ciencias de Benasque Pedro Pascual. Thank you to Simon Willerton, Michael Beverland, Mike Vasmer, Anthony Leverrier, Barbara Terhal and Ben Brown for conversations and comments on the manuscript. Referee 2 is thanked for their diligent attention to detail.

-
- [1] H. Bombin, Phys. Rev. X **5**, 031043 (2015).
 [2] H. Bombin, Phys. Rev. X **6**, 041034 (2016).
 [3] H. Bombin, New Journal of Physics **17**, 083002 (2015).
 [4] E. Dennis, A. Kitaev, A. Landahl, and J. Preskill, Journal of Mathematical Physics **43**, 4452 (2002).
 [5] O. Fawzi, A. Grospellier, and A. Leverrier, to appear in FOCS 2018.
 [6] Y. Fujiwara, Phys. Rev. A **90**, 062304 (2014).
 [7] A. Ashikhmin, C.-Y. Lai, and T. A. Brun, in *Information Theory (ISIT), 2016 IEEE International Symposium on* (IEEE, 2016), pp. 2274–2278.
 [8] B. J. Brown, N. H. Nickerson, and D. E. Browne, Nat Commun **7** (2016).
 [9] N. P. Breuckmann, K. Duivenvoorden, D. Michels, and B. M. Terhal, Quant. Inf. and Comp. **17**, 0181 (2017).
 [10] D. Aharonov and L. Eldar, SIAM Journal on Computing **44**, 1230 (2015).
 [11] M. B. Hastings, in *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)* (2017), vol. 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pp. 25:1–25:26.
 [12] R. Alicki, M. Horodecki, P. Horodecki, and R. Horodecki, Open Systems & Information Dynamics **17**, 1 (2010).
 [13] B. M. Terhal, Rev. Mod. Phys. **87**, 307 (2015).
 [14] B. J. Brown, D. Loss, J. K. Pachos, C. N. Self, and J. R. Wootton, Rev. Mod. Phys. **88**, 045005 (2016).
 [15] D. Bacon, Phys. Rev. A **73**, 012340 (2006).
 [16] J.-P. Tillich and G. Zémor, IEEE Transactions on Information Theory **60**, 1193 (2014).
 [17] O. Fawzi, A. Grospellier, and A. Leverrier, in *Proc. STOC* (ACM, 2018), pp. 521–534.
 [18] I. Dumer, A. A. Kovalev, and L. P. Pryadko, Phys. Rev. Lett. **115**, 050502 (2015).
 [19] A. A. Kovalev, S. Prabhakar, I. Dumer, and L. P. Pryadko, Physical Review A **97**, 062320 (2018).
 [20] S. Bravyi and M. B. Hastings, in *Proceedings of the forty-sixth annual ACM symposium on Theory of computing* (ACM, 2014), pp. 273–282.
 [21] B. Audoux and A. Couvreur, arXiv preprint arXiv:1512.07081 (2015).
 [22] S. Bravyi, D. Poulin, and B. Terhal, Phys. Rev. Lett. **104**, 050503 (2010).
 [23] N. Delfosse, in *Information Theory Proceedings (ISIT), 2013 IEEE International Symposium on* (IEEE, 2013), pp. 917–921.
 [24] N. P. Breuckmann, Ph.D. thesis, Aachen, arXiv preprint arXiv:1802.01520 (2018).
 [25] S. Bravyi and B. Terhal, New Journal of Physics **11**, 043029 (2009).
 [26] J. W. Harrington, Ph.D. thesis (2004), http://thesis.library.caltech.edu/1747/1/jimh_thesis.pdf.
 [27] M. Herold, E. T. Campbell, J. Eisert, and M. J. Kastoryano, npj Quantum Information **1**, 15010 (2015).
 [28] M. Herold, M. J. Kastoryano, E. T. Campbell, and J. Eisert, New Journal of Physics **19**, 063012 (2017).
 [29] E. T. Campbell and D. E. Browne, Phys. Rev. Lett. **104**, 030503 (2010).
 [30] A. Hatcher, Cambridge UP, Cambridge **606** (2002).

- [31] W. Zeng and L. P. Pryadko, arXiv preprint arXiv:1810.01519 (2018).
- [32] A. A. Kovalev and L. P. Pryadko, in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on* (IEEE, 2012), pp. 348–352.
- [33] J. Haah, Phys. Rev. A **83**, 042330 (2011).
- [34] S. Bravyi and J. Haah, Phys. Rev. Lett. **111**, 200501 (2013).
- [35] H. Bombin and M. A. Martin-Delgado, Phys. Rev. Lett. **97**, 180501 (2006).
- [36] H. Bombin, R. Chhajlany, M. Horodecki, and M. Martin-Delgado, New Journal of Physics **15**, 055023 (2013).
- [37] F. H. Watson, E. T. Campbell, H. Anwar, and D. E. Browne, Phys. Rev. A **92**, 022312 (2015).
- [38] A. Kubica, B. Yoshida, and F. Pastawski, New Journal of Physics **17**, 083026 (2015).
- [39] A. Kubica and M. E. Beverland, Phys. Rev. A **91**, 032330 (2015).
- [40] M. Vasmer and D. E. Browne, arXiv preprint arXiv:1801.04255 (2018).
- [41] E. T. Campbell, B. M. Terhal, and C. Vuillot, Nature **549**, 172 (2017).
- [42] A. A. Kovalev and L. P. Pryadko, Phys. Rev. A **87**, 020304 (2013).

Appendix A: A simple proof of relation between Betti numbers

We give a simple proof that $k_j = k_j^T$ as defined in Eq. (34) and Eq. (35). The proof uses simple linear algebra rather than sophisticated homological techniques that are needed in more exotic settings. We use the rank-nullity theorem that for any matrix A ,

$$\text{rank}(A) + \text{nullity}(A) = n, \quad (\text{A1})$$

where n is the number of columns in A . This entails that

$$\text{rank}(\delta_j) + \text{nullity}(\delta_j) = n_j, \quad (\text{A2})$$

$$\text{rank}(\delta_{j-1}^T) + \text{nullity}(\delta_{j-1}^T) = n_j. \quad (\text{A3})$$

Taking the definition of k_j^T (recall Eq. (35)) and using Eq. (A3) to eliminate the dependence on $\text{nullity}(\delta_{j-1}^T)$, we obtain

$$k_j^T = n_j - \text{rank}(\delta_{j-1}^T) - \text{rank}(\delta_j^T). \quad (\text{A4})$$

Using that for any matrix $\text{rank}(A) = \text{rank}(A^T)$, we deduce

$$k_j^T = n_j - \text{rank}(\delta_{j-1}) - \text{rank}(\delta_j). \quad (\text{A5})$$

Using Eq. (A2) to eliminate $\text{rank}(\delta_j)$, we get

$$\begin{aligned} k_j^T &= n_j - \text{rank}(\delta_{j-1}) - [n_j - \text{nullity}(\delta_j)] \\ &= \text{nullity}(\delta_j) - \text{rank}(\delta_{j-1}), \end{aligned} \quad (\text{A6})$$

which is precisely the definition of k_j given in Eq. (34). This completes this simple but educational proof.

Appendix B: Further notation

1. Vector reshaping

Throughout the appendices we often reshape vectors into matrices. If we have a vector v belonging to some tensor product space $A \otimes B$, then we can reshape v into a matrix V . We always use lower-case symbols for vectors and upper-case for the resulting matrix after reshaping. Let $\{\hat{a}_i\}$ and $\{\hat{b}_j\}$ be unit basis vectors for A and B , respectively. Then any vector v can be decomposed in this basis as

$$v = \sum_{i,j} V_{i,j} \hat{a}_i \otimes \hat{b}_j, \quad (\text{B1})$$

where the coefficients $V_{i,j}$ are elements of the matrix representation. That is, $V_{i,j}$ is the entry in the i^{th} row and j^{th} column of matrix V . Furthermore, given matrices $M : A \rightarrow A$ and $N : B \rightarrow B$ we will rewrite equations as follows

$$(M \otimes N)v \rightarrow MVN^T, \quad (\text{B2})$$

which is easily verified.

2. Matrix support

We further introduce the notion of column and row support. Given any matrix X we let $\text{colsupp}(X)$ denote the set of columns in X with at least one nonzero entry. Given any matrix X we let $\text{rowsupp}(X)$ denote the set of rows in X with at least one nonzero entry. We shall often use $|\dots|$ to denote the number of rows or columns within some support. That is, $|\text{colsupp}(X)|$ is the number of columns in X with at least one nonzero entry. For example, if

$$X = \begin{pmatrix} 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}, \quad (\text{B3})$$

then $\text{colsupp}(X) = \{1, 2, 4, 5\}$ and $\text{rowsupp}(X) = \{1, 2, 3\}$. Furthermore, $|\text{colsupp}(X)| = 4$ and $|\text{rowsupp}(X)| = 3$.

Appendix C: Distance bounds: part one

Here we give proofs of distances associated with length-2 chain complexes constructed using the homological product (see Eqs. (46)-(49)).

1. First bound

We begin by showing that $\tilde{d}_{-1} \geq d_0 d_0^T$. The quantity \tilde{d}_{-1} is the weight of the smallest nonzero vector $r \in C_0 \otimes$

C_1 such that $\tilde{\delta}_{-1}r = 0$. We use that $r \in C_0 \otimes C_1$ can be reshaped into a matrix R ; see B 1 for discussion of reshaping. The condition $\tilde{\delta}_{-1}r$ entails that every column of R must be in $\ker(\delta_0)$ and every row of R must be in $\ker(\delta_0^T)$. Assuming, R is nonzero, there must be at least one non-zero column. Since this column has weight at least d_0 , it follows that there are at least d_0 non-zero rows. Each of these rows has weight at least d_0^T . Therefore, the total weight is at least $d_0d_0^T$ as required. Next, we show $\tilde{d}_{-1} \leq d_0d_0^T$. We assume, $d_0 \neq \infty$ and $d_0^T \neq \infty$ otherwise the inequality is trivially true. Let α be a minimal weight non-zero vector in the kernel of δ_0 , so $|\alpha| = d_0$. Similarly let $\beta \in \ker(\delta_0^T)$ with $|\beta| = d_0^T$. Then $\alpha \otimes \beta \in C_0 \otimes C_1$ has $|\alpha \otimes \beta| = d_0d_0^T$ and is easily verified to satisfy $\tilde{\delta}_{-1}(\alpha \otimes \beta) = 0$. The proof of $\tilde{d}_0^T = d_0d_0^T$ follows by symmetry.

2. Second bound

Next we show that $\tilde{d}_0 \geq \min[d_0, d_0^T]$. Recall, this is the weight of the smallest vector r such that $\tilde{\delta}_0r = 0$ and $r \notin \text{im}(\tilde{\delta}_{-1})$. All r can be decomposed as

$$r = \begin{pmatrix} r_a \\ r_b \end{pmatrix}, \quad (\text{C1})$$

where $\tilde{\delta}_0r = 0$ entails that $(\delta_0 \otimes \mathbb{1})r_a = (\mathbb{1} \otimes \delta_0^T)r_b$. Assuming r is a non-trivial cycle, it follows that there must exist a cocycle $w = (w_a, w_b)$ such that $w^T r = 1$. Therefore, $w_a^T r_a + w_b^T r_b = 1$ and either $w_a^T r_a = 1$ or $w_b^T r_b = 1$. We proceed assuming $w_a^T r_a = 1$ and further note that the cocycle can always be assumed to have the form $w = (e \otimes f) \oplus 0$. This is a good place to remind the reader that \oplus is the direct product and when applied to columns vectors means that we stack the columns. Since w ought to be a cocycle it must satisfy $\tilde{\delta}_{-1}^T w = 0$ which entails that $\delta_0 f = 0$. The relation $w^T r = 1$ then becomes $(e^T \otimes f^T)r_a = 1$. We can reshape some vectors into matrices, and these equations become

$$(e^T \otimes f^T)r_a = 1 \implies e^T R_a f = 1 \quad (\text{C2})$$

$$(\delta_0 \otimes \mathbb{1})r_a = (\mathbb{1} \otimes \delta_0^T)r_b \implies \delta_0 R_a = R_b \delta_0 \quad (\text{C3})$$

We consider the vector $R_a f$. From $\delta_0 R_a = R_b \delta_0$ we infer that $\delta_0(R_a f) = R_b \delta_0 f$. Using also that $\delta_0 f = 0$ we have a proof that $\delta_0(R_a f) = 0$ and so $R_a f \in \ker(\delta_0)$. However, $R_a f \neq 0$ otherwise it would be impossible to satisfy $e^T R_a f = 1$. It follows that $d_0 \leq |R_a f|$. Since $R_a f$ is formed from linear combinations of columns from R_a , we have $|R_a f| \leq |R_a|$ and hence $d_0 \leq |R_a|$. It follows that $d_0 \leq |r|$ in this case. For the $w_b^T r_b = 1$ case, a similar argument follows but giving a lower bound of $d_0^T \leq |r|$. Therefore, the actual lower bound on $|r|$ is the minimum of these two cases.

Appendix D: Soundness proof: part one

Here we prove Lem. 5 for $\tilde{\delta}_0^T$, with the $\tilde{\delta}_{-1}$ proof following a similar fashion. Recalling the definition of soundness, we consider $s \in \tilde{C}_0$ such that $s \in \text{im}(\tilde{\delta}_0^T)$ and $|s| < t = \min(d_0^T, d_0)$. Therefore, both $|s| < d_0^T$ and $|s| < d_0$ hold. There must exist at least one $r \in \tilde{C}_1 = C_1 \otimes C_0$ such that $s = \tilde{\delta}_0^T r$. This will not be the only possible solution, but let us begin by exploring the relationship between $|s|$ and $|r|$.

The vector s has two components $s = s_L \oplus s_R$ and breaking $s = \tilde{\delta}_0^T r$ into components, we have

$$\begin{aligned} s_L &= (\delta_0^T \otimes \mathbb{1})r, \\ s_R &= (\mathbb{1} \otimes \delta_0)r. \end{aligned} \quad (\text{D1})$$

Next, we reshape r , s_L and s_R into matrices (see B 1 for discussion of reshaping) so that

$$\begin{aligned} s_L &= (\delta_0^T \otimes \mathbb{1})r \implies S_L = \delta_0^T R, \\ s_R &= (\mathbb{1} \otimes \delta_0)r \implies S_R = R \delta_0^T. \end{aligned} \quad (\text{D2})$$

In terms of support (recall notation of App. B 2) the above equations entail that

$$\begin{aligned} \text{colsupp}(S_L) &\subseteq \text{colsupp}(R), \\ \text{rowsupp}(S_R) &\subseteq \text{rowsupp}(R). \end{aligned} \quad (\text{D3})$$

In general, this means that

$$|\text{colsupp}(S_L)| \leq |\text{colsupp}(R)|, \quad (\text{D4})$$

$$|\text{rowsupp}(S_R)| \leq |\text{rowsupp}(R)|. \quad (\text{D5})$$

Using $|X|$ to denote the number of 1s contained in a binary matrix X , we remark that $|\text{colsupp}(X)| \leq |X|$ and $|\text{rowsupp}(X)| \leq |X|$ for any X , and so

$$|S_L| \geq |\text{colsupp}(S_L)|, \quad (\text{D6})$$

$$|S_R| \geq |\text{rowsupp}(S_R)|.$$

Combined with $|s| = |S_L| + |S_R|$ we find

$$|s| \geq |\text{colsupp}(S_L)| + |\text{rowsupp}(S_R)|. \quad (\text{D7})$$

Squaring both sides and using $(a+b)^2/4 \geq ab$ for integer a and b , we obtain

$$|s|^2/4 \geq |\text{colsupp}(S_L)| \cdot |\text{rowsupp}(S_R)|. \quad (\text{D8})$$

We would like to substitute in Eqs. (D4)-(D5) but the inequality signs do not align correctly. We would be able to proceed if Eqs. (D4)-(D5) held with strict equality, but this is not always the case.

To proceed we use that the above R is not the only possible solution. Given an initial R we can transform to obtain a new R so that Eqs. (D2) are preserved, but so that also Eq. (D4) and Eq. (D5) become equalities. In particular, given a pair of vectors $a \in \ker \delta_0^T$ and $b \in \ker \delta_0$ we can perform $R \rightarrow R + ab^T$ and Eqs. (D2) will

$$\begin{array}{c}
 \text{columns for which } \delta_0^T a \neq 0 \\
 \downarrow \\
 \begin{array}{ccc}
 \text{nontrivial} & & \text{trivial columns} \\
 \text{columns in } \ker \delta_0^T & & \\
 \downarrow & & \downarrow \\
 R = \begin{pmatrix} \color{red}{0} & B & 0 \\ A & C & 0 \\ 0 & 0 & 0 \end{pmatrix} \begin{array}{l} \leftarrow \text{nontrivial rows in } \ker \delta_0 \\ \leftarrow \text{rows for which } \delta_0 b \neq 0 \\ \leftarrow \text{trivial rows} \end{array}
 \end{array}
 \end{array}$$

FIG. 5: The form of R after the repeated $R \rightarrow R + ab^T$ process has terminated. We have taken the liberty of permuting columns and rows, such that: any column of R in $\ker(\delta_0)$ will intersect block A ; any row (transposed) of R in $\ker(\delta_0^T)$ will intersect block B . Since the aforementioned $R \rightarrow R + ab^T$ process has terminated, there are no more column and row pairs such that they are in the relevant kernel and they intersect. Therefore, the upper-left block must be all-zero as shown otherwise there would still exist such an intersecting pair and the $R \rightarrow R + ab^T$ process ought to continue. Note further that the process must terminate after a finite number of rounds since the column and row supports are strictly decreasing with each transform of R . Since the middle block of columns are those that do not vanish under δ_0^T , we have that $|S_L| \geq |\delta_0^T R|$ is equal to the number of columns in the middle block of columns. Similarly, $|S_R|$ is lower bounded by the number of rows in the middle block of rows.

be preserved. We assume for now that neither Eq. (D4) nor Eq. (D5) are strict equalities, and so we may take both a and b^T to be column and row vectors from R . It follows that the new $R + ab^T$ has column and row support strictly contained within that of R , and the support may even reduce in size. Notice that adding ab^T will add a to every column in R on which b^T is supported. So if the support of a and b^T intersect in R , we can strictly decrease the number of columns and the number of rows in R . By intersect in R we mean that if b is the i^{th} row of R and a is the j^{th} column of R , then $R_{i,j} = 1$. Let us consider an example,

$$R = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & \color{red}{1} & 0 \end{pmatrix}, \quad (\text{D9})$$

so that $\text{colsupp}(R) = \{1, 2, 3, 4\}$ and $\text{rowsupp}(R) = \{1, 2, 3, 4, 5\}$. Let $a = (1, 1, 1, 1, 1)^T$ be the fourth column vector and $b = (1, 1, 1, 1, 0)$ be the last row vector. They intersect since $R_{5,4} = 1$ and we emphasize this by highlighting the intersecting element in bold and red. We find that

$$R' = R + ab^T = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \quad (\text{D10})$$

Notice that $\text{colsupp}(R) = \{1, 2, 3\}$ and $\text{rowsupp}(R) = \{1, 2, 3, 4\}$, so that the supports have strictly decreased. Also note that the intersection property was crucial. If we had instead considered

$$R = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & \color{red}{0} & 0 \end{pmatrix}, \quad (\text{D11})$$

with non-intersecting $a = (1, 1, 1, 1, 0)^T$ and $b = (1, 1, 1, 0, 0)$ then we would find

$$R' = R + ab^T = \begin{pmatrix} 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \end{pmatrix}. \quad (\text{D12})$$

The column and row support is completely unchanged. The key point is that when a and b^T intersect in R , we will add a to a set of columns including the column equal to a . Since we do this modulo 2, at least one column is removed. Similarly, at least one row will be removed.

Repeating this $R \rightarrow R + ab^T$ process must terminate when there are no remaining column/row pairs that intersect and are elements of the relevant kernels. After termination the matrix R was a special form best illustrated using a block matrix equation shown in Fig. 5 with further comment in the figure caption. Having transformed into this special form, we next use additional

assumptions under which A and B blocks vanish and so Eq. (D4) and Eq. (D5) become strict equalities. Assume A is nonzero so there exists a column vector c intersecting block A . Since $c \in \ker(\delta_0^T)$ and $c \neq 0$ we have $|c| \geq d_0^T$. Furthermore, since column vector c intersects block A we conclude that the middle block of rows in R must contains at least $|c|$ nonzero rows and consequently, $|\text{rowsupp}(S_R)| \geq |c|$ (see Fig. 5 and caption for more intuition) and consequently $|\text{rowsupp}(S_R)| \geq d_0^T$. Next, we use our assumption that $|s| < d_0^T$ that was asserted at the very start of this proof, which we combine with Eq. (D7) to conclude that

$$d_0^T > |\text{colsupp}(S_L)| + |\text{rowsupp}(S_R)|, \quad (\text{D13})$$

and so $d_0^T > |\text{rowsupp}(S_R)|$. Having proved both $|\text{rowsupp}(S_R)| \geq d_0^T$ and $d_0^T > |\text{rowsupp}(S_R)|$. We have a contradiction that is only resolved if such a column vector c does not actually exist and therefore $A = 0$.

Using a nonzero row vector in $\ker(\delta_0)$, a similar argument entails that $|\text{colsupp}(S_L)| \geq d_0$, which contradicts $|s| < d_0$, and so we conclude $B = 0$ also. Therefore, we see that the above transformation must yield a form with where Eqs. (D4)-(D5) hold with strict equality. This can be combined with Eq. (D8) to conclude that

$$|s|^2/4 \geq |\text{colsupp}(R)| \cdot |\text{rowsupp}(R)|. \quad (\text{D14})$$

Furthermore, if R is supported on a submatrix of size $|\text{colsupp}(R)|$ by $|\text{rowsupp}(R)|$ then the size of this submatrix gives an upper bound on $|R| = |r|$ so that

$$|\text{colsupp}(R)| \cdot |\text{rowsupp}(R)| \geq |r|. \quad (\text{D15})$$

Combining Eq. (D14) and Eq. (D15) produces the desired bound $|s|^2/4 \geq |r|$.

Appendix E: Distance bounds: part two

Here we prove Eqs. (63).

1. First bound

We begin with

$$\check{d}_0 \geq \min[\check{d}_{-1}, \max[\check{d}_0, \check{d}_{-1}^T], \check{d}_0^T] \quad (\text{E1})$$

and remark that the proof for \check{d}_{-1}^T will follow a similar fashion. Recall that \check{d}_0 is the weight of the smallest vector r such that $\check{\delta}_0 r = 0$ and $r \notin \text{im}(\check{\delta}_{-1})$. All r can be decomposed as

$$r = \begin{pmatrix} r_a \\ r_b \\ r_c \end{pmatrix}, \quad (\text{E2})$$

where $\check{\delta}_0 r = 0$ requires that

$$\begin{aligned} (\mathbb{1} \otimes \check{\delta}_{-1}^T) r_b &= (\check{\delta}_{-1} \otimes \mathbb{1}) r_a, \\ (\check{\delta}_0 \otimes \mathbb{1}) r_b &= (\mathbb{1} \otimes \check{\delta}_0^T) r_c. \end{aligned} \quad (\text{E3})$$

Taking the components of r and reshaping into a matrices, the vector equations transform into matrix equations as follows

$$(\check{\delta}_0 \otimes \mathbb{1}) r_b = (\mathbb{1} \otimes \check{\delta}_0^T) r_c \implies \check{\delta}_0 R_b = R_c \check{\delta}_0, \quad (\text{E4})$$

$$(\mathbb{1} \otimes \check{\delta}_{-1}^T) r_b = (\check{\delta}_{-1} \otimes \mathbb{1}) r_a \implies R_b \check{\delta}_{-1} = \check{\delta}_{-1} R_a. \quad (\text{E5})$$

Assuming r is a non-trivial cycle, it follows that there must exist a nontrivial cocycle $w = w_a \oplus w_b \oplus w_c$ such that $w^T r = 1$. Furthermore, the cocycle can be assumed to be of the form $w = (e_a \otimes f_a) \oplus (e_b \otimes f_b) \oplus (e_c \otimes f_c)$ since the span of such vectors encompasses all nontrivial cocycles.

Therefore, $w_a^T r_a + w_b^T r_b + w_c^T r_c = 1$ and at least one of these terms must equal 1 and there are four cases to consider

1. $w_a^T r_a = 1$ and $w_b^T r_b = w_c^T r_c = 0$, in which case we may assume $w = (e_a \otimes f_a) \oplus 0 \oplus 0$;
2. $w_c^T r_c = 1$ and $w_a^T r_a = w_b^T r_b = 0$, in which case we may assume $w = 0 \oplus 0 \oplus (e_c \otimes f_c)$;
3. $w_b^T r_b = 1$ and $w_a^T r_a = w_c^T r_c = 0$, in which case we may assume $w = 0 \oplus (e_b \otimes f_b) \oplus 0$;
4. $w_a^T r_a = w_b^T r_b = w_c^T r_c = 1$; in which case we can find a new w satisfying one of the above 3 cases.

We again remind the reader that all vectors are column vectors. Furthermore, \oplus is the direct product and when applied to columns vectors means that we stack the columns.

We first consider case 1. For $w = (e_a \otimes f_a) \oplus 0 \oplus 0$ to be a cocycle requires that $\check{\delta}_{-1} f_a = 0$. Furthermore, the condition $w^T r = (e_a^T \otimes f_a^T) r_a = 1$ in reshaped form becomes $e_a^T R_a f_a = 1$. We consider the vector $R_a f_a$, and find

$$\check{\delta}_{-1} R_a f_a = R_b \check{\delta}_{-1} f_a = 0, \quad (\text{E6})$$

where we have used Eq. (E5) and $\check{\delta}_{-1} f_a = 0$. In other words, $R_a f_a \in \ker(\check{\delta}_{-1})$. However, $R_a f_a$ is non-zero otherwise it would be impossible to satisfy $e_a^T R_a f_a = 1$. It follows that $\check{d}_{-1} \leq |R_a f_a|$. Since $R_a f_a$ is formed from linear combinations of columns from R_a , we have $|R_a f_a| \leq |R_a|$ and hence $\check{d}_{-1} \leq |R_a|$. It follows that $\check{d}_{-1} \leq |r|$ in case 1.

Next, we consider case 2. The proof method is essentially the same but we repeat for completeness. For $w = 0 \oplus 0 \oplus (e_c \otimes f_c)$ to be a cocycle requires that $\check{\delta}_0^T e_c = 0$. Furthermore, the condition $w^T r = (e_c^T \otimes f_c^T) r_c = 1$ in reshaped form becomes $e_c^T R_c f_c = 1$. We consider the vector $R_c^T e_c$, and find

$$\check{\delta}_0^T R_c^T e_c = (R_c \check{\delta}_0)^T e_c = (\check{\delta}_0 R_b)^T e_c = R_b^T \check{\delta}_0^T e_c = 0, \quad (\text{E7})$$

where we have used Eq. (E4) and $\tilde{\delta}_0^T e_c = 0$. In other words, $R_c^T e_c \in \ker(\tilde{\delta}_0^T)$. However, $R_c^T e_c$ is non-zero otherwise it would be impossible to satisfy $e_c^T R_c f_c = 1$. It follows that $\tilde{d}_0^T \leq |R_c^T e_c|$. Since $R_c^T e_c$ is formed from linear combinations of rows from R_c , we have $|R_c^T e_c| \leq |R_c|$ and hence $\tilde{d}_0^T \leq |R_c|$. It follows that $\tilde{d}_0^T \leq |r|$ in case 2.

Next, we consider case 3 then $w = 0 \oplus (e_b \otimes f_b) \oplus 0$. Furthermore, the condition $w^T r = (e_b^T \otimes f_b^T) r_b = 1$ in reshaped form becomes $e_b^T R_b f_b = 1$. The proof is slightly different from the above two cases. The cocycle conditions now tells us that both $\tilde{\delta}_{-1}^T e_b = 0$ and $\tilde{\delta}_0 f_b = 0$. We have

$$\tilde{\delta}_0 R_b f_b = R_c \tilde{\delta}_0 f_b = 0, \quad (\text{E8})$$

$$\tilde{\delta}_{-1}^T R_b^T e_b = (R_b \tilde{\delta}_{-1})^T e_b = (\tilde{\delta}_{-1} R_a)^T e_b = R_a^T \tilde{\delta}_{-1}^T e_b = 0, \quad (\text{E9})$$

where we have used $\tilde{\delta}_0 f_b = 0$ and $\tilde{\delta}_{-1}^T e_b = 0$ as asserted earlier. Furthermore, $R_b f_b \notin \text{im}(\tilde{\delta}_{-1})$ since otherwise $R_b f_b = \tilde{\delta}_{-1} u$ for some u and then $e_b^T R_b f_b = e_b^T \tilde{\delta}_{-1} u = (\tilde{\delta}_{-1}^T e_b)^T u$. However, since $\tilde{\delta}_{-1}^T e_b = 0$ this would entail $e_b^T R_b f_b = 0$ which is a contradiction and so we must have $R_b f_b \notin \text{im}(\tilde{\delta}_{-1})$. Similarly, one has that $R_b^T e_b \notin \text{im}(\tilde{\delta}_0^T)$ otherwise $R_b^T e_b = \tilde{\delta}_{-1} v$ for some v which would again lead to the contradiction $e_b^T R_b f_b = 0$ when combined with the fact that $\tilde{\delta}_0 f_b = 0$. Combining $R_b f_b \in \ker(\tilde{\delta}_0)$ and $R_b f_b \notin \text{im}(\tilde{\delta}_{-1})$ entails that $R_b f_b$ is a nontrivial cycle and so $\tilde{d}_0 \leq |R_b f_b|$. Since $R_b f_b$ is formed from linear combinations of columns from R_b , we have $|R_b f_b| \leq |R_b|$ and hence $\tilde{d}_0 \leq |R_b|$. Similarly, combining $R_b^T e_b \in \ker(\tilde{\delta}_{-1}^T)$ and $R_b^T e_b \notin \text{im}(\tilde{\delta}_0^T)$ leads to $\tilde{d}_{-1}^T \leq |R_b|$. This suffices to prove that in case 3 we have $|r| \geq \max[\tilde{d}_0, \tilde{d}_{-1}^T]$.

Since any one of the three cases may hold, we must take the minimum over the three cases. This yields the distance lower bound on \tilde{d}_0 .

2. Second bound

Here we prove

$$\tilde{d}_1 \geq \min[\tilde{d}_0, \tilde{d}_{-1}^T], \quad (\text{E10})$$

and remark that the proof for \tilde{d}_{-2}^T will follow a similar fashion. Let $s = s_a \oplus s_b \in \tilde{C}_1$ be a minimal distance nontrivial cycle for $\tilde{\delta}_1$. From $\tilde{\delta}_1 s = 0$ we may infer

$$(\tilde{\delta}_0 \otimes \mathbb{1}) s_a = (\mathbb{1} \otimes \tilde{\delta}_{-1}^T) s_b. \quad (\text{E11})$$

Since s is a nontrivial cycle, there must exist a nontrivial cocycle $w = w_a \oplus w_b$ such that $w^T s = 1$. There are two possible cases

1. $w_a^T s_a = 1$ and $w_b^T s_b = 0$, in which case we may assume $w = (e_a \otimes f_a) \oplus 0$;
2. $w_b^T s_b = 1$ and $w_a^T s_a = 0$, in which case we may assume $w = 0 \oplus (e_b \otimes f_b)$;

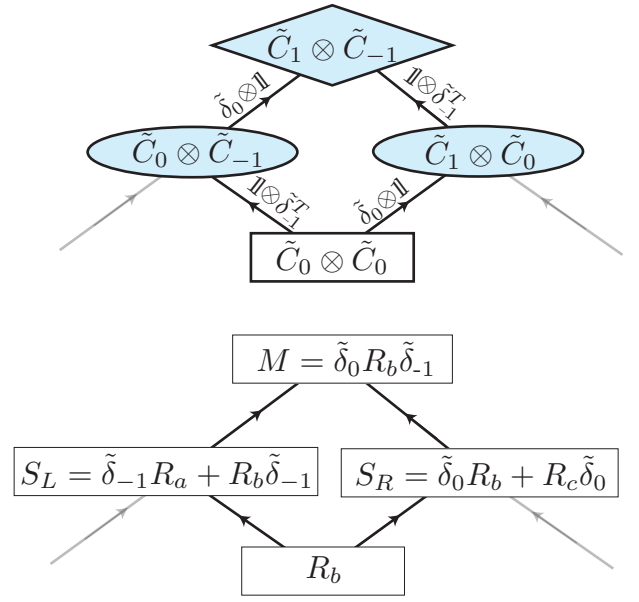


FIG. 6: Top: the relevant subgraph of Fig. 4 reproduced here for convenient reference. Bottom: the relations between different reshaped matrices as given in Eqs. (F2), (F3) and (F4). Here we draw the readers attention to how these two figures are connected. For instance r_b is an element of vector space $\tilde{C}_0 \otimes \tilde{C}_0$, which is then reshaped into R_b .

For case 1, since w is a cocycle $\tilde{\delta}_1^T w = 0$ and so both $\tilde{\delta}_{-1}^T e_a = 0$ and $\tilde{\delta}_{-1} f_a = 0$. However, $e_a \notin \text{im}(\tilde{\delta}_0)$ otherwise w would be a trivial cocycle. As in other proofs, we now reshape into matrix equations

$$w^T s = 1 \implies e_a^T S_a f_a = 1 \quad (\text{E12})$$

$$(\tilde{\delta}_0 \otimes \mathbb{1}) s_a = (\mathbb{1} \otimes \tilde{\delta}_{-1}^T) s_b \implies \tilde{\delta}_0 S_a = S_b \tilde{\delta}_{-1}^T. \quad (\text{E13})$$

Therefore,

$$\tilde{\delta}_0 (S_a f_a) = S_b \tilde{\delta}_{-1}^T f_a = 0 \quad (\text{E14})$$

where we have used Eq. (E13) and $\tilde{\delta}_{-1}^T f_a = 0$. In other words, $S_a f_a \in \ker(\tilde{\delta}_0)$. However, $S_a f_a \notin \text{im}(\tilde{\delta}_{-1}^T)$ otherwise there would exist a u such that $S_a f_a = \tilde{\delta}_{-1}^T u$ and then $e_a^T S_a f_a = e_a^T \tilde{\delta}_{-1}^T u = 0$ by virtue of $\tilde{\delta}_{-1}^T e_a = 0$. This is in contradiction with Eq. (E12) and so $S_a f_a$ is a nontrivial cycle of $\tilde{\delta}_0$ and must satisfy $\tilde{d}_0 \leq |S_a f_a| \leq |s|$. It follows that $\tilde{d}_0 \leq |s_a| \leq |s|$.

For case 2, a similar proof entails that $\tilde{d}_{-1}^T \leq |s_b| \leq |s|$. Since either case may hold the distance is given by the minimum of these two quantities.

Appendix F: Partial soundness

Here we prove Lem. 7, which is a major technical component of Thm. 7. We are working towards a low-weight solution of

$$m = (\tilde{\delta}_0 \otimes \tilde{\delta}_{-1}^T) r_b. \quad (\text{F1})$$

Input: A set of matrices R_b , S_L , S_R , $\tilde{\delta}_0$ and $\tilde{\delta}_{-1}$, with relationships defined in main text.

Output: A new transformed R'_b such that $\tilde{\delta}_0 R'_b \tilde{\delta}_{-1} = \tilde{\delta}_0 R_b \tilde{\delta}_{-1}$ and furthermore R'_b satisfies a set of constraints on its column and row support.

1. While $\text{rowsupp}(R_b \tilde{\delta}_{-1}) - \text{rowsupp}(S_L)$ is nonempty
 - (a) $i \leftarrow \text{SAMPLE}[\text{rowsupp}(R_b \tilde{\delta}_{-1}) - \text{rowsupp}(S_L)]$;
 - (b) $v^T \leftarrow$ the i^{th} row of R_b ;
 - (c) $r^T \leftarrow$ the i^{th} row of $R_b \tilde{\delta}_{-1}$;
 - (d) $j \leftarrow \text{SAMPLE}[\text{colsupp}(r^T)]$;
 - (e) $c \leftarrow$ the j^{th} column of $R_b \tilde{\delta}_{-1}$;
 - (f) $c' \leftarrow$ the j^{th} column of S_L ;
 - (g) $w \leftarrow c + c'$;
 - (h) $R_b \leftarrow R_b + wv^T$;
2. While $\text{colsupp}(R_b \tilde{\delta}_{-1}) - \text{colsupp}(S_L)$ is nonempty
 - (a) $j \leftarrow \text{SAMPLE}[\text{colsupp}(R_b \tilde{\delta}_{-1}) - \text{colsupp}(S_L)]$;
 - (b) $c \leftarrow$ the j^{th} column of $R_b \tilde{\delta}_{-1}$;
 - (c) $k \leftarrow \text{SAMPLE}[\text{rowsupp}(R_b) \cap \text{rowsupp}(c)]$
 - (d) $v^T \leftarrow$ the k^{th} row of R_b ;
 - (e) $R_b \leftarrow R_b + cv^T$;
3. While $\text{rowsupp}(R_b) - \text{rowsupp}(R_b \tilde{\delta}_{-1})$ is nonempty
 - (a) $j \leftarrow \text{SAMPLE}[\text{rowsupp}(R_b) - \text{rowsupp}(R_b \tilde{\delta}_{-1})]$;
 - (b) j^{th} row of $R_b \leftarrow (0, 0, \dots, 0)$
4. While $\text{colsupp}(\tilde{\delta}_0 R_b) - \text{colsupp}(S_R)$ is nonempty
 - (a) $i \leftarrow \text{SAMPLE}[\text{colsupp}(\tilde{\delta}_0 R_b) - \text{colsupp}(S_R)]$;
 - (b) $v \leftarrow$ the i^{th} column of R_b ;
 - (c) $r \leftarrow$ the i^{th} column of $\tilde{\delta}_0 R_b$;
 - (d) $j \leftarrow \text{SAMPLE}[\text{rowsupp}(r)]$;
 - (e) $c \leftarrow$ the j^{th} row of $\tilde{\delta}_0 R_b$;
 - (f) $c' \leftarrow$ the j^{th} row of S_R ;
 - (g) $w^T \leftarrow c + c'$;
 - (h) $R_b \leftarrow R_b + vv^T$;
5. While $\text{rowsupp}(\tilde{\delta}_0 R_b) - \text{rowsupp}(S_R)$ is nonempty
 - (a) $j \leftarrow \text{SAMPLE}[\text{rowsupp}(\tilde{\delta}_0 R_b) - \text{rowsupp}(S_R)]$;
 - (b) $v^T \leftarrow$ the j^{th} row of $\tilde{\delta}_0 R_b$;
 - (c) $k \leftarrow \text{SAMPLE}[\text{colsupp}(R_b) \cap \text{colsupp}(v^T)]$
 - (d) $c \leftarrow$ the k^{th} column of R_b ;
 - (e) $R_b \leftarrow R_b + cv^T$;
6. While $\text{colsupp}(R_b) - \text{colsupp}(\tilde{\delta}_0 R_b)$ is nonempty
 - (a) $j \leftarrow \text{SAMPLE}[\text{colsupp}(R_b) - \text{colsupp}(\tilde{\delta}_0 R_b)]$;
 - (b) j^{th} column of $R_b \leftarrow (0, 0, \dots, 0)^T$

Return: R_b .

FIG. 7: A partial decoder. Certain choices are arbitrary and so we use $\text{SAMPLE}[\dots]$ to mean randomly sample (or use any other criteria) to select one element from a set. For an example of step 1 see transform 1 of toy example 3 in Fig. 10. For an example of step 2 see transform 1 of toy example 1 in Fig. 8. For an example of step 3 see transform 2 of toy example 1 in Fig. 8. For an example of step 5 see transform 1 of toy example 2 in Fig. 9. See the supplementary material for a Mathematica implementation of this partial decoder.

So far we only know that there must be at least one r_b satisfying this equation. We proceed by looking for other r_b consistent with Eq. (F1) that have a low weight and other additional properties. At this point it is convenient to reshape our vectors into matrices (recall App. B 1) and the previous equations become

$$S_L = \tilde{\delta}_{-1}R_a + R_b\tilde{\delta}_{-1}, \quad (\text{F2})$$

$$S_R = \tilde{\delta}_0R_b + R_c\tilde{\delta}_0, \quad (\text{F3})$$

$$M = \tilde{\delta}_0S_L = S_R\tilde{\delta}_{-1} = \tilde{\delta}_0R_b\tilde{\delta}_{-1}. \quad (\text{F4})$$

As a visual aid to understanding these equations we provide Fig. 6.

Notice that if R_b has any columns in the kernel of $\tilde{\delta}_0$, these can be removed without changing M . Similarly, if R_b has any rows in the kernel of $\tilde{\delta}_{-1}^T$, these can be removed without changing M . So we see there are transforms that preserve M but remove elements from R_b .

Our proof is essentially a decoder for R_b . This is a partial decoder as it requires an initial guess for R_b and does not solve for R_a and R_c . We describe the decoder in pseudocode in Fig. 7 and give toy examples of its implementation in Figs. 8, 9 and 10. The rest of this section will discuss the possible transforms of R_b and then an analysis of the partial decoder.

Support inclusions.- Simple matrix algebra (recall notation from App. B 2) leads to the inclusions

$$\text{rowsupp}(R_b\tilde{\delta}_{-1}) \subseteq \text{rowsupp}(R_b), \quad (\text{F5})$$

$$\text{colsupp}(\tilde{\delta}_0R_b) \subseteq \text{colsupp}(R_b), \quad (\text{F6})$$

$$\text{colsupp}(M) \subseteq \text{colsupp}(S_L), \quad (\text{F7})$$

$$\text{rowsupp}(M) \subseteq \text{rowsupp}(S_R). \quad (\text{F8})$$

If we inspect our toy examples (Figs. 8, 9 and 10) we see that these are indeed satisfied before any transformations are performed.

The goal of the partial decoder is to perform a series of transforms such that M is preserved and the final output R_b satisfies the following:

$$\text{rowsupp}(R_b\tilde{\delta}_{-1}) \subseteq \text{rowsupp}(S_L), \quad (\text{F9})$$

$$\text{colsupp}(R_b\tilde{\delta}_{-1}) \subseteq \text{colsupp}(S_L), \quad (\text{F10})$$

$$\text{rowsupp}(R_b\tilde{\delta}_{-1}) = \text{rowsupp}(R_b), \quad (\text{F11})$$

$$\text{colsupp}(\tilde{\delta}_0R_b) \subseteq \text{colsupp}(S_R), \quad (\text{F12})$$

$$\text{rowsupp}(\tilde{\delta}_0R_b) \subseteq \text{rowsupp}(S_R), \quad (\text{F13})$$

$$\text{colsupp}(\tilde{\delta}_0R_b) = \text{colsupp}(R_b), \quad (\text{F14})$$

The partial decoder goes through 6 while loops with each loop aiming to enforce one of these conditions. In each case, the idea is that if the condition is violated this enables us to perform some M preserving transformation that removes columns or rows from R_b .

Overview of R_b transforms.- Next, we give a very general account of how we may transform R_b while preserving M . Given a column vector c such that $\tilde{\delta}_0c = 0$, we

may add c to any of the columns in R_b and M will not change. Furthermore, if $\text{rowsupp}(R_b)$ and $\text{rowsupp}(c)$ have any elements in common, then we can perform a transformation that removes one row from R_b . Also note that if c is itself a column vector of R_b then it is trivially the case that they share row support in common. This is similar to the intersecting argument encountered in the proof in App. D. Let us again illustrate by example. Suppose

$$R_b = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \text{ and } c = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \quad (\text{F15})$$

so that $\text{rowsupp}(R_b) = \{1, 2, 4\}$ and $\text{rowsupp}(c) = \{1, 2, 3\}$. We see that both supports share 1 and 2 in common and so either row could be removed. For example, to remove row 1 we add c to columns 1, 2 and 3, yielding

$$R_b = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix}, \quad (\text{F16})$$

where row 1 is now trivial. Note that while row 1 has been removed, $\text{rowsupp}(R_b)$ now includes row 3, so the total number of supported rows has not decreased. Note the $\text{colsupp}(R_b)$ has not gained any new elements.

Let us now consider another example

$$R_b = \begin{pmatrix} 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix} \text{ and } c = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 0 \end{pmatrix}, \quad (\text{F17})$$

which is similar to the earlier example except now $\text{rowsupp}(R_b)$ is equal to $\text{rowsupp}(c)$. Consequently, when we use c to remove row 1 we obtain

$$R_b = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \quad (\text{F18})$$

so row 1 has been removed but also the total number of rows has decreased. Note again that $\text{colsupp}(R_b)$ has not gained any new elements.

More generally, we have that

Claim 5 (Row removal) *Let c be a column vector such that $\tilde{\delta}_0c = 0$ and let v^T be the j^{th} row vector of R_b where $j \in \text{rowsupp}(R_b) \cap \text{rowsupp}(c)$. Then the transform $R_b \rightarrow R'_b = R_b + cv^T$ satisfies the following*

1. the transform will preserve M ;

2. the new R'_b will have row support in $\text{rowsupp}(R_b) \cup \text{rowsupp}(c) - \{j\}$. If one further has that $\text{rowsupp}(c)$ is contained within $\text{rowsupp}(R_b)$ then the number of rows has strictly decreased.
3. the new R'_b will have column support within the original $\text{colsupp}(R_b)$.

Similarly,

Claim 6 (Column removal) Let v^T be a row vector such that $v^T \tilde{\delta}_{-1} = 0$ and let c be the j^{th} column vector of R_b where $j \in \text{colsupp}(R_b) \cap \text{colsupp}(v)$. Then the transform $R_b \rightarrow R'_b = R_b + cv^T$ satisfies the following

1. the transform will preserve M ;
2. the new R'_b will have column support in $\text{colsupp}(R_b) \cup \text{colsupp}(v^T) - \{j\}$. If one further has that $\text{colsupp}(v^T)$ is contained within $\text{colsupp}(R_b)$ then the number of columns has strictly decreased.
3. the new R'_b will have row support within the original $\text{rowsupp}(R_b)$.

We now proceed to use these ideas in the following way.

While loop 1.- This iteratively reduces the number of elements in $\text{rowsupp}(R_b \tilde{\delta}_{-1}) \cup \text{rowsupp}(S_L)$ until we have $\text{rowsupp}(R_b \tilde{\delta}_{-1}) \subseteq \text{rowsupp}(S_L)$. Whenever this inclusion is false, there exists at least one column, say c , of $R_b \tilde{\delta}_{-1}$ such that $\text{rowsupp}(c)$ is not a subset of $\text{rowsupp}(S_L)$. Furthermore, if c is the j^{th} column of $R_b \tilde{\delta}_{-1}$ let c' be the j^{th} column of S_L . We must have that $c' \neq c$ otherwise $\text{rowsupp}(c)$ would be a subset of $\text{rowsupp}(S_L)$. Since $\tilde{\delta}_0 R_b \tilde{\delta}_{-1} = \tilde{\delta}_0 S_L$ we must have that these matrices are equal on the j^{th} column and so $\tilde{\delta}_0 c = \tilde{\delta}_0 c'$. Therefore, the vector $w = c' - c$ satisfies the following properties:

1. $w \neq 0$ which follows from $c \neq c'$;
2. $w \in \ker(\tilde{\delta}_0)$ which follows from $\tilde{\delta}_0 c = \tilde{\delta}_0 c'$;
3. $\text{rowsupp}(w) \subseteq \text{rowsupp}(R_b \tilde{\delta}_{-1}) \cup \text{rowsupp}(S_L)$ which follows from $\text{rowsupp}(w) \subseteq \text{rowsupp}(c) \cup \text{rowsupp}(c')$.
4. $\text{rowsupp}(w) \cap \text{rowsupp}(R_b)$ is non-empty, because c (and hence w) has row support outside $\text{rowsupp}(S_L)$.

Therefore, we can (by virtue of claim 5) use column vector w to remove a row from R_b . The row removal process is possible for any row in $\text{rowsupp}(R_b) \cap \text{rowsupp}(w)$. However, we want the final row support to be within $\text{rowsupp}(S_L)$ and so from the set of possible rows we choose one outside the set $\text{rowsupp}(S_L)$. In practice, the partial decoder pseudocode make this row selection the first task. Therefore, the set $(\text{rowsupp}(R_b \tilde{\delta}_{-1}) \cup \text{rowsupp}(S_L)) - \text{rowsupp}(S_L)$ strictly decreases in size. Note also that $\text{colsupp}(R_b)$ will not increase (by clause

2 of claim 5). For an example, see transform 1 of toy example 3 in Fig. 10.

While loop 2.- This iteratively reduces the number of rows in R_b until we have $\text{colsupp}(R_b \tilde{\delta}_{-1}) \subseteq \text{colsupp}(S_L)$. First note that if $R_b \tilde{\delta}_{-1}$ has any nonzero columns outside $\text{colsupp}(S_L)$, the column must be in the kernel of $\tilde{\delta}_0$. To prove this, note that if the offending column was outside $\ker(\tilde{\delta}_0)$ then $\text{colsupp}(\tilde{\delta}_0 R_b \tilde{\delta}_{-1})$ would be strictly larger than $\text{colsupp}(\tilde{\delta}_0 S_L)$ which contradicts $\tilde{\delta}_0 R_b \tilde{\delta}_{-1} = \tilde{\delta}_0 S_L$. Since the column is in $\ker(\tilde{\delta}_0)$ and within $\text{colsupp}(R_b \tilde{\delta}_{-1})$, its presence allows us (by virtue of claim 5) to remove a row from R_b . It is crucial that after each iteration of the loop, the column support of R_b strictly decreases (by clause 2 of claim 5), which entails that the while loop must terminate after a finite number of iterations. It is important to comment on what we do not show here; we do not show that each iteration strictly removes columns from $\text{colsupp}(R_b \tilde{\delta}_{-1})$ until it is contained in $\text{colsupp}(S_L)$. Rather the number of rows in R_b are strictly decreased and this process cannot continue without end, so the while loop termination criteria must be satisfied within a finite number of rounds. To be precise, the while loop must terminate, since either (1) after a finite number of loops we obtain some nonzero R_b such that $\text{colsupp}(R_b \tilde{\delta}_{-1}) \subseteq \text{colsupp}(S_L)$; or (2) after a finite number of iterations all rows will be removed from R_b , so that $R_b = 0$, and then $\text{colsupp}(R_b \tilde{\delta}_{-1}) = \text{colsupp}(0) = \emptyset$ is trivially true. Again $\text{colsupp}(R_b)$ will not increase. For an example, see transform 1 of toy example 1 in Fig. 8.

While loop 3.- This iteratively reduces the number of rows in R_b until $\text{rowsupp}(R_b \tilde{\delta}_{-1}) = \text{rowsupp}(R_b)$. This is a fairly straightforward step, since the offending rows must be in the kernel of $\tilde{\delta}_{-1}^T$ they can just be simply removed. Removing rows from R_b leads to rows being removed from $R_b \tilde{\delta}_{-1}$ and the condition established in the previous while loop (that $\text{colsupp}(R_b \tilde{\delta}_{-1})$) will remain true. For an example, see transform 2 of toy example 1 in Fig. 8.

While loop 4.- This is similar to while loop 1, except with roles of rows and columns switched and applied to different matrices. Here we reduce the number of elements in $\text{colsupp}(\tilde{\delta}_0 R_b) \cup \text{colsupp}(S_R)$ until we have $\text{colsupp}(\tilde{\delta}_0 R_b) \subseteq \text{colsupp}(S_R)$, making use of claim 6. Since the process does not introduce any new elements into $\text{rowsupp}(R_b)$, the previously established conditions will continue to hold true.

While loop 5.- This is similar to while loop 2, except with roles of rows and columns switched and applied to different matrices and making use of claim 6. For an example of step 5 see transform 1 of toy example 2 in Fig. 9.

While loop 6.- This is similar to while loop 3, except with roles of rows and columns switched and applied to different matrices.

Analysis.- The above process will terminate because the column and row support of R_b is being gradually reduced. By repeating the above transformations until

the process terminates, we ensure that $R_b \tilde{\delta}_{-1}$ has row and column support strictly within that of S_L . Therefore, the combination $S_L - R_b \tilde{\delta}_{-1}$ also has row and column support strictly within that of S_L . We can infer that $S_L - R_b \tilde{\delta}_{-1} = \sum_i \alpha_i \otimes \hat{a}_i$ where α_i are the column vectors. Since S_L has at most $|S_L|$ columns, there can be at most $|S_L|$ nonzero α_i . Since S_L has at most $|S_L|$ rows, each α_i has weight at most $|S_L|$. This proves the small $|S_L|$ remainder property of our lemma (see property 3). The small $|S_R|$ remainder property holds by a similar fashion

(see property 4). Furthermore, combining Eq. (F9) and Eq. (F11), we conclude that the final R_b has fewer rows than S_L and so no more than $|S_L|$ rows. Similarly, we deduce that the final R_b has fewer columns than S_R and so no more than $|S_R|$ rows. Since the nonzero values of R_b are contained within a submatrix of size $|S_L|$ by $|S_R|$, we know $|R_b| \leq |S_L| \cdot |S_R|$. This proves property 2 of the lemma. It should be clear that property 1 holds because the value of M was initially correct and has been preserved through all transformations.

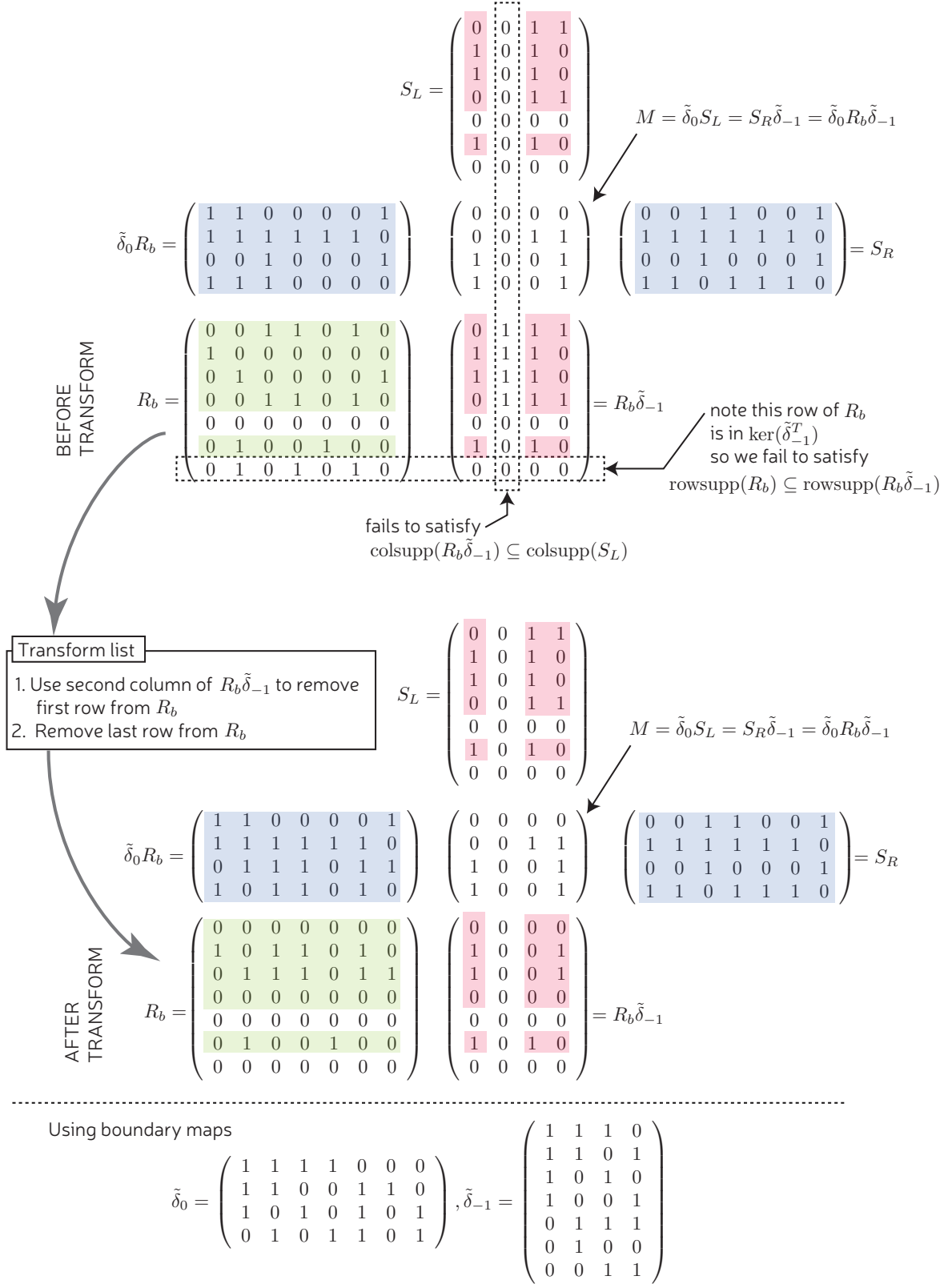


FIG. 8: Toy example 1 showing the form of an initial R_b matrix before any transformations have been performed. The matrix $\tilde{\delta}_0$ was not generated by the homological product but otherwise all features are correct. An actual homological product example would be too large to be instructive and furthermore the partial soundness proof does not use any such properties. The goal is to transform R_b such that M is unchanged, but after the transform R_b , $R_b \tilde{\delta}_{-1}$ and $\tilde{\delta}_0 R_b$ are only supported within the highlighted boxes. The highlighted boxes are themselves derived from the column and row support of S_L and S_R that are fixed.

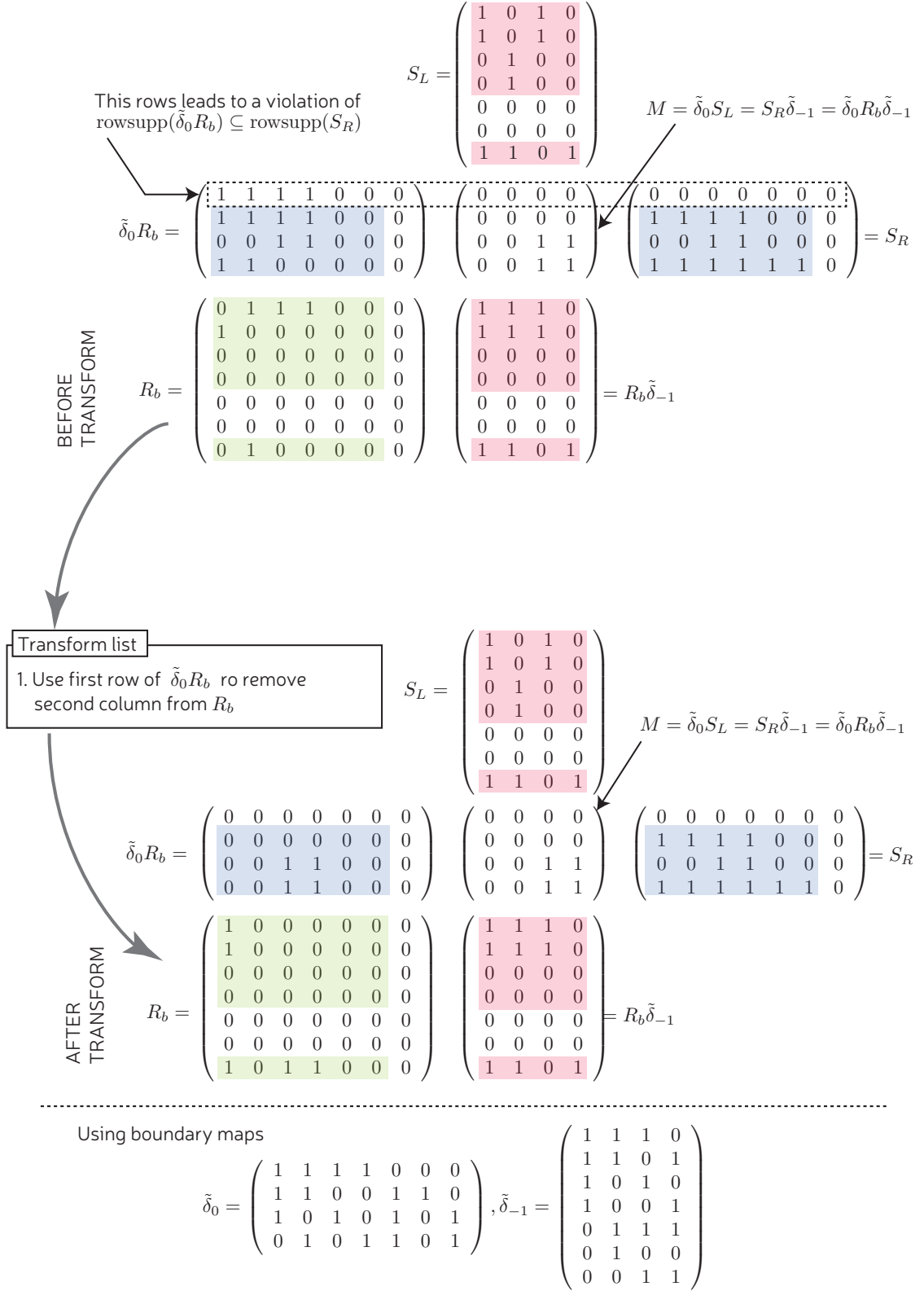


FIG. 9: Toy example 2 showing the form of an initial R_b matrix before any transformations have been performed. All δ boundary maps are the same as in toy example 1 shown in Fig. 8.

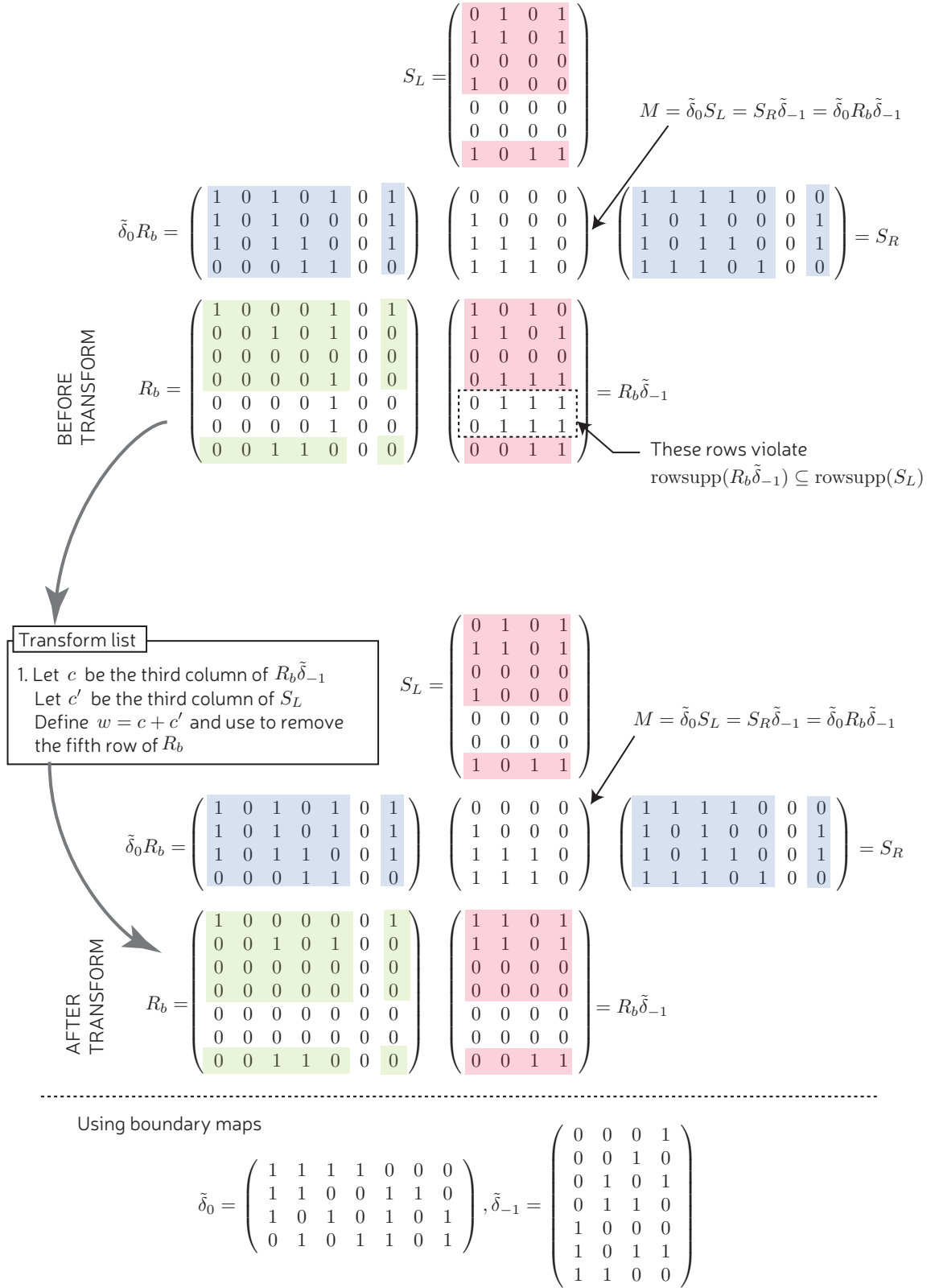


FIG. 10: Toy example 3. Note that in this example we use a different boundary map $\tilde{\delta}_{-1}$ just for the sake of variety.